USING MACHINE LEARNING TO ADVANCE HIGH SCHOOL DROPOUT PREDICTION AND PREVENTION

Anika Alam

A DISSERTATION

in

Education

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2025

Supervisor of Dissertation:

A. Brooks Bowden, Associate Professor of Education

Graduate Group Chairperson:

Matthew Hartley, Professor of Education

Dissertation Committee:

A. Brooks Bowden, Associate Professor of Education

Wendy Chan, Assistant Professor of Education

Alex J. Bowers, Professor of Education Leadership, Teachers College, Columbia University

USING MACHINE LEARNING TO ADVANCE HIGH SCHOOL DROPOUT PREDICTION AND PREVENTION

I dedicate this dissertation to the memory of my grandparents, Badrul and Rezia Alam. Bhaiya and Apu: your love, sacrifices, and profound belief in me have shaped the person I am today. This day would not have been possible without your enduring presence in my life.

ACKNOWLEDGMENT

This research was supported by two sources. The first is a grant is from the American Educational Research Association which receives funds for its "AERA-NSF Grants Program" from the National Science Foundation under NSF award NSF-DRL #1749275. The second source is the Institute of Education Sciences, U.S. Department of Education through Grant R305B200035 to the University of Pennsylvania. The opinions expressed are those of the author and do not represent views of the Institute, the U.S. Department of Education, AERA, or NSF.

First and foremost, I express sincerest gratitude to my incredible advisor, Brooks Bowden. I am grateful for your unwavering support, advocacy, and investment in my journey. Your leadership, work ethic, and dedication to education research have been a source of inspiration. Your guidance – whether about work, insights, or personal advice – has shaped my professional values and motivates me to strive for excellence. I am grateful for the opportunity to learn from you and hope to carry forward the values and knowledge you have shared.

I thank my committee members, Alex Bowers and Wendy Chan, for their invaluable support in the development of this dissertation. Your insightful feedback and thoughtful discussions have shaped the core ideas of this research and provided clarity and direction at critical junctures. I truly appreciate your generosity and effort to ensure that this dissertation meets standards of quality.

I convey appreciation to Ryan Baker, Andres Zambrano, Lidia Rossi, and Annaliese Paulson for their support on this dissertation. Your expertise, guidance, and feedback were instrumental in deepening my understanding of the machine learning landscape and exploring the various directions this work could take.

I sincerely appreciate the mentorship from Erica Greenberg, Lauren Scher, Michael Gottfried, Sade Bonilla, and Sangyoo Lee. Your insights, encouragement, and investment in my professional journey have helped me grow. I am fortunate to have had the opportunity to learn from you.

I am deeply grateful to the Penn community for being my support system. I am especially grateful to Anahita, David, Hanna, Katie, Sam, and Victoria for their moral support. You have become such an important part of my work life, and I am so grateful for your support, wisdom, and the friendship we have built. Thank you for making my day brighter. I look forward to many more years of supporting one another and continuing the tradition of dining at Nando's Peri-Peri at research conferences.

I would not have made it this far without my friends and loved ones, whose support and guidance continue to be my strength. My parents' unwavering optimism and sacrifices have been the foundation of my academic and personal growth. Ammu and Baba - from instilling in me the value of perseverance to providing endless reassurance during challenging times, your words continue to hold me strong. Finally, my deepest gratitude goes towards Samir, the wind beneath my wings. Your belief in me has been the driving force behind this journey. Thank you. To Ammu, Baba, and Samir: this achievement is as much yours as it is mine.

ABSTRACT

USING MACHINE LEARNING TO ADVANCE HIGH SCHOOL DROPOUT PREDICTION AND PREVENTION

Anika Alam

A. Brooks Bowden

The importance of high school completion for jobs and postsecondary opportunities is well-documented. Combined with federal laws where high school graduation rate is a core performance indicator, school, districts, and states face pressure to actively monitor and assess high school completion. This study employs machine learning techniques to identify students at-risk of exiting high school in either 9th or 10th grade. I find increased precision when applying resampling techniques to balance the training data, and that logistic regression performs similarly to more complex algorithms. When assessing the algorithmic fairness of models, I find most models tend to discriminate students with group membership in English proficiency, disability, and economic disadvantage attributes. Post-hoc analyses of the XGboost model reveal that a student's age in 8th grade followed by middle grade absences, especially chronic absenteeism, is predictive of early exit. This study advances the current state of knowledge in the field by (1) generating synthetic data to improve model accuracy, (2) ensuring that model predictions prevent the deepening of structural inequities, and (3) exploring novel approaches to enhance the explainability associated with "black box" models, ultimately generating actionable insights for practitioners and stakeholders.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

This chapter introduces the context of the dissertation, beginning with the significance of high school completion. It then explores how federal laws influence high school completion, followed by a discussion of the emergence of early warning systems as a response to these influences. Finally, the chapter provides a rationale for the necessity of this research, specifically at the intersection of education data science and traditional education research.

1.1 Importance of high school completion

There are economic, social, and civic consequences of dropping out of high school. Compared to high school graduates, adults without a high school diploma earn substantially less in the labor market, experience poorer health, are more likely to engage in criminal behavior, are more likely to require public assistance, and are less likely to vote (Krueger et al., 2015; Belfield & Levin, 2007; Dee, 2004). This is reflected in the labor market, where workers whose highest education level was high school completion typically earn $26,000 more per year than those who did not complete high school. (NCES, 2023). The monetary benefits translate to increased societal efficiency: if the number of students who withdraw from high school was cut in half, then the country can recover over 45 billion dollars that would have otherwise been spent on health care expenditures, lost tax revenues, and social services (Levin et al., 2007). Considering pecuniary and non-pecuniary returns of having a high school degree, on-time high school completion merits serious attention.

In the 2021–22 school year, the national average 4-year high school graduation rate was 87 percent. While this marks a gradual improvement over previous years, significant disparities persist among students from disadvantaged backgrounds. For example, in the same year, economically disadvantaged students had a graduation rate of just 81 percent. Other student groups were well below the national average; English language learners, students with disabilities, and homeless students faced even steeper challenges, with completion rates of 72 percent, 71 percent, and 68 percent, respectively (NCES, 2024). These gaps are not just numbers – they reflect systemic barriers that disproportionately affect students furthest from opportunity.

The 2016 Current Population Survey (CPS) reveals concerning trends in high school dropout rates across different income groups in the United States. In 2016, the dropout rate for 16- to 24-year-olds from families in the highest income quartile was just 3.9 percent, while those from the middle-high income quartile had a slightly lower dropout rate of 3.6 percent. In contrast, students from families in the lowest income quartile faced a dropout rate of 7.2 percent, more than double that of their higher-income peers. Moreover, the number of 16- to 24-year-olds who did not complete high school or were not enrolled in high school was 3.7 times higher in low-income families compared to high-income families (McFarland et al., 2018). This stark contrast underscores the significant disparities in educational access.

These disparities not only limit future economic opportunities for these young people but also contribute to perpetuating cycles of poverty. Without a high school diploma, individuals are less likely to secure stable, well-paying jobs, which in turn exacerbates economic inequality and hinders social mobility. A high school diploma plays a crucial role in enhancing the productivity of the United States in equipping individuals with workforce readiness, economic growth, social stability, and reduced inequality.

1.2 Policy landscape

Federal education laws in the last two decades have emphasized the importance of high school completion. The No Child Left Behind (NCLB) Act, passed in 2002, mandated that states develop and enforce accountability measures to ensure student learning across all public K-12 schools in the United States. As part of the law's comprehensive framework, NCLB required states to administer standardized tests in mathematics and reading in designated grade levels, set rigorous academic standards, and publish annual report cards that detailed student performance (U.S. Department of Education, n.d.). These report cards were designed to provide transparency and ensure that schools were meeting established educational goals. In exchange for receiving federal Title I funding, states were required to oversee the implementation of accountability mechanisms, including sanctions for schools that failed to meet Adequate Yearly Progress (AYP). AYP served as a measure of how well students were performing relative to the state's academic standards, with an emphasis on ensuring that all students, regardless of background, achieved proficiency in core subjects like math and reading. Progress on these standards were tested annually for all students in third through eighth grade and in one grade in high school. Annual test scores were compared to prior years to determine progress on state-determined AYP standards. Schools that did not meet AYP faced escalating consequences, such as developing and implementing two-year improvement plans, additional tutoring services, offering school choice options, or even restructuring efforts.

One significant aspect of NCLB was its focus on the graduation rates of high school students. The law required states to report both aggregate graduation rates and disaggregated graduation rates for specific student subgroups, such as racial and ethnic minorities, low-income students, and those with disabilities. Schools that failed to report graduation rates for even one year could be deemed to have missed AYP, triggering sanctions. This provision underscored NCLB's commitment to addressing achievement gaps and holding schools accountable for the success of all students, particularly those in historically underperforming subgroups. Moreover, recent evidence suggests that federal accountability has substantially increased high school completion rates and human capital (Harris, 2020; Harris et al., 2023). This law places incentive and pressure for school systems to actively monitor, assess, and improve high school completion.

Critics of NCLB argue that the law's provisions disincentivize high school completion and further exacerbate push-out rates for marginalized students. Prior studies have linked the effects of grade retention during NCLB era to higher student exit (or dropout) rates (Darling Hammond, 2006; Rumberger, 2008). For better or for worse, federal accountability in K-12 education is undoubtedly complex and has shaped student learning.

The successor to NCLB, the 2015 Every Student Succeeds Act (ESSA), continues to emphasize high school graduation rate as a core academic performance indicator. ESSA requires that "states and districts are required to intervene in high schools with on-time graduation rates lower than 67 percent" (U.S. Department of Education, 2017). Consequently, states must identify and label high schools for comprehensive support and improvement (CSI) that fail to meet the 67 percent threshold.

Regardless of the impact these laws had on student achievement, both NCLB and ESSA have significantly contributed to the expansion of administrative data in the K-12 education system. The emphasis of standardized testing, reporting, and data collection led to a large increase in the volume of student-level educational information collected by local and state agencies (Figlio & Loeb, 2011). The extensive administrative data collected on students include demographics, attendance, discipline, test scores, and other performance indicators. This data are often systematically aggregated to create longitudinal data systems which collect, store, and analyze educational data over time across various stages of a student's academic journey. By linking multiple data points across years, longitudinal systems enable the identification of early signs of disengagement, allowing for more targeted interventions aimed at improving student outcomes (Allensworth & Easton, 2007; U.S. Department of Education, 2012).

The U.S. Department of Education recognizes the potential to inform data-driven decision-making and strongly encourages the development and use of statewide longitudinal data systems (SLDS). Since 2005, the Education Technical Assistance Act has awarded competitive grants to support the establishment and expansion of SLDS. Every state thus far has received at least one such grant. In 2023, 25 states and the District of Columbia were awarded SLDS grants. This underscores the growing importance and relevance of leveraging administrative data to improve student outcomes.

1.3 Emergence of Early Warning Systems (EWS)

Early warning systems (EWS) are prediction tools that monitor and anticipate which individuals and communities are at risk of an adverse outcome. EWS are used in various public sectors. For instance, EWS have been used by environmental agencies to identify communities at risk of natural disasters (CDC, 2024), in healthcare to predict maternal mortality by monitoring risk factors such as blood pressure, age, and access to prenatal care (NIH, 2019), and in public health to track the spread of infectious diseases (Wu, 2016). In education, there has been a growth of EWS where schools, districts, or states aim to identify students at risk of missing key educational milestones. These include dropping out of high school, experiencing chronic absenteeism, or being held back a grade (AIR, 2010; Bowers et al., 2013).

Unlike SLDS that examine broader trends over time, EWS are predictive tools that identify individuals or groups who may benefit from immediate attention. Most EWS applications in K-12 focus on identifying students at risk of dropping out of school. These applications utilize student educational records (e.g. administrative data that are often linked in SLDS) as risk indicators or predictors to predict which students may need additional support.

Dropout prediction is crucial for early identification of at-risk students because it provides schools and districts with evidence to inform and implement targeted supports and services. The earlier schools can identify at-risk students, the earlier they can provide targeted interventions such as double-dose algebra (Nomi and Raudenbush, 2016; Cortes et al., 2016) or credit recovery programs that let students earn credits for courses they previously failed (Heinrich et al., 2019; Rickles et al., 2018; Viano et al., 2023). A national survey found at least 8 state education agencies have either developed or are currently building statewide EWS and over 20 states are actively supporting the development of EWS in local education agencies (Feathers, 2023).

1.4 Recent EWS Applications

There has been a surge of artificial intelligence (AI) methods used in education settings, particularly in the development of EWS. The allure of AI, driven by its ability to analyze vast amounts of administrative data and provide tailored predictions, has fueled its integration into EWS settings. In EWS settings, there is an increased use of machine learning methods – a subset of AI methods that focus on building prediction models to flag individuals at risk of an adverse outcome. However, recent applications in the K-12 sector revealed deep flaws with these systems, specifically with regard to equity and interpretability.

The first challenge is that EWS may erroneously flag students as at-risk, especially for those from marginalized backgrounds. A notable example is Wisconsin's statewide EWS, which has been under significant scrutiny. A recent audit revealed that students from racial subgroups were disproportionately mislabeled at higher rates than their peers, where over 75 percent of

Hispanic and Black students were misclassified as high-risk. Furthermore, the audit found that the system perpetuated bias in how teachers perceived students of color (Feathers, 2023). Wisconsin's failure to provide equitable and accurate predictions underscores a critical challenge: the potential for EWS to exacerbate existing structural disparities in educational outcomes.

The second major challenge of EWS arises from interpretability of findings generated by models that use machine learning approaches. Models that use such statistical approaches are often coined "black box" models because their decision-making processes are not easily understandable or explainable to users, policymakers, or even the developers themselves. The lack of interpretability makes it difficult to trust and validate the model's predictions, as stakeholders may struggle to understand why certain students are flagged as at risk or why others are not (Prinsloo, 2020; Nussberger et al., 2022; Purcell, 2024). This opacity also complicates efforts to ensure fairness and avoid reinforcing biases, as users may be unable to assess whether the statistical model is inadvertently favoring or disadvantaging certain subgroups.

As a result, the effective use of predictive modeling approaches from AI requires not only strong predictive accuracy but also enhanced transparency, explainability, and interpretability to foster trust and ensure the system's outcomes are both valid and equitable. This is a critical juncture for schools and systems that are either looking to develop an EWS or update their existing system. An EWS has the potential to empower schools and systems with the information needed to take meaningful action. At the same time, there is a need to assess the fairness of prediction models to ensure that model findings are not perpetuating bias and that model findings can be interpreted by a non-technical audience.

1.5 At a Glance

This dissertation employs machine learning algorithms to forecast students at risk of dropping out of high school. I draw data from North Carolina’s statewide longitudinal data system, NCERDC, to develop a prediction model that uses student records from middle grades (grades 6 through 8) as predictors to flag students at risk of dropping out in 9th or 10th grade.[1] The dissertation investigates three key research objectives: (1) compare the predictive accuracy of supervised learning algorithms to traditional methods (i.e., logistic regression) in predicting early high school withdrawal, and examining if addressing imbalanced training data improves model accuracy; (2) conduct a fairness analysis to ensure that models provide equitable predictions across sensitive student attributes; and (3) interpret model findings to understand the underlying factors contributing to high school withdrawal.

I include predictors that capture school engagement through attendance (absence rate and chronic absence), behavior (disciplinary infractions), and coursework (math and reading proficiency). This follows literature recommendations of using attendance, behavior, and course performance, also known as ABC indicators (Balfanz et al., 2007; Mac Iver, 2010; Allensworth & Clark, 2019).

To answer the first research question I compare the prediction accuracy of models that employ the following algorithms: logistic regression, lasso regression, ridge regression, random forests, and extreme gradient boosting (XGBoost). I use these algorithms to train models that rely on imbalanced data, or data where the minority class – number of students who exited early – are largely outnumbered by the majority class, or students who did not exit early. I later

[1] In North Carolina, the legal age for exiting school is 16, which generally aligns with a student's 11th grade year.

address class imbalance using two resampling techniques: oversampling and undersampling the training data. I apply the Synthetic Minority Oversampling Technique (SMOTE) to oversample instances of the minority class, generating synthetic (i.e., artificial) observations to enhance its representation in the dataset (Chawla et al., 2019). Alternatively, I employ undersampling by reducing the number of observations in the majority class to match the size of the minority class. Finally, I evaluate whether training models on either type of resampled data leads to improved prediction accuracy. The subsequent research questions rely on the fifteen models built in the first question.

The second research question assesses algorithmic fairness among the fifteen models using two metrics: the Absolute Between-ROC Area (ABROCA) metric developed by Gardner et al. (2019) and the equalized odds metric developed by Hardt et al. (2016). I examine sensitive attributes by comparing model performance of student subgroups based on gender, race/ethnicity, disability status, financial hardship, and English language proficiency.

The final question interprets a subset of models to gain a deeper understanding of the factors associated with early withdrawal. I assess the consistency of key predictors identified across the models and extract relevant features using methods appropriate for each algorithm. For the regression models (logistic regression, lasso regression, and regression), this involves analyzing non-zero coefficients that exceed a predefined threshold. For ensemble methods, I examine feature importance plots and utilize SHAP (SHapley Additive exPlanations) values to identify significant predictors of early exit. I conclude the analysis with a discussion of tradeoffs associated with the use of complex "black box" models and how to make findings interpretable and actionable for practitioners and stakeholders.

1.6 Study contributions

This dissertation is a conceptual replication of Knowles (2015) and extends Knowles' work by predicting high school exit in a different setting and context. This study advances the field of dropout prediction in four key ways. First, this study is among the few known dropout prediction studies that examine the timing of high school exit to identify an optimal period for delivering targeted interventions and support services. By focusing on the temporal aspects of student withdrawal, this research seeks to inform the strategic allocation of resources and enhance the effectiveness of early intervention efforts.

Second, this study is the only known one to date that evaluates algorithmic fairness in a U.S. context. While there is growing recognition of the potential for algorithmic bias in models trained on historically biased data, most existing efforts to detect and address such biases have taken place in international education settings. This highlights a gap in the application of equitable model predictions within the U.S. educational landscape, where these advancements in data science have yet to be fully realized. Thus, this study contributes to filling a critical gap by introducing how algorithmic fairness can be manifested in the U.S., offering insights that could inform more equitable and transparent decision-making processes in educational systems.

The third contribution of this study is that it tackles class imbalance – an issue often overlooked challenge in U.S. settings, despite its widespread prevalence when predicting adverse outcomes. While class imbalance is a well-recognized issue in many domains, including healthcare and criminal justice, it has received relatively little attention in educational research, particularly in the context of dropout prediction models. By addressing this gap, this study highlights the importance of ensuring balanced representation in predictive models, ultimately contributing to more accurate decision-making in educational policy and practice.

The fourth contribution of this study is its improvement of model generalizability, or the likelihood that the model achieves high predictive accuracy across diverse populations, timeframes, and settings. This study enhances generalizability by adopting a novel methodological approach: it uses data from one student group to establish associations between middle school engagement and high school exit (training data) and then evaluates the model's predictive accuracy on a distinct, separate student population (test data). This approach differs from the conventional method of training models by randomly partitioning data from a single population into training and test subsets. By leveraging a cross-population validation strategy, this study mitigates the risk of overfitting to a specific cohort, ensuring that the model remains robust and applicable across varying demographic and contextual settings. This approach strengthens the model's external validity, making it more useful for real-world applications in diverse educational settings.

In summary, this dissertation seeks to bridge the gap between data science and education research – areas that have historically operated in relative isolation. While the field of data science is actively advancing efforts to enhance data literacy, governance practices, and the ethical use of predictive analytics, these advancements have not yet been widely adopted in educational settings, especially in the U.S. This disconnect has led to missed opportunities for leveraging the full potential of data-driven insights to inform educational policy and practice. By addressing this gap, the dissertation aims to foster a more integrated approach, encouraging the incorporation of best practices from data science to promote more effective, equitable, and transparent decision-making in education.

1.7 Dissertation overview

The goal of this chapter is to establish the context for why high school completion is a critical issue, outline policy and data driven efforts aimed at improving graduation rates, and discuss the challenges associated with these efforts. It also explores the barriers to predict student disengagement such as regards to equity and ease of interpretation by stakeholders. The chapter concludes with a brief overview of the research questions, methods, and the contribution of this work. The remainder of this dissertation is organized as follows: the second chapter is a literature review that situates this work in the intersection of data science and education, establishes the theoretical concepts and framework that underpin this work, and reviews the methodologies used in prior studies. The third chapter covers the research design and provides a detailed overview of the data, sample, and methods employed to address the proposed research questions. The fourth chapter presents the results for the study. The fifth and final chapter is a discussion that discusses the study’s limitations, implications, and suggestions for future research.

CHAPTER 2: LITERATURE REVIEW

Chapter Introduction

The goal of this literature review is to offer a comprehensive and critical overview of existing research on K-12 dropout prediction. This chapter serves several key functions. First, I review the methodologies employed in previous studies and discuss their limitations, highlighting areas for future research to address or improve. Second, I review the theoretical concepts and frameworks that form the foundation of my study. Third, I position my research at the intersection of education and data science, synthesizing relevant scholarship to show how my work extends, fills gaps in, or challenges the current body of knowledge. This includes a discussion of key issues, debates, and trends in the field. Finally, I argue for the significance of my dissertation and its potential contributions to advancing the field. This chapter situates the relevant background knowledge needed to understand my decision to explore the following research questions:

> **1:** How does the prediction accuracy of supervised learning algorithms to predict early exit from high school compare to that of traditional models (i.e., logistic regression)? Additionally, how does model performance vary when resampling techniques are used to address class imbalance?
> **2:** To what extent does each model provide fair predictions across sensitive student attributes such as gender, race/ethnicity, disability status, financial hardship, and English proficiency?
> **3:** What are the most salient predictors of students who exited high school in 9th or 10th grade?

This chapter is organized as follows. First, I summarize existing research on the relationship between student disengagement and high school dropout rates. I then survey the effectiveness of dropout prevention efforts, with a focus on the application of early warning systems. Following this, I describe the theory and standardized methodology for constructing predictive models in data science. Later, I explore the use of machine learning in education, reviewing two common machine learning techniques in dropout prediction – regression models and tree-based models – and the trade-offs involved in using these methods over traditional prediction approaches. Next, I discuss the challenges of applying machine learning to dropout prediction, including issues related to model accuracy and interpretability, generalizability, class imbalance, and potential student discrimination in model outcomes. I also delve into the debate surrounding the inclusion of demographic information in predictive modeling, presenting key arguments from both perspectives. The subsequent section reviews recent advancements in measuring student discrimination in prediction models, often known as algorithmic fairness. I highlight four methods for assessing algorithmic fairness fairness: group differences in performance, the Absolute Between-ROC Area (ABROCA) metric, equalized odds, and demographic parity, discussing the applicability of each in different contexts. Finally, I conclude the chapter by outlining criteria for developing more consistent, reliable, and coordinated early warning systems and justifying the need for my work within the broader field of dropout prediction.

2.1 Factors associated with high school exit

This subsection synthesizes literature that uncovers early warning signs of dropping out and highlights research gaps that remain in this area of research.

2.1.1 Attendance, behavior, and coursework ("ABC") predictors

A student's decision to withdraw from high school can be connected to factors across four domains: community, school, family, and individual (Hammond et al., 2007; Rumberger & Lin, 2008; Shargel, 2013). Empirical research that aims to understand the causes and consequences of dropping out of high school are grounded in two theories: individual perspective and institutional perspective. The individual perspective emphasizes that the decision to drop out depends on individual decision made by the student and is shaped by the individual's school engagement, attitudes, experiences and beliefs. In contrast, the institutional perspective argues that contextual, external factors drive students to permanently discontinue their schooling journey, such as family, school, and community (Rumberger, 2011; Balfanz, 2013; Doll et al., 2013).

This dissertation focuses on the theoretical construct of student engagement individual perspective and specifically, the relationship between student-education engagement and dropping out of high school. Student-education engagement refers to a student's K-12 schooling experience that encompasses behavioral, cognitive, and emotional engagement (Gleason & Dynarski, 2002; Fredricks et al., 2004). Recent literature narrows the scope of student education engagement with a base set of categories: attendance, behavior, and course performance, also known as ABC indicators (Frazelle Nagel, 2015; Allensworth Easton, 2007; Mac Iver, 2010; Balfanz et al., 2007; Allensworth & Clark, 2019).

Prior research has established that early disengagement from school increases the likelihood of dropping out (Balfanz & Bryne 2018; Rumberger, 2020; Casillas et al. 2012). Table 1 presents an overview of studies that have examined ABC engagement as a predictor for early exit. These studies share similar a similar conclusion that grades are associated with on-time grade promotion and high school graduation (Jackson, 2018). Dropout flags focusing on GPA were some of the most accurate dropout flags across the literature (Bowers et al., 2013) and failing grades were strongly predictive of dropping out (Bowers & Sprott, 2012a; Bowers 2010b; Balfanz et al., 2007; Gubbels et al. 2019). This knowledge is ubiquitous in K-12 settings; schools and school systems often rely on grades to identify students in need of additional support and services.

Most studies examining risk factors tend to focus primarily on course performance, or "C" indicators, such as student grades, test scores, and the types of courses taken. However, research that incorporates all three categories of ABC indicators (attendance, behavior, and course performance) in the same prediction model is relatively rare and tends to be concentrated in U.S. K-12 settings (Knowles, 2015; Sorenson, 2019; Sansone, 2019). This is unsurprising, given the limited availability of longitudinal education data systems that integrate such comprehensive information, particularly in post-secondary settings or in education systems outside the United States.

Table 1: Literature summary of ABC predictors

Type of student disengagement	**Factors strongly predictive of high school exit**
Attendance	- Moderate absences in 9th grade (Allensworth & Easton, 2007) - Chronic absenteeism in middle grades (Allensworth et al., 2014; Seeskin et al., 2022) - Middle school attendance (Kieffer et al., 2011) - 9th grade chronic absenteeism (Mac Iver & Messel, 2013)
Behavior	- Out-of-school suspensions (Balfanz et al., 2014)
Coursework	- 4th grade math and reading scores (Kieffer et al., 2011) - 9th grade course failure (Neild & Balfanz, 2006; Mac Iver & Messel, 2013) - 9th grade GPA (Allensworth & Easton, 2007; Bowers & Sprott, 2012b; Allensworth, 2013; Easton et al., 2017) - Teacher assigned grades in 9th and 10th grade (Bowers & Sprott, 2012b)

2.1.2 Gaps in student engagement literature

Although considerable attention has been dedicated to understanding and supporting at-risk students, there are several dimensions of student engagement that are not sufficiently explored. Specifically, areas such as chronic absenteeism, the age of students, rurality, and the timing of student exit from educational systems warrant further investigation.

Recent empirical work has demonstrated a stronger association between poor school attendance and the decision to drop out of school. Specifically, students who are chronically absent – typically defined as students who are absent at least 10 percent of the time – experience lower academic achievement, are more likely to repeat a grade, are more likely to face disciplinary infractions, and are more likely to drop out (Balfanz & Byrnes, 2018; Gottfried, 2017; Humm et al., 2018). Chronic absenteeism in schools has risen sharply in the years following the COVID-19 pandemic, revealing a significant shift in student engagement and attendance patterns. In the 2018-2019 school year, approximately 15 percent of K-12 students across the U.S. were classified as chronically absent. However, by the 2021-2022 school year, that figure had more than doubled, with 30 percent of students falling into this category (White House, 2023). In response to this growing challenge, schools and districts have developed early warning systems that identify students at risk of being chronically absent (Wu & Weiland, 2024). Despite these growing concerns, dropout prediction studies to date have not prioritized chronic absenteeism as a predecessor for dropping out of high school.

Despite the present education policies that impose age restrictions for school entry, there are few known prediction models that examine age as a predictor for early exit. There is previous economic work that exploited age at school entry to examine elementary and middle school engagement. Scholars find that on average, K-12 students whose birth date is just before the school entry cutoff exhibit lower academic performance (Dee and Sievertsen, 2018; Dobkin and Ferreira, 2010; Oliveira and Duque, 2019; McEwan and Shapiro, 2008; Elder and Lubotsky, 2009; Crawford et al., 2014). Among the body of studies that predict high school dropout, only Sorensen (2019) includes a demographic factor if a student is "old-for-grade" but did not describe the parameters describing the age indicator.

Rurality merits important consideration in education research. Studies suggest that there is a negative association between geographic isolation and academic achievement (Drescher and Torrance, 2022; Echazarra and Radinger, 2019; Faggian et al., 2017). Although population density is frequently used when disaggregating high school graduation rates, most studies omit rurality as a predictor for high school dropout.

Although much research has focused on the general effects of dropping out, there is limited exploration into whether early dropout – during 9th or 10th grade – has more profound long-term effects on academic, social, and economic outcomes compared to dropping out in the later years like 11th or 12th grade. The consequences of early dropout may be more severe, given the formative nature of the earlier high school years in terms of academic foundation, social development, and future career prospects. However, the absence of a clear, evidence-based understanding of dropout timing hampers the ability to design targeted interventions and policies that could more effectively address the needs of at-risk students at various stages of their high school education.

In summary, these factors, while integral to the broader landscape of academic achievement, have yet to be examined in dropout prediction studies. The inclusion of these aspects of student engagement in prediction models could provide deeper insights into the complex dynamics that shape the decisions of at-risk students.

2.2 Dropout prediction and prevention efforts

2.2.1 Efficacy of early warning systems

Despite efforts to predict student disengagement using historical data, not much is known about which indicators are the most predictive and how to translate it to actionable evidence that schools and school systems can utilize (Bowers, 2021). This subsection highlights research efforts to develop early warning systems and the effectiveness of such applications.

The Consortium on Chicago School Research (CCSR) has spearheaded the movement of school districts in developing early warning systems. Based on their research in the Chicago Public Schools district, CCSR developed a freshman on-track indicator that flags students as "off track" based on course credits and course failures in 9th grade. The on-track indicator has proven effective in identifying at-risk students in Chicago Public Schools (Allensworth & Easton, 2007; Allensworth, 2013). It has since been adopted by several other districts, which have reported similar positive outcomes. An overview of these applications is presented in Table 2.

CCSR has since developed an additional on-track indicator for students in grades 3 through 8. This indicator, called the 3-8 OnTrack metric*,* is designed to help elementary and middle schools better prepare students for high school. In 2019, Chicago Public Schools incorporated this indicator to account for 10 percent elementary principals' evaluations. A recent study found that the 3-8 OnTrack metric was successful in identifying at-risk students, with low GPA in middle grades to be the most predictive of dropping out (Seeskin et al., 2022). This example highlights the growing interest of districts in understanding and preventing student disengagement.

Table 2: Applications of early warning systems

Study	Setting	Approach	Findings
Allensworth & Easton (2007) Allensworth (2013)	Chicago Public Schools	Freshman on-track indicator	- Indicators were more predictive of HSG than students' background characteristics or middle school test scores
Norbury et al. (2012)	2 urban Midwest districts	Freshman on-track indicator	- On-track indicator was significantly predictive of HSG
Crofton & Neild (2018)	School District of Philadelphia	Freshman on-track indicator	- Failing a course or missing a required course

			- Female students were more often on track than male students
Seeskin et al. (2022)	Chicago Public Schools	Grades 3-8 OnTrack metric	- Strong interaction between being chronically absent & maintaining a 3.0 GPA
Perdomo et al. (2023)	Wisconsin	Dropout early warning system (DEWS)	- Prediction system accurately sorted students by dropout risk - Low implementation in districts - System led to little or no increases in HSG

Notes: HSG is short for high school graduation; GPA is short for grade point average.

A study by Canbolat (2024) stands out as one of the few non-experimental evaluations of an early warning system. In its examination of a system that flags students at risk of being chronically absent, Canbolat found that early identification reduced chronic absenteeism among students who did not receive free- or reduced-price lunch (FRPL) but found no effects among FRPL students. Moreover, a recent evaluation of the Wisconsin statewide early warning system found that early identification, at best, improved high school graduation by single digits (Perdomo et al., 2023). There are two studies to date that have randomly assigned early warning systems to a comparison group. Both studies found the early warning systems reduced chronic absenteeism but had no effects on student suspensions, low GPA, and course credits earned (Faria et al., 2017; Mac Iver et al., 2019). Despite the expansion of studies that evaluate the efficacy of early warning systems and the role of CCSR in providing frameworks for districts to use, further research is essential to fully understand the long-term effectiveness of these systems. Additionally, continued exploration is needed to identify potential improvements as these systems evolve in response to the challenges presented by the pandemic.

2.2.2 Efforts to reengage at-risk students

Aside from the development of early warning systems and federal accountability laws, schools and school systems have made many efforts to improve high school completion. Common reengagement strategies that are typically provided at the high school level include raising the compulsory school age, providing double-dose algebra, and offering credit recovery programs. Table 3 provides a summary of these approaches and their respective ties to empirical research.

Table 3: High school reengagement efforts

Reengagement strategy	Theory of Change	Empirical Evidence
Raising the compulsory school age	Increases the length of education journeys for students who would have otherwise discontinued their schooling	- Increases educational attainment (Oreopoulos, 2009; Cabus & Witte, 2011) - Increases earnings earnings (Angrist & Krueger, 1991) - No changes in high school graduation rates (Landis et al., 2010; Mackey et al. 2013; Raimondi & Vergolini, 2019).
Double-dose algebra	Provides struggling students with twice the amount of algebra instruction to build a strong foundation in algebra	- No short-term changes in 9th grade algebra failure rates (Nomi & Allensworth, 2013) - Substantial and positive long-term impacts of double-dose algebra on high school completion rates, college entrance exam scores, and college enrollment rates (Cortes & Goodman, 2014; Cortes et al., 2013)
Credit recovery programs	Help high school students retake courses and earn credits toward graduation	- Increased the likelihood of graduating high school, especially for economically disadvantaged and Hispanic students (Viano & Henry, 2024; Nomi et al., 2021) - No effects of online credit recovery programs in the short term (Heppen et al., 2017; Rickles et al., 2018)

It is crucial to emphasize that efforts to reengage students are generally provided after at-risk students have been identified. However, proactive measures, such as the development of predictive models and early warning systems, are essential predecessors to forecast which students are at risk of dropping out.

2.2.3 Why develop an early warning system

An early warning system is a structured approach designed to identify individuals at risk of negative outcomes. In K-12 settings, early warning systems provide evidence to enable timely, targeted interventions to support at-risk students. This proactive approach not only prevents dropout but also optimizes resource efficiency.

When students drop out, schools and districts lose funding that is tied to student enrollment. There are also the long-term societal costs associated with lower education levels, including reduced workforce productivity and higher social service needs. By investing in early warning systems, schools can intervene before students disengage, thereby reducing the risk of dropout and ensuring that resources are allocated more efficiently. In the long run, preventing dropouts through early identification is far less expensive than the broader financial and societal costs that arise when students discontinue schooling and do not earn a high school diploma.

2.3 Building a Prediction Model

The first two sections of this chapter focused on the non-technical dimensions of student disengagement by discussing what is known about students who drop out of high school, and what efforts schools and districts have made to proactively support these at-risk students. The remainder of this chapter largely focuses on methodological literature and technical approaches to develop, improve, and evaluate the robustness of early warning systems.

This section introduces the first technical aspect of developing an early warning system: the process of building a prediction model. This section covers four essential steps that are critical to building a robust dropout prediction model: applying a statistical approach to estimate the probability that a student will drop out of high school, which is typically achieved through a logistic regression; identifying a threshold or cutoff value that sets the minimum predicted probability needed to classify a student as being at risk for "early exit"; understanding techniques to improve model generalizability using cross-validation; and lastly, discussing appropriate metrics to evaluate model performance. These steps collectively form the foundation for developing a predictive model that can provide actionable insights for early intervention and dropout prevention.

2.3.1 Logistic regression

Logistic regression is a statistical method commonly used for classification tasks, where the goal is to predict the probability of a particular event occurring. A specific type of logistic regression, a binary logistic regression, focuses on situations where the dependent variable is binary with two possible outcomes: the occurrence or non-occurrence of the event. When the dependent variable takes the values of either '0' or '1', the logistic regression estimates the probability of each observation belonging to one of the two categories, with predicted probability values that range from 0 to 1.

Logistic regression model relies on the following assumptions: including independent observations, no perfect multicollinearity and linearity.

1. Linearity. The relationship between each continuous predictor variable and the log odds of the dependent variable should be linear.
2. There are no outliers, or extreme observations, in the data.
3. No perfect multicollinearity, or when independent predictors are highly correlated with each other.

To satisfy the linearity assumption, the logistic regression output values are transformed from a probability to a logit function, where the outcome is in a log odds unit. For the probability that an event will occur, p, the probability that it will not occur is (1-p). Taking the log odds of the ratio, $Log\left(\frac{p}{(1-p)}\right)$, transforms the outcome distribution from [0,1] to a full range of numbers, (-∞, ∞) (Angrist & Pischke, 2014; Wooldridge, 2019).

This approach is popular in exploratory data analysis (often called descriptive analysis) and remains the standard approach for predicting binary outcomes in education research. It is important to note that logistic regression is one of the many methods that can be used to build a prediction model. A logistic regression typically encompasses the formal specification:

$$Log\left(\frac{p}{(1-p)}\right) = \beta_0 + \beta_1\beta X_1 + \beta_2 X_2 + \dots + \beta_k X_k$$

where there are K predictors with β_k corresponding regression coefficients that are also reported as log odds. The outcome is the expected log of the odds that the student will exit early. To better understand log odds, consider an example of a logistic regression with one predictor, 6th grade standardized math score. A regression coefficient of -2.66 can be interpreted as a one standard deviation increase in math score decreases the odds that a student will exit early by a factor of 2.66.

To determine if a coefficient β_k is statistically different from zero, researchers rely on Wald (z) confidence intervals of and z tests. Similar to F-tests in other regression frameworks, one can conduct the likelihood ratio test to see if a collective set of predictors are not needed.

There is flexibility in determining how many predictors or features the model should have. This can follow a forward selection method where the initial model is parsimonious (with few or no predictors) and later models gradually include more predictors. Conversely, backward selection inputs the largest pool of predictors in the initial model and excludes predictors in later models (James et al., 2021; Hastie et al., 2009).

Regression model fit is often evaluated on its sum of squared errors, also known as the residual sum of squares (RSS) (Kuhn & Johnson, 2013). RSS measures the variance in the residuals (error term). RSS is represented by the formulation:

$$RSS = \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

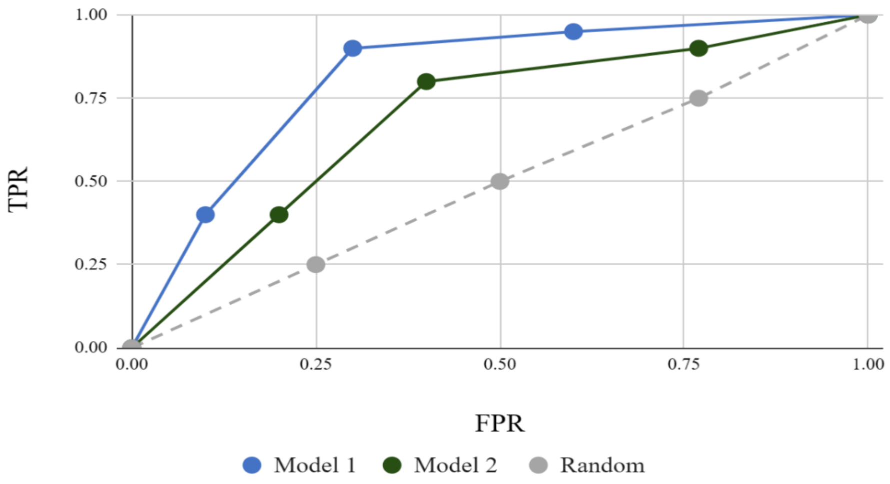
where observation *i*'s difference between observed values and estimated values across *p* predictors are squared and summed across all (*n*) observations. The smaller the RSS, the better model fit. Although a model with an RSS of zero indicates that the predictors perfectly predict the outcome, it can signal two potential issues: overfitting on data and multicollinearity, or correlation among predictors (Fahrmeir et al. 2022; Kuhn & Johnson, 2013). In cases where the RSS is close to zero, researchers rely on approaches that either reduce the number of model predictors or adjust the RSS function to prevent overfitting. Alternative approaches to logistic regression will be extensively discussed in 2.4.

2.3.2 Identifying a decision threshold

This subsection focuses on a tuning parameter known as the decision threshold (i.e., cut-off score), which determines how the model assigns outcomes to each observation. While the primary goal of a prediction model is to estimate the likelihood of an event occurring, it is rare for the model to produce predicted probabilities that are exactly 0 or 1.0. In the context of prediction models, the decision threshold is the optimal cutoff point that differentiates students at risk of early exit from those who are not. Students with predicted probabilities above this threshold would be classified as "exited early," while those with probabilities below it would be categorized as "did not exit early."

A receiver operator characteristic (ROC) curve is regarded as the gold standard for measuring the performance of a probability threshold. Originating in signal detection theory, a ROC curve visualizes the tradeoff between the true positive rate (TPR) – the proportion of instances where the model correctly predicts early exit – and false positive rate (FPR) – the proportion of instances where the model incorrectly predicts early exit. This is done across all classification thresholds (Swets, 1988; Swets et al., 2000; Streiner & Cairney, 2007). Figure 1 presents a hypothetical ROC curve that provides curves for two models, Model 1 and Model 2.The default probability threshold (if not specified in the model) is 0.5; selecting a more relaxed cutoff beyond 0.5 would result in a higher true positive rate but comes with a penalty of a higher false positive rate.

Figure 1: Hypothetical ROC curves



As seen in Figure 1, the default probability threshold of 0.5 is represented with a diagonal line. This line indicates that the model is essentially guessing its predictions and holds no predictive power. The most desirable performance is when the model exhibits a high TPR and low FPR, which is typically achieved when the curve is to the top left corner. In comparison to Model 2, Model 1 demonstrates a steeper curve at all points, suggesting that Model 1 consistently yields a higher number of true positives across the full range of false positive thresholds. This indicates that Model 1 is more effective in correctly identifying positive instances, making it a stronger model in terms of its model performance.

The area under the ROC curve, called the area under curve (AUC), measures a model's ability to discriminate between positive and negative instances across all classification thresholds. An AUC score of 0.5 indicates that the model is no better than random guessing, while an AUC of 1.0 indicates perfect classification. The AUC can be interpreted as the probability that the model will correctly assign a higher score to a randomly chosen student who exits early compared to a student who does not exit early (Bowers & Zhou, 2019; Kroese et al., 2019; Nahm, 2021). For example, an AUC value of 0.75 means indicates that there is a 75 percent probability that the model will correctly identify a student who exits early, as compared to a student who does not. It id important to note that identifying the optimal threshold is not specific to the method discussed so far (i.e., logistic regression) but is applicable for any prediction approach where the outcome is known or observed.

2.3.3 Cross-validation

Cross-validation is an approach to improve model performance. This addresses the issue of overfitting, or when a prediction model performs poorly on unseen, new data (Garrett et al., 2022). Cross-validation is a resampling practice where the data are randomly split into a training set and a test set. The model learns from the training data by identifying patterns, trends, and relationships between the model's predictors and the outcome of interest. It then applies this learned knowledge to generate predictions for each new observation, allowing it to forecast the likelihood of an outcome. The trained model is then evaluated with this new, unseen, data that is referred to as test data (James et al., 2023; Garrett et al., 2022). There are no standardized guidelines on how data should be split, though the majority of the data are typically allocated for the training data to provide the model with a larger dataset for learning. Some researchers follow a more balanced approach of splitting the data to a 60% training subset and 40% testing subset, and others may prefer a 80% training and 20% testing subset. Researchers also can use two sets of observations (one as train data, the other as test data) that differ on one or more characteristics.

There are variations of cross-validation, one of which is *k*-fold cross validation. *K*-fold cross-validation is applicable in settings where a researcher may produce multiple models that each rely on a unique subset of data that are eventually aggregated into one model. This strategy involves the following steps: choosing *k* folds; splitting the data into *k* equal sets with the $\frac{1}{k}$ of the data serves as test data and the remainder as train data; calculating the mean squared error (MSE) within each fold for each model (Kuhn & Johnson, 2013; Fahrmeir et al. 2022).

2.3.4 Evaluating model performance

The approach for evaluating the performance of models with binary outcomes is straightforward: it compares the accuracy of predicted outcomes against the actual observed outcomes. This method provides a clear assessment of how well the model can correctly classify instances, offering valuable insights into its overall effectiveness.

The standard metric of model performance is the accuracy rate, or the proportion of instances in the test data that were correctly classified by the model (Hung et al., 2017; James et al., 2021; Bishop, 2024). An intuitive approach to understanding which instances were correctly classified is with a confusion matrix that compares predicted and true counts for each outcome level. Table 4 provides a hypothetical confusion matrix for a binary classifier (a 2 by 2 matrix) that compares predicted labels (i.e., predicted outcomes) with true labels. In this example, the outcome of interest is early exit, where students who exited early are the positive class, students who did not exit early form the negative class. The matrix disaggregates instances into four groups - true negatives (TN), true positives (TP), false negatives (FN), and false positives (FP).

Table 4: Hypothetical confusion matrix

		TRUE LABELS	
		Exited early	Did not exit early
PREDICTED LABELS	Exited early	TP	FP
	Did not exit early	FN	TN

Notes: TP stands for true positive; FN stands for false negative; FP stands for false positive; and TN stands for true negative.

The accuracy rate can be found by calculating:

$$\frac{TP + TN}{(TP + TN + FP + FN)} \times 100$$

A key limitation of the accuracy rate is that it does not provide insight into how well the model performs for each class label. Specifically, it fails to reveal the model's effectiveness in classifying both the positive and negative classes. There are additional metrics that can be extracted from the confusion matrix. These include precision, which measures the accuracy of positive predictions; recall or sensitivity, which reflects the true positive rate; and specificity, which indicates the true negative rate. Table 5 presents the formulas used to calculate these additional metrics (Kroese et al., 2019; James et al., 2023).

Table 5: Formulas to calculate performance metrics

Metric	**Formula**
Accuracy	$\frac{TP + TN}{(TP + TN + FP + FN)}$
Recall or sensitivity	$\frac{TP}{(TP + FN)}$

Specificity	$\frac{TN}{(FP + TN)}$

Notes: TP stands for true positive; FN stands for false negative; FP stands for false positive; and TN stands for true negative.

2.3.5 Summary of process to build prediction models

This subsection focuses on diagnostic measures and parameters used for building a robust prediction model. So far, 2.3 covered the logistic regression as a standard approach to predict a binary outcome, selection of a decision threshold, cross-validation, and metrics to evaluate model performance. These guidelines are rooted in prediction literature and are applicable to settings where outcomes for each observation are known. The observed outcomes are later compared with predicted outcomes to measure model accuracy. Based on prior literature, the standard approach to building a prediction system can be summarized as a similar process to below:

1. Retrieve data that includes observed outcome for each unit.
2. Split the data into a training set and testing set.
3. Select characteristics or predictors to include in the model using either forward selection or backward selection.
4. Identify the optimal decision threshold using a receiver operator characteristic (ROC) curve.
5. Build a prediction model using training data.
6. Assess model accuracy by running the model on test data.
7. Compare the predicted probabilities (from test data) with its observed outcomes to calculate metrics such as accuracy, sensitivity, and specificity.
8. If the model needs to be improved, revisit forward selection or backward selection and repeat parts 2 through 7.

It is important to note steps to building a prediction model can be applied to models that apply logistic regression or machine learning methods - a set of quantitative methods that will be described in the next section.

2.4 Machine learning

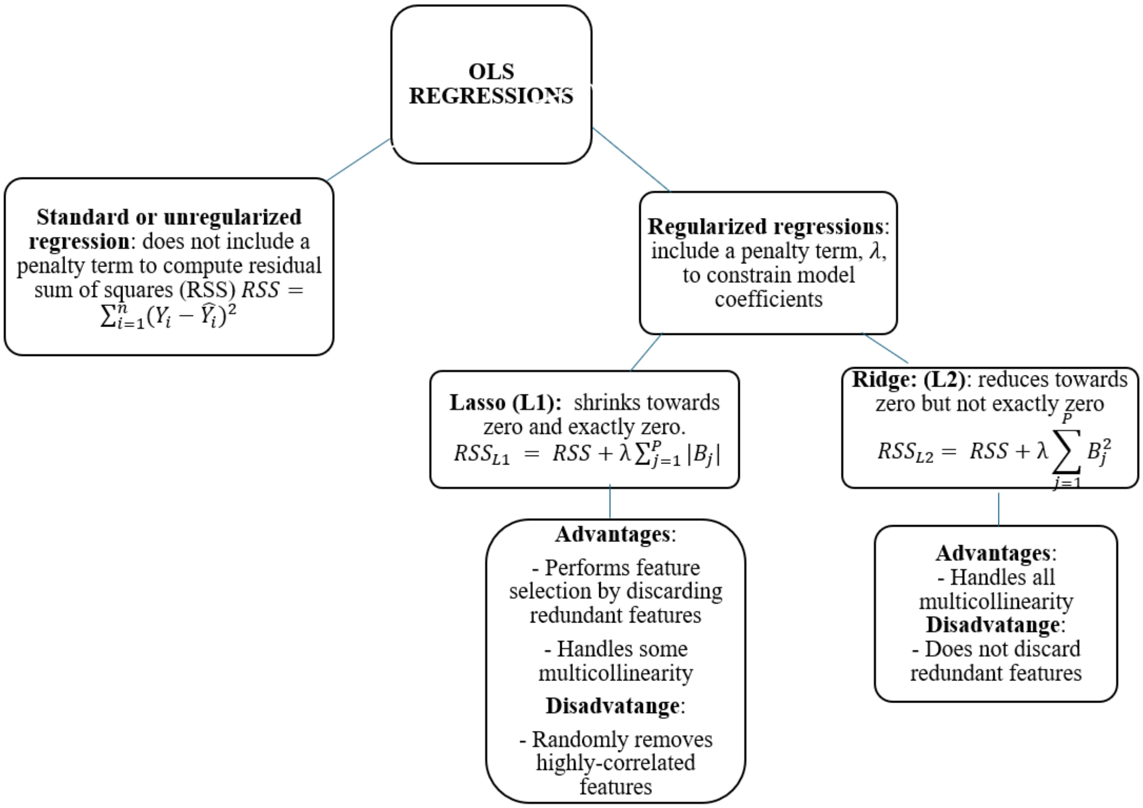
2.4.1 What is machine learning?

Machine learning is a subset of artificial intelligence methods that learn from historical data to make predictions. Machine methods, or algorithms, examine characteristics to “learn” about the relationship between predictors, or model features, and the outcome of interest (Bishop, 2024). There are two categories of machine learning algorithms: supervised and unsupervised. Supervised methods rely on labeled data, or data where the outcome of interest is already known. The algorithm produces a model to predict the outcome and then compares predicted outcomes with actual outcomes. In contrast, unsupervised methods employ unlabeled data, or raw data where the outcome is unknown. The goal of unsupervised learning is to draw conclusions in given data (Jordan & Mitchell, 2015). This section briefly describes the anatomy of select machine learning algorithms and compares it to that of logistic regression.

2.4.2 Regression approaches

As discussed in 2.3.1, logistic regressions – regressions where the outcome of interest is binary – rely on maximum likelihood estimation (MLE) assumptions. The residual sum of squares (RSS) function is a metric that indicates the model’s discrepancy between predicted probabilities and true outcomes. Drawbacks to this “unregularized” regression is the inability to detect multicollinearity (i.e., when two or more model features are highly correlated to each other) and overfitting. For this reason, there are alternative to unregularized regressions that aim to address these concerns. This subset of regression methods, called regularized regressions, adds a penalty term to the RSS function. This penalty helps shrink the coefficients toward zero, addressing issues like multicollinearity and overfitting (Hastie et al., 2009; Friedman, 2023). Figure 2 provides a graphical overview of 2 common types of regularized regression, each differing in how penalty term is calculated. The figure also highlights the key advantages and drawbacks of each regression method, offering a concise comparison to help clarify their respective strengths and limitations.

Figure 2: Comparison of unregularized versus regularized regressions


OLS REGRESSIONS
Standard or unregularized regression: does not include a penalty term to compute residual sum of squares (RSS) $RSS = \sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2$
Regularized regressions: include a penalty term, λ, to constrain model coefficients
Lasso (L1): shrinks towards zero and exactly zero. $RSS_{L1} = RSS + \lambda\sum_{j=1}^{P}|B_j|$
Ridge: (L2): reduces towards zero but not exactly zero $RSS_{L2} = RSS + \lambda\sum_{j=1}^{P} B_j^2$
Advantages:
- Performs feature selection by discarding redundant features
- Handles some multicollinearity
Disadvatange:
- Randomly removes highly-correlated features
Advantages:
- Handles all multicollinearity
Disadvatange:
- Does not discard redundant features

2.4.4 Tree-based methods

Decision tree is an algorithm that uses a tree-like structure to make predictions by sequentially asking questions based on model features, or predictors it received. The algorithm follows a straightforward "if-then" rule system, progressively asking questions that divide the data into smaller and groups based on different features. While decision trees are easy to interpret and understand, they are prone to overfitting, which can hinder the model's ability to generalize on new, unseen data (Brown, 2017; Bishop, 2024). For this reason, rather than building a single decision tree, model performance is generally improved when multiple models are aggregated to create a final model, also known as an ensemble model. This subsection focuses on ensemble learning in the context of tree-based methods, or methods that use a decision tree to represent how different predictors can be used to predict an outcome.

The motivation behind averaging many models is that it reduces the prediction error from a single model, leading to higher overall accuracy than relying on a single model (Breiman, 2001; Hastie et al., 2009; Brown, 2017). Two common approaches to ensemble individual models are bagging (short for bootstrap aggregation) and boosting are widely used approaches to ensemble individual models. Table 6 provides a high-level comparison of bagging and boosting.

Table 6: Bagging versus boosting approaches

	Bagging	**Boosting**
Creation	Breiman (1996)	Freund & Schapire (1999)
Function	Relies on subsamples of data randomly drawn from sample with replacements. Each model is trained on a subsample of data	Builds an individual model with all training data and follows a reiterative correction process until a predefined number of iterations is reached or a maximum training error is met.
Learning process	“Parallel” learning since all models learn independently	“Sequential” learning where each subsequent model corrects the errors made by previous models (Polikar, 2012; Mienye & Sun, 2022).
Final decision	Majority voting among all models where the class with the most votes (i.e. the mode) is chosen as the final prediction for student *i*.	The algorithm repeats steps 2 and 3 until instances of training errors are below a certain threshold.
Example algorithms	Random forest utilizes bagging to create a random, uncorrelated “forest” or a collection of decision trees	There are variations of boosting such as AdaBoost, gradient boost, and extreme gradient boost (XGboost)

The two most prominent machine learning algorithms that apply boosting are AdaBoost and gradient boost. AdaBoost adds a weighted error to the previous model's erroneous prediction samples, forcing the next model to prioritize the misclassified sample. The weighted error is computed with the specification:

$$F(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

where for *T* iterations, a weak hypothesis h_t receives a weighted error (Kunapuli, 2023; Polikar, 2012). On the other hand, gradient boost trains new models with residual errors from a previous model. Gradient boost has been widely used in Kaggle and other machine learning competitions.

Developed by Chen & Guestrin (2016), XGBoost is a gradient boost method that offers improvement from gradient boost in the following ways: XGBoost uses lasso (L_1) and ridge (L_2) regularizations to prevent overfitting and can handle missing values in data, reducing the time towards data preparation (Chen & Guestrin, 2016; Khan et al., 2024). Since its introduction, XGBoost has gained widespread popularity in the data science community. It has played a key role in nearly all instances where individuals and teams have won prestigious machine learning competitions such as Kaggle, where data miners and statisticians compete to develop the most accurate models for predicting and analyzing data provided by researchers and companies (NVIDIA, 2023).

2.4.5 Advantages of machine learning

In predictive analytics, alternative methodologies to machine learning are frequently employed, with survival analysis being one of the most prominent. Unlike machine learning models, which primarily predict the occurrence of an event, survival analysis extends this by not only assessing whether an event occurred but also examining when that event transpired – also known as the time-to-event. This makes survival analysis particularly valuable in fields such as health sciences, where it is often used to predict events like heart failure or the likelihood of relapse following the initiation of a new treatment (Schober & Vetter, 2018). Commonly used survival analysis methods include the Cox Proportional-Hazards model and the Kaplan-Meier estimator, both of which are adept at handling censoring, a situation where the time to the event is unknown or incomplete (i.e., censored data).

Survival analysis is typically not employed in dropout prediction contexts, as the focus in such settings is primarily on whether dropout occurs, rather than on the duration leading up to it. However, despite its applicability in time-to-event problems, survival analysis presents several limitations when used for dropout prediction.

The first drawback is the reliance on assumptions that may not always hold in real-world applications. For instance, the Cox proportional-hazards model assumes that the hazard ratio (i.e., risk ratio of dropping out) remains constant over time – an assumption that may not be valid in many situations, as a student's hazard likelihood may fluctuate over time. Research in dropout prediction has consistently shown that academic disengagement in the years leading up to high school, particularly during 9th grade, is a strong predictor of early school exit (Neild et al., 2008; Bowers & Sprott, 2012a, 2012b; Allensworth et al., 2013; Knowles, 2015), suggesting that the assumption of proportional hazards may not be appropriate in this context.

A second limitation is that traditional survival analysis techniques struggle to manage high-dimensional data (Huang et al., 2023). In contrast, machine learning algorithms are better equipped to handle datasets where the number of predictors exceeds the number of observations. In recent studies comparing machine learning approaches with survival analysis models, machine learning techniques have demonstrated superior predictive performance (Gong et al., 2018; Spooner et al., 2020; Srujana et al., 2024; Kolasseri, 2024).

In summary, while survival analysis provides valuable insights for time-to-event predictions, machine learning methods enhance these approaches by leveraging complex algorithms capable of managing larger, more intricate datasets, thereby improving the prediction of survival outcomes.

Compared to conventional prediction methods, such as logistic regression, machine learning offers several advantages. First, machine learning algorithms can handle a much larger number of predictors. Second, machine learning enables process automation. Models can be trained with hyperparameters, or optimal parameters that control the learning process to achieve a desired accuracy. Hyperparameters reduce the need for human intervention and to run multiple models to achieve successful predictions. Machine learning offers ease in dimension reduction; especially for systems and states that are overwhelmed with years of education records, certain algorithms can reduce the number of predictors or dimensions while retaining as much information as possible (Brown, 2017). And lastly, logistic regression struggles with understanding complex relationships. For example, a logistic regression cannot identify sub-predictors for specific student subgroups, making it more challenging to detect heterogeneity (Kroese et al., 2019).

2.5 Challenges with dropout prediction

This section confronts several barriers to building a robust dropout prediction model. The barriers described in this section include the tension between model accuracy and interpretability; issues with model generalizability; complications arising when training data exhibit unequal representation by outcome label; and when models inadvertently perpetuate discriminatory practices that are often embedded in the data.

To explore these barriers in depth, I begin by outlining their theoretical foundations, drawing on empirical research that either validates, challenges, or offers solutions to these issues. The remainder of this chapter embeds a synthesis of 19 select dropout prediction studies as a comparison group. These studies are referred to as recent studies. The goal of this synthesis is to survey the strengths and limitations of recent dropout prediction efforts. These studies meet three key criteria: 1) they have been completed or published in the last 10 years (i.e., since 2015), 2) they have either been published in a peer-reviewed journal or presented as a doctoral dissertation, and 3) they leveraged student-level data to predict the likelihood of dropout, either at the K-12 level or at the postsecondary level. The purpose of reviewing these studies is to critically examine their methodological rigor, assess how well they address the inherent challenges in dropout prediction, and identify best practices that can inform future research in this field.

2.5.1 Model accuracy versus model interpretation

Breiman, the pioneer behind the random forest machine learning approach, delineated two contrasting approaches in prediction modeling: data modeling, which prioritizes model interpretability, and algorithmic modeling, which emphasizes predictive performance. In his seminal 2001 paper, Breiman posited that only 2% of statisticians adhered to the algorithmic modeling culture, while approximately 98% remained aligned with the data modeling tradition (Breiman, 2001). This dichotomy was confirmed in an extensive review of over 100 dropout flags by Bowers (2013), which revealed that none of the dropout prediction studies had reported accuracy and instead reported *p*-values and model fit. Table 7 summarizes how recent studies have assessed model performance.

Table 7: Evaluation Metrics of Prior Studies

Study	AUC/ ROC	Accuracy	Sensitivity	Specificity	F-1 score	PR curve	P - value	Root MSE
Anderson et al. (2019)	✓							
Cannistrà et al. (2022)	✓	✓	✓	✓				
Chen & Ding (2023)		✓						
Gardner et al. (2019)	✓							
Gutierrez-Pachas et al. (2022)	✓							✓
Knowles (2015)	✓							
Kruger (2023)			✓	✓		✓		
Lee & Chung (2019)	✓					✓		
Lee & Kizilcec (2020)		✓	✓	✓	✓			
Nájera & Ortega (2022)			✓	✓	✓			
Nascimiento et al. (2022)	✓						✓	✓
Oz et al. (2023)	✓		✓	✓				
Sansone (2019)	✓	✓	✓					
Selim & Rezk (2023)	✓		✓					
Sha et al. (2022)	✓							
Sorenson (2019)	✓		✓					
Weissman (2022)	✓		✓					
Yu et al. (2021)			✓				✓	

Notes: AUC stands for Area Under Curve; ROC stands for Receiver Operating Characteristic; AUC/ROC refers to the metric that the study reported either the AUC for its models or that it presented ROC curves to demonstrate model performance. Accuracy refers to the proportion of correct predictions out of all model predictions. Sensitivity, or recall, is the true positive rate, or the proportion of true positives out of all actual positives. Specificity is the true negative rate, or the proportion of true negatives out of all actual negatives. PR curve is short for precision-recall curve and refers to the metric that the study presented PR curves to demonstrate model performance. P-value stands for probability value; it refers to the estimated probability of rejecting the null hypothesis that there is no difference between two or more reported values. MSE stands for mean squared error; root MSE is the standard deviation of the residuals, or prediction errors.

I find that many dropout prediction studies since Bowers' (2013) review have placed considerable emphasis on prediction accuracy. Of the 19 dropout prediction studies, all went beyond *p*-values to assess model performance. Most included AUC values or ROC curves, with around a quarter reporting prediction accuracy. However, a significant majority did focus on sensitivity, reflecting an increasing emphasis on accurately identifying students who are likely to drop out. This shift highlights a growing recognition of the importance of predicting the positive class – students who are at risk of exiting early. In contrast with findings presented by Bowers (2013), there is evidence suggesting that recent studies have moved away from relying on *p*-value significance, instead prioritizing the reporting of prediction accuracy. Additionally, I find that the models in these studies performed better than a random 50-50 guess.

Despite their impressive predictive capabilities, there remains a critical gap in understanding the inner workings of prediction models. The lack of transparency in how models arrive at their decisions raises concerns, especially in high-stakes domains such as education, where the implications of AI-driven predictions can be profound (Bowers, in press; Du et al., 2021; Feng & Law, 2021; Herodotou et al., 2020). In my review of recent studies, fewer than half of them included an interpretation of their model findings (see Table 8).

Table 8: Accuracy versus Interpretation of Prior Studies

Study	Accuracy	Interpretation	How models were interpreted
Anderson et al. (2019)	✓		
Cannistrà et al. (2022)	✓	✓	Variable importance, partial dependence plots, regression coefficients
Chen & Ding (2023)	✓		
Gardner et al. (2019)			
Gutierrez-Pachas et al. (2022)	✓	✓	Variable importance
Knowles (2015)	✓	✓	Variable importance
Kruger (2023)	✓	✓	SHAP value; variable importance
Lee & Chung (2019)	✓		
Lee & Kizilcec (2022)	✓		
Nájera & Ortega (2022)	✓	✓	Variable importance
Nascimiento et al. (2022)			
Oz et al. (2023)	✓	✓	SHAP value
Sansone (2019)	✓	✓	Variable importance
Selim & Rezk (2023)	✓	✓	Regression coefficients
Sha et al. (2022)	✓		
Sorenson (2019)	✓	✓	Variable importance
Weissman (2022)	✓		
Yu et al. (2021)	✓		

Among those that did report features associated with dropout, most used variable importance plots to interpret the models. These plots show the mean decrease in accuracy for each feature, illustrating the mean decrease in accuracy of each model feature. In other words, variable importance plots describe the contribution of each feature to the model's predictive performance. Cannistrà et al. (2022) also included partial dependence plots, a graph that shows the functional form that links the feature to the outcome without posing parametric assumptions. Merely two studies have included SHapley Additive Explanations (SHAP) value plots, a visualization that relies on horizontal bars to represent the magnitude and direction of each model feature (Johnson, 2023). As mentioned in 2.3.1, regression coefficients are typically provided for regression-based models such as logistic regression, lasso, and ridge regression. This reflects the prioritization of algorithmic prediction over model interpretation, signaling a potential reversal of the 2:98 ratio posited by Breiman (2001).

Model interpretation is essential for three reasons. First, it refines and optimizes model performance by addressing biases and disparities that may be inherent in the data. A clear understanding of how a model makes its predictions will allow users to scrutinize the data inputs and model behavior, helping them detect and mitigate biases that may otherwise go unnoticed.

Second, it improves trust and accountability. If these systems are perceived as "black boxes," where predictions are made without clear explanations, it can erode confidence in the school or district's recommendations. Many machine learning approaches, especially newer ensemble approaches that incorporate boosting, are perceived as both "complex and largely unknown" (Parker et al., 2017; Nussberger et al., 2022; Purcell, 2024)

Third, model interpretation is key to ensuring that the next steps for dropout prevention are aligned with the students' needs. By understanding the model's reasoning, users can design and deliver targeted supports and services that align to the student's needs. For example, a prediction might indicate that a student is at risk of dropping out of high school, but without understanding whether it is driven by academic challenges, attendance, or student behavior, decision-makers cannot provide the appropriate support or services that are needed.

Although recent dropout prediction studies demonstrate prioritization of model accuracy over model interpretation, I argue that there is a need for dropout prediction work to place a similar emphasis on model interpretation and contribute to understanding what factors contribute to student withdrawal. Improved model interpretation in early warning systems is essential to foster accountability, fairness, and efficacy.

2.5.2 Generalizability

A significant obstacle that early warning systems face is generalizability, or the extent to which predictive models maintain their accuracy and validity across varying contexts, populations, and time periods. In the context of machine learning applications, this issue becomes particularly pronounced, as models often require frequent retraining using data from different school systems and different academic years to ensure robustness. For instance, a predictive model developed with data from students graduating in 2018 may not yield the same level of accuracy for students graduating in the post-pandemic era, as shifts in educational conditions and student behavior could influence the factors contributing to graduation outcomes.

Existing research on the generalizability of early warning systems has yielded limited success in overcoming this challenge. Studies such as those by Stuit et al. (2016), Coleman et al. (2019), and Coleman (2021) have demonstrated that attempts to apply these systems across varied contexts often fail to produce consistent or reliable results. This raises important questions about the ability of early warning systems to adapt to changing educational environments and highlights the need for further exploration into methods for improving their generalizability.

2.5.3 Imbalanced data

In prediction science, the categorical outcomes (also referred to as classes) are imbalanced if a dataset has many more instances of some classes than others. Class imbalance poses a challenge in building prediction models because when using an imbalance dataset, the predictive accuracy is poorer for minor classes and higher for the major classes (Bishop, 2024; James et al., 2021; Ali et al., 2013). Moreover, when class imbalance is not addressed, performance metrics often overlook misclassifications of the minority class, which can lead to an increased false positive rate (Fernandez et al., 2018; Kuhn, Johnson, et al., 2013). This issue is particularly prevalent in dropout prediction, where the number of students who drop out is substantially outweighed by those who persist in school. To address this well-known issue, scholar recommendations fall in four areas: examine precision-recall curves, oversample the minority class, undersample the majority class, and assign class weights. Table 9 provides a breakdown of these four approaches.

Table 9: Approaches to address class imbalance

	Precision-recall (PR) curves	**Oversample minority instances**	**Undersample majority instances**	**Class weights**
Development	Manning & Schutze (1999)	Chawla et al. (2002)	Many since 1976 (see examples)	Not applicable
Objective	Increased attention to false positive rates	Balance the ratio between majority and minority instances	Balance the ratio between majority and minority instances	Assign class weights so that the model will give more importance to minority instaces
Approach	An alternative to ROC curves that instead examine the tradeoff between true positives (recalls) and true negatives (precision)	Synthetic Minority Oversampling Technique (SMOTE) finds k-nearest neighbors from the minority class and takes a weighted average between the instance and its neighbors.	Reduces majority instances by random or by selecting *n* instances from the majority class that are closest to the minority class. Then, for each of the *k* instances in the minority class, the technique resamples the majority class to include *k* × *n* observations.	Typically involves inverse probability weighting, or class weights that are inversely proportional to their respective frequencies (Géron, 2022; Krawcyzk, 2016). Class weights are analogous to sample weights which are often employed in survey analysis.
Examples	PR-curves have found more promising results than ROC curves in (Davis et al., 2006; Cook & Ramadas, 2020; Sofaer et al., 2019).	ADASYN SMOTE by He et al. (2008); Borderline-SMOTE by Han et al. (2005); and density-based SMOTE, (Bunkhumpornpat et al., 2012).	Near-Miss by Zhang and Mani (2003); Tomek Links by Tomek (1976); and cluster centroids by Lin et al. (2017).	Not applicable

Among the 19 dropout prediction studies I reviewed, two examined Precision-Recall (PR) curves (Lee & Chung, 2019; Kruger, 2023). While Lee and Chung (2019) examined both

Receiver Operating Characteristic (ROC) and PR curves and found that both approaches identified the same model with the highest accuracy, they concluded that PR curves offered more distinct AUC values, enabling clearer differentiation of model performance.

Since its introduction in 2002, SMOTE has proven successful across various domains and is integrated into a wide range of software packages, both commercial and open-source (Fernández et al., 2018). However, in dropout prediction settings, SMOTE remains underutilized. Among the recent dropout prediction studies reviewed, only Lee & Chung (2019) and Sha et al. (2022) addressed class imbalanced with SMOTE.

2.5.4 Model discrimination

Early warning systems are trained on historical data. There is evidence that historical data, especially student education records, can capture systemic inequities ingrained in student experiences. These inequities are reflected in settings such as disproportionate rates of failure or disengagement among students from marginalized backgrounds (Perdomo et al., 2023; Baker, 2023). It is critical to ensure that models are transparent and are not perpetuating or worse, amplifying inequities. This can lead to harmful consequences like misidentifying certain groups as at-risk when they might not be or failing to identify students who genuinely need attention. I highlight two such failures of early warning system that perpetuated existing inequities: in Wisconsin and in the United Kingdom.

The failure of Wisconsin's statewide early warning system (DEWS) to provide racially equitable predictions shocked the education and the data science field. Despite the predictive prowess of DEWS demonstrated by Knowles (2015), a 2021 equity analysis conducted by Wisconsin's Department of Public Instruction (DPI) equity found that the system's false negative rate – how often a student who did graduate on time was misclassified as high-risk – was disproportionately higher for students of color. Specifically, the false negative rate for Hispanic and Black students was 18 and 42 percentage points higher, respectively, compared to their White peers (Feathers, 2023a).

Another notorious failure is the 2020 United Kingdom (UK) A-Level grading controversy that replaced standardized tests with teacher predictions adjusted based on the “quality of the school.” The poorly designed algorithm systematically lowered grades for students at lower-income schools (Baker, in press; Smith, 2020; Idowu, 2024).

Both operational failures led students, communities, institutions, and stakeholders to be skeptical of the fairness around complex artificial intelligence models – and with good reason. The following subsection will briefly cover the debate around the inclusion of demographic information in education settings.

2.6 Use of demographic information in prediction models

2.6.1 Debate of using demographic information in education

The use of demographic data to understand student engagement and decision-making is controversial and complex. This review follows the sociological definition of a demographic as a characteristic of an individual that, while subject to manipulation in an experimental sense, may evolve over time as the individual’s self-awareness develops (Davis & Museus, 2019). While there are many types of demographic characteristics, I interrogate literature related to four commonly utilized categories in education: gender, race and ethnicity, financial hardship, English proficiency, and disability status**.** Proponents for the inclusion of demographics in education argue that it identifies racial disparities in educational outcomes, which can in turn be used to close the achievement gap. Opponents against the consideration of demographics argue that 1) it reinforces a focus on individual deficiencies rather than strengths, a mindset commonly referred to as deficit-based thinking, and 2) there are legal and ethical considerations regarding the protection of student privacy that are overlooked, and 3) the ability to act on predictions based on demographics.

First, supporters contend that examining disparities in education is critical to understanding and closing the achievement gap. The achievement gap refers to the persistent disparity in educational outcomes for students from historically marginalized backgrounds, typically defined by race and ethnicity, socioeconomic status, and English proficiency. Though racial disparities in educational attainment have reduced over the years, there is overwhelming evidence that children living in poverty, Black and Hispanic students, and students who are not English proficient are significantly less likely to succeed compared to their advantaged peers (NCES, 2024; Reardon, 2019). Understanding minority students' engagement and achievement can ensure that services are accessible to all racial/ethnic groups and allocate a fair distribution of funding and resources. The emphasis on ensuring success for students from all backgrounds is reflected in accountability laws like No Child Left Behind (NCLB), which mandates that schools report high school graduation rates broken down by student subgroups. This requirement aims to hold schools accountable for the academic progress of all students, ensuring that disparities in graduation rates are recognized and addressed. By tracking performance across different demographic groups, the law seeks to promote transparency and encourage targeted interventions to support underserved students. Furthermore, it enables a fairer allocation of funding and resources, ensuring that underserved communities receive the support they need to succeed.

The first argument to exclude demographic considerations in education is that it promotes deficit-based thinking. Deficit-based thinking is the belief that students from marginalized backgrounds (e.g. Black, Hispanic, access to limited financial resources, and non-English proficient) are inherently lacking in abilities, skills, or potential. This mindset blames minority students' failures in school as an inherent "deficit" rather than recognizing inherent inequities enacted by social policies and practices at play (Gorski, 2010; Davis & Museus, 2019). This sets up a dangerous complex where, if adopted by teachers and school leaders, subjects students to racism and xenophobia. For this reason, some advocates recommend ignoring or downplaying demographic characteristics in educational settings, suggesting that all students should be treated identically—an approach commonly referred to as adopting a "colorblind" lens.

The second argument for excluding demographic characteristics in educational settings revolves around ethical and legal concerns. Opponents argue that student privacy must be safeguarded, as the misuse of sensitive data without proper protection can violate ethical standards and undermine trust between students, families, and institutions. In the U.S., laws such as the Family Educational Rights and Privacy Act (FERPA) protect the privacy of student records in all educational agencies and institutions receiving federal funds (Department of Education, 2022). However, historical analyses have revealed unintended consequences and lapses in FERPA's enforcement, leading to violations of student privacy (Vance & Waughn, 2020). Recent studies have highlighted several shortcomings in FERPA, including its negative impact on students' access to opportunities and its role in the overrepresentation of Black children in school disciplinary actions (Peter, 2021); for inadequately protecting female student privacy (Daggett, 2020); failing to address issues related to facial recognition technology, which may marginalize students of color, women, and people with disabilities (Galligan et al., 2020; Bala, 2019). There is sufficient evidence that underscores the urgency to develop fair and balanced educational privacy legislation.

A third reason to exclude demographic characteristics in prediction models is actionability. Actionability refers to the extent to which schools and systems can provide supports and services based on a student's individual needs, rather than on factors outside of their control. If a student is labeled as high-risk primarily because of something that is either not malleable or beyond the individual's control (such as gender, race and ethnicity, or socioeconomic status) then it would not be appropriate to provide an intervention based on these sensitive attributes (Fassett et al., 2022; Baker, 2023a; Paquette et al., 2020). For example, a student's race or family income may correlate with certain risks but does not directly indicate specific actions that can be taken to improve the student's educational outcomes. Instead, focusing on factors that can be changed or influenced, such as academic performance, engagement, and behavioral patterns, allows educators to implement interventions that are more actionable and responsive to the student's actual needs, without perpetuating biases or unfairly targeting certain groups.

2.6.2 Inclusion of demographics in dropout prediction

Since machine learning algorithms are designed to identify implicit patterns within the data they are given, these models can often capture reflect the social biases that are embedded within the data (Rosenbaum, 2001; Feldman et al., 2015; Baker, 2023a). This subsection examines the extent to which recent studies have included demographic characteristics to predict student outcomes.

A survey of educational data mining studies published between 2015 and 2020 in the Journal of Educational Data Mining (EDM) found that 15 percent of publications associated with included demographic information in their analyses, and that the frequency of reporting different types of demographic data is uneven (Paquette et al., 2020). In my review, I find that twelve of the recent studies included gender as a predictor, and of those twelve, eight also included race and ethnicity as predictors (see Table 10). Socioeconomic status, disability status, and English proficiency were included in far fewer studies.

Table 10: Demographic information used in prior studies

Study	Gender	Race/ ethnicity	Economically disadvantaged	Having a disability	English Proficiency
Anderson et al. (2019)	✓	✓			
Cannistrà et al. (2021)	✓	✓	✓		
Chen & Ding (2023)	--	--	--	--	--
Gardner et al. (2019)					
Gutierrez-Pachas et al. (2022)	✓				
Knowles (2015)	✓	✓		✓	✓
Kruger (2023)					
Lee & Chung (2019)					
Lee & Kizilcec (2022)	✓	✓			
Nájera & Ortega (2022)	✓				
Nascimiento et al. (2022)					
Sansone (2019)	✓	✓			
Selim & Rezk (2023)	✓				
Sha et al. (2022)	✓				
Sorenson (2019)	✓	✓	✓	✓	✓
Weissman (2022)	✓	✓	✓	✓	
Yu et al. (2021)	✓	✓	✓		

Notes: Oz et al. (2023) was not included in this table because they examined household-level characteristics. Chen & Ding (2023) did not describe what predictors used in their model, therefore values in that row are denoted with “--".

It is worth noting that Yu et al. (2021) examined model fairness of “aware” models that did include sensitive attributes as predictors, and “blind” models that did not include sensitive attributes as predictors. This approach strengthened its conclusions about the role of demographic information in prediction models. I echo the points made in Baker et al. (2023a) and argue that future studies should take a clearer stance on the use of demographic information. This could either come from the comparison of "aware" or "blind" models, or by the use of demographic characteristics to assess model fairness, rather than incorporating it as a model predictor.

2.7 Metrics to evaluate algorithmic fairness

This remainder of this section discusses algorithmic bias, or the systematic discrimination of models to produce predictions that disadvantages observations who come from protected attributes (i.e., historically marginalized backgrounds). Since data capture the historical inequities embedded within the education system, it follows that machine models that are trained on this data will internalize and perpetuate these biases in model predictive models (Jiang & Pardos, 2021; Idowu, 2024; Yu et al., 2020). Algorithmic bias is evident in prediction model settings, with Wisconsin's and the UK's early warning system demonstrating the profound effects of discriminatory predictions (Baker & Hawn, 2022; Feathers, 2023a; 2023b). I review four criteria to evaluate algorithmic bias: group differences, Absolute Between-ROC Area (ABROCA) method, equalized odds, and demographic parity. Table 11 organizes the key points of each criterion and compares its approach, strengths, and potential drawbacks. While there is no universal agreement on an optimal fairness metric, these efforts collectively demonstrate a commitment to ensuring that model predictions do not reinforce systemic inequities embedded in the data.

Table 11: Metrics to evaluate algorithmic fairness

Approach	Description	Key Metrics	Strengths	Limitations
Group differences	Measures disparities in model performance across subgroups based on protected attributes.	AUC, false positive rate, false negative rate, accuracy, recall, precision, etc.	Provides direct insight into model disparities across groups (e.g., gender, race, etc.).	Lacks standardization across studies, making it difficult to generalize findings or compare results.
ABROCA (Gardner et al., 2019)	Evaluates the difference in model performance between baseline	ABROCA value, calculated as the integral of the absolute difference	Captures fairness across all thresholds, not just specific values,	Complex to calculate and interpret; not widely adopted in research.

	and comparison groups using ROC curves.	between ROC curves for different subgroups.	offering a more nuanced evaluation.	
Equalized Odds (Hardt et al., 2016)	Ensures that both true positive and false positive rates are equal across groups.	True Positive Rate (TPR), False Positive Rate (FPR) across groups.	Guarantees fairness in terms of decision-making outcomes, ensuring no systematic bias in predictions.	Does not account for varying levels of accuracy across subgroups, which can affect fairness in resource allocation.
Demographic Parity (Dwork et al., 2012)	Requires equal probability of a positive outcome across groups, independent of protected attributes.	Probability of positive prediction across groups.	Ensures equal chances of positive outcomes for all groups.	May ignore individual fairness and is less suitable in contexts where group membership should affect outcomes.

Despite the importance of fairness evaluation in predictive modeling, relatively few dropout prediction studies have incorporated such steps in their analyses. Among the studies I reviewed, only Sha et al. (2022) incorporated an ABROCA slicing. In contrast, several other studies have reported model performance across student subgroups, providing insights into how different groups of students fare in terms of dropout risk (Anderson et al., 2019; Lee & Kizilcec, 2022; Weissman, 2022; Yu et al., 2021). This disparity highlights a significant gap in dropout prediction research, where more comprehensive fairness and equity analyses are needed to ensure that predictive models are not only accurate but also equitable across diverse student populations. The absence of such analysis risks reinforcing existing disparities or overlooking the specific needs of marginalized student groups.

2.8 Standards for early warning systems

While there have been no established standards for what defines a high-quality, equitable early warning system, recent efforts have proposed criteria to guide its development. Bowers (2021) outlines four key characteristics of robust, equitable early warning systems: they should be accurate, accessible, actionable, and accountable. 'Accurate' in that the model provides accurate predictions for new data instances. 'Accessible' refers to the ability for the public to understand how predictions are made, including opportunities for replication and reproducibility. 'Actionable' means the identified predictors are malleable and can be used to tailor targeted supports, services, and interventions. 'Accountable' ensures that models are routinely evaluated for algorithmic bias and fairness. (Bowers, 2021). In my review, I find that all but one study meets the accuracy criteria for model performance (see Table 12). These studies assessed model performance using at least one of the following methods: receiver operating characteristic (ROC) curve analysis, precision-recall curve analysis, or accuracy rate.

Table 12: 4 “A”s in prior studies

Study	Accurate	Accessible	Actionable	Accountable
Anderson et al. (2019)	✓			✓
Cannistrà et al. (2022)	✓		✓	
Chen & Ding (2023)	✓			
Gardner et al. (2019)	✓			✓
Gutierrez-Pachas et al. (2023)	✓		✓	
Knowles (2015)	✓	✓	✓	
Kruger (2023)	✓		✓	
Lee & Chung (2019)	✓		✓	
Lee & Kizilcec (2022)	✓			✓
Nájera & Ortega (2022)			✓	
Nascimiento et al. (2022)	✓			
Oz et al. (2023)	✓		✓	
Sansone (2019)	✓		✓	
Selim & Rezk (2023)	✓		✓	
Sha et al. (2022)	✓			✓
Sorenson (2019)	✓			
Weissman (2022)	✓			✓
Yu et al. (2021)	✓			✓

Just over half of the reviewed studies go a step further by interpreting model findings and identifying key features associated with early exit, making their findings actionable for practitioners. One-third of the studies promote accountability by incorporating fairness analyses, employing methods such as differences in group performance, ABROCA slicing analysis, equalized odds, or demographic parity. However, it is disappointing that, to date, Knowles (2015) remains the only study to have fully published their algorithm and code, enabling critique and replication by other researchers and practitioners.

I argue that future dropout prediction work should strive to ensure that the work is accessible. Most dropout prediction studies to date have not replicated established models using new, unseen data. This lack of validation on external datasets limits the generalizability and robustness of findings. A few notable exceptions (Knowles, 2015; Bowers et al., 2013; Coleman et al., 2019) have made their code publicly available, thus contributing to the open science movement, which advocates for the sharing of code to support the confirmation, reproducibility, and further extension of research findings (Agasisti & Bowers, 2017; Bowers et al., 2019; Bowers, 2021). By sharing code, researchers enable others to test, adapt, and build upon their work, fostering a more collaborative and reliable scientific community.

2.9 Rationale for this work

This dissertation strives to meet the 4A criteria proposed by Bowers (2021): accurate, actionable, accountable, and accessible. First, I demonstrate accuracy of my models using multiple techniques: cross-validation on a new, unseen student population and explore multiple resampling techniques to address class imbalance. Second, I apply various approaches to interpret the predictors identified in the models, including feature importance plots and SHapley Additive Explanations (SHAP) values. To enhance the actionability of my work, I explore strategies for communicating these results to a non-technical audience and simplifying the interpretation of "black box" models. Third, I ensure that my work is accountable by assessing algorithmic fairness, reporting the Absolute Between-ROC Area (ABROCA) and equalized opportunity metrics. Finally, I promote transparency and accessibility by providing open access code and output, inviting discussion, critique, and reproducibility. This would be the only known study since Knowles (2015) that ensures findings are accessible to the public.

Another contribution of this work is its focus on the temporal aspect of student dropout, specifically by analyzing patterns of student exits over time. This approach not only enhances the precision of at-risk identification but also allows for more targeted and timely support, potentially reducing dropout rates before students reach a critical point.

Chapter summary

Understanding prior efforts in dropout prediction highlights several gaps in this area of research. First, these efforts rarely address class imbalance and its potential to improve model accuracy. Second, although these studies seek identify factors that are predictive of student exit, recent studies have not put in effort to ensure that model findings are interpretable to a non-technical audience. Third, few studies to date have assessed if models discriminate on student attributes, using new approaches to evaluate algorithmic fairness.

This underscores a need to extend existing knowledge of machine methods to monitor student engagement without associating outcomes with structural inequities. This, in turn, can help school leaders, policy makers, and stakeholders learn about student disengagement and how it may shape student persistence in high school.

CHAPTER 3: METHODS

Chapter introduction

The goal of this dissertation is to develop a predictive model that leverages middle school engagement data to identify students at risk of dropping out of high school during 9th or 10th grade. This chapter provides an overview of the analytic approach for my dissertation. First, I provide a justification of the decision-making process to use middle school data to predict high school exit, followed by a brief description of the data from NCERDC and the training and testing sample. After describing the outcome of interest – whether a student exits high school in 9th or 10th grade – as well as describing the middle school engagement data that are used as predictors, I outline the empirical strategy to answer each research question.

For the first question, I create prediction models that employ one of the following approaches: logistic regression, lasso regression, ridge regression, random forest, and XGboost. I test if the prediction accuracy of these models can be improved by applying resampling techniques such as undersampling majority instances (i.e., instances who do not exit early) and oversampling minority instances (i.e., instances who do not exit early) so that their representation in the training data is equal. I evaluate the predictive accuracy of these models using multiple metrics, such as, accuracy, sensitivity, precision, and specificity.

The second question assesses the extent to which the models developed in the first question provide fair predictions for students in marginalized subgroups, or protected attributes. The protected attributes of focus are having a race or ethnicity that is not White, being economically disadvantaged, having an IEP, and being Limited English Proficient. I assess algorithmic fairness using two approaches: the ABROCA metric (Gardner et al., 2019) and equalized odds metric (Hardt et al., 2016).

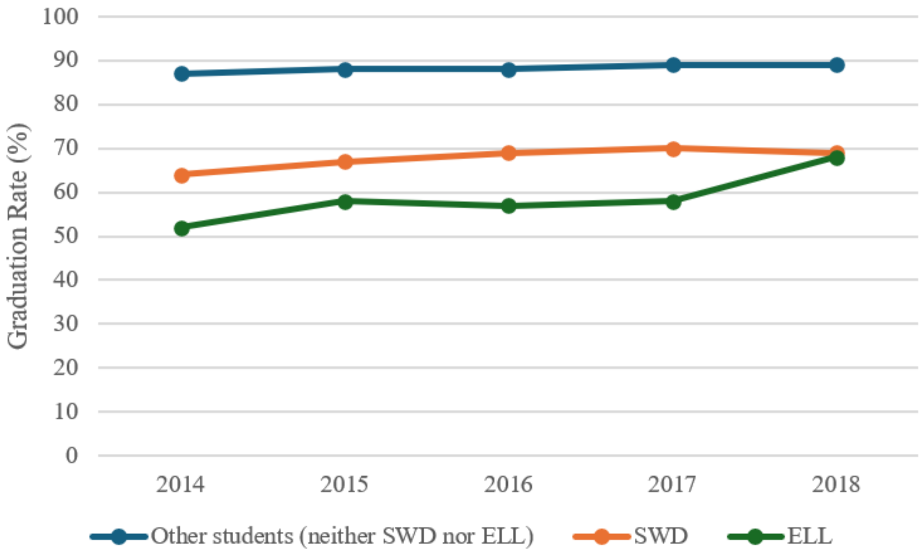
For the third research question, I interpret model findings for selected models that demonstrated model accuracy (as examined in the first research question) and algorithmic fairness (that was explored in the second research question). I then rank the salient predictors from the subset of models to better understand which factors are associated with early exit, as well as to understand potential overlap in relevant predictors across all the models. I rely on regression coefficients, feature importance plots, and Shapley Additive exPlanations (SHAP) plots to understand which features are predictive of early exit.

3.1 Conceptual framework

This analysis examines the academic trajectory of traditional public school students in North Carolina. North Carolina's overall 4-year graduation rate for traditional public school students, regardless of student characteristics, has remained steady between 84 to 87 percent in the years 2014 to 2018. Further examination of high school graduation rates reveals significant variation across student subgroups. Figure 3 illustrates the graduation rates over time for three distinct student profiles: students who are classified as English Language Learner (ELL), students with disabilities (SWD), and students who are neither ELL nor SWD. SWD are defined as students who receive services through the Individual Education Program (IEP).[2]

[2] This figure includes graduation rates for students who earn two types of diplomas. The first is the Future-Ready Course of Study which is North Carolina's minimum graduation requirements to earn a diploma. The second is the Occupational Course of Study which is offered for students with disabilities who have been identified for the program. This option adapts course requirements and requires the name number of credits as Future-Ready Course of Study (NC DPI, n.d.).

Figure 3: North Carolina graduation rates over time

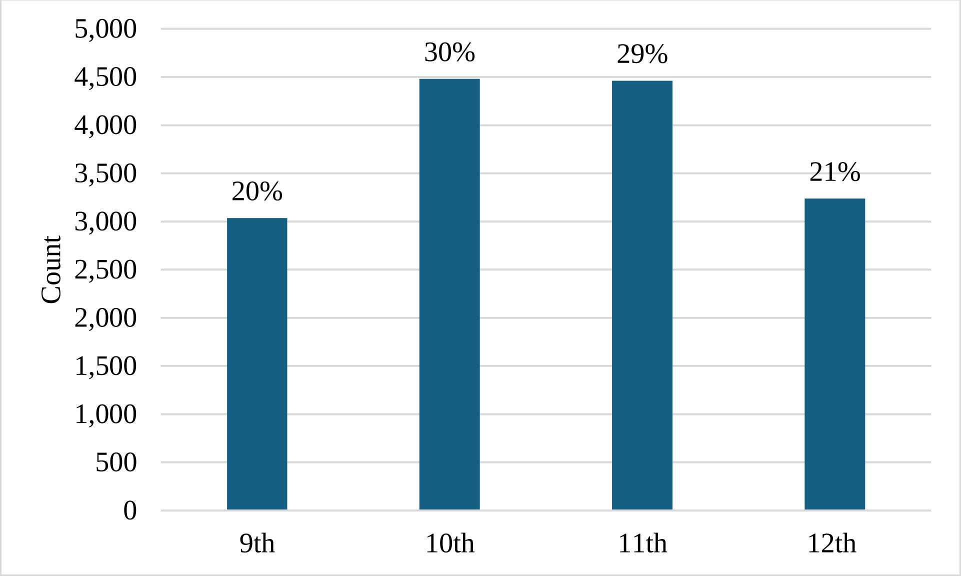


Notes: This graph captures the four-year cohort-adjusted graduation rates for students who attended traditional public schools in North Carolina. This was calculated using NCERDC administrative data and may vary slightly from North Carolina Department of Public Instruction (DPI) reported estimates.

Compared to peers who do not have ELL status or a disability, both SWD and ELL consistently exhibit lower graduation rates. In 2014, the graduation rates for SWD and ELL students were 64 percent and 52 percent, respectively. Although ELL students made substantial progress in high school completion between 2014 and 2018, the gap in graduation rates remained significant, with a difference of nearly 20 percentage points by 2018. Prior research demonstrates that these gaps in high school completion rates are pervasive in later life outcomes and are associated with profound disparities in long-run economic and social well-being (Belfield & Levin, 2007; Rumberger, 2020). Investigating the academic trajectory of students, especially those from marginalized backgrounds, is essential for schools and school systems. Early identification of at-risk students enables schools and districts to provide targeted interventions, supports, and services that could ultimately improve their academic trajectory and increase the likelihood of graduating high school.

North Carolina is a unique observational setting because of its school age policies, where the compulsory school starting age is 7 years old until 16 years old. In settings where the compulsory attending span is 9 years, many high school students turn 18 prior to entering 12th grade. This pattern is evident in North Carolina. Figure 4 presents the exit counts for first-time 12th graders across the state who were expected to graduate in 2018. Students are considered to have dropped out, or exited, if they have withdrawn from the North Carolina traditional public school system.[3]

Figure 4: Temporal patterns of early exit for class of 2018



Notes: (*N* = 15,214). This graph captures counts and percents of the 6th grade students in 2011-2012 cohort who exit high school in either 9th or 10th grade. Among the 15,214 students who exited early, 20 percent of these students exited in 9th grade; 30 percent exited in 10th grade; 29 percent exited in 11th grade; and 21 percent exited in 12th grade.

[3] The counts of students presented in Figure 4 does not include students who temporarily exit the school system and return in a subsequent year (often referred to as a stopout). A detailed definition and breakdown of how a student is classified as an "early exit" is provided in 3.5.1.

Of the 15,214 students in the 2018 cohort who dropped out of high school between 9th and 10th grade, 20 percent of these students exited in 9th grade; 30 percent exited in 10th grade; 29 percent exited in 11th grade; and 21 percent exited in 12th grade. These figures are concerning, as half of the students who permanently discontinue their schooling journey do so in the first two years of entering high school.

This dissertation predicts high school exit in the state of North Carolina for traditional public school students who would have completed high school in 2018. Given the high exit rates in 9th and 10th grade in this context, I argue that an early warning system should flag students before they enter high school.

As discussed in Chapter 2, there is a lack of research on the timing of providing interventions aimed at preventing high school dropout. To date, no studies have explored whether there is a critical window during which prevention efforts would be the most effective. This presents a challenge for early warning system applications, which typically wait until students enter high school before exhibit signs of disengagement. Given the limited understanding of how identification timing impacts effectiveness, waiting until high school may result in missed opportunities for timely intervention.

Evidence that half of all dropouts in 2018 exited in the first two years of high school, coupled with the absence of evidence around timing of at-risk identification, motivates this dissertation to rely on use middle school engagement data to predict early exit. Early identification of at-risk students before they enter high school provides schools and districts more time to implement targeted interventions and support for these students, possibly decreasing the likelihood of dropping out. This dissertation uses supervised learning algorithms, or approaches to predict an outcome when the actual outcome is observable. These algorithms learn and build associations between model predictors and the observed outcome so that they can eventually apply these associations and patterns to new, unseen data.

As highlighted in Chapter 2, few studies to date have evaluated if models provide equitable predictions for students from marginalized backgrounds. My earlier discussion of Wisconsin's statewide early warning system failure highlights the need to ensure that prediction tools do not perpetuate or exacerbate existing structural disparities in educational outcomes. Despite this, however, many dropout prediction studies continue to include sensitive student attributes (e.g., gender, race and ethnicity) as model predictors (see 2.6.2, Table 9). In agreement with recent empirical efforts suggesting that demographics should be excluded from prediction models because of its lack of actionability – characteristics that are neither malleable nor within the individual's control (described in further detail in 2.6.1; Paquette et al., 2020; Bowers, 2021; Baker, 2023a) – there is a need to assess model fairness using demographics rather than incorporating it as a predictor.

Lastly, there is not enough work done to interpret model findings that can be understood by a non-technical audience. As seen in Table 7 (in 2.5.1), studies that include a subset of relevant predictors associated with dropout generally rely on variable importance plots, which can be difficult to quantify for practitioners, policymakers, and stakeholders. This dissertation extends beyond traditional forms of model interpretation to simplify the interpretation of complex, statistical models often called "black box" models.

3.2 Research questions

This study investigates the following research questions to predict early exit from high school:

> **1:** How does the prediction accuracy of supervised learning algorithms to predict early exit from high school compare to that of traditional models (i.e., logistic regression)?

Additionally, how does model performance vary when resampling techniques are used to address class imbalance?
2: To what extent do models provide fair predictions across sensitive student attributes such as gender, race/ethnicity, disability status, financial hardship, and English proficiency?
3: What are the most salient predictors of students who exited high school in 9th or 10th grade?

3.3 Data

I use administrative and longitudinal data from the Department of Public Instruction in North Carolina that are available through the North Carolina Education Research Data Center (NCERDC). The data include student-level data spanning 2011 to 2018 and grades 6 through 12. North Carolina's data do not include student home addresses or zip codes, so the most localized geographic information available is at the school level. To understand urbanicity, I incorporate data from the National Center for Education Statistics (NCES) Education Demographic and Geographic Estimates (EDGE), a publicly available source that provides data to understand the spatial and social context of education in the United States. It uses data from the U.S. Census Bureau’s American Survey to create school, district, and state level indicators of economic, housing, and social conditions for public schools. All NCES data are linked using school identifiers that align with school identifiers in most state longitudinal data systems. I merge EDGE data with NCERDC data by matching these school identifiers, which are consistent across both EDGE and NCERDC.

3.4 Sample

The analytic sample is first-time sixth-grade traditional public school students in North Carolina during the 2011-2012 school year. 94% of students in the population sample (*n* = 107,602) persist in the school system beyond 10th grade, compared to 6% who exited in either 9th or 10th grade.

3.4.1 Data decisions

I restrict the sample based on four characteristics. First, students must have a graduation or exit record provided by NCERDC between 2014 and 2019. Second, students must not have repeated a middle grade, or a grade between 6th to 8th grade. Third, students must attend districts that follow a compulsory school age of 16.[4] Fourth, students must have at least some attendance and state test score data from a middle grade to be included in the sample. This requires that students must have at least one year of attendance data and one year of test score data to be included in the sample.

In cases where a student has missing data for one or two years in either attendance or test scores, I impute the missing values using the student's unique middle school median. A student’s unique middle school median is calculated by taking the student’s available data in each category (English language arts test scores, and math test scores), finding its median, and imputing missing values in that category with this median. For example, if a student did not have 6th grade attendance information and had attended 90 percent of total days enrolled in 7th grade and 94 percent of total days enrolled in 8th grade, that student’s 6th grade attendance would be imputed using the median of their available attendance data, in this case, 92 percent. The described criteria are not dependent on the school or district that the student attended.[5]

3.4.2 Training and testing sample

This dissertation utilizes data from two different populations – a training cohort and a test cohort. This is to address the pervasive challenge of overfitting, which occurs when a model essentially

[4] SL2016-94 is a pilot program where four school districts in North Carolina raised the compulsory school age from 16 to 18 beginning in the 2016-2017 school year. These four districts are excluded in this analysis.

[5] Mobility, or when students change schools during the school year, is described in detail in section 3.6.2.

"memorizes" the data it was given rather than learning generalizable patterns. This typically happens when the model relies too heavily on the training data, or the data that the model learns patterns, relationships from to make predictions. Extensive research has shown that cross-validation is instrumental in enhancing a model's generalizability, or its ability to perform well on new, unseen data (Kuhn & Johnson, 2013; Kroese et al., 2019; Bishop, 2024).

As described in 2.3.3, cross-validation is a technique that splits data into subsets, trains the model on some subsets, and evaluates the model on a subset that was not used for training. The test data – the subset of data that was not used for training – is used to assess how well the model generalizes to new data. I cross-validate models in this analysis by using data of 6th grade students in Fall 2010 as a training sample and 6th grade students in Fall 2011 as a test sample.

A consequence of the decisions imposed on the sample (described earlier in 3.4.1) is that it reduces the sample sizes of both the training and test cohorts. By restricting the data to students with non-missing middle school records, I am unable to observe middle school engagement for a subset of students in both cohorts. Furthermore, the data decisions disproportionately reduce the number of observations of students who exited early. After imposing these data restrictions, the resulting sample sizes are as follows: the final train data have 89,716 observations, of which 1,551 students (1.7 percent) exited in 9th or 10th grade and 88,165 students (98.3 percent) persisted beyond 10th grade. Similarly, the final test data have 95,077 observations, of which 2,404 students (2.5 percent) exited early and 92,673 students (97.5 percent) persisted beyond 10th grade.

To better understand the composition of students across both samples, I present characteristics of the training and test samples in Table 13. The table displays the proportion of the sample represented by each student attribute, along with its intersection with exit status, and the corresponding standard deviation. For instance, a value of 0.441 for females who exited early in the training data indicates that 44.1 percent of all students who exited early in the train data were female. The reported characteristics are binary, meaning that students who do not identify in the same group are assumed to be the counterfactual. This enables the extension of this table to provide information such that, in the prior example, 55.9 percent of all students who exited early in the train data were male. The other race category are students who identified in one of the following categories: 2 or more races, American Indian or Alaskan Native, and Native Hawaiian or other Pacific Islander.[6]

Table 13: Descriptive statistics for training and test sample

	Exited early		**Did not exit early**	
	Train	Test	Train	Test
N	1,551	2,404	88,165	92,673
Female	0.411 (0.492)	0.374 (0.484)	0.505 (0.500)	0.496 (0.500)
Asian	0.006 (0.080)	0.004 (0.064)	0.027 (0.162)	0.027 (0.161)
White	0.487 (0.500)	0.497 (0.500)	0.540 (0.498)	0.524 (0.499)
Black	0.296 (0.457)	0.260 (0.439)	0.264 (0.441)	0.263 (0.440)
Other race	0.082 (0.274)	0.042 (0.200)	0.054 (0.227)	0.041 (0.198)
Economically disadvantaged	0.787 (0.409)	0.794 (0.405)	0.446 (0.497)	0.469 (0.499)
Age	14.580 (0.682)	14.425 (0.648)	13.693 (0.464)	13.697 (0.463)
Had a disability	0.284 (0.451)	0.283 (0.450)	0.114 (0.318)	0.119 (0.324)

[6] I do not report disaggregated characteristics for students in the “other race” category as these students reported in each of these categories make up less than 2 percent of the student population.

Had a disability in a middle grade	0.322 (0.468)	0.315 (0.465)	0.137 (0.344)	0.141 (0.348)
Limited English proficient	0.080 (0.271)	0.102 (0.303)	0.045 (0.208)	0.042 (0.200)
Limited English proficient in a middle grade	0.095 (0.293)	0.113 (0.316)	0.055 (0.229)	0.053 (0.224)

Notes: Standard deviations are reported in parentheses. Data are from students' 8th grade records. Age is reported in years and captures a student's age on October 17th of the year they are in 8th grade. Disability status is approximated with a student having an Individual Education Plan (IEP). Middle grades refer to grades 6 through 8. Other race captures students who identified in one of the following categories: 2 or more races, American Indian or Alaskan Native, and Native Hawaiian or other Pacific Islander.

Among the students who exited early, the training sample exhibits a slight overrepresentation of female, other race, and Black students. In contrast, the characteristics of students who did not exit early appear to be balanced between the training and testing sample.

In examining the composition of students who exited early, there are some emerging patterns that are the same in both training and test sample. This table highlights that among students who exited early, just over half are White, more than three-fourths are economically disadvantaged (i.e. come from households that face financial hardship), approximately 60 percent are male, and more than 25 percent had a disability. I cannot directly compare these proportions to national-level proportions of student exits as there is very limited nationwide reporting and understanding of students who drop out of high school. However, this evidence suggests that exited students from the 2017 and 2018 cohorts had an overrepresentation of vulnerable characteristics, such as being economically disadvantaged and having a disability.

3.5 Measures

3.5.1 Outcome

The outcome of interest is high school exit in 9th or 10th grade. This binary outcome takes a value of "1" if a student exits in either 9th or 10th grade, and "0" if a student persists in school beyond 10th grade. North Carolina monitors student exit and provides detailed records on when students leave the school system.[7] The tracking of students across the state system is facilitated by encrypted identifiers created and provided by the NCERDC. Each individual student is assigned a unique identifier that enables linking of student level data across schools, grades, and years.

In this dissertation, I use the terms "dropout," "student withdrawal," or "early exit" to refer to a permanent discontinuation of schooling from the North Carolina public school system. This is distinct from "stopout," which refers to students who temporarily discontinue schooling. In this study, stop outs are students who drop out in 9th or 10th grade and return to school in a subsequent school year. To ensure that my outcome captures dropout and not stopouts, I cross-check exit records from spring 2017, 2018, and 2019 records, flagging students as dropouts if they left school in 9th or 10th grade and did not return in later school years.[8]

I refer to the students who exited early as the “positive” or “minority” class, as they represent both the affirmative outcome and make up less than half of the testing sample. Conversely, I refer to students who did not exit early as the “negative” or “majority” class.

Although the goal of a prediction model is to forecast the likelihood of an event occurring, the model may rarely estimate probability outputs that are exactly 0 or exactly 1. For this reason, standard machine learning practices recommend the identification of a decision threshold, or probability cutoff for which a model classifies the probability into a class label (e.g., "exiting early" or "not exiting early") with a receiver operator characteristic (ROC) curve analysis (Kroese et al., 2019). The ROC curve essentially plots the tradeoff between true positives (proportion of correctly predicted for positive cases) and false positives (units for

[7] North Carolina records include an exit survey that students who withdrew from the school system are required to complete. The survey asks questions about the students’ decision to exit. The survey data are not examined in this analysis due to a high rate of missing responses.

[8] This analysis is unable to observe the trajectory of exited students, such as if they transferred or enrolled in a charter school or a private school. The implications of this are discussed in Chapter 5.1.

which the model incorrectly predicted the proportion of incorrectly predicted positive cases) at random classification thresholds (Streiner & Cairney, 2007; Bowers & Zhou, 2019). I map a ROC curve for each model to identify the optimal probability threshold at which a student with a predicted probability above this threshold will be classified as likely to exit early.

3.5.2 Model features

I use data on student engagement from grades 6 to 8 as predictors to flag students at risk of exiting early. These predictors – also known as model features – data include End-of Grade (EOG) test scores, attendance, disciplinary infractions, school mobility, and student characteristics. I follow prior research that recommends that a base set of indicators should include ABC – attendance, behavior, and course performance (Frazelle & Nagel, 2015; Allensworth & Easton, 2007; Balfanz et al., 2007; Mac Iver, 2010). I exclude gender, race and ethnicity as model features because these student attributes are not actionable and typically are not used as justification for why a student is classified at-risk or for why a student is eligible for intervention, supports, and services. Instead, I use gender and racial identification to assess whether the model provides equitable predictions a discussion that will be further explored in Section 3.7.

It is important to note that none of the model features in this analysis are categorical in nature (e.g., represent more than two categories or non-numeric groups). This is because some of the analytical methods that I employ either 1) assume that features with numeric values have an inherent order or ranking of values, or 2) are not equipped to directly handle non-numeric features.

Table 14 provides an overview of the model features that will be used for all models. This subsection provides a detailed description of how these features were created and organizes them into four categories: student characteristics, attendance, academic performance, and discipline information.

Table 14: Model features

Group	Variable	Description
Student characteristics (extracted from 8th grade records)	Financial hardship	1: Student is economically disadvantaged* 0: Not economically disadvantaged
	Limited English Proficient	1: Student is Limited English Proficient (LEP) 0: Not LEP
	Limited English Proficient in a middle grade	1: Student was LEP in grades 6-8 0: Not LEP in grades 6-8
	Age	Continuous value capturing age in October of 8th grade. Reported in years, rounded to the nearest tenth
	Disability status	1: Student has an Individualized Education Plan (IEP) 0: No IEP
	Disability status in a middle grade	1: Student had an IEP in grades 6-8 0: No IEP in grades 6-8
	Urban	1: Attended a school in that locale 0: Did not attend a school in that locale
	Suburban	
	Town	
	Rural	
Attendance information	6th grade absence	Total days absent divided by total days enrolled
	7th grade absence	
	8th grade absence	
	Chronic absence in 6th grade	1: Student is chronically absent - has missed 10% or more of total enrolled days 0: Student is not chronically absent
	Chronic absence in 7th grade	
	Chronic absence in 8th grade	
	Chronic absence in all middle grades	1: Student was chronically absent in all grades 6-8 0: Student not chronically absent in all grades 6-8
	Chronic absence in a middle grade	1: Student was chronically absent in grades 6-8 0: Student not chronically absent in grades 6-8
	School mobility in 6th grade	Number of times a student changed schools within the school year
	School mobility in 7th grade	
	School mobility in 8th grade	

	School mobility in middle grades	Number of times a student changed schools within the school year in grades 6-8
Academic information	6th grade math proficiency	1: Student did not score proficient on grade-level EOG
	6th grade reading proficiency	0: Student scored proficient or higher on grade-level End-of-Grade (EOG) test
	7th grade math proficiency	
	7th grade reading proficiency	
	8th grade math proficiency	
	8th grade reading proficiency	
	Math proficiency in middle grades	1: Student was not proficient in all grades 6-8 EOG tests
	Reading proficiency in middle grades	0: Student was proficient in at least 1 grade 6-8 EOG tests
Discipline information	6th grade ISS	1: Received in-school suspension (ISS)
	7th grade ISS	0: Did not receive ISS
	8th grade ISS	
	ISS in a middle grade	1: Received ISS in grades 6-8 0: Never received ISS in grades 6-8
	6th grade OSS	1: Received out-of-school suspension (OSS)
	7th grade OSS	0: Did not receive OSS
	8th grade OSS	
	OSS in a middle grade	1: Received OSS in grades 6-8 0: Never received ISS in grades 6-8
	Suspended in a middle grade	1: Received either OSS or ISS in grades 6-8 0: Never received OSS or ISS in grades 6-8
	ST suspension in a middle grade	1: Received a short-term suspension in grades 6-8 0: Never received a short-term suspension in grades 6-8
	LT suspension in a middle grade	1: Received a long-term suspension in grades 6-8 0: Never received a long-term suspension in grades 6-8

Notes: Gender, race/ethnicity, economic, and disability status are based on 8th grade student records.

*North Carolina Department of Public Instruction (DPI) defines an economically disadvantaged student as a child meeting one or more of the following criteria: direct certification from food assistance programs (SNAP, TANF, FDPIR); runaway, homeless, foster, Medicaid recipient, enrolled in Head Start or state-funded pre-kindergarten, or migrant status; and community eligibility provision (CEP).

Student characteristic indicators

Student characteristics include English proficiency status, disability status, age, and urbanicity at the school level. English proficiency and disability status are extracted from 8th grade records.

Using birth month and birth year provided by NCERDC, I follow Bowden et al. (2023) and calculate a student's age based on the kindergarten entry cutoff date of October 17th. Given that the data do not provide full dates of birth, there may be a margin of error of up to one month in calculating the exact age.

Urbanicity (i.e., urban, suburban, town, and rural) is measured at the school level using publicly available data from NCES Education Demographic and Geographic Estimates (EDGE) data. The indicator follows NCES classifications of locales into four categories: city, a territory inside an urbanized area and inside a principal city; suburban, a territory outside of a principal city and inside an urbanized area; town, a territory inside an urban cluster; and rural, a territory that is 2.5 or more or miles from an urban cluster or 5 or more miles away from an urbanized area. Additional information on locale classifications can be found in Giverdt (2017). Locale classifications were matched to the school that the student attended in 8th grade. For students that attended multiple schools in 8th grade, urbanicity was matched to the school in which the student was enrolled the longest.

I follow North Carolina Department of Public Instruction (DPI)'s definition of economic disadvantage. A student is considered economically disadvantaged if they meet one or more of the following criteria: direct certification from food assistance programs (SNAP, TANF, FDPIR); runaway, homeless, foster, Medicaid recipient, enrolled in Head Start or state-funded pre-kindergarten, or migrant status; and community eligibility provision (NC DPI, 2018).[9]

There are two indicators for limited English proficiency (LEP): being classified as LEP in 8th grade or ever been classified as LEP in middle grades (between 6th through 8th grade). LEP

[9] The criteria for Economically disadvantaged Status (EDS) students are publicly available on the NC DPI website, eliminating concerns about data privacy on how EDS is defined.

students are those with limited abilities in speaking, writing, reading, or understanding English. This classification is equivalent to what other states refer to as English Language Learners (ELL). Under the Every Student Succeeds Act (ESSA), provisions are in place to offer supplemental services aimed at improving the English language proficiency and academic performance of LEP students (U.S. Department of Education, 2016). Eligibility for these services is determined through a standardized statewide annual language proficiency assessment, followed by additional entrance and exit procedures.

Student disability is determined by the student having an Individualized Education Plan (IEP), also known as the Individualized Education Program. Students with an IEP receive special education, services, and support tailored to their needs. IEPs are mandated by the Individuals with Disabilities Education Act (IDEA), a special education law. Eligibility for IEP services requires identification and evaluation, which may vary across school systems and states. In this analysis, students who have an IEP in 8th grade are classified as having a disability. Similar to the LEP indicators, there are two indicators for student disability: having a disability in 8th grade or having ever had a disability in a middle grade.

Attendance indicators

Attendance is measured with three types of indicators: absence rate, incidence of chronic absenteeism, and school mobility. A student’s weighted absence rate is calculated by dividing the total number of absences across all schools attended in a grade by the total enrollment days for that grade. To minimize potential outliers or errors in administrative data, I restrict the analysis to students whose annual total enrollment days in the public school system is more than 10 days but do not exceed 210 days.

The models include a binary indicator for being chronically absent in a specific middle grade (6th through 8th) and across all middle grades. This follows North Carolina Department of Public Instruction's definition of being chronically absent as missing at least 10 percent of instructional days, regardless if the absence is excused (NC DPI, 2019).

School mobility captures if a student changed school in a school year. Empirical evidence suggests that within-year school moves disrupt school experiences in ways that are associated with lower student achievement (Hanushek et al., 2004; Burkam et al., 2009; Welsh, 2017). For this reason, this indicator focuses on within-year mobility rather than between-year mobility. The indicators capture two pieces of information: the number of times a student has changed schools within a grade, and the number of within-year moves across 6th to 8th grade.

Discipline indicators

Student behavior is captured through four categories of disciplinary infraction data: out-of-school suspensions (OSS), in-school suspensions (ISS), short-term (ST) suspensions, and long-term (LT) suspensions. The North Carolina middle school discipline data do not include records of every student in the system; the data only include information on students who have received a disciplinary infraction. Therefore, this analysis assumes that a student with no discipline record has received neither an OSS nor an ISS.

The OSS indicators examine whether a student received an OSS at each grade level, and whether the student has ever received an OSS in a middle grade. Similarly, there are 4 ISS indicators – whether a student received an ISS at each grade level, and whether the student has ever received an ISS in a middle grade.

The ST and LT suspension indicators examine if a student has received a ST or a LT suspension in a middle grade. This follows North Carolina's definition that a short-term suspension as one that is 10 days or fewer, while a long-term suspension is at least 11 days.[10]

The North Carolina middle school discipline data are not consistent across the years of analysis; some years lack information on the number of times a student was suspended or the reasons for suspension. Consequently, this analysis is unable to observe details of disciplinary infractions.

Academic indicators

I measure academic engagement using student performance in English language arts and math tests. NC's End-of-Grade (EOG) tests are administered annually for students in grades 3 through 8. EOG tests are scored on a scale of 1 to 5 where students receiving a score between levels 3 to 5 indicate proficiency, while scores below 3 are considered below proficient (NC DPI, 2017).

I transform proficiency scores into binary indicators that represent two forms of non-proficiency: (1) when a student is not proficient in either math or English Language Arts in a specific grade, and (2) when a student fails to demonstrate proficiency in a subject across all middle school years. The latter indicator applies to students who were not proficient in a given subject throughout all three years of middle school.

I do not include continuous measures of EOG tests for two reasons. First, even if the scores were standardized to a scale where they could be interpreted as units of standard deviation, such an interpretation would be difficult for school leaders and educators. These stakeholders typically focus on improving test performance, rather than on increasing standard deviations in

[10] NCERDC data do not include the length of a short-term or long-term suspension. This limits the ability to observe the length of suspension in this analysis.

test scores. Second, using raw test scores as a continuous measure would complicate the interpretation of small changes in performance, as these changes may not correspond to a meaningful shift in the number of correct answers on the exam. North Carolina's middle school data do not include report card level transcript information. Thus, I am unable to examine coursework performance or the types of courses taken in grades 6 through 8.[11]

3.5.3 Descriptive statistics of test sample

This subsection examines how model features vary by exit status. Specifically, I analyze the mean and distribution of each model feature for the testing sample provided in Table 15.

Compared to students who persisted in high school beyond 10th grade, students who exited early were more likely to be economically disadvantaged, be older, have an IEP, be Limited English Proficient, not meet grade-level proficiency in reading or math, have higher absence rates, be chronically absent, and receive some form of school suspension. This aligns with the risk indicators highlighted in prior empirical work (Allensworth & Easton, 2007; Balfanz et al., 2014; Allensworth et al., 2014) . However, there are no observable differences by exit status with regards to school mobility and school-level urbanicity.

Table 15: Descriptive statistics of model features for test sample

	Outcome		
	Did not exit early	Exited early	Total
N	92,673 (97.5%)	2,404 (2.5%)	95,077 (100.0%)
Student characteristics			
Economically disadvantaged	0.469 (0.499)	0.794 (0.405)	0.478 (0.499)
Age	13.697 (0.463)	14.425 (0.648)	13.716 (0.482)
IEP	0.119 (0.324)	0.283 (0.450)	0.123 (0.328)
Ever had an IEP in a middle grade	0.141 (0.348)	0.315 (0.465)	0.145 (0.352)
Limited English proficient	0.042 (0.200)	0.102 (0.303)	0.043 (0.203)

[11] This analysis is unable to examine or approximate course offerings in middle grades. There is no centralized oversight of course offerings statewide, as middle school course selection is provided and managed at the county level.

Ever limited English proficient in 8th grade	0.053 (0.224)	0.113 (0.316)	0.055 (0.227)
Urban	0.282 (0.450)	0.295 (0.456)	0.283 (0.450)
Suburban	0.155 (0.362)	0.147 (0.354)	0.155 (0.362)
Town	0.120 (0.325)	0.114 (0.317)	0.120 (0.325)
Rural	0.443 (0.497)	0.445 (0.497)	0.443 (0.497)
Academic information			
Not proficient in 6th grade math	0.161 (0.368)	0.465 (0.499)	0.169 (0.375)
Not proficient in 7th grade math	0.575 (0.494)	0.889 (0.314)	0.583 (0.493)
Not proficient in 8th grade math	0.547 (0.498)	0.895 (0.306)	0.555 (0.497)
Not proficient in all middle grade math	0.155 (0.362)	0.430 (0.495)	0.162 (0.368)
Not proficient in 6th grade reading	0.217 (0.412)	0.511 (0.500)	0.224 (0.417)
Not proficient in 7th grade reading	0.490 (0.500)	0.800 (0.400)	0.497 (0.500)
Not proficient in 8th grade reading	0.428 (0.495)	0.727 (0.446)	0.435 (0.496)
Not proficient in all middle grade reading	0.193 (0.394)	0.417 (0.493)	0.198 (0.399)
Attendance information			
Absence rate in 6th grade	0.034 (0.040)	0.094 (0.092)	0.036 (0.043)
Absence rate in 7th grade	0.040 (0.044)	0.110 (0.101)	0.042 (0.048)
Absence rate in 8th grade	0.037 (0.039)	0.105 (0.102)	0.039 (0.043)
Chronically absent in 6th grade	0.047 (0.211)	0.255 (0.436)	0.052 (0.222)
Chronically absent in 7th grade	0.066 (0.248)	0.353 (0.478)	0.073 (0.260)
Chronically absent in 8th grade	0.012 (0.108)	0.078 (0.269)	0.013 (0.115)
Ever chronically absent in a middle grade	0.102 (0.303)	0.494 (0.500)	0.112 (0.316)
Chronically absent in all middle grades	0.001 (0.032)	0.016 (0.125)	0.001 (0.038)
School mobility in 6th grade	0.006 (0.075)	0.017 (0.128)	0.006 (0.077)
School mobility in 7th grade	0.013 (0.112)	0.057 (0.232)	0.014 (0.117)
School mobility in 8th grade	0.000 (0.003)	0.000 (0.000)	0.000 (0.003)
School mobility in all middle grades	0.185 (0.418)	0.350 (0.577)	0.189 (0.423)
Discipline information			
OSS in 6th grade	0.084 (0.277)	0.334 (0.472)	0.090 (0.287)
OSS in 7th grade	0.101 (0.301)	0.400 (0.490)	0.108 (0.311)
OSS in 8th grade	0.087 (0.281)	0.391 (0.488)	0.094 (0.292)
OSS in a middle grade	0.185 (0.388)	0.611 (0.488)	0.195 (0.397)
ISS in 6th grade	0.115 (0.319)	0.342 (0.474)	0.121 (0.326)
ISS in 7th grade	0.132 (0.339)	0.395 (0.489)	0.139 (0.346)
ISS in 8th grade	0.105 (0.306)	0.334 (0.472)	0.111 (0.314)
ISS in a middle grade	0.194 (0.395)	0.492 (0.500)	0.201 (0.401)
Ever suspended in a middle grade	0.291 (0.454)	0.730 (0.444)	0.302 (0.459)

Notes: In the first row, *N* indicates the sample size and proportion of the test data represented by the subgroup. Standard errors are reported in other parentheses. Student characteristics are extracted from 8th grade records. Detailed information about each indicator can be found in 3.5.2.

3.6 Empirical strategy for research question 1

The first question examines the following: 1) how the prediction accuracy of supervised learning algorithms to predict high school exit compares to that of traditional approaches, and 2) how model performance may vary when the model includes resampling techniques to address class imbalance. This question builds, optimizes, and evaluates the prediction accuracy of fifteen prediction models.

To answer these questions, I follow four key steps discussed in this section. 3.6.1 focuses on selecting five supervised learning algorithms that use middle school engagement data to predict whether a student will exit school in 9th or 10th grade. The second step, described in 3.6.2, follows field norms to identify a decision threshold to set a minimum predicted score that will classify a student as likely to exit early. This step also includes the parameters and strategies used to optimize model performance. Step 3.6.3 discusses statistical techniques to handle class imbalance, where the data contains significantly more instances of one outcome (e.g., persisting beyond 10th grade) than the other (e.g., early exit). The final step, 3.6.4, evaluates the performance of all models, comparing those trained on balanced data with those trained on imbalanced data.

3.6.1 Supervised algorithms

Supervised algorithms are a class of machine learning methods where the outcome is known (often referred to as class “labels”) and models are trained on data that include class labels. These algorithms learn from the patterns in the relationships in the training data. Once a model is trained it can make predictions on unseen data that was not used during the training process, also known as test data.

I follow the approach seen in earlier dropout prediction studies where the prediction accuracy of a logistic regression model is compared to that of more advanced supervised algorithms. This dissertation treats the logistic regression as a baseline model and compares its performance to models that use lasso regression, ridge regression, random forest, and extreme gradient boosting (XGboost). The remainder of 3.6.1 describes each of the five supervised learning algorithms.

Logistic regression

A logistic regression estimates the likelihood of a binary outcome using the maximum likelihood estimation (MLE) framework. A key feature of the logistic regression approach is its residual sum of squares (RSS), which quantifies the model's error (Kuhn & Johnson, 2013). The RSS is represented by the following formulation:

$$RSS = \sum_{i=1}^{n} (Y_i - \widehat{Y_\iota})^2$$

where observation *i*'s difference between actual values Y_i and predicted values $\widehat{Y_\iota}$ are squared and summed across *n* observations. A smaller RSS indicates lower variance, which is associated with better model fit (James et al., 2023). Understanding (RSS) is crucial, as it optimizes model parameters. The second and third algorithm used in this analysis depend on RSS in its objective function (i.e., model specification). Thus, a comprehensive understanding of how RSS functions is essential for interpreting model performance.

Regularized regressions

Regularized regressions, often known as shrinkage methods, are regression methods that prevent overfitting by adding a penalty term to the RSS function. Regularized regressions diverge from the standard regression approach by shrinking the coefficients towards zero, which can improve the model's ability to generalize to unseen or new data. This dissertation applies two types of regularized regression: lasso regression and ridge regression. Both methods follow similar formulations – it applies a tuning parameter, λ, to create a penalty term that is added in the RSS function. The regressions vary in how the penalty term is calculated.

Ridge regressions, also known as L2 regularizations, add a squared magnitude penalty term to the loss function (Tibshirani, 1996; Hastie et al., 2015; James et al., 2023). Building on the use of RSS to evaluate model linear fit, the L2 penalty term is added to the end of the RSS function with the following formulation:

$$RSS_{L2} \ = \ RSS + \lambda \sum_{j=1}^{P} B_j^2$$

where lambda (λ) controls the sum of squared regression coefficients across *p* predictors. It is important to note that ridge regressions include all predictors in the final model and will not set any of them to exactly zero (unless λ is infinite). The smaller λ is, the closer the function resembles a logistic regression model.

Least absolute shrinkage and selection operator (lasso) regressions, or L1 regularizations, add an absolute value of magnitude penalty term. Its RSS function is denoted by:

$$RSS_{L1} \ = \ RSS + \lambda \sum_{j=1}^{P} |B_j|$$

Lasso regressions can force coefficient estimates to be exactly zero when λ is sufficiently large, forming a parsimonious model (Kroese et al., 2019; Friedman et al., 2023).

Random forest

The fourth supervised approach is random forest. This approach creates a collection (or "forest") of multiple models with random subsets of data, hence the name "random forest." Random forest is a tree-based method, so its models are an aggregate of decision trees. As described in 2.4.4, decision trees follow a tree-like structure to ask “if-then” structured questions about model features to split the data into smaller groups based on different features. Random forest has three key features to arrive at a final prediction: bagging, feature randomness, and majority voting.

First, it uses bagging (bootstrap aggregation) to create a diverse set of decision trees. Bagging involves drawing random subsets of the training data with replacement, and each tree is independently trained on a different subset. The second attribute is feature randomness, where each tree randomly selects a subset of features (predictors) at each split, reducing correlation between trees. The third feature is its use of voting to make final predictions. Each tree casts a 'vote' for the predicted outcome for an instance *i* and the outcome with the most (majority) vote across all the decision trees determines the final prediction for *i* (Breiman, 2001; Hastie et al., 2009a, 2009b; Cutler et al., 2012).

XGboost

XGboost, short for extreme gradient boosting, is another tree-based method that follows a sequential process of building trees that penalizes incorrect predictions. The core of this method relies on a gradient boosting algorithm. I first describe gradient boosting and then introduce a new technique that makes it “extreme.”

Gradient boosting offers an improvement to bagging by which builds one tree at a time and assigns higher weights to incorrect predictions (often called “weak learners”) from earlier models. The algorithm aggregates the residual errors – the difference between actual values Y_i and predicted values $\widehat{Y_i}$ – to score each model with a loss function, θ. As explained in 2.4.4, θ guides how the model will adjust its parameters to improve accuracy.

The technique that makes gradient boosting “extreme” is the addition of a regularization term. This is the same regularization or penalty term that is added to regularized regressions, lasso and ridge regressions. XGboost’s objective function follows the specification:

$$obj(\theta) \; = \sum_{i}^{n} l\big(Y_i - \widehat{Y_\iota}\big) + \sum_{k=1}^{K} \lambda(F_k)$$

where $l\big(Y_i - \widehat{Y_\iota}\big)$ represents the loss function and $\lambda(F_k)$ is a regularization term for *K* number of decision trees. The goal of this function is to minimize this “loss” to achieve more accurate predictions (Chen & Guestrin, 2016). The final prediction for instance *i* is a weighted sum of all tree predictions.

3.6.2 Optimizing model performance

This subsection outlines two techniques employed in this analysis to ensure that each model is performing at a desired performance level. It is important to emphasize that the techniques discussed in this subsection – hyperparameter tuning and the identification of an optimal decision threshold – are applied during the training phase of model development. This process typically involves initially running a model with training data using default parameters, followed by iterative adjustments to the model parameters. These techniques, once applied, increase the likelihood that the model will generate precise predictions for new, unseen data.

Tuning hyperparameters

Hyperparameter tuning is an approach to optimize machine models by imposing parameters or specifications to improve model behavior, learning, and performance. Hyperparameters typically need to be manually tuned or specified. In cases where the model is not assigned or specified with hyperparameters, models are tuned on default settings built into the statistical command, package, or library.[12] There is a broad range of hyperparameters that can be applied to machine learning models. For instance, in tree-based methods like random forest and XGboost, one can specify the number of features to consider when determining the best split of the training data, using the built-in hyperparameter *max_features*. This can help control model complexity and improve generalization by limiting the number of features evaluated at each decision point.

There are several approaches to tune hyperparameters. This can be a manual search that involves trial and error or a grid search that automates the search process to identify a hyperparameters that meet a manually specified level of prediction error. While the manual search approach can be time-consuming and inefficient, grid search offers a more structured and systematic approach, ensuring a more comprehensive exploration of the hyperparameter space. For this reason, I opt to use grid search to identify a combination of hyperparameters to optimize the performances of the random forest and XGboost models.

I rely on a slightly varied grid search to identify a level of regularization for regularized regressions (i.e. lasso and ridge regression). I follow standard practices to identify the optimal λ in both regressions using k-fold cross-validation. This strategy involves the following steps: choosing *k* folds; splitting the data into *k* equal sets with the $\frac{1}{k}$ of the data serves as test data and the remainder as train data; calculating the mean squared error (MSE) within each fold for each λ; calculating the overall cross-validation MSE for each λ; and plotting cross-validation MSE for each λ to identify the minimum cross-validation λ (Tibshirani & Friedman, 2001; Hastie et al., 2009a; 2009b; Friedman et al., 2023).

[12] Commands, packages, and libraries refer to features in statistical programming languages (R, Stata, Python, etc.) to initiate actions or perform specific tasks.

Decision threshold

Decision threshold refers to the cut-off score at which a model assigns an outcome (often referred to as class) to each prediction. Binary classification models generate a predicted probability score ranging from 0 to 1 for each observation. In this analysis, this predicted probability score indicates the likelihood that a student exited in either 9th or 10th grade. However, to assign class labels, the model requires a decision threshold – a specified probability value that determines whether a student is classified as 'exited early' or 'did not exit early.' For instance, a model with a decision threshold of 0.7 will classify all predictions with a probability score of 0.7 or higher with an 'exited early' label, while scores below 0.7 would be classified otherwise. The decision threshold is a hyperparameter in that models do not internally select a threshold; it must be manually specified. In data science, the standard approach to identify a decision threshold is through the analysis of the Receiver Operating Characteristic (ROC) curve (Swets, 1988; Swets et al., 2000; Streiner & Cairney, 2007).

A ROC curve is a visual representation that shows the model's ability to distinguish between classes. It displays the trade-off between the true positive rate (TPR) and false positive rate (FPR). In this analysis, positive class refers to students actually exited early, while the negative class refers to students who did not exit early. The TPR is the model's ability to correctly identify true positives and is calculated as the proportion of predicted true positives (i.e., students who were correctly labeled 'early exit') over all true positives (i.e., all students who exited early). On the other hand, the FPR measures when a model misclassifies a negative class as a positive class. It is calculated as the proportion of false positive (i.e., students who were incorrectly labeled 'early exit') over true negatives, or students who did not exit early.

Often referred to as the false alarm, the FPR signals the cost of misclassifications (Bowers & Zhou, 2019; Nakas et al., 2023).

Selecting a decision threshold is a critical decision that has consequences on model performance. For instance, increasing the threshold (e.g., from 0.3 to 0.8) makes the model more stringent as fewer instances can be labeled in the positive class. This approach reduces the TPR as the model would misclassify more instances in the positive class that have lower scores, but at the same time this reduces the FPR, which is arguably good for preventing misclassification of negative instances. Conversely, lowering the threshold (e.g., from 0.8 to 0.3) would increase the TPR at the sacrifice of increasing the FPR. Although a combination of high TPR and low FPR are ideal, there is no standardized practice on how to select a threshold. The chosen threshold may depend on which metric is the most important to the specific setting.

3.6.3 Class imbalance

In classification methods, the classes (i.e., outcomes) are imbalanced when the data have many more instances of one outcome (the majority class) compared to another outcome (the minority class). Class imbalance is a challenge in prediction model development. During the learning phase – when the model is learning and building associations from training data – models tend to prioritize correctly predicting the majority class. This is often at the expense of incorrectly predicting outcomes for the minority class (He & Garcia, 2009; Krawczyk, 2016; Fernández et al., 2018). This issue is common in dropout prediction and is evidenced in this analysis, where the training data outcomes follow a 98:2 ratio. This means that for every 98 students who continue beyond 10th grade, only 2 students drop out in either 9th or 10th grade. I address class imbalance with two techniques: undersampling and oversampling.

Oversampling

The first approach to balance classes is increasing, or oversampling instances in the minority class. I do this with Synthetic Minority Oversampling Technique (SMOTE), a technique that generates synthetic (i.e., artificial) minority class observations based on the *k*-nearest neighbor for the minority class (Chawla et al. 2002; Anis & Ali, 2017; Fernandes et al. 2018). This synthetic data generation process continues until the number of minority instances in the training data matches that of majority instances. The test data do not need to be resampled since the test data is not used for learning which features are predictive of early exit. Rather, the test data is used at the final stage of model development to assess model accuracy.

There are several packages and variations of SMOTE available across different programming languages. I utilize the "smotefamily" package in R developed by Siriseriwan (2024) to generate synthetic instances. This package is an ensemble of various SMOTE techniques that have been introduced to the data science community over the past two decades. Among the different SMOTE variations, I apply safe-level-SMOTE, proposed by Bunkhumpornpat and authors (2009).

Safe-level-SMOTE aims to improve the general SMOTE method (Chawla et al., 2002) by first assigning a safe level, or weight degree, to each minority stance. The safe level is the number of minority instances in *k*-nearest neighbors. A safe level close to *k* indicates that the instance is “safe”, whereas a safe level closer to 0 indicates that the instance is noisy. After assigning a safe level, synthetic instances are positioned around safe regions. Compared to general SMOTE, safe-level-SMOTE ensures that synthetic instances do not overlap with majority instances, ultimately leading to improved model performance (Bunkhumpornpat et al., 2009). This synthetic data generation process is used to balance the training data of first-time 6th grade students in the 2010-2011 school year. The SMOTE training data consist of 175,021 instances, where 50.4 percent of instances did not exit early and 49.6 percent exited early.

After applying Safe-level-SMOTE to oversample train data, I find that the synthetic minority sample closely mirrors the original minority sample on all characteristics except school mobility. Appendix Table A1 compares minority instances in the training data both before and after applying SMOTE, focusing on demographic characteristics (that are not used in model predictions) and model features (that serve as predictors in the models).

Undersampling

The second approach to address class imbalance is by reducing instances of the majority class, or students who did not exit early. I do this with an undersampling technique proposed by Menardi & Torelli (2014) that randomly undersamples instances of the majority class without replacement.[13]

I utilize the "ROSE" package in R developed by Lunardon et al. (2014) to randomly drop a subsample of the majority class from the training data. The downsized training data have 3,102 observations with an equal number of class instances.

After undersampling the training data, I find that the downsized majority sample closely resembles the original majority sample. Appendix Table A2 presents descriptive statistics of the majority class before and after undersampling. Similar to the synthetic training data built via oversampling, I find that the reduced and original majority samples are comparable across all demographics and model features, except for school mobility.

[13] I am unable to apply undersampling approaches that follow mathematical formulas (such as Near Miss, Condensed Nearest Neighbors (CNN), and Tomek Links), the software packages for these techniques are no longer accessible on the Comprehensive R Archive Network (CRAN).

3.6.4 Evaluating model performance

I evaluate the performance of fifteen models, each constructed using one of five supervised learning algorithms. These models are trained using one of the following data: highly imbalanced original training data, balanced training data that incorporates synthetic instances of the minority class, or balanced training data with reduced instances of the majority class. Table 16 provides a breakdown of the models by its training data and algorithm. All the models employ cross-validation where each model is trained on the training data (i.e., 6th grade students in fall 2010) and evaluated on a separate test data (i.e., 6th grade students in fall 2011). I assess model performance by examining how well the trained models make predictions on test data. The remainder of this dissertation categorizes and refers to these models by their respective panel. Models are assigned to a panel on the type of training data used in the learning phase. Panel A encompasses models developed with highly imbalanced original training data; Panel B includes models where the data were balanced through oversampling; and Panel C focuses on models with balanced training data achieved with downsampling.

Table 16: Prediction model overview

Training data	Algorithm	Outcome: early exit
Panel A: Highly imbalanced (98:2)	Logistic regression	1: Student dropped out of high school in 9th or 10th grade 0: Student completed 10th grade or more
	Lasso regression	
	Ridge regression	
	Random forest	
	Extreme gradient boosting (XGboost)	
Panel B: Balanced with oversampling of minority class (1:1)	SMOTE Logistic Regression	
	SMOTE Lasso Regression	
	SMOTE Ridge Regression	
	SMOTE Random Forest	
	SMOTE Extreme Gradient Boosting (XGboost)	
Panel C: Balanced with undersampling of majority class (1:1)	US Logistic Regression	
	US Lasso Regression	
	US Ridge Regression	
	US Random Forest	
	US Extreme Gradient Boosting (XGboost)	

Notes: The outcome, early exit, is consistent across all 15 models. SMOTE stands for Synthetic Minority Oversampling Technique and refers to training data that include synthetic instances of the minority class. US stands for undersampling and refers to training data that reduce instances of the majority class.

An intuitive approach to evaluating model performance is to compare the predicted outcomes (i.e., the model's predictions) with the actual, observed outcomes. This is analogous to the standard metric of model performance, the accuracy rate. The accuracy rate is computed as the proportion of instances in the test data that the model correctly classified (Hung et al., 2017; James et al., 2023; Bishop, 2024). A key limitation of the accuracy rate is that it does not provide insight into how well the model performs for each class label. Specifically, it fails to reveal the model's effectiveness in classifying both the positive and negative classes. To address this, I extract additional metrics from the confusion matrix.

I use the example confusion matrix provided in Table 16 to illustrate the process of calculating multiple performance metrics. A confusion matrix is a table that compares predicted and true labels for each class. Because this analysis uses a binary classifier, the comparison of predicted labels (i.e., predicted outcomes) with true (i.e., observed) labels is depicted by a 2 by 2 matrix. The outcome of interest is if a student exited high school in 9th or 10th grade, where instances who exited early form the positive class, and instances otherwise form the negative class. The example matrix compares predicted labels and true labels for 1,000 instances (n = 1,000). The matrix disaggregates instances into four key groups - true negatives (900), true positives (50), false negatives (40), and false positives (50).

Table 17: Example confusion matrix

		TRUE LABELS	
		Exited early	Did not exit early
PREDICTED LABELS	Exited early	10	50
	Did not exit early	40	900

Notes: This table is built for a binary outcome where a student who drops out of high school in 9th or 10th grade is labeled "early exit" and "did not exit early."

The formulas for the accuracy rate and other performance metrics were previously described in Table 5. In this example, the accuracy rate is 91 percent $\left(100 * \frac{900+10}{(900+10+40+50)}\right)$. The true positive rate, also known as sensitivity, is 20 percent $\left(100 * \frac{10}{(10+40)}\right)$. The true negative rate, or specificity, is approximately 95 percent $\left(100 * \frac{900}{(900+50)}\right)$. The precision is 17 percent $\left(100 * \frac{10}{(10+50)}\right)$.

This example highlights the importance of evaluating additional metrics beyond the accuracy rate. Although the model achieves a relatively high accuracy of 91 percent, its sensitivity is notably low at 20 percent. This indicates that the model correctly identifies only 20 percent of the positive instances, or students who exited early.

The analysis findings will focus on sensitivity. Models with low sensitivity are susceptible to misidentifying the target population (students who withdraw), therefore reducing model validity. Conversely, students under false positive counts are less of a concern for stakeholders. The overidentification of students who exit early can still be beneficial for students who marginally persist in school longer, as these students may still benefit from additional interventions and support.

The final metric to evaluate model performance is the AUC score, which captures the area under the ROC curve (see 3.6.2 for additional information about the ROC curve). AUC scores can range from 0 to 1; an AUC score of 0.5 indicates that the model is guessing close to random and a score of 1 indicates perfect model performance. The AUC is interpreted as the probability that the model will provide a higher output for a randomly chosen student who exits early compared to a randomly chosen student who does not exit early (Kroese et al., 2019; Nahm, 202l). For instance, an AUC value of 0.33 indicates that there is a 33 percent probability that the model will correctly identify a student who exits early. I follow the approach of Bowers & Zhou (2019) and apply the Wilcoxon rank sum test to test if AUC values in a panel are statistically significant from that in other panels.

3.7 Empirical strategy for research question 2

The goal of the second research question is to assess whether the fifteen models yield fair predictions for students from marginalized backgrounds, also known as protected attributes. As discussed in 2.6, 2.7 and 3.1, it is essential to ensure that algorithmic decisions are not perpetuating potential biases present in the data itself. I examine the accuracy of model predictions for students grouped by following protected attributes: gender, race and ethnicity, English learner status, disability status, and economic disadvantage. Data on these protected attributes are extracted from students' 8th grade records provided by NCERDC. I evaluate algorithmic fairness with two criteria: Absolute Between-ROC Area (ABROCA) metric and equalized odds metric.

3.7.1 ABROCA slicing analysis

The ABROCA slicing analysis developed by Gardner et al. (2019) detects differential accuracy between student subgroups for a protected attribute. This method proposes "slicing" the data into multiple subgroups and evaluating model performance across these subgroups. The data should be categorized where for every protected attribute, there exists a baseline (or majority) group *b*, and a comparison (or minority) group *c*. For example, to evaluate algorithmic fairness on disability status, the data should be partitioned in such a way that the baseline group are individuals that do not have an identified disability, and the comparison group are individuals with an identified disability.

The metric is calculated by taking the absolute value of the difference in area between the ROC curve of the baseline group, ROC_b, and the ROC curve of the comparison group, ROC_b. The metric is not threshold dependent and captures the divergence in performance across all possible thresholds, t. A lower ABROCA value indicates a smaller difference in predictions, suggesting that there is less unfairness in the model. The ABROCA statistic is formally defined by the following formula:

$$\int_0^1 |\, \mathrm{ROC}_b(\mathrm{t}) - ROC_c(t)| dt$$

I follow a systematic approach for calculating the ABROCA statistic. First, I segment the data into subgroups based on protected attributes, designating the first group as the comparison group and the second group as the baseline group. These attributes include gender (female and male), disability status (having an IEP and not having an IEP), English learner status (Limited English Proficient and not Limited English Proficient), financial hardship (economically disadvantaged and not economically disadvantaged), and race or ethnicity (non-White and White). It is important to note that these subgroups are not mutually exclusive; a student's classification in a baseline or comparison group of an attribute does not influence their classification in any other attribute. Next, I assess model performance for each subgroup by computing their respective ROC curves. Last, I calculate the difference between the subgroups for an attribute by subtracting their respective ROC curves. The ABROCA value is then derived for each attribute *A* across all fifteen supervised learning models developed in the first research question. Each model will have one ABROCA statistic per attribute. Given that there are 5 protected attributes, this analysis will present in a total of 75 ABROCA statistics.

I rely on several statistical tests to determine if the ABROCA values are statistically significant. I follow the approach of Gardner et al. (2019) and perform a Kruskal-Wallis test, a non-parametric test used to assess whether there are significant differences between two or more independent groups. The Kruskal-Wallis test is an omnibus test, meaning it can indicate that at least two values are different, but it cannot specify which values differ from one another (Okoye, & Hosseini, 2024). I use this test to compare all ABROCA values in one attribute. For instance, one Kruskal-Wallis test will compare all 15 ABROCA statistics across the gender attribute, while another test will assess the same statistics grouped in the disability attribute, and so on for each relevant attribute. When the Kruskal-Wallis test rejects the null hypothesis that the

ABROCA statistics are the same for an attribute, I will follow the approach of Xu et al. (2024) and perform a Wilcoxon signed-rank test. The Wilcoxon signed-rank test will be used to compare ABROCA values within the same attribute and compare whether the ABROCA value differs against zero.

3.7.2 Equalized odds metric

The second approach to evaluate algorithmic fairness is by computing the equalized odds metric. Hardt et al. (2016)'s seminal work argues that equalized odds are achieved when both the true positive rate and false positive rate are equal among the baseline and comparison groups of attribute *A*, A_b and A_c. This criterion is formally satisfied by when:

$$P\left\{\hat{Y} = 1 \,\middle|A = A_c\,, Y = 1\right\} = P\left\{\hat{Y} = 1 \,\middle|A = A_b\,, Y = 1\right\} \quad (1)$$

$$P\left\{\hat{Y} = 1 \,\middle|A = A_c\,, Y = 0\right\} = P\left\{\hat{Y} = 1 \,\middle|A = A_b\,, Y = 0\right\} \quad (2)$$

where the first condition is that the predicted true positive rate is equal across both groups pf protected attribute *A*, and the second condition is the same but for the false positive rate (Hardt et al., 2016; Dunkelau, J., & Duong, 2022). Equalized odds relies on a fixed tolerance, meaning that it is dependent on the decision threshold. To ensure that this specification is not dependent on one threshold, I compute equalized odds metrics for the two models that apply a different decision threshold.

To calculate the equalized odds criterion, I first calculate the sensitivity (i.e., true positive rate) for every A_b and A_c in attribute *A*. To compare the sensitivity for each subgroup, I create a sensitivity equalized odds ratio that is the quotient of $\frac{sensitivity_b}{sensitivity_c}$. I repeat this process to calculate the false alarm (i.e., false positive rates) and arrive at the false alarm equalized odds ratio.

3.8 Empirical strategy for research question 3

The third research question interprets model findings to understand which aspects of middle school engagement are associated with early exit. To achieve this, I rely on post-hoc explainability techniques to analyze and interpret the decision-making process of machine learning models. These post-hoc methods differ depending on the algorithm utilized. For regression models (such as those employing logistic, lasso, or ridge regression), I interpret the coefficients produced by the models. For tree-based models (random forest and XGBoost), I present visual representations of key model features with feature importance plots and SHAP plots.

This question utilizes post-hoc explainability methods to interpret model findings. Rather than focusing on a single model, I include findings from all undersampled models to examine overlap in features that are predictive of early exit. Next, I take one of the undersampled models – the XGboost model – and use additional approaches to interpret what features are predictive of early exit. Specifically, I use a feature importance plot with gain values and a Shapley Additive exPlanations (SHAP) beeswarm plot.

3.8.1 Interpreting regression models

For models that were built using logistic regression, I rely on Wald (*z*) confidence intervals of and *z*-tests to determine if a coefficient statistically differs from zero. It is important to note that the objective functions of regularized regressions – lasso and ridge regressions – include a penalty term to shrink coefficients toward zero. As such, regularized regressions do not provide Wald confidence intervals or *z*-tests. In the case of lasso regression, some coefficients are shrunk precisely to zero, resulting in a sparse model that includes only non-zero coefficients (Hastie et al., 2009a; 2009b). Therefore, for models that use lasso regression, I extract all predictors with non-zero coefficients and rank them according to their magnitude. In ridge regression, the penalty term reduces coefficients toward zero but does not shrink them to exactly zero. Consequently, for models that use ridge regression, I extract predictors with coefficients exceeding a value of 0.05 and rank them based on their magnitude.

3.8.2 Interpreting tree-based models

For tree-based models, I follow standards of quality and create a feature importance plot that ranks features by the extent to which that feature was used to optimize accurate predictions (Cutler et al., 2012; Khan et al., 2024). The feature importance for random forest models is computed with the percent increase in node purity. Node purity captures the reduction in sum of squared errors when a feature is chosen to split.[14] For XGboost models, the feature importance plot ranks features by its gain. The gain metric is calculated by averaging each feature's contribution across all trees in the model and is interpreted as the proportion of accurate predictions influenced by that feature. For example, a Gain value of 0.2 indicates that 20 percent of accurate predictions were driven by the optimization of that feature. A drawback of the Gain metric is that it is not additive; the optimization of one feature is dependent on the other features present in this model. For this reason, I explore other approaches to interpret tree-based models.

The second post-hoc analysis for tree-based models is a Shapley Additive exPlanations (SHAP) analysis. Derived from game theory, a Shapley value averages differences in predictions over all combinations of model features. This is different from the gain method that instead builds an order of model features based on their position in the tree (Shapley, 1953). An advantage of the SHAP analysis is that it breaks down feature importance by the categories of

[14] Increase in node purity is analogous to the Gini Index, which is considered the standard metric for evaluating features of random forest models (James et al.., 2023).

the outcome variable, where users can understand the directionality of the association between the model feature and outcome.[15] The magnitude of the SHAP value reflects the strength of the feature. The SHAP value is represented in the same unit as the outcome of interest (i.e., likelihood that a student exited early) (Štrumbelj and Kononenko, 2010; Kunapuli, 2023; Khan et al., 2024). Another advantage of this approach is that SHAP values are additive, meaning that the contribution of each feature can be computed independently and then summed up. Similar to regression models, a SHAP analysis provides coefficients to describe the magnitude and directionality of each mode feature. With this approach, one can compute the predicted likelihood of a student exiting early with the formula:

$$\log odds(Y_i) = base\ value + \sum_{i=1}^{k} \beta_k X_k$$

where the base value represents the mean value of the outcome (in binary classification, this would be the equivalent of the proportion of instances of the minority class). The base value can be added to the sum of SHAP values, allowing for a local and precise interpretation of model outputs (Awan, 2023).

Chapter Summary

This chapter reviewed the analytic approach for my dissertation. I rely on North Carolina administrative student records from 2011 to 2018 to build a prediction model that flags students who are at risk of dropping out of high school in 9th or 10th grade. I employ data science methods to explore three research questions 1) develop prediction models with various statistical approaches and techniques and evaluate model accuracy, 2) examine each model's ability to

[15] A recent study found that the SHAP value method was less biased than the gain metric (Lundberg & Lee, 2017).

provide equitable predictions for students from marginalized backgrounds, and 3) interpret model findings to identify salient predictors of early exit.

For the first question, I develop 15 prediction models that employ one of the following approaches: logistic regression, lasso regression, ridge regression, random forest, and XGboost. Each model will have three variations based on the type of training data that was used in the model development phase: original, highly imbalanced data, balanced data with oversampling of minority instances, and balanced data with undersampling of majority instances. I test model accuracy of these models using AUC values, accuracy, sensitivity, and specificity.

The second question assesses the extent to which the models developed in the first question provide fair predictions for students in protected attributes. I assess algorithmic fairness using two approaches: the ABROCA metric (Gardner et al., 2019) and equalized odds metric (Hardt et al., 2016).

For the third research question, I interpret model findings for a subset of models. I extract and rank the salient predictors from the subset of models to better understand which factors are associated with early exit, as well as to understand potential overlap in relevant predictors across all the models. I rely on approaches such as feature importance plots and Shapley Additive exPlanations (SHAP) plots to understand which features are predictive of early exit

CHAPTER 4: RESULTS

Chapter Introduction

This chapter presents the findings of this dissertation. The findings are organized by the three research questions that guide this analysis.

The first question evaluates the performance of 15 models that vary by machine learning algorithm (i.e., statistical approach) and by the data used for model training, or the process of teaching a model to recognize patterns in data. Model performance is evaluated using multiple criteria, such as the area under the receiver operating characteristic (ROC) curve, accuracy rate, and accuracy rates for student subgroups based on their outcome label (i.e., “exited early” or “did not exit early”). The findings suggest that models that are trained with original, imbalanced data have high prediction accuracy of 97 percent but exhibit very low sensitivity, or proportion of correct predictions for minority instances. I find that the models that include resampling techniques – either oversampling minority instances or undersampling majority instances in the training data – greatly increases the sensitivity but at the cost of a lower specificity and a generally lower accuracy. Regression-based models tend to perform similarly across both types of resampled training data, whereas tree-based models tend to have higher sensitivity when trained with downsized training data than those with oversampled training data.

The second question assesses the extent to which the same models are providing fair predictions that do not reinforce or create discriminatory practices. I use the ABROCA slicing analysis and equalized odds metric to evaluate algorithmic fairness. The ABROCA statistics suggest that all models, regardless of the data it was trained on or by its algorithm, tend to discriminate student predictions based on English proficiency status and disability status. The equalized odds metrics reveal that the undersampled logistic regression, compared to the undersampled XGboost model, provides higher sensitivity (i.e., true positive rate) at the penalty of higher false alarms (i.e., false positive rates), with both models exhibiting the closest equalized odds ratios for the gender attribute.

The third question seeks to analyze and interpret the features that were identified from the models developed in the first question. First, a comparison of model features from all undersampled models suggests a strong overlap in which features are predictive of early exit. The models collectively rank age as the strongest predictor of early exit, followed by middle school absences and being chronically absent. Additional post-hoc analyses of the undersampled XGboost model reveal heterogeneity in the association between age and early exit, and that the association between binary features and early exit may be more precise.

4.1 Question 1 findings

This question evaluates the predictive accuracy of the fifteen models described in Table 15. The primary objective is to develop prediction models capable of identifying students at risk of dropping out of high school during their 9th or 10th grade years. To address this, I employ several supervised learning algorithms, which are statistical techniques that learn from input (or training) data where the outcome, referred to as the 'label,' is known. These algorithms construct inferences, recognize underlying patterns, and establish associations from the training data. In educational research, logistic regression is commonly employed as the standard method for predicting binary outcomes. However, this analysis also incorporates more sophisticated learning algorithms that are less frequently utilized in educational contexts but are more prevalent in data science. Specifically, I include two regression-based methods—lasso and ridge regression—as well as two tree-based approaches—random forest and extreme gradient boosting (XGboost).

All models were trained using data from first-time 6th grade students in the 2010-11 school year. The first 5 models in Panel A were trained using the original training data which exhibited class imbalance, a common issue in predictive modeling where minority instances (e.g., students who exited early) are significantly outnumbered by majority instances (e.g., students who did not exit early). In the training data, students who exited comprised only 2 percent of the sample, while those who did not exit early accounted for 98 percent. Class imbalance can be a challenge for prediction models as they typically prioritize predictions for majority class, often at the cost of inaccurately predicting the labels for the minority class (He & Ma, 2013; James, 2023).

To address this issue, I develop two additional sets of models that employ the same supervised learning algorithms but with balanced training data, ensuring an equal ratio (1:1) of students who exited early to those who did not. The second set of models in Panel B address class imbalance using a resampling technique called SMOTE, (short for Synthetic Minority Oversampling Technique) that essentially clones the minority instances in the training data so that both classes are balanced (Chawla et al., 2002; Bunkhumpornpat et al., 2012). The third and final set of models in Panel C resamples by reducing instances in majority class until it is the same size as the minority class (Menardi & Torelli, 2014; Lunardon et al., 2014).

The goal of a prediction model is to provide accurate predictions for new, unseen data. Therefore, the standard way to evaluate model performance is to measure its accuracy in classifying labels for test data, a sample that was not used during the training phase. All models were assessed with a separate testing sample of first-time 6th grade students in Fall 2011.

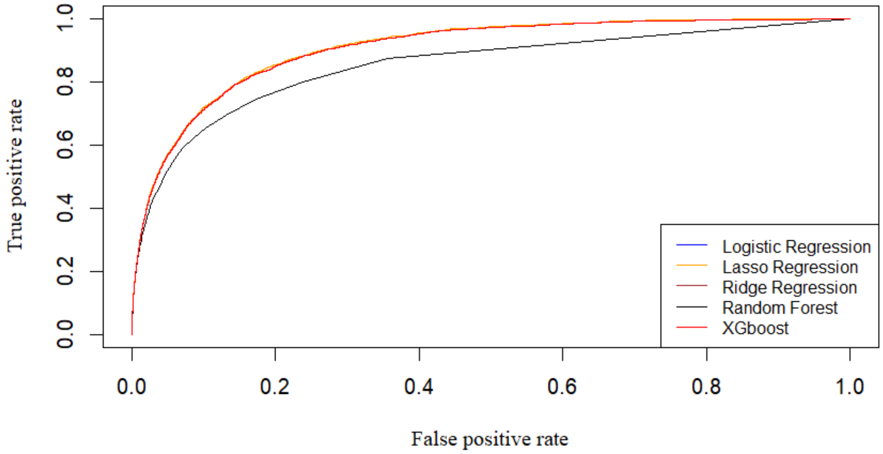
The following two subsections summarize the findings for the first research question; the first subsection examines the ROC curves, and the second subsection examines the accuracy rates for each model.

4.1.1 ROC curves

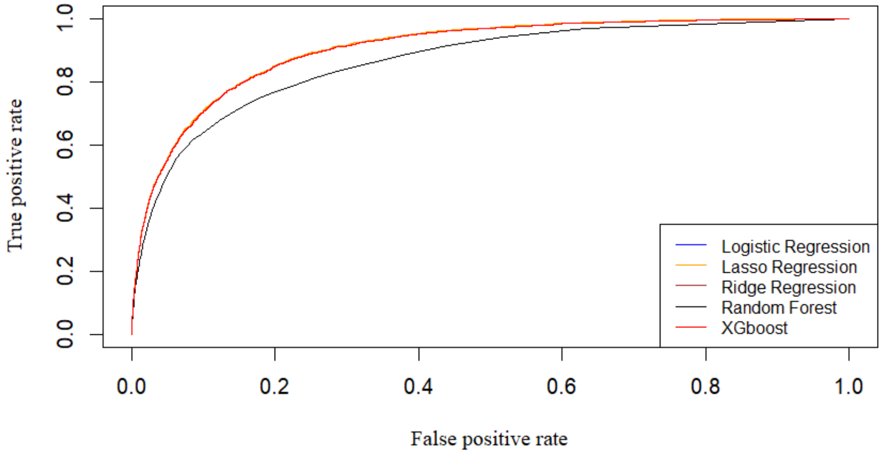
I examine receiver operating characteristic (ROC) curves as visual evidence of model performance. ROC curves illustrate the trade-off between true positives and false positives, offering a clear view of how well the model distinguishes between classes. By showing model performance across various decision thresholds, ROC curves are particularly valuable since they are not reliant on a single threshold. In contrast, other performance metrics are tied to a specific threshold, making ROC curves helpful in depicting a more comprehensive evaluation of model performance (Streiner & Cairney, 2007; Bowers & Zhou, 2019; Nakas et al., 2023). Figure 5 presents the ROC curves of all fifteen models.

Figure 5: ROC curves

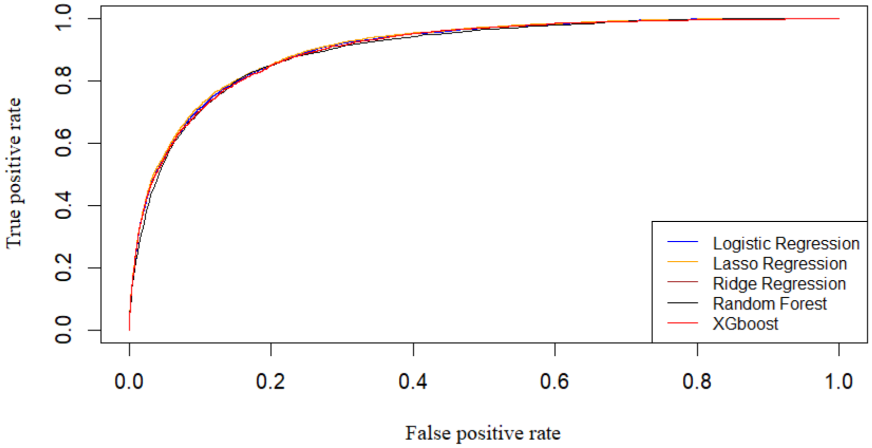
Panel A: Models trained with original, imbalanced training data



Panel B: Models trained with oversampled, balanced training data



Panel C: Models trained with undersampled, balanced training data



The ROC curves in Panel A demonstrate that most of the machine approaches perform similarly across various thresholds, with the XGboost and lasso regression models exhibiting slightly higher sensitivity (true positive rate) than the other models. These curves help guide the tuning of Panel A models. Based on these ROC curves, I decided to set the decision threshold for the Panel A models at 0.2, where instances with predicted probabilities of 0.2 or higher are classified as 'early exit.'

Panel B ROC curves illustrate steeper curves than that of Panel A, demonstrating an increase in the true positive rate (i.e., y-axis) across all potential false positive rates. The convergence of the ROC curves in Panel B beyond where x = 0.5 suggests that Panel B models will exhibit similar tradeoffs between the two criteria at a threshold of 0.5 or above. I set the decision threshold to 0.4 for all models except for random forest. Because the random forest curve is lower than that of other curves, that means that it is not performing as well as other models. For this reason, I assign a lower decision threshold of 0.1 for random forest.

Panel C ROC curves strongly overlap, suggesting that the five models can similarly distinguish between classes. In fact, it is challenging to identify an optimal model based solely on these curves. Additionally, the shape of the curves in Panel C closely resembles those from all the models in Panel B, with the exception of the random forest model. This reinforces the need to further evaluate model performance using additional metrics.

4.1.2 Model performance

I evaluate model performance using four metrics: area under curve (AUC), accuracy, sensitivity, specificity, and precision. The AUC score ranges from 0 to 1 and measures the area under the receiver operating characteristic (ROC) curve, capturing the model’s ability to distinguish between classes. Accuracy is the proportion of correct predictions. Sensitivity indicates the proportion of predicted instances for the minority instances, or students who exited early. In contrast, specificity measures the proportion of correct predictions for the majority class (i.e., students who did not exit early). Precision is the proportion of true positives out of all predicted positive instances. Table 18 presents the performance metrics for each model.

Table 18: Model Performance

Model	**AUC**	**Accuracy**	**Sensitivity**	**Specificity**	**Precision**
Panel A: Models trained with original imbalanced data (*n* = 89,716)					
Logistic Regression	0.90	0.97	0.20	0.99	0.49
Lasso Regression	0.90	0.97	0.19	0.99	0.50
Ridge Regression	0.91	0.97	0.17	0.99	0.51
Random Forest	0.85	0.97	0.31	0.99	0.37
XGboost	0.91	0.97	0.33	0.99	0.39
Panel B: Models trained with oversampled data (*n* = 175,021)					
SMOTE Logistic Regression	0.90	0.83	0.81	0.83	0.11
SMOTE Lasso Regression	0.91	0.79	0.86	0.79	0.09
SMOTE Ridge Regression	0.91	0.82	0.84	0.81	0.06
SMOTE Random Forest	0.86	0.82	0.75	0.83	0.10
SMOTE XGboost	0.90	0.86	0.73	0.86	0.12
Panel C: Models trained with undersampled data (*n* = 3,102)					
US Logistic Regression	0.90	0.81	0.84	0.81	0.14
US Lasso Regression	0.91	0.82	0.84	0.82	0.17
US Ridge Regression	0.91	0.81	0.84	0.81	0.10
US Random Forest	0.90	0.74	0.89	0.74	0.08
US XGboost	0.91	0.79	0.87	0.79	0.10

Notes: SMOTE stands for Synthetic Minority Oversampling Technique and refers to synthetic data used to upsample the minority class in the training data. US stands for undersampling and refers to training data that reduce instances of the majority class.

In examining Panel A model performance, the first two metrics – AUC and accuracy – suggest that all models are making highly accurate predictions. The AUC values reflect an 85 to

91 percent likelihood that each model will correctly identify a random student who exited early as having a higher probability of dropping out than a randomly chosen student who did not. Moreover, all models correctly predicted exit status for 97 percent of the test sample. While these results may initially seem promising, the subsequent three metrics reveal that model performance is not as strong as it appears. The specificity metric of 0.99 indicates that the models are accurately predicting the exit status of 99 percent of students who remained in high school beyond 10th grade. However, Panel A models fall short in terms of sensitivity. All models provide poor sensitivity that falls between 17 to 33 percent. Notably, regression models – logistic, lasso, and ridge regression – demonstrate the lowest sensitivity, while tree-based models (XGBoost and random forest) provide slightly higher sensitivity of 31 to 33 percent. This indicates that a Panel A model, at best, will not correctly label over two-thirds of all students who had exited early. A precision of 0.49 reflects that among all the instances predicted to have exited early, 49 percent of them were accurate. The precision rates suggest that regression models have higher precision than tree models, although all models overclassified at least half of the true at-risk population.

Panel B presents performance of models that were trained using balanced data that included synthetic instances of the minority class achieved via SMOTE. Compared to Panel A models, Panel B models observe similar AUC values but offer significant improvements in sensitivity. However, this increase in sensitivity comes at the expense of reduced overall accuracy and lower specificity. The pattern of model performance remains consistent across algorithm types, with regression-based models maintaining higher sensitivity and tree-based models exhibiting greater specificity. However, it is worth noting that the random forest models in Panel A and B have the lowest AUC values, suggesting a lower model performance.

Panel C presents the performance of models trained using balanced data created by downsampling majority instances to match the number of minority instances. Compared to the models in Panel B, those in Panel C exhibit slightly lower accuracy rates, with tree-based models experiencing the most pronounced decline of 8 to 9 percentage points. However, both XGboost and random forest models demonstrate substantial improvements to sensitivity relative to their counterparts in first two panels. Notably, these two models surpass the sensitivity of regression-based models in Panel B, albeit at the cost of a slight reduction in specificity. Among all 15 models, undersampled XGboost random forest exhibit the highest sensitivity of 89 percent with SMOTE XGboost providing both the highest accuracy and specificity.

In summary, regression-based models generally exhibit consistent performance across both types of resampled training data. The Wilcoxon rank sum test confirms that models trained on imbalanced data have lower AUC values ($p < .001$), and that there are no statistically significant differences in AUC values in Panel B and C models ($p > 0.05$). A closer comparison of models in Panels B and C suggest that tree-based models demonstrate greater sensitivity to reductions in training data size, often experiencing a more pronounced decline in performance with downsized datasets compared to oversampled ones. This discrepancy highlights the inherent robustness of regression models to variations in data availability, while tree-based models may be more reliant on larger sample sizes to maintain predictive accuracy. This comes at a cost where both Panel B and C models exhibit lower precision, with Panel C models offering slightly higher precision.

Although the findings from this analysis offer valuable insights, further information is required to evaluate model performance across various student subgroups, with particular attention given to students from marginalized backgrounds. The evaluation of model fairness is essential to ensure that the models are equitable and provide accurate predictions for all students, especially those who may be at higher risk of exiting early.

4.2 Question 2 findings

This question evaluates the extent to which the fifteen models provide fair predictions for students from protected student attributes (i.e., historically marginalized backgrounds). I first "slice" the test data into subgroups for each attribute, designating one as the baseline (non-protected) group and the other as the comparison (protected) group. These attributes include gender (female and male), disability status (having an IEP and not having an IEP), English learner status (Limited English Proficient and not Limited English Proficient), financial hardship (economically disadvantaged and not economically disadvantaged), and race or ethnicity (non-White and White). I assess algorithmic fairness across subgroups using two fairness metrics: Area Between ROC Curves (ABROCA) value and the equalized odds statistic.

4.2.1 ABROCA findings

Table 19 presents the ABROCA values for each of the five protected attributes. Each ABROCA value indicates the difference in AUC between that of the baseline group and the comparison group for that attribute. In Panel A, the first value for logistic regression under the gender attribute – 0.006 – indicates the difference in AUC between model performance for male and female students. This difference translates to a 0.6 percent higher likelihood that the Panel A logistic regression will misclassify an instance as an early exit based on the instance's gender.

Table 19: ABROCA Statistics

	PROTECTED ATTRIBUTE				
MODEL	Gender	English Learner Status	Disability Status	Economic Disadvantage	Race/ Ethnicity
Panel A: Models trained with original data (98:2)					
Logistic Regression	0.006	0.090	0.058	0.045	0.033
Lasso Regression	0.005	0.087	0.060	0. 045	0.031
Ridge Regression	0.006	0.089	0. 058	0. 045	0.032
Random Forest	0.021	0.077	0.041	0.034	0.018
XGboost	0.006	0.095	0.060	0.048	0.034
Panel B: Models trained with SMOTE data (1:1)					
SMOTE Logistic Regression	0.004	0.082	0.059	0.042	0.029
SMOTE Lasso Regression	0.004	0.080	0.061	0.042	0.023
SMOTE Ridge Regression	0.004	0.080	0.061	0.043	0.029
SMOTE Random Forest	0.017	0.090	0.043	0.038	0.027
SMOTE XGboost	0.006	0.100	0.063	0.032	0.035
Panel C: Models trained with undersampled data (1:1)					
US Logistic Regression	0.003	0.086	0.056	0.042	0.027
US Lasso Regression	0.005	0.092	0.056	0. 043	0.040
US Ridge Regression	0.007	0.101	0.046	0.043	0.041
US Random Forest	0.013	0.093	0.059	0.044	0.033
US XGboost	0.007	0.100	0.062	0.048	0.038

Notes: SMOTE stands for Synthetic Minority Oversampling Technique and refers to synthetic data used to upsample the minority class in the training data. Undersampling refers to training data with reduced instances of the majority class in the training data. The ABROCA value is calculated by taking the difference in AUC between that of the baseline (majority) group and that of the comparison (minority) group of that attribute. A Kruskal-Wallis was used to detect differences in ABROCA values within a protected attribute. For attributes that rejected the null hypothesis of the Kruskal-Wallis test, a Wilcoxon signed-rank test identified which values were statistically significant from zero. ***$p < 0.01$

First, the ABROCA findings generally demonstrate little variability in statistics within an attribute. I rely on the Kruskal-Wallis test that tested the null hypothesis that the ABROCA values are the same within attribute. I conduct the Kruskal-Wallis test once for each attribute. The tests confirmed that there are no differences in subgroup performance for each attribute ($p >$ 0.05). This means that that all models perform similarly for any one attribute (e.g., there is no statistically significant difference between an ABROCA values of 0.003 and 0.017 under the gender attribute).

Second, the results reveal substantial variability in ABROCA values across protected attributes. The highest ABROCA values were observed for the English proficiency attribute, ranging from 0.077 to 0.101, followed by the values under the disability attribute. Meanwhile, the gender attribute exhibited the smallest differences in subgroup performance. Notably, any ABROCA value for English proficiency is at least 3.5 times larger than those for gender.

There is no known empirical guidance that suggests a threshold for ABROCA values that would designate a protected attribute as discriminatory. Despite this not being discussed in the field, I argue that ABROCA values alone do not provide sufficient information to determine whether a model discriminates based on a protected attribute. Further analysis is necessary to examine the directionality of model performance – for example, whether the model performs worse for Limited English Proficient students or non-Limited English Proficient students.

In summary, the ABROCA slicing analysis reveals two key insights: 1) it did not detect variation in algorithmic fairness across models for any single attribute, indicating that all models performed similarly for each attribute, and 2) it uncovered significant disparities in model performance across subgroups, raising the need for additional exploration of certain protected attributes.

I contend that the ABROCA slicing results, in isolation, do not provide a comprehensive understanding of algorithmic fairness. The ABROCA statistics are derived from AUC (Area

Under the Curve) values, which, as discussed in Section 4.1.2, should not be used as the sole metric to evaluate model performance. Although a model may exhibit a high AUC, it aggregates performance across all decision thresholds, which may not necessarily reflect fairness in decision-making. Since model accuracy depends on the decision threshold used to assign labels to instances, it is plausible that algorithmic fairness could vary not only across different thresholds but also between algorithms. Therefore, I extend the analysis to examine how models perform across subgroups defined by protected attributes, providing a more nuanced evaluation of fairness.

4.2.2 Equalized odds findings

The second fairness criteria, the equalized odds metric, is achieved when both subgroups of an attribute share the same sensitivity (i.e., true positive rate) and false alarm rate (i.e., false positive rate) for each subgroup (Hardt et al., 2016). This metric differs from the ABROCA slicing analysis in that the metric relies on a decision threshold to compare model performance across subgroups. I present abridged results that only share equalized odds metrics for a subset of the fifteen models.

Among the fifteen models, I selected the two models that met two criteria. I rely on earlier findings from this analysis and select models that 1) provided the highest mean of sensitivity and sensitivity (derived from Table 18 performance metrics), and 2) had smallest ABROCA values across all attributes. This decision-making process narrowed it down to two models: the undersampled logistic regression model and the undersampled XGBoost model.

Table 20 provides the equalized odds metrics for the two selected models.[16] As described in 3.7.2, the equalized odds criterion compares the sensitivity (i.e., the true positive rate) and the false alarm rate (i.e., the false positive rate) between the baseline (A_b) and comparison (A_c) subgroups in each attribute (Hardt et al., 2016; Wadsworth et al., 2018; Baker & Hawn, 2021). The baseline and comparison groups correspond to the same subgroups established in the ABROCA slicing analysis, with the baseline capturing students from the non-protected group and the comparison consisting of students from the protected group. For each attribute *A*, sensitivity ratio is computed by dividing the sensitivity of A_b with the sensitivity of A_c. The false alarm ratio is computed with the same steps but using the false alarm rates. The equalized odds ratio is the quotient of the sensitivity ratio divided by that of the false alarm ratio.

The sensitivity and false alarm rate ratios can be interpreted as follows: a ratio of exactly 1 indicates that the respective metric (either sensitivity or false alarm rate) is equal across subgroups, thus meeting one of the criteria for equalized opportunity. A ratio below 1 suggests bias towards the comparison group, while a ratio above 1 indicates bias towards the baseline group. The closer a ratio is to 1, the closer the model is to achieving equalized odds.

Table 20: Equalized odds findings

	US XGBoost			US Logistic Regression		
	Baseline	Comparison	Equalized Odds Ratio	Baseline	Comparison	Equalized Odds Ratio
Gender						
Sensitivity	0.89	0.82	1.09	0.93	0.89	1.04
False alarm rate	0.27	0.17	1.59	0.35	0.25	1.40
English Learner Status						

[16] The decision thresholds for these models follow what was used in the analysis of the first research question. The decision thresholds for the undersampled XGBoost and undersampled logistic regression are 0.4 and 0.6, respectively.

Sensitivity	0.87	0.82	1.06	0.92	0.89	1.03
False alarm rate	0.20	0.37	0.54	0.30	0.47	0.64
Disability Status						
Sensitivity	0.84	0.93	0.90	0.89	0.97	0.92
False alarm rate	0.17	0.48	0.35	0.26	0.63	0.41
Economic disadvantage						
Sensitivity	0.72	0.90	0.79	0.84	0.94	0.89
False alarm rate	0.08	0.36	0.21	0.19	0.44	0.43
Race/Ethnicity						
Sensitivity	0.86	0.88	0.98	0.91	0.92	0.99
False alarm rate	0.15	0.27	0.56	0.25	0.36	0.69

Notes: US is short for undersampling and refers to training data with reduced instances of the majority class. The sensitivity is the true positive rate, or the proportion of the subgroup that was correctly labeled as an "early exit." The false alarm rate is the false positive rate, or the proportion of the subgroup that was mislabeled as an "early exit." The equalized ratio in columns 3 and 6 is computed by dividing the baseline metric (either sensitivity or false alarm rate) by same comparison group metric.

The findings in Table 20 demonstrate that the logistic regression model provides higher sensitivity rates for subgroups across all attributes. This suggests that the logistic regression model is more effective than the XGboost model at correctly labeling students as ‘exited early.’ However, this comes with a trade-off: the logistic regression model also exhibits a relatively higher false alarm rate for subgroups in all attributes. This indicates that the logistic regression model is more susceptible to misclassifying students who did not exit early as having ‘exited early,’ leading to an increased number of false positives. This higher mislabeling suggests that while the logistic regression model is better at detecting early exits, it sacrifices some degree of precision in its predictions, leading to more instances of incorrect classifications.

The equalized odds ratios in columns 3 and 6 reflect the model's balance, or its ability to maintain similar sensitivity and false alarm rates across both the baseline and comparison subgroups of an attribute. In examining the odds ratios, I find that both the XGboost and logistic regression models provide similar sensitivity ratios across most attributes, with sensitivity ratios in the logistic regression model providing ratios that are slightly closer to 1.

In comparison to the XGboost model, the logistic regression model provides false alarm equalized odds ratios that are closer to 1 across all attributes except for gender. The economic disadvantage false alarm ratio for the XGboost model, however, is significantly lower than that for the logistic regression model (0.21 and 0.43, respectively).

The equalized odds criterion is theoretically achieved when the ratios between the sensitivity and false alarm ratios in an attribute are both equal. The ratios provided in Table 19 suggest that although the equalized odds criterion is not formally satisfied by any attribute, model predictions tend to discriminate the least for students based on their gender, race and ethnicity. Moreover, the ratios in logistic regression exhibit a smaller gap between sensitivity and false alarm ratios, exhibiting a more "balanced" approach in its model performance between non-protected groups and protected groups of an attribute.

4.3 Research Question 3 findings

This question utilizes post-hoc explainability methods to identify salient predictors of students who exited high school in 9th or 10th grade. Rather than focusing on a single model, I first examine model features from all undersampled models.

The second part of this research question takes a deep dive of the undersampled XGboost model and uses additional post-hoc approaches such as feature importance plot and Shapley Additive exPlanations (SHAP) beeswarm plot to rank features that are predictive of early exit.

4.3.1 Predictors of early exit across undersampled models

I extract relevant predictors from the logistic regression models by first selecting coefficients that reject the null hypothesis of the Wald *z*-test that the coefficient is statistically different from zero at the 95 percent confidence interval or higher (where $p > 0.05$). For regularized regression (i.e., lasso and ridge regression) models, I extract predictors with coefficients larger than 0.1. For tree-based (i.e., random forest and XGBoost) models, I rely on feature importance plots that ranks model features by a given metric. The random forest plot provides the percent increase in node purity and the XGBoost feature importance plot provides the gain metrics. Finally, I rank the extracted predictors from each model by their magnitude.

Table 21 provides a color-coded chart to organize predictors of early exit, where predictors are disaggregated into four categories: strongly predictive, moderately predictive, weakly predictive, and not predictive. The darkest shade of blue reflects the indicators most predictive of early exit, while teal reflects the ones that are moderately predictive, and light blue reflects the indicators that are weakly predictive. Predictors with an unshaded box were either not identified as predictive, had a coefficient magnitude less than 0.1, or were not statistically significant at the 95 percent confidence level (statistical significance was only observed for the logistic regression models).

Features for the logistic, lasso, and ridge regressions were ranked by coefficient magnitude where coefficients with an absolute value between 0.1 and 0.33 were weakly predictive; those between 0.33 and 1 were moderately predictive; and those at 1 or above were strongly predictive. Features for the random forest were ranked by the percent increase for node purity where features at or above 100 percent were strongly predictive, and features between 30 to 50 percent were moderately predictive, and those between 10 to 30 percent were weakly predictive. Finally, features for XGboost models were categorized using the gain metric provided by feature importance plots where gain values greater than or equal to 0.1 were strongly predictive; gain values between 0.05 and 0.1 were moderately predictive; and gain values between 0.03 and 0.05 were weakly predictive.

Table 21: Features predictive of early exit from high school

Legend

■ Strongly Predictive ▩ Moderately Predictive ░ Weakly Predictive

	Panel C: Undersampled Models				
	Logistic Regression	Lasso Regression	Ridge Regression	Random Forest	XGBoost
Student characteristics					
Economically disadvantaged	▩	▩	▩	▩	░
Age	■	■	■	■	■
IEP					
Ever had an IEP in a middle grade			░		
Limited English proficient					
Ever limited English proficient in 8th grade	■	▩	▩		
Urban					
Suburban					
Town		░	░		
Rural					
Academic information					
Not math proficient in 6th grade	▩	▩	▩	▩	
Not math proficient in 7th grade	▩	▩	▩	▩	░
Not math proficient in 8th grade	▩	░	▩		
Not math proficient in all middle grades	░				
Not reading proficient in 6th grade			░	▩	
Not reading proficient in 7th grade				▩	
Not reading proficient in 8th grade		░	░	▩	░
Not reading proficient in all middle grades					
Attendance information					
Absence rate in 6th grade	■	■	■	■	░

Absence rate in 7th grade	■	■	■	■	▨
Absence rate in 8th grade	■	■	■	■	▨
Chronically absent in 6th grade					
Chronically absent in 7th grade					
Chronically absent in 8th grade			▨	▨	░
Ever chronically absent in a middle grade	▨	▨	▨	■	■
Chronically absent in all middle grades					
School mobility in 6th grade		░	░		
School mobility in 7th grade					
School mobility in 8th grade		▨	▨		
School mobility in a middle grade	▨	▨	░	▨	
Discipline information					
OSS in 6th grade					
OSS in 7th grade		░	░		░
OSS in 8th grade		░	░		░
OSS in a middle grade			░	▨	
ISS in 6th grade		░	░		
ISS in 7th grade					
ISS in 8th grade	▨	▨	▨		░
ISS in a middle grade					
Ever suspended in a middle grade				▨	
ST suspension in a middle grade		▨	░	▨	░
LT suspension in a middle grade			▨		

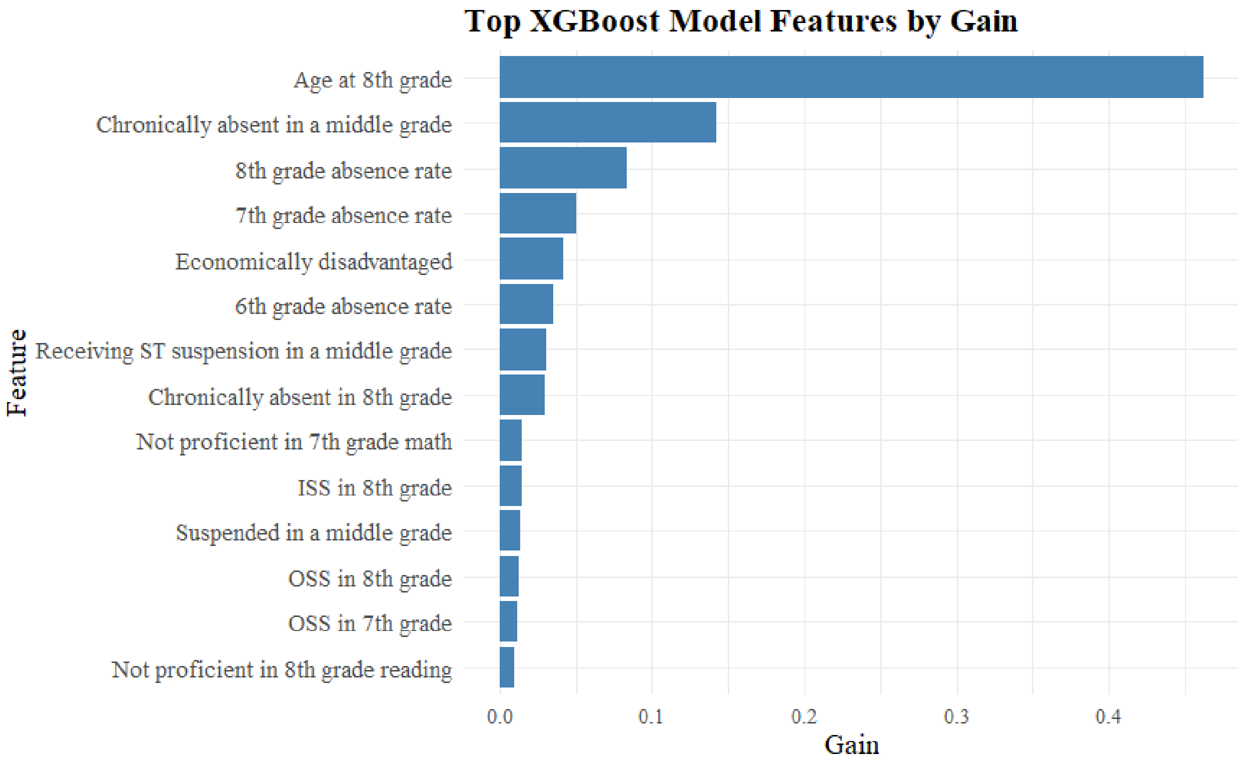
Notes: Undersampling refers to resampled training data with downsized majority instances to match the number of instances the minority class. The models presented in this table correspond to Models 10 to 15 reported in Table 17. Unshaded boxes indicate that the feature had a coefficient below value 0.1, and for logistic regression, were not significant at the 95 percent confidence level.

Table 21 reveals that age is unanimously a strong predictor of early exit, followed by 6th, 7th, 8th grade absence rates, and later trailed by being chronically absent in a middle grade. Receiving a short-term suspension in a middle grade, being chronically absent in 8th grade, and not being proficient in 7th grade math are moderately to weakly predictive of early exit. The findings suggest strong agreement between the lasso regression and ridge regression output, with both models ranking the predictive importance of features most similarly. All but one feature identified by XGboost (being chronically absent) were also identified by at least 2 other models. Compared to the other undersampled models, the XGboost model provides a very sparse model by identifying a small number of features. Conversely, the ridge regression model provides the greatest number of predictive features.

4.3.2 Deep dive of undersampled XGBoost model

The undersampled XGboost model is further explored with two approaches: a feature importance plot and a SHAP beeswarm plot. Figure 6 illustrates the XGboost model's feature importance plot. It displays gain values, or values that indicate the proportion of accurate predictions that optimized that feature. For instance, a gain value of 0.13 indicates that 13 percent of all correct predictions utilized that model feature.

Figure 6: Undersampled XGboost importance plot



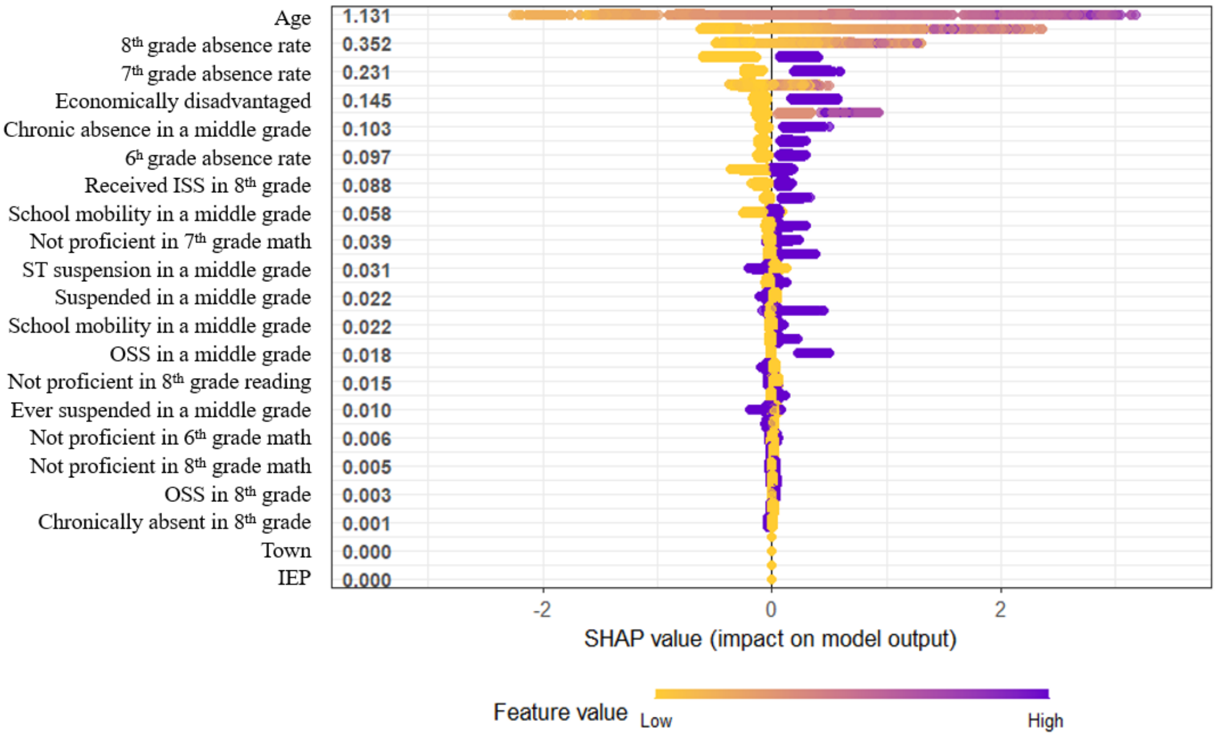
Notes: Gain represents the improvement in accuracy brought on by that specific feature; it provides the proportion of accurate predictions that optimized that feature.

The feature importance plot reveals that age in 8th grade, being chronically absent in a middle grade, followed by 8th grade and 7th grade absences were utilized the most in optimizing predictions. Notably, the inclusion of age results in an average gain of 0.45 splits that use this feature. A drawback of this approach is that decision trees are biased towards features that have more split points. Features are ranked based on the number of splits the feature is involved in. Because continuous features can be split into more levels compared to binary features, continuous features (i.e., age and absence rates) tend to be ranked higher in feature importance. Moreover, this plot does not reveal the directionality of the association between model features and early exit. Recognizing the limitations of gain values, I employ other approaches to interpreting the XGboost model.

I explore the SHAP Beeswarm plot, as presented in Figure 7, which shows how each feature independently influences the final prediction. The features are ranked in descending order. The reported values near the feature name represent the mean SHAP value. The x-axis provides SHAP values that are analogous to the predicted probability provided by the model, or the log-odds that an instance will exit early. The magnitude of the SHAP value indicates the strength of the feature's contribution, where purple dots indicate that the feature pushes the model towards predicting a higher likelihood of exiting early, whereas values in yellow push the model towards a lower likelihood (Cooper, 2021). The directionality of the feature's contribution is captured by the SHAP values, where a positive SHAP value suggest a positive contribution to the prediction, whereas negative values indicate a negative contribution. As described in 3.8.2, the final prediction can be computed using a fixed base value added to the sum of computed SHAP values. The base value is the proportion of instances in the test data who exited early and the mean SHAP value for each value is multiplied by instance *i*'s value for that feature. The model specification is similar to that of a logistic regression. The mean SHAP values provided in Figure 7 allow a formal specification of the undersampled XGboost model, where:

$$\log odds(Y_i) = 2.5 + \ 1.131(Age_i) + 0.352(Grade\ 8\ absence\ rate_i) \ldots + 0.001\ (Grade\ 8\ chronic\ absence_i)$$

Figure 7: Undersampled XGBoost SHAP Beeswarm Plot



The beeswarm plot provides a very similar ranking of model features to that of the feature importance plot. Both plots rank age as the strongest predictor, followed by middle school absenteeism and economic disadvantage.

Figure 7 depicts that when binary features (e.g., math or reading proficiency, chronic absence, or receiving a form of suspension) take a value of 1, then the SHAP value (i.e., log-odds of exit early) is high. The continuous features – age and absence rates for each middle grade – demonstrate significant variation in magnitude and directionality. For middle grade absence rate features, we see that lower values of an absence rate have negative SHAP values (the points extending towards the left are increasingly yellow), but the reverse is not observed. A strong, positive association between absence rate and early exit would be depicted by purple dots for positive SHAP values, but this is not the case. I observe that positive SHAP values for absence rates and age contain many yellow dots with no clear gradient that becomes increasingly purple. This indicates that instances that may be an older age or that exhibit higher absence rates do not have high contributions towards predicting early exit status.

Chapter Summary

This chapter presents the results of this dissertation, organized according to the three research questions guiding the analysis. The first question examines the performance of 15 models that differ in machine learning algorithms (i.e., statistical methods) and in the data used for training. I evaluate model performance with various criteria, such as the area under the curve (AUC) value, overall accuracy, and accuracy rates for student subgroups based on their outcome label (i.e., "exited early" or "did not exit early"). The results indicate that models trained on the original, imbalanced data achieve a high prediction accuracy but have very low sensitivity, meaning they struggle to make accurate predictions for minority instances. I find that models incorporating resampling techniques – either oversampling minority instances or undersampling majority instances – usignificantly improve sensitivity, though at the expense of lower specificity and reduced overall accuracy.

The second question evaluates the fairness of these models. I use ABROCA slicing analysis and the equalized odds metric to assess algorithmic fairness. The ABROCA statistics reveal that all models, regardless of their training data or algorithm, tend to discriminate based on students' English proficiency and disability status. The equalized odds metrics show that the undersampled logistic regression model provides higher sensitivity (true positive rate) than the undersampled XGBoost model, but at the cost of more false positives (higher false positive rate). Both models, however, exhibit similar equalized odds ratios concerning gender.

The third question interprets model findings. A comparison of features across all undersampled models reveals strong consistency in the predictors of early exit. Across the models, age is ranked as the strongest predictor of early exit, followed by middle school absences and chronic absenteeism. Further post-hoc analysis of the undersampled XGBoost model uncovers variation in the relationship between age and early exit, as well as more precise associations between binary features and early exit.

CHAPTER 5: CONCLUSION

Chapter introduction

The final chapter synthesizes the key lessons from this dissertation. It acknowledges the limitations of the analysis and their implications for interpreting findings; discusses the results of each research question and its significance for future research; outlines potential avenues to enhance or extend this work; and offers recommendations that contribute to the development of a robust early warning system.

5.1 Limitations

This section outlines the limitations of this dissertation, which can be grouped into three categories: technical limitations, referring to the drawbacks associated with the choice of statistical software; data limitations, which pertain to student engagement indicators that are not captured in the data; and design limitations, which involve the decisions made during the analysis that influence the interpretation and implications of the findings.

5.1.1 Technical limitations

The decision to conduct this analysis using R programming software limits the potential for broader scientific exploration. Unlike Python, which offers powerful data science tools such as Scikit-learn, R does not provide the same range of capabilities for tasks like hyperparameter tuning and model visualization. Libraries like Scikit-learn offer more advanced options that are crucial for refining models and ensuring they are optimized to their full potential. As a result, the decision to use R constrains the flexibility and depth of analysis that could have been achieved through Python's more robust data science ecosystem.

5.1.2 Data limitations

This subsection discusses limitations related to the administrative data leveraged in this dissertation. First, this analysis does not include coursework engagement data as model features. These data are typically captured in transcript data such as course grades, GPA, or the types of courses taken. This is a significant limitation as prior empirical work have underscored the critical role of academic performance – especially course failure and GPA – in predicting high school dropout (Balfanz, 2009; Bowers & Sprott, 2012a, 2012b; Bowers et al., 2013). Indeed, most studies that predict high school dropout have included course performance as a model feature and have consistently identified it as one of the strongest predictors of early exit (Knowles, 2015; Sorenson, 2019; Sansone, 2019; Lee & Chung, 2019). This gap in my data could disproportionately impact the accuracy of predictions for students who face additional academic challenges. Given the strong relationship between academic performance and the likelihood of dropping out, the absence of more detailed coursework information may drive the lack of predictive power observed in my models, especially for students who are Limited English Proficient.

There is an increasing recognition that attendance, behavior, and coursework (ABC) indicators may not fully capture all dimensions of student's schooling experience. Recent replication studies have found that certain ABC indicators, particularly absence and suspension indicators, may exhibit low accuracy and sensitivity (Bowers & Zhou, 2019). These limitations raise concerns about the validity of using such indicators for early warning systems or predictive modeling, as misclassification or underestimation of at-risk students could lead to ineffective or misguided interventions. This reinforces the importance of considering alternative or complementary metrics to improve the predictive power of dropout prediction models.

Alternative types of data, such as school climate, students’ sense of belonging, socioemotional well-being, and the quality of relationships with teachers, may better capture schooling experiences. Research has shown that these factors significantly influence student engagement and, ultimately, academic outcomes (Jackson, 2018; Balfanz, 2018). The COVID-19 pandemic has further amplified these concerns by introducing a host of challenges for both students and educators, including heightened feelings of loss, alienation, depression, and other mental health struggles. Such issues have profound implications for students' well-being, which in turn affects their academic engagement and success (Snyder, 2022; Su et al., 2022). The disruption caused by the pandemic has underscored the importance of considering mental health and emotional support as indicators in early warning systems. Neglecting these factors may result in an incomplete understanding of the challenges students face in terms of academic disengagement and potential dropout. This gap in the data is a common limitation in research settings that primarily rely on administrative records to build generalizations about students' overall educational experiences. Moreover, this analysis relies on statewide administrative data, further exacerbates this issue. It is very likely that schools, districts, and counties have rich, contextual data that can provide deeper insights into student engagement levels. In short, the lack of contextual data (such as those reported by students and teachers) prevents a holistic understanding of the factors associated with student disengagement.

Lastly, it would be extremely valuable to have data that identifies whether a student has been flagged as at-risk by their school or district, along with key details such as when they were first identified, the specific interventions they received, and if their risk status changed. The inclusion of such information could significantly enhance the development of predictive models and early warning systems because it could illuminate if early identification works, for whom, and under what conditions. Moreover, it can highlight discrepancies in disparities between early identification via statistical modeling (i.e., an early warning system) versus current applications in a K-12 setting. Such data could help to determine whether early identification truly leads to better outcomes and could offer a clearer picture of which strategies work best for supporting at-risk students.

5.1.3 Design limitations

First, this analysis assumes that students who exited the North Carolina public school system have permanently discontinued their schooling journey. This assumption cannot be confirmed as I am unable to observe the trajectory of students who have withdrawn from the state's public system. The outcome being observed could signal various decisions. One possibility is that students had exited the public school system to complete their education in another setting, such as a charter school, a private school, or to take the General Education Development (GED) test and earn a high school equivalency diploma. This possibility may be particularly relevant for studies that predict high school dropout in post-pandemic school years, as recent studies suggest that declining public school enrollment is associated with increases in private school enrollment (Dee, 2023; Lieberman & Riser-Kositsky, 2024).

Second, the analysis includes "stopouts", or students who temporarily discontinue schooling in either 9th or 10th grade and return to the public school system in a subsequent school year. I assign the outcome of "early exit" status to only capture students who do not return to school for the following 4 or 5 school years. As a result, the analysis assigns stopouts – students who leave school temporarily but eventually return – with the outcome "did not exit early". There is a lack of empirical evidence exploring whether student engagement differs between stopouts and those who have remained continuously enrolled in the school system. A lack of distinction between stopouts and students who never stopped may mask patterns of student disengagement that contribute to early exit.

Third, the analysis is unable to build assumptions of what a student's age in 8th grade is indicative of. In other words, it is possible that student age captures institutional decisions that were made prior to 6th grade, such as repeating or skipping a grade.[17] Because the analysis does not examine school engagement before a middle grade, the age feature could be a function of unobserved student behavior.

Fourth, the consistent prioritization of age in generating accurate predictions may be due to the structure of the age feature. With the exception of age, absence rates, and school mobility features, most of the model features are binary. Among the few features that are continuous in nature, the age feature is the largest age spread of values (i.e., standard deviation) amongst all model features (see Table A1 and A2 for descriptive statistics of model features). A decision tree (i.e., a single model used in tree-based models) tends to assign greater importance to continuous features over binary ones. This is because continuous variables can be split at multiple cut points, offering more flexibility in partitioning the data (Zhou & Hooker, 2021). For instance, a feature like age can be split into intervals (e.g., students aged between 13.5 and 14.2 years or those older than 13.7 years), enabling the tree to maximize information gain based on the most relevant cut-off points. In contrast, binary features are limited to just two potential splits, reducing their ability to capture nuanced patterns in the data.

Lastly, a limitation of this analysis lies in the choice of approach for addressing the third research question. While SHAP beeswarm plots were used to interpret models, alternative

[17] North Carolina is one of the few states where kindergarten is not compulsory. Students enrolled in kindergarten can move to the next grade at the discretion of the school principal.

methods, such as partial dependence plots (PDPs), could have provided more insightful results. PDPs are graphical tools that illustrate the relationship between a model feature and the outcome variable, while controlling for other variables. These plots reveal both the importance of a feature and the nature of its relationship with the outcome (e.g., linear, quadratic, monotonic). Unlike linear regression coefficients, PDPs can be applied to more complex models, including tree-based algorithms. One key advantage of PDPs is their ability to explore individual features and their interactions with others, which enhances model interpretability. As demonstrated by Cannistrà et al. (2022), PDPs can improve the communication of model insights, making complex models more accessible to non-technical audiences.

5.2 Discussion of findings

The goal of this dissertation is to leverage middle school data to predict if a student is at risk of dropping out of high school in 9th or 10th grade. I employ data science methods to explore three areas relevant to dropout prediction: 1) develop prediction models with various statistical approaches and techniques, 2) examine each model's ability to provide equitable predictions for students from marginalized backgrounds, and 3) interpret model findings to identify salient predictors of early exit.

5.2.1 Discussion of model performance

The evaluation of model performance reveals several noteworthy patterns. First, models trained on highly imbalanced data – where the number of students who exited early (the minority class) is vastly outnumbered by those who did not (the majority class) – tend to neglect the minority instances, essentially classifying all observations as 'did not exit early.' The model's inability to identify the target population (i.e., students who exited early) is masked by several metrics, such as high accuracy and AUC values (97 percent and 0.90, respectively). This highlights the necessity to examine additional performance metrics and model performance for student subgroups.

In contrast, I find that resampling techniques applied to the training data – whether through oversampling minority instances or undersampling majority instances – are effective in identifying the target population. However, this improvement comes at a cost, where models trained on balanced data tend to mislabel a higher proportion of students as having 'exited early' (see Table 17). Methods that utilize tree-based models, such as random forests and XGBoost, appear to be more effective at accurately predicting minority instances when trained on undersampled data, compared to oversampled data. This assessment of model performance highlights the potential for balanced training data to deliver precise predictions and that logistic regression performs similar to more complex statistical approaches. There is no definitive “winner” among the models, as some perform better at identifying majority instances, while others excel at identifying minority instances. The most suitable model depends on the context and purpose, which can vary by the researcher, school or district, and setting.

It is important to note that, across all panels, the random forest model demonstrated the lowest performance. Specifically, the random forest models in Panels A and B exhibited the lowest AUC values, while the model in Panel C recorded the lowest specificity at 74 percent. These results underscore the distinct roles of bagging and boosting techniques in managing the tradeoff between bias and variance. The random forest model, which employs the bagging approach, reduces variance by constructing independent trees and aggregating their predictions through majority voting. In contrast, the boosting technique, as utilized by XGBoost, mitigates bias by sequentially constructing trees, where each new tree corrects errors made by the preceding models. This distinction is reflected in the superior performance of XGBoost over the random forest model across all panels.

The model predictions in this analysis were more precise than those found in several studies (Sansone, 2019; Weissman, 2022; Oz et al., 2022; Selim & Rezk, 2023). Although my findings did not supersede the reported accuracies of the Chicago on-track indicator (Allensworth et al., 2013) or the Growth Mixture Model approach (Bowers & Sprott, 2012a), I contend that my dissertation cannot be directly compared to the aforementioned studies. The Allensworth et al. (2013) study included detailed middle school coursework data, such as course grades and GPA, while the Bowers & Sprott (2012a) study included 9th grade GPA, a well-established predictor of high school completion and even postsecondary success. Because this analysis does not examine 9th grade GPA or include coursework data, the predictive power of models in this study cannot be directly compared to theirs.

5.2.2 Discussion of algorithmic fairness

Exploration of the second research question finds that all models, regardless of the level of imbalance in the training data, disproportionately misclassify student subgroups based on their English learner status, race/ethnicity, disability status. My examination of two models, the undersampled XGboost and undersampled logistic regression, reveal that the logistic regression provides a more “balanced” model performance in ensuring that between non-protected groups and protected groups of an attribute (e.g., students without a disability and students with an identified disability) are similar. However, this comes at a penalty of the logistic regression exhibiting higher misclassification rates for both non-protected and protected subgroups. Based on these results, I argue that the undersampled XGBoost model is a 'safer' choice because it demonstrates lower misclassification rates compared to the undersampled logistic regression model. Recall that the backfire of Wisconsin’s early warning system, which disproportionately mislabeled Black and Hispanic students as “exiting early”, was due to high misclassification rates. While the models in this analysis do not misclassify as high as Wisconsin’s model, I contend that the XGboost model still exhibits algorithmic bias, as its predictions for many subgroups remain unsatisfactory.

The lack of equitable predictions for English learner, disability, and race/ethnicity subgroups warrants further investigation. If this prediction model were to have application in a school, district, or state setting, I propose that students with these protected attributes should be excluded from the analytic sample. Instead, separate models should be developed for each specific student subgroup, allowing for more tailored and careful consideration of their unique needs and challenges. This approach would ensure that these groups are not overlooked and that predictions are more accurately aligned with their educational contexts.

Further investigation is needed to understand why both the random forest model and XGboost models exhibit similar bias issues in the ABROCA analysis, especially for English learners and economic disadvantage. Given that the ABROCA statistics are independent of a single decision threshold and rather aggregate model performance across all thresholds, one hypothesis I have is that certain cutoff values in the XGBoost model might lead to significant misclassification of students within these protected attributes. This could, in turn, amplify the bias observed in the difference in AUC values.

5.2.3 Discussion of model interpretation

The third research question interpreted model findings for the five undersampled models. This question had two goals: to examine the extent of overlap in relevant predictors identified by each model and to improve the explainability of model findings so that it is accessible to non-technical audiences.

The findings reveal a converged narrative where the models generally identified and ranked predictors similarly. The models are in agreement that student age, middle school absences (especially being chronically absent) and economic disadvantage are the strongest predictors of early exit. With the exception of 7th and 8th grade absence rates, the features identified by the XGboost model were also identified in the three regression-based models, indicating strong overlap. Moreover, XGboost provides sparsity in its selection of model features, whereas ridge regression provided the greatest number of predictors associated with early exit.

New post-hoc approaches, like Shapley Additive Explanations (SHAP) plots, have enhanced the explainability of complex machine learning models. One key advantage of SHAP plots is that they provide insights similar to those provided by regressions. The mean SHAP values act as coefficients for each model feature, enabling users to calculate the likelihood of a student exiting early. A further examination of the undersampled XGboost model with a SHAP plot reveals similar ranking of features that were used to optimize predictions, reinforcing the presence of student age and chronic absence in a middle grade in making accurate predictions. The SHAP plot, however, depicts the large variance in age and absence rates, suggesting that the associations between these features and early exit are multifaceted. As described in the 5.1, the prioritization of continuous model features such as age should be interpreted with caution.

5.3 Next steps

There are several avenues to enhance or extend the work presented in this dissertation. One potential direction is exploring strategies to mitigate algorithmic bias, such as adjusting threshold values for students in protected attributes, as demonstrated by Lee & Kizilec (2020). While this approach may not fully address all aspects of fairness at the same time, it offers a valuable starting point for considering additional methods to mitigate algorithmic bias.

Second, it could be interesting to complement the use of supervised learning with clustering approaches. In particular, a growth mixture model (GMM) could be helpful in identifying multiple sub-populations and examining longitudinal differences within each sub-population (similar to Bowers & Sprott, 2012). Assuming that the data contain sufficient information to separate students into distinct clusters, it would be interesting to identify and understand students whose school engagement declines during middle grades.

Third, I acknowledge the additional value in including additional model features to understand math and reading proficiency. Although this dissertation does not examine math and reading scores as continuous measures, the inclusion of such features could improve model performance. Future work could further investigate math and reading proficiency by developing additional indicators of students whose math or reading scores are "slightly below" and "slightly above" the proficiency threshold. This feature could be instrumental in identifying students who may benefit from targeted academic interventions and support services and could also be used to evaluate the effectiveness of early warning systems.

Fourth, this research should investigate prediction accuracy at a more localized level. Building on the analytic approach of Coleman (2021), it would be valuable to examine prediction accuracy across the counties in North Carolina.[18] I hypothesize that, similar to Coleman (2021)’s findings, there may be significant variation in both the prediction accuracy

[18] In North Carolina most public school districts are county school units, meaning that the county board of education generally serves as the administrative unit of schools in its jurisdiction (North Carolina General Assembly, n.d.).

and the factors associated with early exit. Furthermore, it is likely that the findings derived from statewide early warning systems may not be generalized in most local contexts.

Fifth, this topic would be greatly improved from the addition of studies that apply economic evaluation methods to better understand the impact of high school dropout prediction efforts. Specifically, studies that assess the cost-effectiveness of implementing early warning systems would provide invaluable insights into the economic benefits of such interventions. Additionally, examining the costs and benefits associated with resource allocation – particularly how resources are utilized before and after the implementation of an early warning system – could offer a clearer picture of its long-term financial implications. Such research could significantly strengthen the case for early warning systems as a tool for improving resource efficiency, demonstrating how proactive interventions not only support student success but also lead to more sustainable and cost-effective outcomes for schools, districts, and society at large.

Sixth, future work can focus on the characteristics of students who were either under-identified or overidentified by the model. This analysis could be initiated by providing detailed descriptive statistics that differentiate these two subgroups of students. Gaining a deeper understanding of students who have been over-identified is particularly valuable, as it can guide decision-makers in establishing an appropriate threshold for overidentification for the prediction model, ensuring that the model's predictions align with realistic educational contexts and outcomes. Conversely, investigating students who have been under-identified offers crucial insights into the diversity of students who exited early. Such an investigation would elucidate the types of students that the model is either able to predict accurately or, conversely, fail to identify as at-risk of early exit. Specifically, exploring these under-identified cases can provide a more nuanced understanding of the typology of high school dropouts that the model has both successfully identified and overlooked.[19] By delving into the size, composition, and characteristics of the dropout profiles proposed by Bowers & Sprott (2012b), researchers can uncover important nuances about at-risk student populations. These insights could ultimately inform the design of more tailored and effective intervention strategies that better address the needs of diverse at-risk populations (Menzer & Hampel, 2009; Bowers & Sprott, 2012a, 2012b; Ogresta et al., 2021).

5.4 Additional work needed in the field

Developing an effective early warning system involves much more than just developing a prediction model. There are several components that must be in place before developing an early warning system. I outline four prerequisites that should be met prior to the development of an early warning system: improving research data systems, helping schools and districts move towards evidence-based decision-making, establishing specific parameters for model accuracy and misclassification, and dealing with generalizability issues.

First, schools, districts, and states should actively build equitable data infrastructures. Data systems should strive alignment with FAIR (Finding, Accessible, Interoperable, and Reusable) guiding principles (Bowers, & Choi, 2023).

Second, school districts and systems should have research personnel who can make “data sense”, or those who are skilled in not only developing and refining prediction models but also in conveying findings to a non-technical audience. This includes building dashboards, creating data visualization, and collaboration with leaders to inform decision-making. (Schutt & O'Neil, 2013; Krumm & Bowers, 2022; Bowers, in press). Emphasis should be placed on building

[19] Bowers & Sprott (2012b) dispels the monolithic tale that students who drop out of high school share the same characteristics. The authors examine the typology of high school dropouts to categorize them into three subgroups: quiet, jaded, and involved.

comprehensibility of model findings, fostering dialogue among stakeholders, and guiding collaborative efforts to determine next steps for dropout prevention. There is a growing body of literature that encourages collaboration with leaders, stakeholders, and communities for which the system is designed to serve (Lee, 2018; Bowers & Krumm, 2021; Bowers, 2021a).[20]

Third, when planning or refining an early warning system, schools, districts, and policymakers should establish a clear threshold for model accuracy. This helps promote transparency and accountability during model development. Furthermore, it ensures that the model meets key criteria, such as sensitivity (i.e., true positive rate) and specificity (i.e., false alarm rate), with particular attention to students in protected attributes. These discussions help minimize the risk of early warning system failures, like the one experienced in Wisconsin.

The final area of focus that is needed in the field is model generalizability. In machine learning applications, the issue of generalizability is defined as the extent to which predictive models maintain their accuracy and validity across varying contexts, populations, and time periods. For instance, changes in educational conditions and student behavior observed since the COVID-19 pandemic could impact the factors that influence graduation outcomes. In this case, the prediction models developed in this dissertation may not hold validity if it were to predict early exit for students graduating in 2023. Although prior attempts to generalize early warning system findings across varied contexts have not been successful (Stuit et al., 2016; Coleman, 2021), more work needs to be done to understand how early warning systems can adapt to changing education environments.

[20] An example of this is Hawn-Nelson et al. (2020)'s toolkit and guidance on how community stakeholders and data analysts can work together to build data systems that serve the community.

5.5 Dissertation summary

There is growing public concern about the fairness of K-12 early warning systems. The 2023 investigation of Wisconsin’s statewide early warning system revealed that the system disproportionately mislabeled students of color as “high-risk” and negatively influenced how teachers perceive these students. The challenge of providing fair predictions is further exacerbated by the increasing integration of artificial intelligence (AI) methods in educational settings, raising concerns about the interpretability of models for practitioners and stakeholders. As AI approaches become more prevalent in education, it is crucial to ensure that these methods are designed and implemented in ways that promote fairness and provide explanations of the decision-making process for complex AI models that are often labeled as “black box.”

This dissertation uses North Carolina state longitudinal data to examine middle school engagement and predict the likelihood that a student will drop out of high school in either 9th or 10th grade. Although I was able to develop models that demonstrate high predictive accuracy, the fairness analysis finds that these models are susceptible to model discrimination. It is important to note that the machines are not inherently biased, but are reflective of structural inequities in the education system (Baker & Hawn, 2021; Baker, 2023; Bowers, in press).

The findings of this dissertation underscores the need to develop approaches to mitigate bias in the model development phase. As AI models are becoming increasingly integral to various public sectors, the field of data science is actively developing approaches to demystify and provide explanations for the decisions made by black box models. However, the work is far from complete. As methodologies continue to evolve, it is important to stay informed about the latest developments and incorporate new techniques that address emerging challenges.

Continuous monitoring and adaptation are essential to maintain the integrity and validity of early warning systems in public applications.

This dissertation is a conceptual replication study that tests the same hypotheses across diverse settings and contexts. While this study leverages data from pre-pandemic school years, there is an urgent need to replicate these findings in post-pandemic environments. Given the significant impact of the pandemic on student engagement and the added challenges that have emerged, early warning systems have become an even more critical tool for schools and educational systems aiming to reengage at-risk students. The ongoing development, evaluation, and refinement of early warning systems can foster essential collaboration and dialogue among researchers, decision-makers, and policymakers. Such continued research will not only enhance the effectiveness of these systems but also ensure their adaptability to the evolving educational landscape in a post-pandemic world.

APPENDIX A: ADDITIONAL DESCRIPTIVE STATISTICS

Table A1: Descriptive statistics for SMOTE minority and original minority instances

	Students who exited early	
	Original train	SMOTE train
N	1,551 (1.7%)	86,856 (49.6%)
Student characteristics		
Female	0.411 (0.492)	0.407 (0.451)
Asian	0.006 (0.080)	0.004 (0.056)
White	0.487 (0.500)	0.513 (0.479)
Black	0.296 (0.457)	0.299 (0.438)
Other race	0.082 (0.274)	0.063 (0.219)
Economically disadvantaged	0.787 (0.409)	0.810 (0.352)
Age	14.580 (0.682)	14.571 (0.611)
IEP	0.284 (0.451)	0.273 (0.428)
Ever had an IEP in a middle grade	0.322 (0.468)	0.304 (0.438)
Limited English proficient	0.080 (0.271)	0.075 (0.256)
Ever limited English proficient in 8th grade	0.095 (0.293)	0.087 (0.273)
Urban	0.268 (0.443)	0.270 (0.418)
Suburban	0.231 (0.422)	0.226 (0.394)
Town	0.124 (0.330)	0.110 (0.287)
Rural	0.376 (0.485)	0.394 (0.462)
Academic information		
Not math proficient in 6th grade	0.433 (0.496)	0.421 (0.471)
Not math proficient in 7th grade	0.468 (0.499)	0.459 (0.473)
Not math proficient in 8th grade	0.892 (0.310)	0.908 (0.268)
Not math proficient in all middle grades	0.295 (0.456)	0.300 (0.445)
Not reading proficient in 6th grade	0.503 (0.500)	0.496 (0.482)
Not reading proficient in 7th grade	0.567 (0.496)	0.561 (0.473)
Not reading proficient in 8th grade	0.839 (0.367)	0.856 (0.327)
Not reading proficient in all middle grades	0.391 (0.488)	0.401 (0.478)
Attendance information		
Absence rate in 6th grade	0.103 (0.104)	0.100 (0.090)
Absence rate in 7th grade	0.115 (0.113)	0.111 (0.099)
Absence rate in 8th grade	0.136 (0.128)	0.133 (0.111)
Chronically absent in 6th grade	0.230 (0.421)	0.210 (0.381)
Chronically absent in 7th grade	0.309 (0.462)	0.297 (0.431)
Chronically absent in 8th grade	0.428 (0.495)	0.426 (0.471)
Ever chronically absent in a middle grade	0.598 (0.491)	0.583 (0.477)
Chronically absent in all middle grades	0.088 (0.283)	0.081 (0.258)
School mobility in 6th grade	0.019 (0.135)	1.013 (0.094)
School mobility in 7th grade	0.020 (0.140)	1.014 (0.097)
School mobility in 8th grade	0.050 (0.219)	1.040 (0.172)
School mobility in a middle grade	0.401 (0.608)	1.359 (0.530)
Discipline information		

OSS in 6th grade	0.268 (0.443)	0.259 (0.415)
OSS in 7th grade	0.349 (0.477)	0.348 (0.455)
OSS in 8th grade	0.397 (0.489)	0.408 (0.472)
OSS in a middle grade	0.575 (0.494)	0.571 (0.492)
ISS in 6th grade	0.269 (0.444)	0.263 (0.419)
ISS in 7th grade	0.337 (0.473)	0.332 (0.444)
ISS in 8th grade	0.346 (0.476)	0.343 (0.445)
ISS in a middle grade	0.442 (0.497)	0.435 (0.479)
Ever suspended in a middle grade	0.687 (0.464)	0.674 (0.463)

Notes: In the first row, *N* indicates the sample size and proportion of the train data represented by the subgroup. Standard errors are reported in other parentheses. Student characteristics are extracted from 8th grade records. Detailed information about each indicator can be found in 3.5.2.

Table A2: Descriptive statistics for undersampled majority and original majority instances

	Students who did not exit early	
	Original train	Undersampled train
N	88,165 (98.3%)	1,551 (50.0%)
Student characteristics		
Female	0.505 (0.500)	0.500 (0.500)
Asian	0.027 (0.162)	0.029 (0.168)
White	0.540 (0.498)	0.533 (0.499)
Black	0.264 (0.441)	0.275 (0.446)
Other race	0.054 (0.227)	0.052 (0.221)
Economically disadvantaged	0.446 (0.497)	0.427 (0.495)
Age	13.693 (0.464)	13.693 (0.456)
IEP	0.114 (0.318)	0.108 (0.311)
Ever had an IEP in a middle grade	0.137 (0.344)	0.132 (0.338)
Limited English proficient	0.045 (0.208)	0.049 (0.216)
Ever limited English proficient in 8th grade	0.055 (0.229)	0.058 (0.234)
Urban	0.267 (0.442)	0.270 (0.444)
Suburban	0.240 (0.427)	0.248 (0.432)
Town	0.100 (0.300)	0.090 (0.286)
Rural	0.393 (0.488)	0.393 (0.488)
Academic information		
Not math proficient in 6th grade	0.146 (0.353)	0.139 (0.346)
Not math proficient in 7th grade	0.142 (0.349)	0.136 (0.343)
Not math proficient in 8th grade	0.618 (0.486)	0.628 (0.483)
Not math proficient in all middle grades	0.089 (0.284)	0.092 (0.288)
Not reading proficient in 6th grade	0.202 (0.402)	0.196 (0.397)
Not reading proficient in 7th grade	0.272 (0.445)	0.275 (0.447)
Not reading proficient in 8th grade	0.554 (0.497)	0.542 (0.498)
Not reading proficient in all middle grades	0.165 (0.371)	0.155 (0.362)
Attendance information		
Absence rate in 6th grade	0.034 (0.039)	0.033 (0.039)
Absence rate in 7th grade	0.035 (0.041)	0.033 (0.038)
Absence rate in 8th grade	0.041 (0.044)	0.039 (0.041)
Chronically absent in 6th grade	0.045 (0.207)	0.041 (0.197)
Chronically absent in 7th grade	0.048 (0.214)	0.045 (0.208)
Chronically absent in 8th grade	0.068 (0.252)	0.063 (0.242)
Ever chronically absent in a middle grade	0.122 (0.327)	0.104 (0.305)
Chronically absent in all middle grades	0.007 (0.084)	0.007 (0.084)
School mobility in 6th grade	0.005 (0.071)	1.006 (0.080)
School mobility in 7th grade	0.006 (0.076)	1.006 (0.076)
School mobility in 8th grade	0.010 (0.100)	1.006 (0.076)
School mobility in a middle grade	0.207 (0.447)	1.190 (0.425)
Discipline information		

OSS in 6th grade	0.078 (0.268)	0.073 (0.260)
OSS in 7th grade	0.088 (0.283)	0.072 (0.259)
OSS in 8th grade	0.091 (0.288)	0.086 (0.281)
OSS in a middle grade	0.180 (0.384)	0.164 (0.371)
ISS in 6th grade	0.101 (0.301)	0.090 (0.287)
ISS in 7th grade	0.125 (0.331)	0.121 (0.326)
ISS in 8th grade	0.124 (0.329)	0.108 (0.310)
ISS in a middle grade	0.188 (0.391)	0.171 (0.377)
Ever suspended in a middle grade	0.291 (0.454)	0.275 (0.446)

Notes: In the first row, *N* indicates the sample size and proportion of the train data represented by the class. Standard errors are reported in other parentheses. Student characteristics are extracted from 8th grade records. Detailed information about each indicator can be found in 3.5.2.

APPENDIX B: CODE AND OUTPUT

Figure B1: Code for research question 1

Data cleaning

```
#cleaning train data
train <- read.csv("D:/NCERDC_DATA/Alam/ML/Training sample/Data/trainingpanel.csv")
summary(train)
# str(train) this showed that almost no variables were factors
train <- train %>% mutate_if(is.integer, as.factor)
train = subset(train, select = -c(mastid) )
train <- train %>% as_tibble  %>% mutate(across(c(40:43), as.numeric))
str(train)

#cleaning test data
test <- read.csv("D:/NCERDC_DATA/Alam/ML/Testing sample/Data/testingpanel.csv")
test <- test %>% mutate_if(is.integer, as.factor)
test = subset(test, select = -c(mastid) )
train <- train %>% as_tibble  %>% mutate(across(c(40:43), as.numeric))
str(test)

write.csv(train,'train.csv', row.names=FALSE)
write.csv(test,'test.csv', row.names=FALSE)
```

Model 1: Logistic regression

```
train <- read.csv("train.csv")
test <- read.csv("test.csv")

# Fit the logistic regression model
log1.m <- glm(dropout ~ ., data = subset(train, select = -c(female, hispanic, asian, black, white,
other_race)), family = 'binomial')
summary(log1.m)
```

```
##
## Call:
## glm(formula = dropout ~ ., family = "binomial", data = subset(train,
##     select = -c(female, hispanic, asian, black, white, other_race)))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7671  -0.1243  -0.0764  -0.0527   3.8438
##
## Coefficients: (1 not defined because of singularities)
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -33.741987   0.801317 -42.108  < 2e-16 ***
## ever_stsusp_middle       10.940859 145.265122   0.075 0.939963
## ever_ltsusp_middle           0.254953   0.343387   0.742 0.457805
## ever_OSS_6                  -0.052846   0.093110  -0.568 0.570327
## ever_OSS_7                   0.154329   0.091206   1.692 0.090626 .
## ever_OSS_8                   0.296006   0.093909   3.152 0.001621 **
## ever_OSS_middle            -10.797742 145.265154  -0.074 0.940747
## ever_ISS_middle              0.055268   0.127531   0.433 0.664744
## ever_ISS_6                   0.102078   0.110516   0.924 0.355671
## ever_ISS_7                   0.068679   0.083340   0.824 0.409893
## ever_ISS_8                   0.236480   0.077540   3.050 0.002290 **
## not_math_proficient_6        0.309266   0.100846   3.067 0.002164 **
## not_math_proficient_7        0.546723   0.093213   5.865 4.48e-09 ***
## not_math_proficient_8        0.309054   0.112036   2.759 0.005806 **
## no_math_proficiency_middle  -0.452462   0.135109  -3.349 0.000811 ***
## not_read_proficient_6        0.451764   0.112643   4.011 6.06e-05 ***
## not_read_proficient_7        0.054599   0.092597   0.590 0.555437
## not_read_proficient_8        0.152022   0.098090   1.550 0.121184
## no_read_proficiency_middle  -0.537957   0.136329  -3.946 7.95e-05 ***
## eds                          0.414365   0.073239   5.658 1.53e-08 ***
## age_eighthfall1              1.925457   0.050317  38.267  < 2e-16 ***
## ever_swd                     0.051677   0.170880   0.302 0.762336
## swd_8                       -0.387899   0.177370  -2.187 0.028746 *
## ever_lep                     0.406953   0.249231   1.633 0.102504
## lep_8                       -0.450850   0.269412  -1.673 0.094237 .
## absence_rate_6               1.767911   0.640226   2.761 0.005756 **
## absence_rate_7               2.610089   0.594187   4.393 1.12e-05 ***
## absence_rate_8               5.207350   0.535821   9.718  < 2e-16 ***
## chrabsent_6                  0.249767   0.136201   1.834 0.066682 .
## chrabsent_7                  0.293176   0.119291   2.458 0.013985 *
## chrabsent_8                  0.375300   0.133803   2.805 0.005034 **
## ever_chrabsent_middle        0.262627   0.147092   1.785 0.074186 .
## chrabsent_middle            -0.399342   0.206058  -1.938 0.052622 .
## school_mobility_middle       0.193665   0.058851   3.291 0.000999 ***
## school_mobility_8            0.225752   0.166380   1.357 0.174831
## school_mobility_7           -0.080500   0.234909  -0.343 0.731835
## school_mobility_6            0.279470   0.239537   1.167 0.243327
## urban                       -0.006347   0.074854  -0.085 0.932428
## suburban                    -0.050352   0.078166  -0.644 0.519466
## town                         0.214412   0.098451   2.178 0.029417 *
## rural                              NA         NA      NA       NA
## ever_suspended               0.182487   0.132669   1.376 0.168973
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 15662.2  on 89715  degrees of freedom
## Residual deviance:  9830.8  on 89675  degrees of freedom
## AIC: 9912.8
```

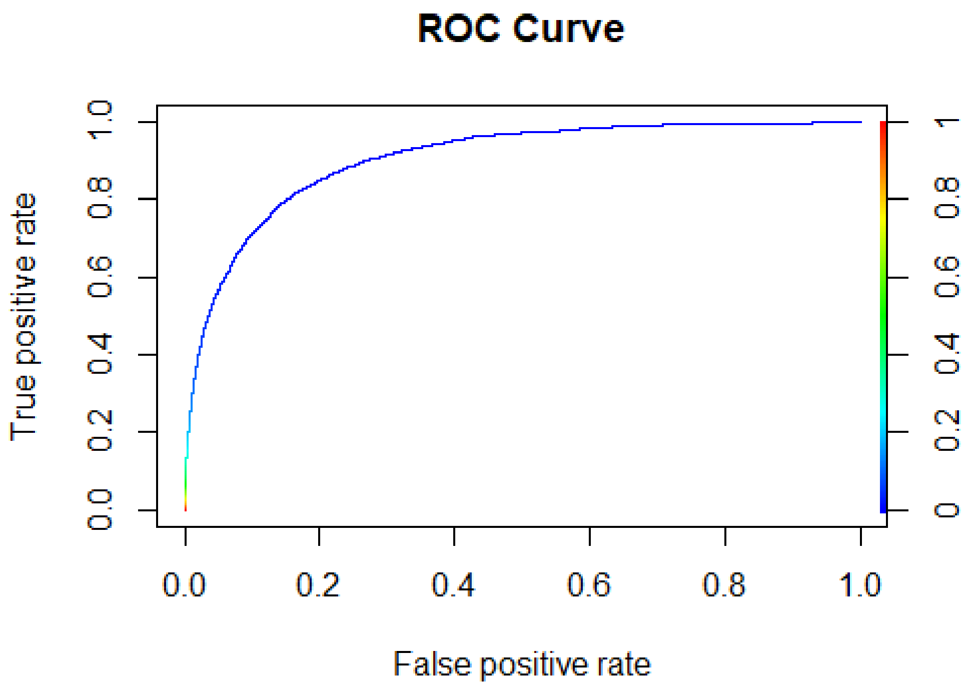
```
##
## Number of Fisher Scoring iterations: 13
```

```
# Predict on the TEST data
predict_log <- predict(log1.m, test[,-1], type = 'response')
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
# Create a prediction object for ROCR
pred <- prediction(predict_log, test$dropout)
```

```
# Create a performance object for ROC curve
perf_log <- performance(pred, "tpr", "fpr")
# Plot the first ROC curve (perf_log)
plot(perf_log, colorize = TRUE, main = "ROC Curve")
```



```
# AuC score
auc <- performance(pred, measure = "auc")
auc@y.values[[1]]
```

```
## [1] 0.9044747
```

```
# Convert predictions to factors (assuming binary classification)
predict_log_class <- as.factor(ifelse(predict_log >= 0.2, 1, 0))
test$dropout <- as.factor(test$dropout)
```

```
# Create confusion matrix
cm <- confusionMatrix(data = predict_log_class, reference = test$dropout, positive = "1")
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 92184  1927
##          1   489   477
##
##                Accuracy : 0.9746
##                  95% CI : (0.9736, 0.9756)
##     No Information Rate : 0.9747
##     P-Value [Acc > NIR] : 0.6031
##
##                   Kappa : 0.2725
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.198419
##             Specificity : 0.994723
##          Pos Pred Value : 0.493789
##          Neg Pred Value : 0.979524
##              Prevalence : 0.025285
##          Detection Rate : 0.005017
##    Detection Prevalence : 0.010160
##       Balanced Accuracy : 0.596571
##
##        'Positive' Class : 1
##
```

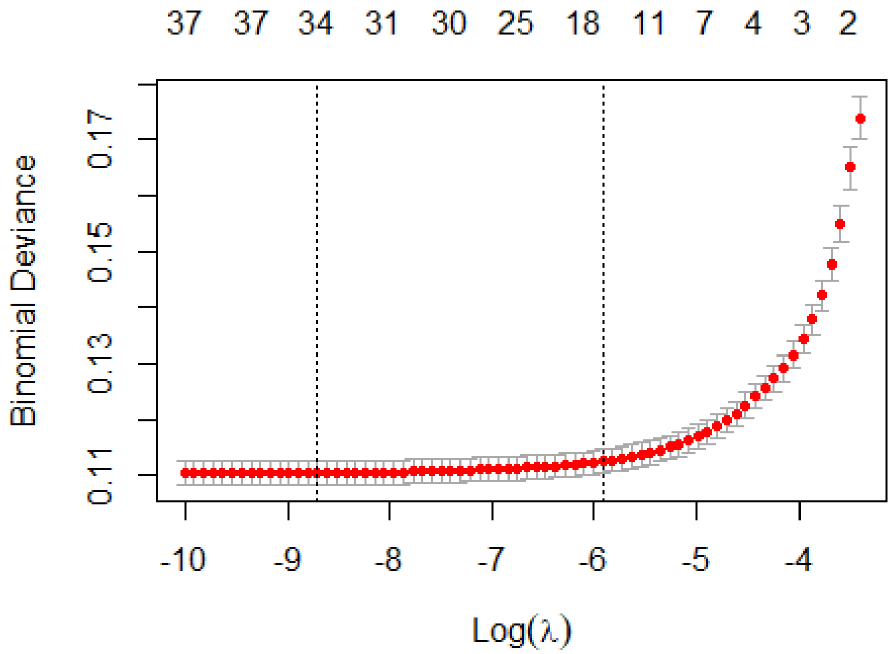
Preparing for Lasso and Ridge

```
train <- read.csv("train.csv")
test <- read.csv("test.csv")

train = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )

y.train = train$dropout %>% unlist() %>% as.numeric()
y.test = test$dropout %>% unlist() %>% as.numeric()
x.train = model.matrix(dropout~., train)[,-1]
x.test = model.matrix(dropout~., test)[,-1]

dim(x.train)
dim(x.test)
```

```
write.csv(x.train,'x.train.csv', row.names=FALSE)
write.csv(x.test,'x.test.csv', row.names=FALSE)
write.csv(y.train,'y.train.csv', row.names=FALSE)
write.csv(y.test,'y.test.csv', row.names=FALSE)
```

Model 2: Lasso

```
#CV to estimate best lambda
set.seed(2023)
cv.lasso <- cv.glmnet(x.train, y.train, alpha = 1, family='binomial') # Fit lasso regression model on training data
#Display MSE vs log-lambda plot
plot(cv.lasso) # Draw plot of training MSE as a function of lambda
```

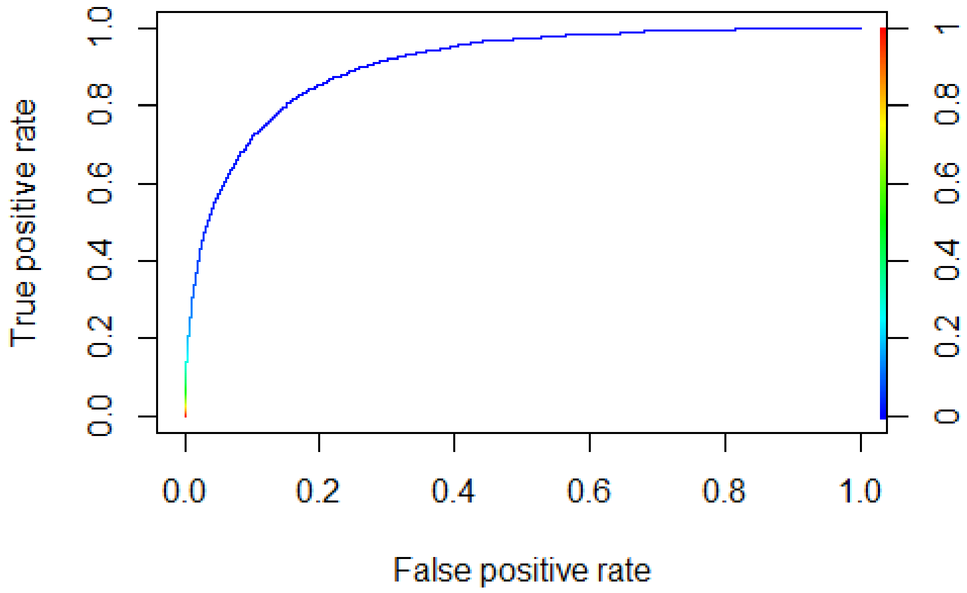


```
# ROC analysis to identify optimal threshold
lasso.pred <- predict(cv.lasso, newx=x.test, s = "lambda.min", type="response")
# Ensure lasso.pred is a numeric vector
lasso.pred <- as.numeric(lasso.pred)
print(length(lasso.pred))  # Check length of lasso.pred
```

```
## [1] 95077
```

```
#Create ROC curve
pred_lasso <- prediction(lasso.pred, y.test)
```

```
y.test <- as.matrix(y.test)
perf_lasso <- performance(pred_lasso , "tpr", "fpr")
plot(perf_lasso, colorize=TRUE)
```



```
# AuC score
auc <- performance(pred_lasso, measure = "auc")
auc@y.values[[1]]
```

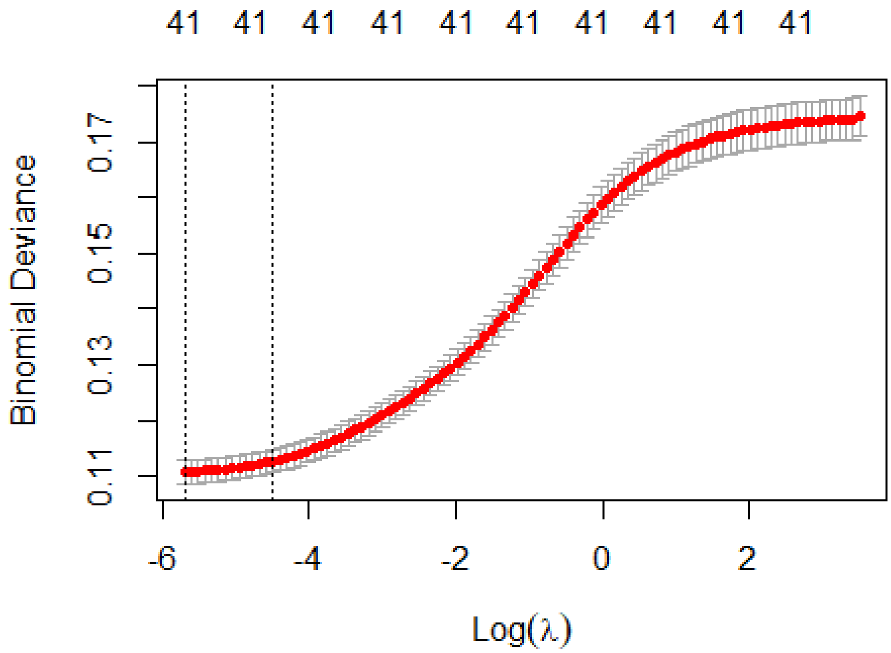
```
## [1] 0.9070521
```

```
# Convert predictions to factors
predict_lasso_class <- as.factor(ifelse(lasso.pred >= 0.2, "1", "0"))
# Ensure test$dropout is a factor with the same levels
test$dropout <- as.factor(test$dropout)
levels(predict_lasso_class) <- levels(test$dropout)  # Ensure factor levels match

# Create confusion matrix
cm <- confusionMatrix(data = predict_lasso_class, reference = test$dropout, positive = "1")
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 92204  1940
##          1   469   464
##
##                Accuracy : 0.9747
##                  95% CI : (0.9736, 0.9757)
##     No Information Rate : 0.9747
##     P-Value [Acc > NIR] : 0.5465
##
##                   Kappa : 0.2677
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.193012
##             Specificity : 0.994939
##          Pos Pred Value : 0.497320
##          Neg Pred Value : 0.979393
##              Prevalence : 0.025285
##          Detection Rate : 0.004880
##    Detection Prevalence : 0.009813
##       Balanced Accuracy : 0.593975
##
##        'Positive' Class : 1
##
```

Model 3: Ridge regression

```
#CV to estimate best lambda
set.seed(2023)
cv.ridge <- cv.glmnet(x.train, y.train, alpha = 0, family='binomial') # Fit ridge regression model on training data
#Display MSE vs log-lambda plot
plot(cv.ridge) # Draw plot of training MSE as a function of lambda
```



```
ridge.pred <- predict(cv.ridge, newx=x.test, s = "lambda.min", type="response")
# Ensure lasso.pred is a numeric vector
ridge.pred <- as.numeric(ridge.pred)
print(length(ridge.pred))  # Check length of lasso.pred
```

```
## [1] 95077
```

```
# Extract the coefficients at the best lambda (lambda.min or lambda.1se)
ridge.coefs <- coef(cv.ridge, s = "lambda.min")  # or use lambda.1se for a more regularized solution

# View the coefficients
print(ridge.coefs)
```

```
## 42 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)        -29.97623269
## ever_stsusp_middle   0.11726085
## ever_ltsusp_middle   0.22822519
## ever_OSS_6          -0.02043158
## ever_OSS_7           0.14816966
## ever_OSS_8           0.26383650
## ever_OSS_middle      0.10679842
## ever_ISS_middle      0.10545648
## ever_ISS_6           0.07008450
## ever_ISS_7           0.07658856
## ever_ISS_8                   0.20170969
## not_math_proficient_6        0.24231139
## not_math_proficient_7        0.42358656
## not_math_proficient_8        0.22506985
## no_math_proficiency_middle  -0.25729181
## not_read_proficient_6        0.25829594
## not_read_proficient_7        0.04419397
## not_read_proficient_8        0.16275189
## no_read_proficiency_middle  -0.26823950
## eds                          0.34786841
## age_eighthfall1              1.66895716
## ever_swd                    -0.02462989
## swd_8                       -0.18973493
## ever_lep                     0.17261443
## lep_8                       -0.12728764
## absence_rate_6               2.12497782
## absence_rate_7               2.70725959
## absence_rate_8               4.67291775
## chrabsent_6                  0.19859994
## chrabsent_7                  0.26235226
## chrabsent_8                  0.37056709
## ever_chrabsent_middle        0.30418184
## chrabsent_middle            -0.32989299
## school_mobility_middle       0.16607225
## school_mobility_8            0.24358069
## school_mobility_7           -0.02817470
## school_mobility_6            0.27816572
## urban                       -0.02107455
## suburban                    -0.04304323
## town                         0.17340319
## rural                       -0.01534907
## ever_suspended               0.14243790
```

```
# To view the coefficients in a more readable format (as a dataframe):
ridge.coefs_df <- as.data.frame(as.matrix(ridge.coefs))
print(ridge.coefs[ridge.coefs != 0]) # Display only non-zero coefficients
```

```
## <sparse>[ <logic> ]: .M.sub.i.logical() maybe inefficient
```

```
##  [1] -29.97623269   0.11726085   0.22822519  -0.02043158   0.14816966
##  [6]   0.26383650   0.10679842   0.10545648   0.07008450   0.07658856
## [11]   0.20170969   0.24231139   0.42358656   0.22506985  -0.25729181
## [16]   0.25829594   0.04419397   0.16275189  -0.26823950   0.34786841
## [21]   1.66895716  -0.02462989  -0.18973493   0.17261443  -0.12728764
## [26]   2.12497782   2.70725959   4.67291775   0.19859994   0.26235226
## [31]   0.37056709   0.30418184  -0.32989299   0.16607225   0.24358069
## [36]  -0.02817470   0.27816572  -0.02107455  -0.04304323   0.17340319
## [41]  -0.01534907   0.14243790
```

```
ridge.coefs_df <- ridge.coefs_df %>%
  arrange(desc(s1))
print(ridge.coefs_df)
```

```
##                                      s1
## absence_rate_8               4.67291775
## absence_rate_7               2.70725959
## absence_rate_6               2.12497782
## age_eighthfall1              1.66895716
## not_math_proficient_7        0.42358656
## chrabsent_8                  0.37056709
## eds                          0.34786841
## ever_chrabsent_middle        0.30418184
## school_mobility_6            0.27816572
## ever_OSS_8                   0.26383650
## chrabsent_7                  0.26235226
## not_read_proficient_6        0.25829594
## school_mobility_8            0.24358069
## not_math_proficient_6        0.24231139
## ever_ltsusp_middle           0.22822519
## not_math_proficient_8        0.22506985
## ever_ISS_8                   0.20170969
## chrabsent_6                  0.19859994
## town                         0.17340319
## ever_lep                     0.17261443
## school_mobility_middle       0.16607225
## not_read_proficient_8        0.16275189
## ever_OSS_7                   0.14816966
## ever_suspended               0.14243790
## ever_stsusp_middle           0.11726085
## ever_OSS_middle              0.10679842
## ever_ISS_middle              0.10545648
## ever_ISS_7                   0.07658856
## ever_ISS_6                   0.07008450
## not_read_proficient_7        0.04419397
## rural                       -0.01534907
## ever_OSS_6                  -0.02043158
## urban                       -0.02107455
## ever_swd                    -0.02462989
## school_mobility_7           -0.02817470
## suburban                    -0.04304323
## lep_8                       -0.12728764
## swd_8                       -0.18973493
## no_math_proficiency_middle  -0.25729181
## no_read_proficiency_middle  -0.26823950
## chrabsent_middle            -0.32989299
## (Intercept)                -29.97623269
```

```r
write.csv(ridge.coefs_df, "ridge.coefs.csv", row.names = TRUE)


#Create ROC curve
pred_ridge <- prediction(ridge.pred, y.test)
y.test <- as.matrix(y.test)
perf_ridge <- performance(pred_ridge , "tpr", "fpr")
#plot_ridge <- plot(perf_ridge, colorize=TRUE) #lasso prob threshold should be 0.2

# AuC
perf_ridge <- performance(pred_ridge,"auc")
auc <- as.numeric(perf_ridge@y.values)
auc
```

```
## [1] 0.908553
```

```r
# Convert predictions to factors
predict_ridge_class <- as.factor(ifelse(ridge.pred >= 0.2, "1", "0"))
# Ensure test$dropout is a factor with the same levels
test$dropout <- as.factor(test$dropout)
levels(predict_ridge_class) <- levels(test$dropout)  # Ensure factor levels match

# Create confusion matrix
cm <- confusionMatrix(data = predict_ridge_class, reference = test$dropout, positive = "1")
print(cm)
```

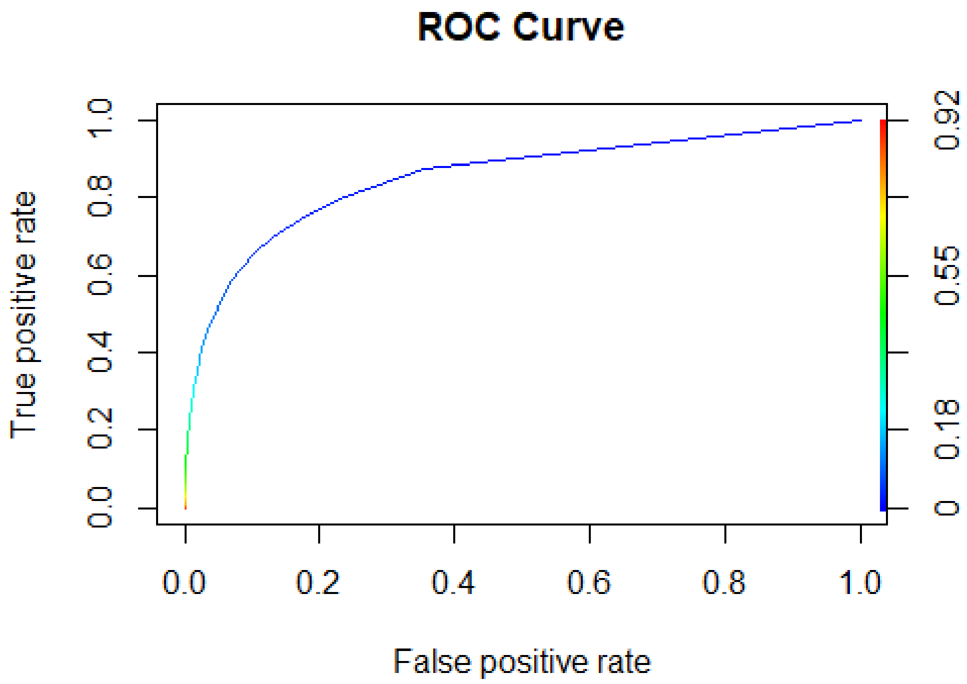
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 92280  1990
##          1   393   414
##
##                Accuracy : 0.9749
##                  95% CI : (0.9739, 0.9759)
##     No Information Rate : 0.9747
##     P-Value [Acc > NIR] : 0.3369
##
##                   Kappa : 0.2483
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.172213
##             Specificity : 0.995759
##          Pos Pred Value : 0.513011
##          Neg Pred Value : 0.978890
##              Prevalence : 0.025285
##          Detection Rate : 0.004354
##    Detection Prevalence : 0.008488
##       Balanced Accuracy : 0.583986
##
##        'Positive' Class : 1
##
```

```
f1_score <- cm$byClass["F1"]
print(f1_score)
```

```
##        F1
## 0.2578636
```

Model 4: Random forest

```
train <- read.csv("train.csv")
test <- read.csv("test.csv")
train_nodem = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )
train_nodem$dropout <- as.factor(train_nodem$dropout)

set.seed(2023)
RF.dropout <- randomForest(dropout ~ ., data = train_nodem, ntree = 100, importance = TRUE)


# Predict on the TEST data
rf.pred <- predict(RF.dropout, newdata = test[,-1], type = "prob")[,2]

# Create a prediction object for ROCR
rf_pr_test <- prediction(rf.pred, test$dropout)

# Create a performance object for ROC curve
perf_rf <- performance(rf_pr_test, "tpr", "fpr")

# Plot the ROC curve
plot(perf_rf, colorize = TRUE, main = "ROC Curve")
```



```
# Calculate AUC
auc <- performance(rf_pr_test, measure = "auc")
print(auc@y.values[[1]])
```

```
## [1] 0.8521343
```

```
# Convert predictions to binary class (assuming binary classification)
predict_rf_class <- as.factor(ifelse(rf.pred >= 0.22, 1, 0))
test$dropout <- as.factor(test$dropout)

# Create confusion matrix
cm <- confusionMatrix(data = predict_rf_class, reference = test$dropout, positive = "1")
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 91379  1652
##          1  1294   752
##
##                Accuracy : 0.969
##                  95% CI : (0.9679, 0.9701)
##     No Information Rate : 0.9747
##     P-Value [Acc > NIR] : 1
```

```
##
##                  Kappa : 0.3222
##
##  Mcnemar's Test P-Value : 4.789e-11
##
##             Sensitivity : 0.312812
##             Specificity : 0.986037
##          Pos Pred Value : 0.367546
##          Neg Pred Value : 0.982242
##              Prevalence : 0.025285
##          Detection Rate : 0.007909
##    Detection Prevalence : 0.021519
##       Balanced Accuracy : 0.649424
##
##        'Positive' Class : 1
##
```

Preparing for XGboost

```
train <- read.csv("train.csv")
test <- read.csv("test.csv")
str(train)
str(test)

train = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )

y.train = train$dropout %>% unlist() %>% as.numeric()
y.test = test$dropout %>% unlist() %>% as.numeric()
x.train = model.matrix(dropout~., train)[,-1] #data should only be predictors
x.test = model.matrix(dropout~., test)[,-1]

# Check the structure of data
str(x.train)
str(x.test)
str(y.train)
str(y.test)

# eta controls the learning rate, which scales the contribution of each tree. A smaller value (e.g., 0.1) can lead to more robust models but requires more boosting rounds. The default value of 0.3 is more aggressive

# eval_metric specifies the metric to evaluate during training. The detailed parameter set explicitly specifies "logloss", which is useful for binary classification tasks.

# gamma specifies the minimum loss reduction required to make a further partition. Setting gamma to 0 means no regularization is applied to the tree splitting, which may lead to more complex trees.
# min_child_weight is the minimum sum of instance weight (hessian) needed in a child. It controls
overfitting; higher values prevent the model from learning overly specific patterns.

# Data preparation
dtrain <- xgb.DMatrix(data = x.train, label = y.train)
dtest <- xgb.DMatrix(data = x.test, label = y.test)
ts_label <- test$dropout


# Initial parameter setup (if needed)
initial_params <- list(
  booster = "gbtree",
  objective = "binary:logistic",
  eval_metric = "logloss",
  eta = 0.3,
  max_depth = 6, gamma = 3
)

# Cross-validation to find optimal rounds of boosting
cv_results <- xgb.cv(
  params = initial_params,
  data = dtrain,
  nrounds = 100,
  nfold = 5,
  early_stopping_rounds = 20,
  verbose = 1
)

# Extract the Best Number of Rounds
best_nrounds <- cv_results$best_iteration

# Train the Final Model with Optimal Parameters
set.seed(2023)
final_model <- xgb.train(
  params = initial_params,
  data = dtrain,
  nrounds = best_nrounds
)

# Grid search for hyperparameter tuning
search_grid <- expand.grid(
  max_depth = c(3, 6),
  eta = c(0.01, 0.1),
  colsample_bytree = c(0.5, 0.7)
)

best_auc <- Inf  # Use Inf for minimization
best_params <- list()
```

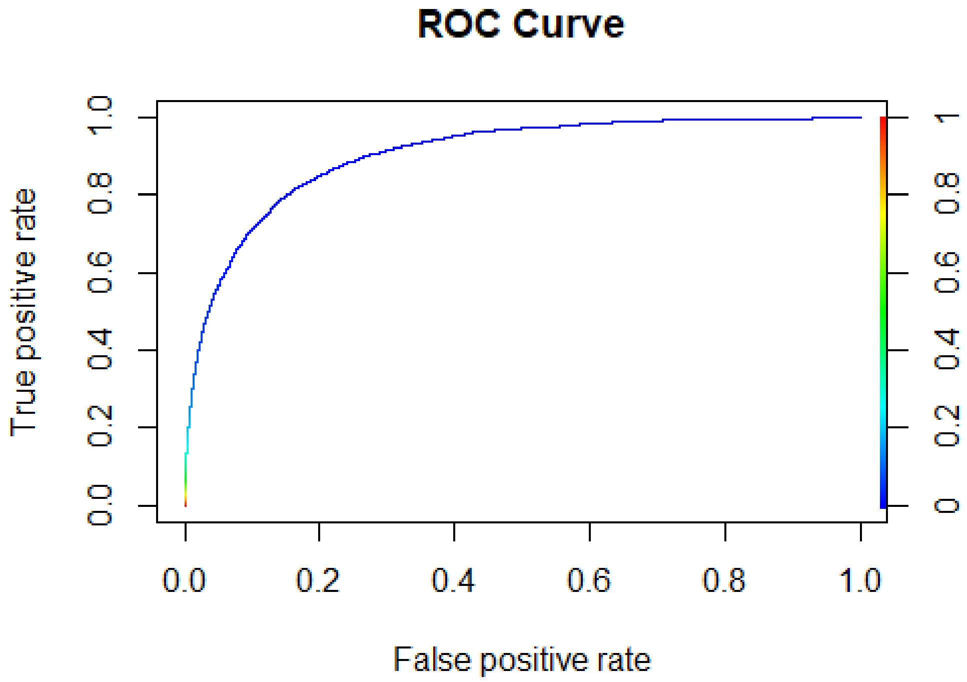
```
for (i in 1:nrow(search_grid)) {
  params <- list(
    objective = "binary:logistic",
    eval_metric = "logloss",
    max_depth = search_grid$max_depth[i],
    eta = search_grid$eta[i],
    colsample_bytree = search_grid$colsample_bytree[i]
  )

  cv_results <- xgb.cv(
    params = params,
    data = dtrain,
    nfold = 5,
    nrounds = 100,
    early_stopping_rounds = 10,
    verbose = 1
  )

  mean_logloss <- min(cv_results$evaluation_log$test_logloss_mean)

  if (mean_logloss < best_auc) {
    best_auc <- mean_logloss
    best_params <- params
    best_nrounds <- cv_results$best_iteration
  }
}
```

Model 5: XGboost

```
# Train the final model with the best parameters
dtest <- xgb.DMatrix(data = x.test, label = y.test)
set.seed(2023)
xgb1 <- xgb.train (params = best_params, data = dtrain, watchlist = list(val=dtest,train=dtrain),
print_every_n = 10, nrounds = best_nrounds)
```

```
## [1]  val-logloss:0.604779    train-logloss:0.603645
## [11] val-logloss:0.222354    train-logloss:0.209417
## [21] val-logloss:0.125100    train-logloss:0.104524
## [31] val-logloss:0.095889    train-logloss:0.071148
## [41] val-logloss:0.087154    train-logloss:0.060079
## [51] val-logloss:0.084379    train-logloss:0.056018
## [61] val-logloss:0.083503    train-logloss:0.054327
## [71] val-logloss:0.083082    train-logloss:0.053517
## [81] val-logloss:0.082899    train-logloss:0.052925
## [91] val-logloss:0.082737    train-logloss:0.052555
## [100]    val-logloss:0.082580    train-logloss:0.052259
```

```
#model prediction
xgbpred <- predict (xgb1,dtest)
xgbpred <- ifelse (xgbpred > 0.18,"1", "0")

y.test <- as.factor(y.test)
xgbpred <- as.factor(xgbpred)
y.test = test$dropout %>% unlist() %>% as.factor()

predict_xgboost <- predict(xgb1, dtest, type = 'response')
pred_xgboost <- prediction(predict_xgboost, test$dropout)
# Create a performance object for ROC curve
perf_xgboost <- performance(pred, "tpr", "fpr")
# Plot the first ROC curve (perf_log)
plot(perf_xgboost, colorize = TRUE, main = "ROC Curve")
```



```
# Create confusion matrix
cm <- confusionMatrix(data = xgbpred, reference = y.test, positive = "1")
print(cm)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 91447  1621
##          1  1226   783
```

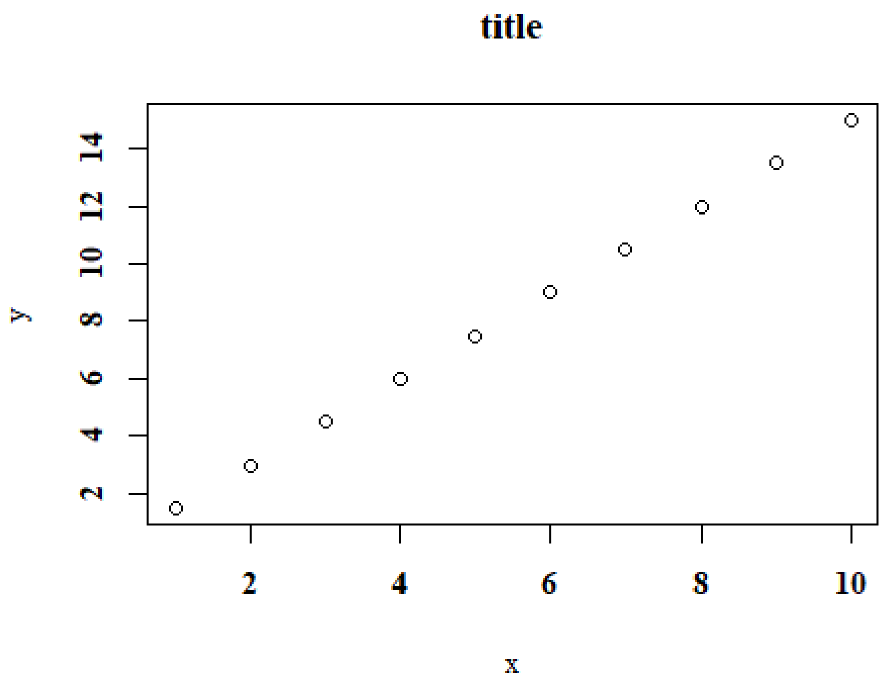
```
## 
##                Accuracy : 0.9701
##                  95% CI : (0.969, 0.9711)
##     No Information Rate : 0.9747
##     P-Value [Acc > NIR] : 1
## 
##                   Kappa : 0.3397
## 
##  Mcnemar's Test P-Value : 1.534e-13
## 
##             Sensitivity : 0.325707
##             Specificity : 0.986771
##          Pos Pred Value : 0.389746
##          Neg Pred Value : 0.982583
##              Prevalence : 0.025285
##          Detection Rate : 0.008235
##    Detection Prevalence : 0.021130
##       Balanced Accuracy : 0.656239
## 
##        'Positive' Class : 1
## 
```

```
auc <- performance(pred_xgboost, measure = "auc")
auc@y.values[[1]]
```

```
## [1] 0.9090673
```

plotting all ROC curves in one graph

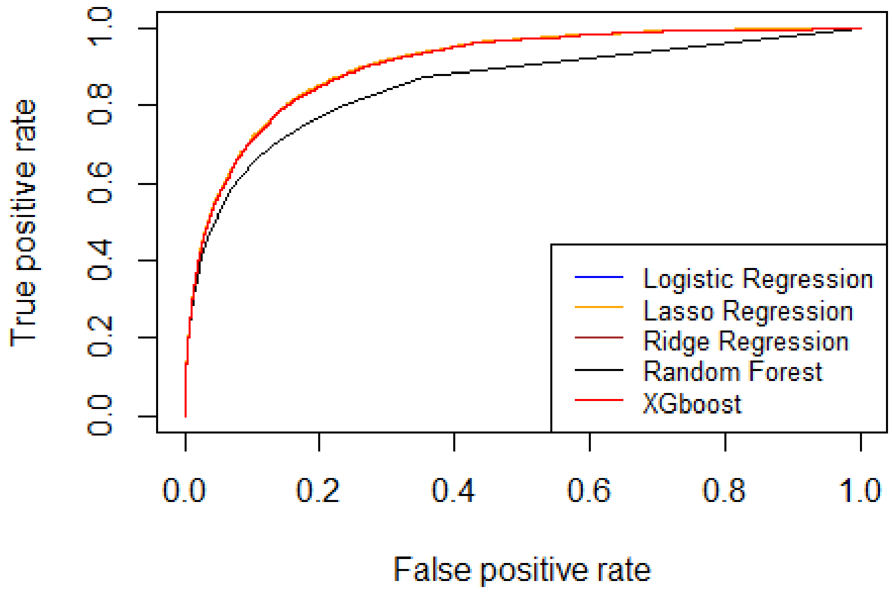
```
x = seq(1,10,1)
y = 1.5*x
windowsFonts(A = windowsFont("Times New Roman"))
plot(x, y,
  family="A",
  main = "title",
  font=2)
```



```
windowsFonts(Times=windowsFont("Times New Roman"))

# Set the font to Times New Roman in the plots
plot(perf_log, colorize = FALSE, col = "blue", family = "Times New Roman")
```

```
## Warning in title(...): font family not found in Windows font database

## Warning in title(...): font family not found in Windows font database
```

```
# Add the second ROC curve for perf_lasso with a different color
plot(perf_lasso, colorize = FALSE, col = "orange", add = TRUE, family = "Times New Roman")
#plot(perf_ridge, colorize = FALSE, col = "brown", add = TRUE, family = "Times New Roman")
plot(perf_rf, colorize = FALSE, col = "black", add = TRUE, family = "Times New Roman")
plot(perf_xgboost, colorize = FALSE, col = "red", add = TRUE, family = "Times New Roman")

# Add a legend to the plot with Times New Roman font
legend("bottomright",  # Position of the legend (can change to topright, top, etc.)
       legend = c("Logistic Regression", "Lasso Regression", "Ridge Regression", "Random Forest",
"XGboost"),
       col = c("blue", "orange", "brown", "black", "red"),  # Colors of the curves
       lty = 1,  # Line type for the curves (solid line)
       cex = 0.8)  # Text size for the legend
```



```
    #  family = "Times New Roman")  # Font family for the legend
```

Oversampling train data (SMOTE)

```
#trying SMOTE
library(smotefamily)
train <- read.csv("train.csv")
library(caret)
library(nnet)
# Convert dropout to a factor
train$dropout <- as.numeric(train$dropout)
# Apply SMOTE
set.seed(123) # For reproducibility
smote_result <- SMOTE(X = train, target = train$dropout,
                      K = 4, dup_size = 0)

# Combine the SMOTE result into a new data frame
smotetrain <- data.frame(smote_result$data)

# Check the distribution of the target variable after SMOTE
table(smotetrain$dropout)
```

```
##
##     0     1
## 88165 86856
```

```
table(train$dropout)
```

```
##
##     0     1
## 88165  1551
```

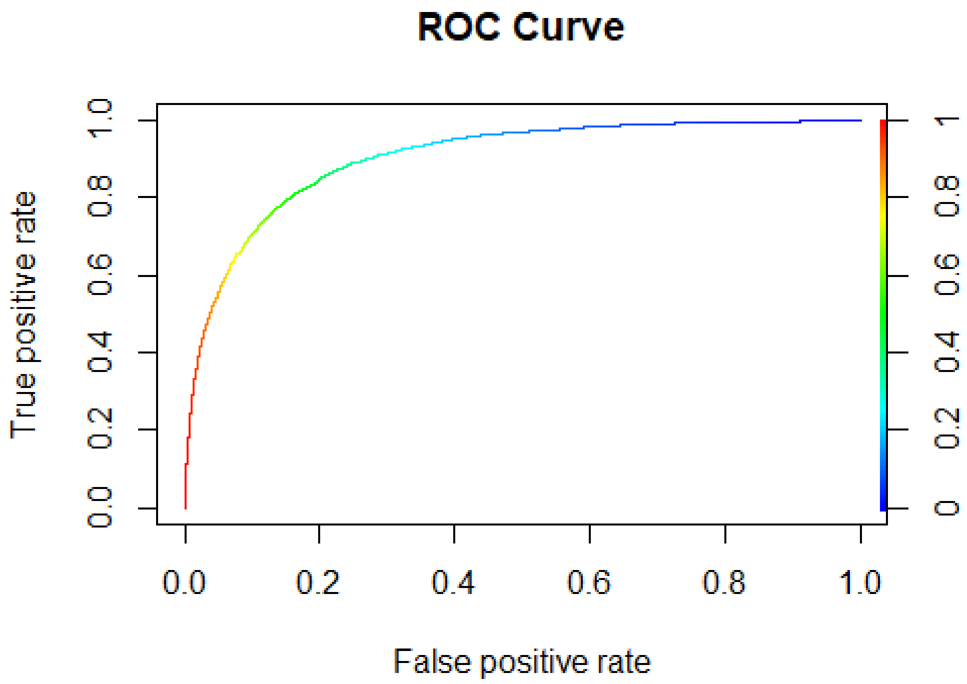
```
smotetrain <- smotetrain[,-49]
write.csv(smotetrain,"oversampletrain.csv", row.names=FALSE)
```

Model 6: SMOTE logistic regression

```
smotetrain <- read.csv("oversampletrain.csv")
test <- read.csv("test.csv")

# Fit the logistic regression model
log1.m <- glm(dropout ~ ., data = subset(smotetrain, select = -c(female, hispanic, asian, black, white,
other_race)), family = 'binomial')

# Predict on the TEST data
predict_log <- predict(log1.m, test[,-1], type = 'response')
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
# Create a prediction object for ROCR
pred <- prediction(predict_log, test$dropout)

# Create a performance object for ROC curve
perf_log <- performance(pred, "tpr", "fpr")

# Plot the ROC curve
plot(perf_log, colorize = TRUE, main = "ROC Curve")
```



```
# AuC score
auc <- performance(pred, measure = "auc")
auc@y.values[[1]]

## [1] 0.9027851

# Convert predictions to factors (assuming binary classification)
predict_log_class <- as.factor(ifelse(predict_log >= 0.5, 1, 0))
test$dropout <- as.factor(test$dropout)

# Create confusion matrix
cm <- confusionMatrix(data = predict_log_class, reference = test$dropout, positive = "1")
print(cm)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 77366   456
##          1 15307  1948
##
##                Accuracy : 0.8342
##                  95% CI : (0.8318, 0.8366)
##     No Information Rate : 0.9747
##     P-Value [Acc > NIR] : 1
```

```
## 
##                  Kappa : 0.1609
## 
##  Mcnemar's Test P-Value : <2e-16
## 
##             Sensitivity : 0.81032
##             Specificity : 0.83483
##          Pos Pred Value : 0.11289
##          Neg Pred Value : 0.99414
##              Prevalence : 0.02528
##          Detection Rate : 0.02049
##    Detection Prevalence : 0.18148
##       Balanced Accuracy : 0.82257
## 
##        'Positive' Class : 1
## 
```

Preparing for SMOTE lasso and ridge

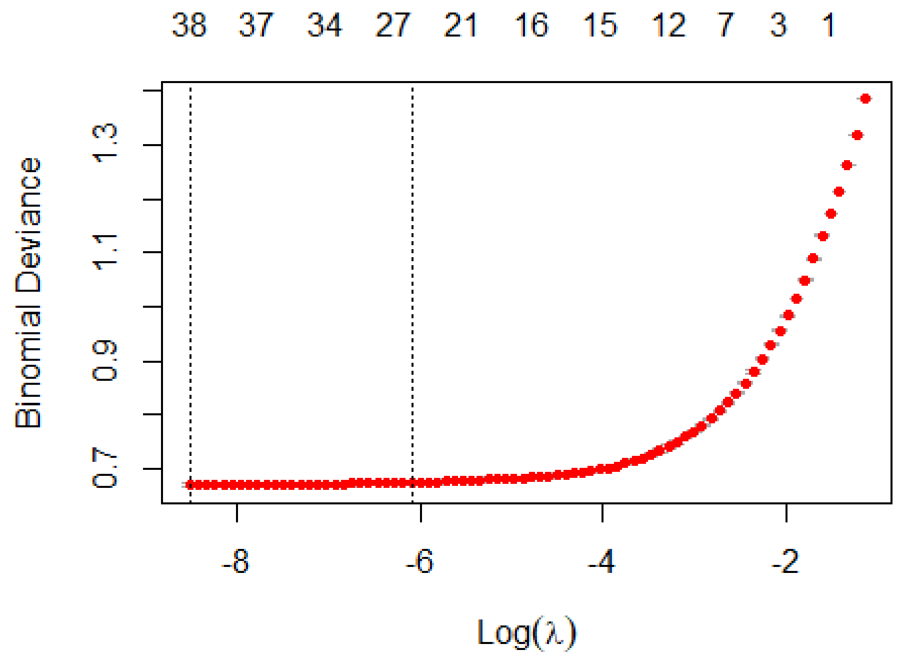
```
train <- read.csv("oversampletrain.csv")
test <- read.csv("test.csv")



train = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )

y.train = train$dropout %>% unlist() %>% as.numeric()
y.test = test$dropout %>% unlist() %>% as.numeric()
x.train = model.matrix(dropout~., train)[,-1] #data should only be predictors
x.test = model.matrix(dropout~., test)[,-1]

dim(x.train)
dim(x.test)



write.csv(x.train,'x.train.csv', row.names=FALSE)
write.csv(x.test,'x.test.csv', row.names=FALSE)
write.csv(y.train,'y.train.csv', row.names=FALSE)
write.csv(y.test,'y.test.csv', row.names=FALSE)
```

Model 7: SMOTE lasso regression

https://www.r-bloggers.com/2021/05/class-imbalance-handling-imbalanced-data-in-r/

```
set.seed(2023)
cv.lasso <- cv.glmnet(x.train, y.train, alpha = 1, family='binomial') # Fit lasso regression model on training data
```

```
#Display MSE vs log-lambda plot
plot(cv.lasso) # Draw plot of training MSE as a function of lambda
```



```
# ROC analysis to identify optimal threshold
lasso.pred <- predict(cv.lasso, newx=x.test, s = "lambda.min", type="response")
# Ensure lasso.pred is a numeric vector
lasso.pred <- as.numeric(lasso.pred)
print(length(lasso.pred))  # Check length of lasso.pred
```

```
## [1] 95077
```

```
#Create ROC curve
pred_lasso <- prediction(lasso.pred, y.test)
y.test <- as.matrix(y.test)
perf_lasso <- performance(pred_lasso , "tpr", "fpr")
#plot(perf_lasso, colorize=TRUE) #lasso prob threshold should be 0.2
#abline(h = 0.8, col = "red", lty = 2)  # Add threshold line
```

```
# AuC score
auc <- performance(pred_lasso, measure = "auc")
auc@y.values[[1]]
```

```
## [1] 0.9051221
```

```
# Convert predictions to factors
predict_lasso_class <- as.factor(ifelse(lasso.pred >= 0.4, "1", "0"))
```

```
# Ensure test$dropout is a factor with the same levels
test$dropout <- as.factor(test$dropout)
levels(predict_lasso_class) <- levels(test$dropout)  # Ensure factor levels match

# Create confusion matrix
cm <- confusionMatrix(data = predict_lasso_class, reference = test$dropout, positive = "1")
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 72832   327
##          1 19841  2077
##
##                Accuracy : 0.7879
##                  95% CI : (0.7853, 0.7905)
##     No Information Rate : 0.9747
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1312
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.86398
##             Specificity : 0.78590
##          Pos Pred Value : 0.09476
##          Neg Pred Value : 0.99553
##              Prevalence : 0.02528
##          Detection Rate : 0.02185
##    Detection Prevalence : 0.23053
##       Balanced Accuracy : 0.82494
##
##        'Positive' Class : 1
##
```

```
f1_score <- cm$byClass["F1"]
print(f1_score)
```

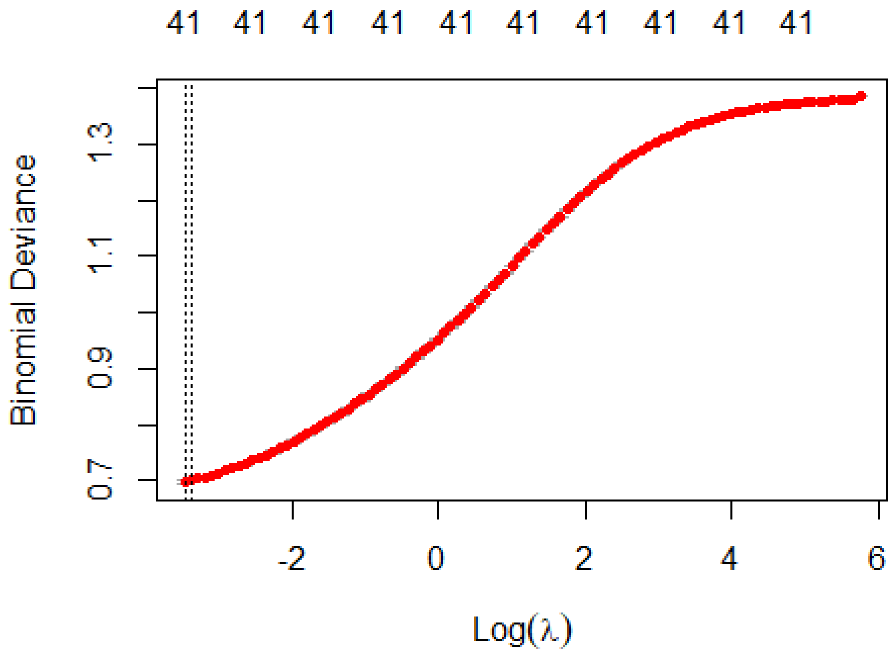
```
##        F1
## 0.1707919
```

Model 8: SMOTE ridge regression

https://www.r-bloggers.com/2021/05/class-imbalance-handling-imbalanced-data-in-r/

```
#CV to estimate best lambda
set.seed(2023)
cv.ridge <- cv.glmnet(x.train, y.train, alpha = 0, family='binomial') # Fit ridge regression model on
```

```
training data
#Display MSE vs log-lambda plot
plot(cv.ridge) # Draw plot of training MSE as a function of lambda
```



```
# Extract the coefficients at the best lambda (lambda.min or lambda.1se)

ridge.pred <- predict(cv.ridge, newx=x.test, s = "lambda.min", type="response")
# Ensure lasso.pred is a numeric vector
ridge.pred <- as.numeric(ridge.pred)
print(length(ridge.pred))  # Check length of lasso.pred
```

```
## [1] 95077
```

```
#Create ROC curve
pred_ridge <- prediction(ridge.pred, y.test)
y.test <- as.matrix(y.test)
perf_ridge <- performance(pred_ridge , "tpr", "fpr")
#plot_ridge <- plot(perf_ridge, colorize=TRUE) #lasso prob threshold should be 0.2
#abline(h = 0.8, col = "red", lty = 2)  # Add threshold line

# AuC
perf_ridge <- performance(pred_ridge,"auc")
auc <- as.numeric(perf_ridge@y.values)
auc
```

```
## [1] 0.9085196
```

```
# Convert predictions to factors
predict_ridge_class <- as.factor(ifelse(ridge.pred >= 0.2, "1", "0"))
# Ensure test$dropout is a factor with the same levels
test$dropout <- as.factor(test$dropout)
levels(predict_ridge_class) <- levels(test$dropout)  # Ensure factor levels match

# Create confusion matrix
cm <- confusionMatrix(data = predict_ridge_class, reference = test$dropout, positive = "1")
print(cm)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction     0     1
##          0 57299   117
##          1 35374  2287
## 
##                Accuracy : 0.6267
##                  95% CI : (0.6236, 0.6298)
##     No Information Rate : 0.9747
##     P-Value [Acc > NIR] : 1
## 
##                   Kappa : 0.07
## 
##  Mcnemar's Test P-Value : <2e-16
## 
##             Sensitivity : 0.95133
##             Specificity : 0.61829
##          Pos Pred Value : 0.06073
##          Neg Pred Value : 0.99796
##              Prevalence : 0.02528
##          Detection Rate : 0.02405
##    Detection Prevalence : 0.39611
##       Balanced Accuracy : 0.78481
## 
##        'Positive' Class : 1
## 
```

Model 9: SMOTE random forest

https://www.r-bloggers.com/2021/05/class-imbalance-handling-imbalanced-data-in-r/

```
train <- read.csv("oversampletrain.csv")
test <- read.csv("test.csv")
train_nodem = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )
```

```r
str(train_nodem)
```

```
## 'data.frame':    175021 obs. of  42 variables:
##  $ dropout                   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ ever_stsusp_middle        : num  1 1 1 1 1 1 0 1 0 1 ...
##  $ ever_ltsusp_middle        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ever_OSS_6                : num  0 1 0 0 1 1 0 0 0 0 ...
##  $ ever_OSS_7                : num  0 0 1 0 1 1 0 1 0 1 ...
##  $ ever_OSS_8                : num  1 1 0 1 0 0 0 0 0 1 ...
##  $ ever_OSS_middle           : num  1 1 1 1 1 1 0 1 0 1 ...
##  $ ever_ISS_middle           : num  0 0 0 0 1 1 0 1 0 1 ...
##  $ ever_ISS_6                : num  0 0 0 0 1 1 0 0 0 1 ...
##  $ ever_ISS_7                : num  0 0 1 0 1 1 0 1 0 1 ...
##  $ ever_ISS_8                : num  0 0 1 0 1 0 0 0 0 1 ...
##  $ not_math_proficient_6     : num  1 0 1 1 1 0 0 0 0 0 ...
##  $ not_math_proficient_7     : num  1 0 1 0 1 1 0 1 0 0 ...
##  $ not_math_proficient_8     : num  1 1 1 1 1 1 0 1 1 1 ...
##  $ no_math_proficiency_middle: num  1 0 1 0 1 0 0 0 0 0 ...
##  $ not_read_proficient_6     : num  1 0 1 0 1 1 0 1 0 1 ...
##  $ not_read_proficient_7     : num  1 1 1 1 1 1 0 1 0 1 ...
##  $ not_read_proficient_8     : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ no_read_proficiency_middle: num  1 0 1 0 1 1 0 1 0 1 ...
##  $ eds                       : num  1 0 1 0 1 1 0 1 1 1 ...
##  $ age_eighthfall1           : num  14.6 13.5 14.9 14.2 14.3 15.6 13.9 14.8 14.2 14.2 ...
##  $ ever_swd                  : num  1 0 0 0 1 0 0 0 0 1 ...
##  $ swd_8                     : num  1 0 0 0 1 0 0 0 0 1 ...
##  $ ever_lep                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ lep_8                     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ absence_rate_6            : num  0.05 0.02 0.03 0.01 0.04 ...
##  $ absence_rate_7            : num  0.07 0.02 0 0.03 0.03 ...
##  $ absence_rate_8            : num  0.13 0.07 0.06 0.13 0.13 ...
##  $ chrabsent_6               : num  0 0 0 0 0 1 1 0 0 0 ...
##  $ chrabsent_7               : num  0 0 0 0 0 1 1 0 1 1 ...
##  $ chrabsent_8               : num  1 0 0 1 1 0 1 1 1 1 ...
##  $ ever_chrabsent_middle     : num  1 0 0 1 1 1 1 1 1 1 ...
##  $ chrabsent_middle          : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ school_mobility_middle    : num  1 1 1 2 2 2 1 2 1 3 ...
##  $ school_mobility_8         : num  1 1 1 1 1 1 1 2 1 2 ...
##  $ school_mobility_7         : num  1 1 1 1 1 1 1 1 1 2 ...
##  $ school_mobility_6         : num  1 1 1 1 1 2 1 1 1 1 ...
##  $ urban                     : num  1 0 0 0 0 0 1 0 0 0 ...
##  $ suburban                  : num  0 1 0 0 0 1 0 0 0 0 ...
##  $ town                      : num  0 0 1 0 0 0 0 0 0 1 ...
##  $ rural                     : num  0 0 0 1 1 0 0 1 1 0 ...
##  $ ever_suspended            : num  1 1 1 1 1 1 0 1 0 1 ...
```

```r
train_nodem$dropout <- as.factor(train_nodem$dropout )
# Running RF
```

```
set.seed(2023)
RF.dropout <- randomForest(dropout ~ ., data = train_nodem, ntree = 100, importance = TRUE)


# Predict on the TEST data
rf.pred <- predict(RF.dropout, newdata = test[,-1], type = "prob")[,2]

# Create a prediction object for ROCR
rf_pr_test <- prediction(rf.pred, test$dropout)

# Create a performance object for ROC curve
perf_rf <- performance(rf_pr_test, "tpr", "fpr")

# Plot the ROC curve
#plot(perf_rf, colorize = TRUE, main = "ROC Curve")
#abline(h = 0.8, col = "red", lty = 2)  # Adjust according to your needs

# Calculate AUC
auc <- performance(rf_pr_test, measure = "auc")
print(auc@y.values[[1]])
```

```
## [1] 0.864723
```

```
# Convert predictions to binary class (assuming binary classification)
predict_rf_class <- as.factor(ifelse(rf.pred >= 0.1, 1, 0))
test$dropout <- as.factor(test$dropout)

# Create confusion matrix
cm <- confusionMatrix(data = predict_rf_class, reference = test$dropout, positive = "1")
print(cm)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction     0     1
##          0 76482   612
##          1 16191  1792
## 
##                Accuracy : 0.8233
##                  95% CI : (0.8208, 0.8257)
##     No Information Rate : 0.9747
##     P-Value [Acc > NIR] : 1
## 
##                   Kappa : 0.1373
## 
##  Mcnemar's Test P-Value : <2e-16
## 
##             Sensitivity : 0.74542
##             Specificity : 0.82529
##         Pos Pred Value : 0.09965
##         Neg Pred Value : 0.99206
##             Prevalence : 0.02528
##         Detection Rate : 0.01885
##   Detection Prevalence : 0.18914
##      Balanced Accuracy : 0.78536
##
##       'Positive' Class : 1
##
```

```
# F1 Score
f1_score <- cm$byClass["F1"]
print(f1_score)
```

```
##        F1
## 0.1757983
```

Preparing for SMOTE xgboost

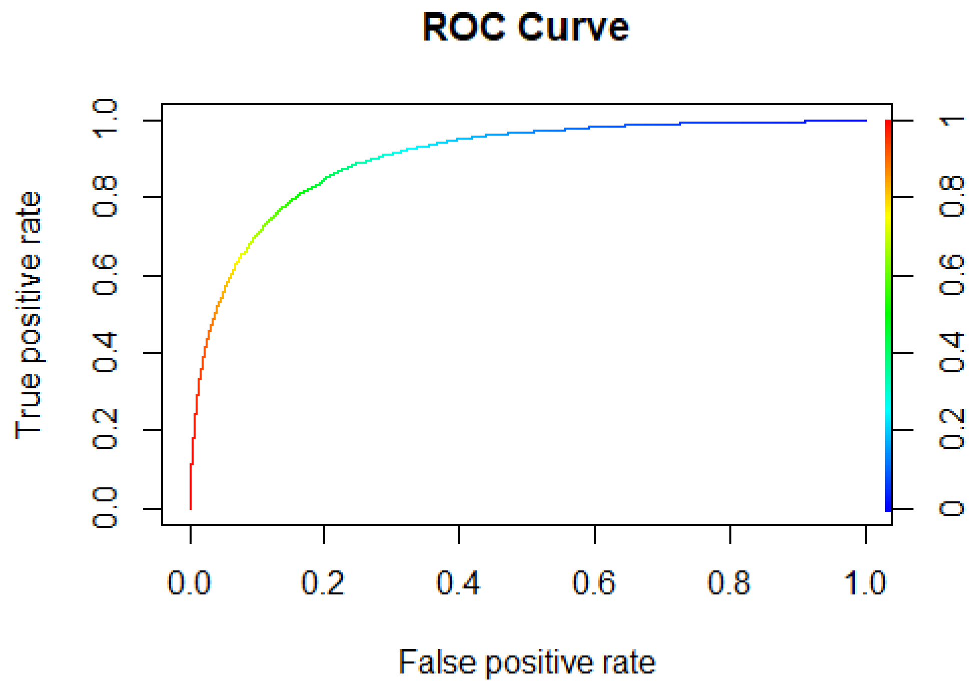
```
train <- read.csv("oversampletrain.csv")
test <- read.csv("test.csv")
train = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )

str(train)
summary(train$dropout)
str(test)

y.train = train$dropout %>% unlist() %>% as.numeric()
y.test = test$dropout %>% unlist() %>% as.numeric()
x.train = model.matrix(dropout~., train)[,-1] #data should only be predictors
x.test = model.matrix(dropout~., test)[,-1]



# Check the structure of data
str(x.train)
str(x.test)
str(y.test)

dtrain <- xgb.DMatrix(data = x.train, label = y.train)
dtest <- xgb.DMatrix(data = x.test, label = y.test)
ts_label <- test$dropout



# Initial parameter setup (if needed)
initial_params <- list(
  booster = "gbtree",
  objective = "binary:logistic",
  eval_metric = "logloss",
  eta = 0.3,
  max_depth = 6, gamma = 3
)

# Cross-validation to find optimal rounds of boosting
cv_results <- xgb.cv(
  params = initial_params,
  data = dtrain,
  nrounds = 100,
  nfold = 5,
  early_stopping_rounds = 20,
  verbose = 1
)

# Extract the Best Number of Rounds
best_nrounds <- cv_results$best_iteration

# Train the Final Model with Optimal Parameters
set.seed(2023)
final_model <- xgb.train(
  params = initial_params,
  data = dtrain,
  nrounds = best_nrounds
)

# Grid search for hyperparameter tuning
search_grid <- expand.grid(
  max_depth = c(3, 6),
  eta = c(0.01, 0.1),
  colsample_bytree = c(0.5, 0.7)
)

best_auc <- Inf  # Use Inf for minimization
best_params <- list()

for (i in 1:nrow(search_grid)) {
  params <- list(
    objective = "binary:logistic",
    eval_metric = "logloss",
    max_depth = search_grid$max_depth[i],
    eta = search_grid$eta[i],
    colsample_bytree = search_grid$colsample_bytree[i]
  )

  cv_results <- xgb.cv(
    params = params,
    data = dtrain,
    nfold = 5,
    nrounds = 100,
    early_stopping_rounds = 10,
    verbose = 1
  )

  mean_logloss <- min(cv_results$evaluation_log$test_logloss_mean)

  if (mean_logloss < best_auc) {
    best_auc <- mean_logloss
    best_params <- params
    best_nrounds <- cv_results$best_iteration
  }
}
```

Model 10: SMOTE xgboost

```
# Train the final model with the best parameters
dtest <- xgb.DMatrix(data = x.test, label = y.test)
set.seed(2023)
xgb1 <- xgb.train (params = best_params, data = dtrain, watchlist = list(val=dtest,train=dtrain),
print_every_n = 20, nrounds = best_nrounds)
```

```
## [1]  val-logloss:0.631128    train-logloss:0.631665
## [21] val-logloss:0.242482    train-logloss:0.208547
## [41] val-logloss:0.166629    train-logloss:0.118527
## [61] val-logloss:0.132950    train-logloss:0.082313
## [81] val-logloss:0.114859    train-logloss:0.063310
## [100]    val-logloss:0.103824    train-logloss:0.051447
```

```
#model prediction
xgbpred <- predict (xgb1,dtest)
xgbpred <- ifelse (xgbpred > 0.1,"1", "0")

y.test <- as.factor(y.test)
xgbpred <- as.factor(xgbpred)
y.test = test$dropout %>% unlist() %>% as.factor()

predict_xgboost <- predict(xgb1, dtest, type = 'response')
pred_xgboost <- prediction(predict_xgboost, test$dropout)
# Create a performance object for ROC curve
perf_xgboost <- performance(pred, "tpr", "fpr")
# Plot the first ROC curve (perf_log)
plot(perf_xgboost, colorize = TRUE, main = "ROC Curve")
```

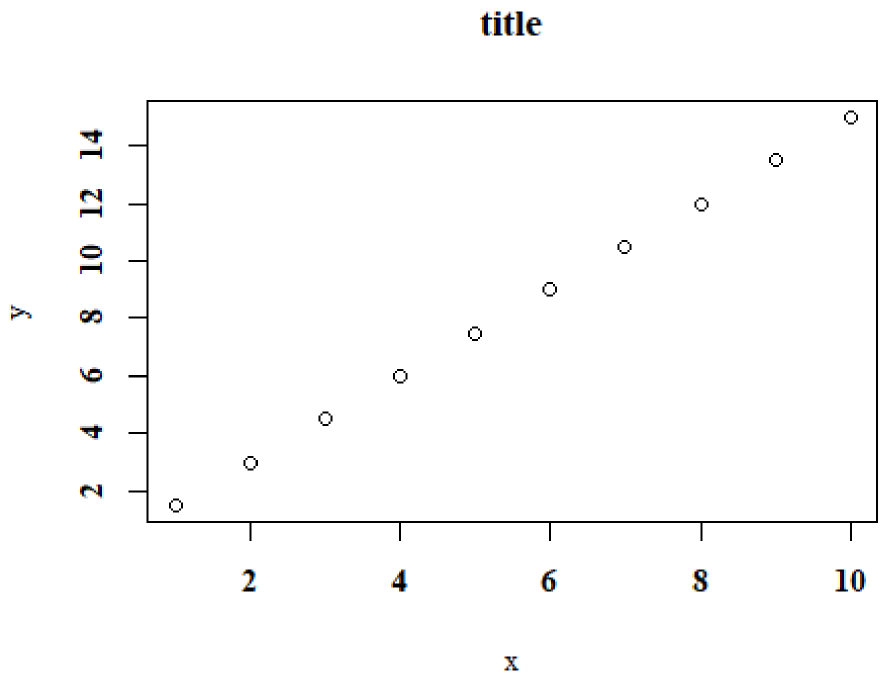


```
# Create confusion matrix
cm <- confusionMatrix(data = xgbpred, reference = y.test, positive = "1")
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 79909   656
##          1 12764  1748
##
##                Accuracy : 0.8589
##                  95% CI : (0.8566, 0.8611)
##     No Information Rate : 0.9747
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1707
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.72712
##             Specificity : 0.86227
##          Pos Pred Value : 0.12045
##          Neg Pred Value : 0.99186
##              Prevalence : 0.02528
##          Detection Rate : 0.01839
##    Detection Prevalence : 0.15263
##       Balanced Accuracy : 0.79469
##
##        'Positive' Class : 1
##
```

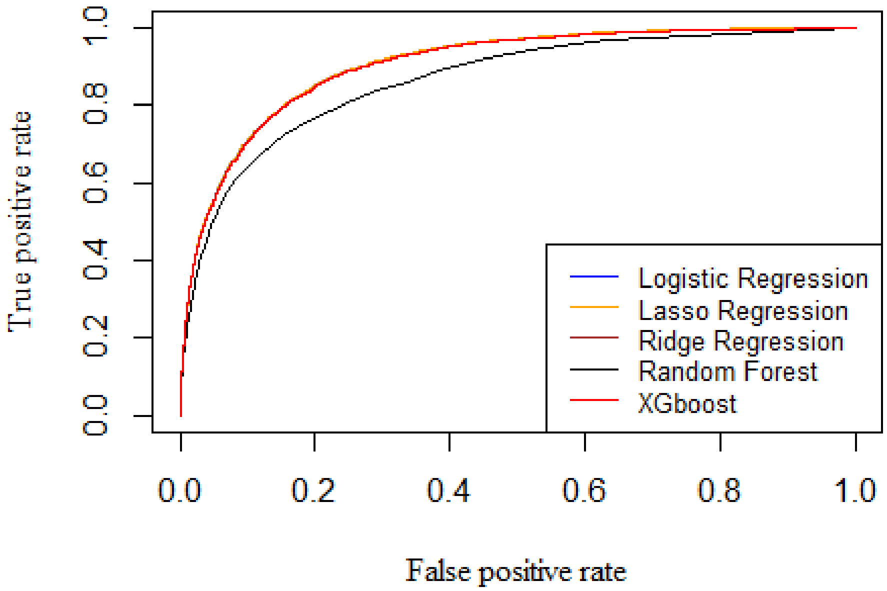
plotting SMOTE ROC curves in one graph

```
x = seq(1,10,1)
y = 1.5*x
windowsFonts(A = windowsFont("Times New Roman"))
plot(x, y,
  family="A",
  main = "title",
  font=2)
```



```
windowsFonts("Times New Roman" = windowsFont("Times New Roman"))

# Set the font to Times New Roman in the plots
plot(perf_log, colorize = FALSE, col = "blue", family = "Times New Roman")

# Add the second ROC curve for perf_lasso with a different color
plot(perf_lasso, colorize = FALSE, col = "orange", add = TRUE, family = "Times New Roman")
#plot(perf_ridge, colorize = FALSE, col = "brown", add = TRUE, family = "Times New Roman")
```

```
plot(perf_rf, colorize = FALSE, col = "black", add = TRUE, family = "Times New Roman")
plot(perf_xgboost, colorize = FALSE, col = "red", add = TRUE, family = "Times New Roman")

# Add a legend to the plot with Times New Roman font
legend("bottomright",  # Position of the legend (can change to topright, top, etc.)
       legend = c("Logistic Regression", "Lasso Regression", "Ridge Regression", "Random Forest", "XGboost"),
       col = c("blue", "orange", "brown", "black", "red"),  # Colors of the curves
       lty = 1,  # Line type for the curves (solid line)
       cex = 0.8)  # Text size for the legend
```



```
   #  family = "Times New Roman")  # Font family for the legend
```

Undersampling

```
train <- read.csv("train.csv")
test <- read.csv("test.csv")
str(train)
str(test)
library(ROSE)
```

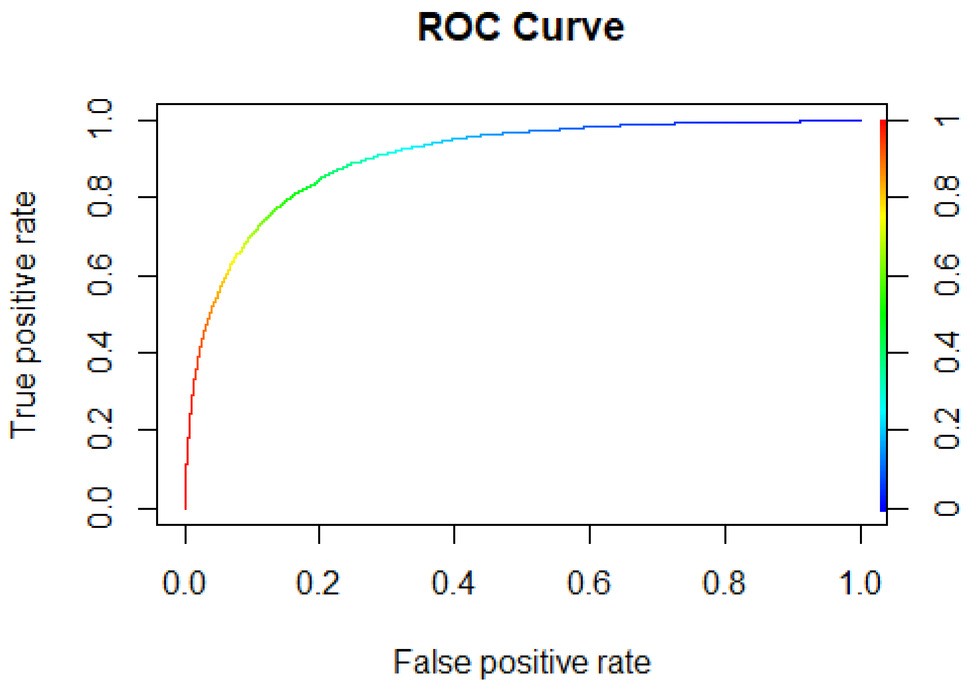
```
## Loaded ROSE 0.0-4
```

```
table(train$dropout)
```

```
# Perform undersampling
undersample_result <- ovun.sample(dropout ~ ., data = train, method = "under", N = 3102, seed = 1)

# Convert the result to a data frame
undersampletrain <- undersample_result$data

# Check the first few rows of the undersampled data
head(undersampletrain)

table(undersampletrain$dropout)
write.csv(undersampletrain,'undersampletrain.csv', row.names=FALSE)
```

Model 11: Undersample logistic regression

```
train <- read.csv("D:/NCERDC_DATA/Alam/ML/undersampletrain.csv")
test <- read.csv("test.csv")

log1.m <- glm(dropout ~ ., data = subset(train, select = -c(female, hispanic, asian, black, white,
other_race)), family = 'binomial')

# Create a prediction object for ROCR
pred <- prediction(predict_log, test$dropout)

# Create a performance object for ROC curve
perf_log <- performance(pred, "tpr", "fpr")

# Plot the ROC curve
plot(perf_log, colorize = TRUE, main = "ROC Curve")
```



```
# AuC score
auc <- performance(pred, measure = "auc")
auc@y.values[[1]]
```

```
## [1] 0.9027851
```

```
# Convert predictions to factors (assuming binary classification)
predict_log_class <- as.factor(ifelse(predict_log >= 0.6, 1, 0))
test$dropout <- as.factor(test$dropout)

# Create confusion matrix
cm <- confusionMatrix(data = predict_log_class, reference = test$dropout, positive = "1")
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 81108   595
##          1 11565  1809
##
##                Accuracy : 0.8721
##                  95% CI : (0.87, 0.8742)
##     No Information Rate : 0.9747
##     P-Value [Acc > NIR] : 1
```

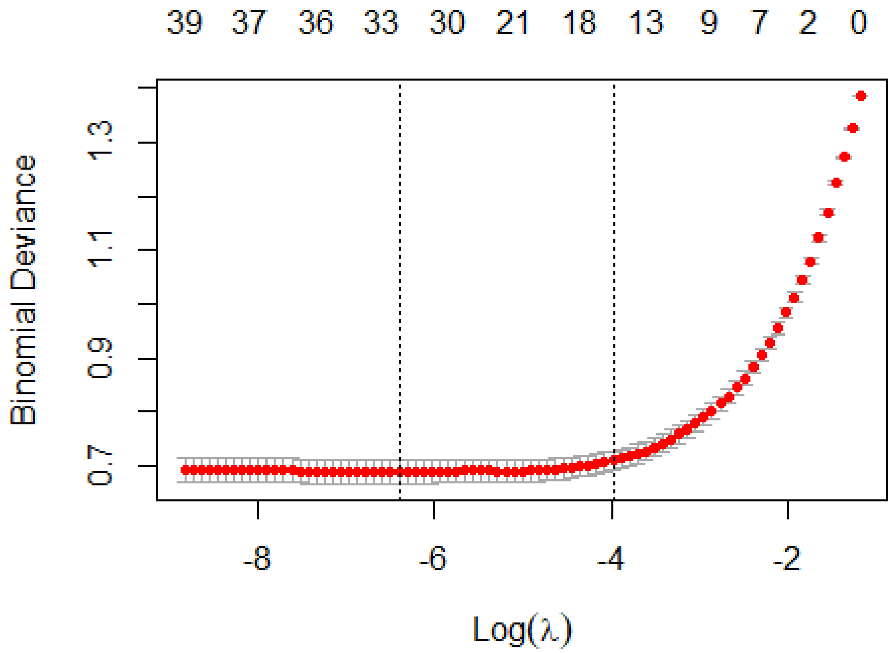
```
##
##                   Kappa : 0.1948
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.75250
##             Specificity : 0.87521
##          Pos Pred Value : 0.13526
##          Neg Pred Value : 0.99272
##              Prevalence : 0.02528
##          Detection Rate : 0.01903
##    Detection Prevalence : 0.14066
##       Balanced Accuracy : 0.81385
##
##        'Positive' Class : 1
##
```

Preparing for undersampled lasso and ridge

```
train <- read.csv("D:/NCERDC_DATA/Alam/ML/undersampletrain.csv")
test <- read.csv("test.csv")
#str(train)
#str(test)

train = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )

y.train = train$dropout %>% unlist() %>% as.numeric()
y.test = test$dropout %>% unlist() %>% as.numeric()
x.train = model.matrix(dropout~., train)[,-1] #data should only be predictors
x.test = model.matrix(dropout~., test)[,-1]

dim(x.train)
dim(x.test)

write.csv(x.train,'x.train.csv', row.names=FALSE)
write.csv(x.test,'x.test.csv', row.names=FALSE)
write.csv(y.train,'y.train.csv', row.names=FALSE)
write.csv(y.test,'y.test.csv', row.names=FALSE)
```

Model 12: Undersampled lasso regression

```
set.seed(2023)
cv.lasso <- cv.glmnet(x.train, y.train, alpha = 1, family='binomial') # Fit lasso regression model on training data
#Display MSE vs log-lambda plot
plot(cv.lasso) # Draw plot of training MSE as a function of lambda
```



```
# Extract the coefficients at the best lambda (lambda.min or lambda.1se)
lasso.coefs <- coef(cv.lasso, s = "lambda.min")  # or use lambda.1se for a more regularized solution


# ROC analysis to identify optimal threshold
lasso.pred <- predict(cv.lasso, newx=x.test, s = "lambda.min", type="response")
# Ensure lasso.pred is a numeric vector
lasso.pred <- as.numeric(lasso.pred)
print(length(lasso.pred))  # Check length of lasso.pred
```

```
## [1] 95077
```

```
#Create ROC curve
pred_lasso <- prediction(lasso.pred, y.test)
y.test <- as.matrix(y.test)
perf_lasso <- performance(pred_lasso , "tpr", "fpr")
#plot(perf_lasso, colorize=TRUE) #lasso prob threshold should be

# Convert predictions to factors
predict_lasso_class <- as.factor(ifelse(lasso.pred >= 0.4, "1", "0"))
# Ensure test$dropout is a factor with the same levels
test$dropout <- as.factor(test$dropout)
levels(predict_lasso_class) <- levels(test$dropout)  # Ensure factor levels match

# AuC score
```

```
auc <- performance(pred_lasso, measure = "auc")
auc@y.values[[1]]
```

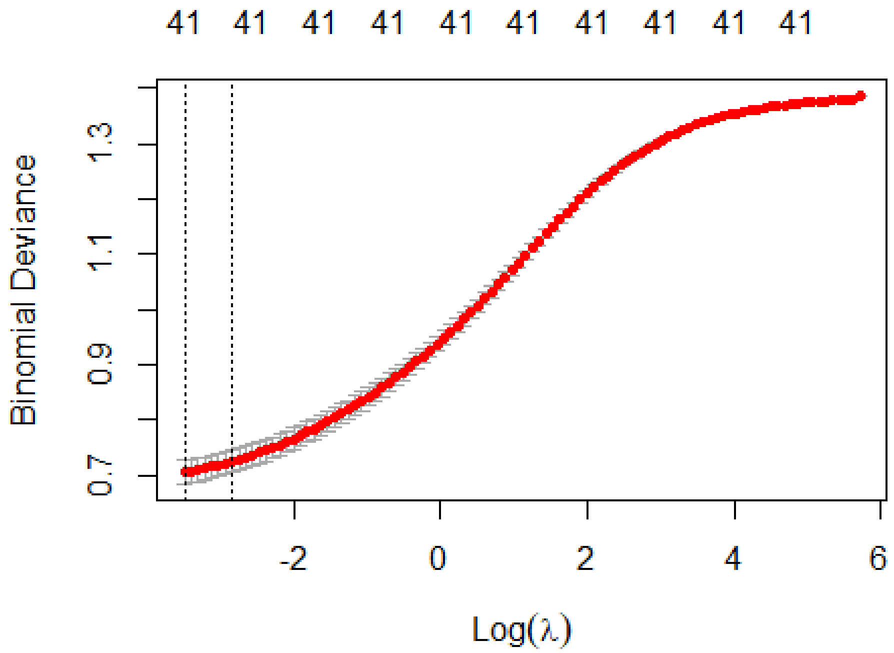
```
## [1] 0.9073362
```

```
# Create confusion matrix
cm <- confusionMatrix(data = predict_lasso_class, reference = test$dropout, positive = "1")
print(cm)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction     0     1
##          0 84621   749
##          1  8052  1655
## 
##                Accuracy : 0.9074
##                  95% CI : (0.9056, 0.9093)
##     No Information Rate : 0.9747
##     P-Value [Acc > NIR] : 1
## 
##                   Kappa : 0.2426
## 
##  Mcnemar's Test P-Value : <2e-16
## 
##             Sensitivity : 0.68844
##             Specificity : 0.91311
##          Pos Pred Value : 0.17050
##          Neg Pred Value : 0.99123
##              Prevalence : 0.02528
##          Detection Rate : 0.01741
##    Detection Prevalence : 0.10210
##       Balanced Accuracy : 0.80077
## 
##        'Positive' Class : 1
## 
```

Model 13: Undersampled ridge regression

```
#CV to estimate best lambda
set.seed(2023)
cv.ridge <- cv.glmnet(x.train, y.train, alpha = 0, family='binomial') # Fit ridge regression model on training data
#Display MSE vs log-lambda plot
plot(cv.ridge) # Draw plot of training MSE as a function of lambda
```

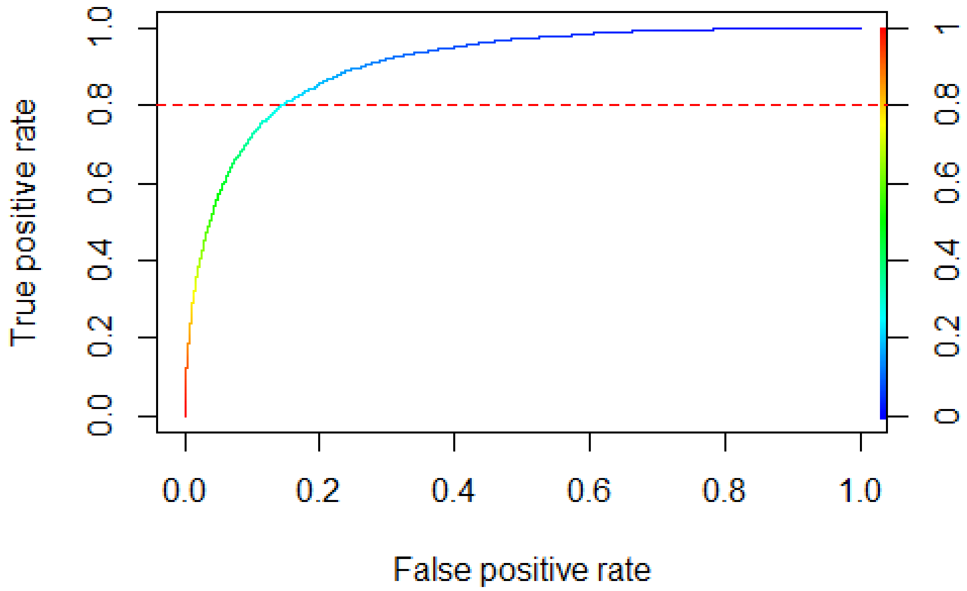


```
# Extract the coefficients at the best lambda (lambda.min or lambda.1se)
ridge.coefs <- coef(cv.ridge, s = "lambda.min")  # or use lambda.1se for a more regularized solution
```

```
ridge.pred <- predict(cv.ridge, newx=x.test, s = "lambda.min", type="response")
# Ensure lasso.pred is a numeric vector
ridge.pred <- as.numeric(ridge.pred)
print(length(ridge.pred))  # Check length of lasso.pred
```

```
## [1] 95077
```

```
#Create ROC curve
pred_ridge <- prediction(ridge.pred, y.test)
y.test <- as.matrix(y.test)
perf_ridge <- performance(pred_ridge , "tpr", "fpr")
plot_ridge <- plot(perf_ridge, colorize=TRUE) #lasso prob threshold should be 0.2
abline(h = 0.8, col = "red", lty = 2)  # Add threshold line
```



```
# AuC
perf_ridge <- performance(pred_ridge,"auc")
auc <- as.numeric(perf_ridge@y.values)
auc

## [1] 0.9075588

# Convert predictions to factors
predict_ridge_class <- as.factor(ifelse(ridge.pred >= 0.2, "1", "0"))
# Ensure test$dropout is a factor with the same levels
test$dropout <- as.factor(test$dropout)
levels(predict_ridge_class) <- levels(test$dropout)  # Ensure factor levels match

# Create confusion matrix
cm <- confusionMatrix(data = predict_ridge_class, reference = test$dropout, positive = "1")
print(cm)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 75133   376
##          1 17540  2028
##
##                Accuracy : 0.8116
##                95% CI : (0.8091, 0.814)
##     No Information Rate : 0.9747
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1461
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.84359
##             Specificity : 0.81073
##          Pos Pred Value : 0.10364
##          Neg Pred Value : 0.99502
##              Prevalence : 0.02528
##          Detection Rate : 0.02133
##    Detection Prevalence : 0.20581
##       Balanced Accuracy : 0.82716
##
##        'Positive' Class : 1
##
```

```
f1_score <- cm$byClass["F1"]
print(f1_score)
```

```
##        F1
## 0.1845986
```

Model 14: Undersampled random forest

```
train <- read.csv("undersampletrain.csv")
test <- read.csv("test.csv")
train_nodem = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )

str(train_nodem)
```

```
## 'data.frame':    3102 obs. of  42 variables:
##  $ dropout                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ever_stsusp_middle        : int  0 0 0 0 0 1 1 1 0 0 ...
##  $ ever_ltsusp_middle        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ever_OSS_6                : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ ever_OSS_7                : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ ever_OSS_8                : int  0 0 0 0 0 1 1 1 0 0 ...
##  $ ever_OSS_middle           : int  0 0 0 0 0 1 1 1 0 0 ...
##  $ ever_ISS_middle           : int  1 0 0 0 0 0 1 0 0 0 ...
##  $ ever_ISS_6                : int  1 0 0 0 0 0 1 0 0 0 ...
##  $ ever_ISS_7                : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ ever_ISS_8                : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ not_math_proficient_6     : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ not_math_proficient_7     : int  1 0 0 0 1 0 0 1 0 0 ...
##  $ not_math_proficient_8     : int  1 0 0 0 1 1 1 1 1 0 ...
##  $ no_math_proficiency_middle: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ not_read_proficient_6     : int  1 0 0 0 0 0 0 1 1 0 ...
##  $ not_read_proficient_7     : int  1 0 0 0 0 0 0 1 1 0 ...
##  $ not_read_proficient_8     : int  1 0 0 0 1 0 0 1 1 0 ...
##  $ no_read_proficiency_middle: int  1 0 0 0 0 0 0 1 1 0 ...
##  $ eds                       : int  0 0 1 0 0 1 1 1 0 0 ...
##  $ age_eighthfall1           : num  14 13.9 14 13.8 13.7 13.2 13.4 13.6 13.5 13.3 ...
##  $ ever_swd                  : int  0 0 0 0 0 0 0 1 1 0 ...
##  $ swd_8                     : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ ever_lep                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ lep_8                     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ absence_rate_6            : num  0 0.06 0.08 0.01 0.04 ...
##  $ absence_rate_7            : num  0 0.05 0.09 0 0 ...
##  $ absence_rate_8            : num  0 0.02 0.08 0.06 0.03 ...
##  $ chrabsent_6               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ chrabsent_7               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ chrabsent_8               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ever_chrabsent_middle     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ chrabsent_middle          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ school_mobility_middle    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ school_mobility_8         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ school_mobility_7         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ school_mobility_6         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ urban                     : int  0 1 0 0 0 1 0 1 0 0 ...
##  $ suburban                  : int  0 0 0 0 0 0 0 0 1 1 ...
##  $ town                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ rural                     : int  1 0 1 1 1 0 1 0 0 0 ...
##  $ ever_suspended            : int  1 0 0 0 0 1 1 1 0 0 ...
```

```r
train_nodem$dropout <- as.factor(train_nodem$dropout )
# Running RF
set.seed(2023)
RF.dropout <- randomForest(dropout ~ ., data = train_nodem, ntree = 100, importance = TRUE)


# Predict on the TEST data
rf.pred <- predict(RF.dropout, newdata = test[,-1], type = "prob")[,2]

# Create a prediction object for ROCR
rf_pr_test <- prediction(rf.pred, test$dropout)

# Create a performance object for ROC curve
perf_rf <- performance(rf_pr_test, "tpr", "fpr")

# Plot the ROC curve
#plot(perf_rf, colorize = TRUE, main = "ROC Curve")
#abline(h = 0.8, col = "red", lty = 2)  # Adjust according to your needs
```

```r
# Calculate AUC
auc <- performance(rf_pr_test, measure = "auc")
print(auc@y.values[[1]])
```

```
## [1] 0.899018
```

```r
# Convert predictions to binary class (assuming binary classification)
predict_rf_class <- as.factor(ifelse(rf.pred >= 0.4, 1, 0))
test$dropout <- as.factor(test$dropout)

# Create confusion matrix
cm <- confusionMatrix(data = predict_rf_class, reference = test$dropout, positive = "1")
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 68125   260
##          1 24548  2144
##
##                Accuracy : 0.7391
##                  95% CI : (0.7363, 0.7419)
##     No Information Rate : 0.9747
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1059
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.89185
##             Specificity : 0.73511
##          Pos Pred Value : 0.08032
##          Neg Pred Value : 0.99620
##              Prevalence : 0.02528
##          Detection Rate : 0.02255
##    Detection Prevalence : 0.28074
##       Balanced Accuracy : 0.81348
##
##        'Positive' Class : 1
##
```

```r
# F1 Score
f1_score <- cm$byClass["F1"]
print(f1_score)
```

```
##        F1
## 0.1473742
```

Preparing for Undersampled XGboost

```
train <- read.csv("undersampletrain.csv")
test <- read.csv("test.csv")
str(train)
str(test)

train = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )

y.train = train$dropout %>% unlist() %>% as.numeric()
y.test = test$dropout %>% unlist() %>% as.numeric()
x.train = model.matrix(dropout~., train)[,-1] #data should only be predictors
x.test = model.matrix(dropout~., test)[,-1]

# Check the structure of data
str(x.train)
str(x.test)
str(y.train)
str(y.test)

# Data preparation
dtrain <- xgb.DMatrix(data = x.train, label = y.train)
dtest <- xgb.DMatrix(data = x.test, label = y.test)
ts_label <- test$dropout


# Initial parameter setup (if needed)
initial_params <- list(
  booster = "gbtree",
  objective = "binary:logistic",
  eval_metric = "logloss",
  eta = 0.3,
  max_depth = 6, gamma = 3
)

# Cross-validation to find optimal rounds of boosting
cv_results <- xgb.cv(
  params = initial_params,
  data = dtrain,
  nrounds = 100,
  nfold = 5,
  early_stopping_rounds = 20,
  verbose = 1
)

# Extract the Best Number of Rounds
best_nrounds <- cv_results$best_iteration
```

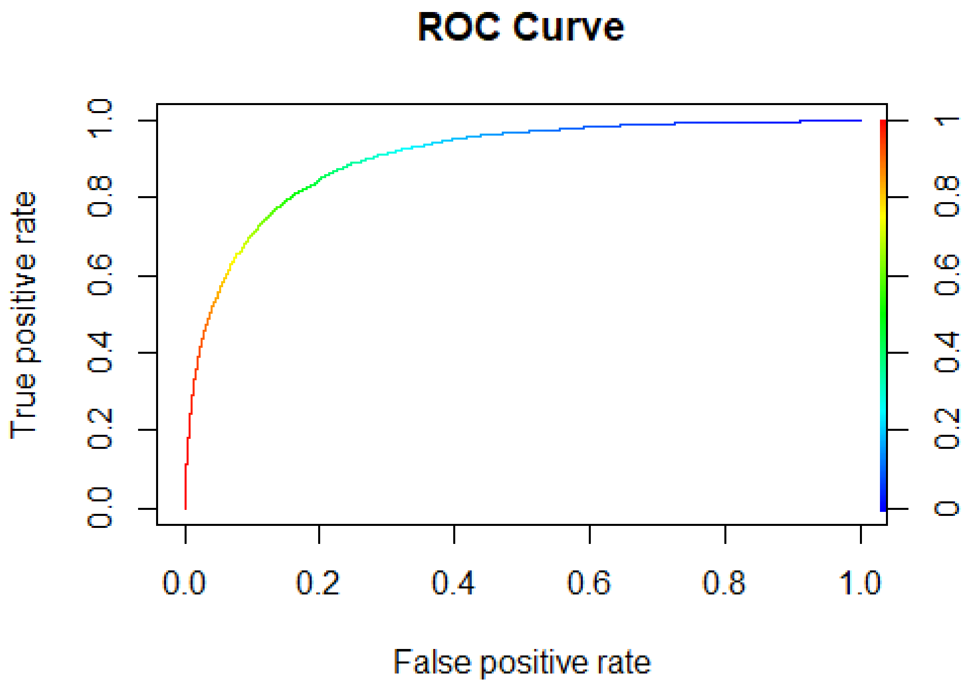
```
# Train the Final Model with Optimal Parameters
set.seed(2023)
final_model <- xgb.train(
  params = initial_params,
  data = dtrain,
  nrounds = best_nrounds
)

# Grid search for hyperparameter tuning
search_grid <- expand.grid(
  max_depth = c(3, 6),
  eta = c(0.01, 0.1),
  colsample_bytree = c(0.5, 0.7)
)

best_auc <- Inf  # Use Inf for minimization
best_params <- list()

for (i in 1:nrow(search_grid)) {
  params <- list(
    objective = "binary:logistic",
    eval_metric = "logloss",
    max_depth = search_grid$max_depth[i],
    eta = search_grid$eta[i],
    colsample_bytree = search_grid$colsample_bytree[i]
  )

  cv_results <- xgb.cv(
    params = params,
    data = dtrain,
    nfold = 5,
    nrounds = 100,
    early_stopping_rounds = 10,
    verbose = 1
  )

  mean_logloss <- min(cv_results$evaluation_log$test_logloss_mean)

  if (mean_logloss < best_auc) {
    best_auc <- mean_logloss
    best_params <- params
    best_nrounds <- cv_results$best_iteration
  }
}
```

Model 15: Undersampled XGboost

```
# Train the final model with the best parameters
dtest <- xgb.DMatrix(data = x.test, label = y.test)
set.seed(2023)
xgb1 <- xgb.train (params = best_params, data = dtrain, watchlist = list(val=dtest,train=dtrain),
print_every_n = 10, nrounds = best_nrounds)
```

```
## [1]  val-logloss:0.646212    train-logloss:0.647242
## [11] val-logloss:0.460663    train-logloss:0.441257
## [21] val-logloss:0.411490    train-logloss:0.370018
## [31] val-logloss:0.392529    train-logloss:0.340938
## [41] val-logloss:0.376640    train-logloss:0.325796
## [51] val-logloss:0.371436    train-logloss:0.317605
## [61] val-logloss:0.367045    train-logloss:0.311442
## [71] val-logloss:0.365427    train-logloss:0.306871
## [81] val-logloss:0.364583    train-logloss:0.303276
## [91] val-logloss:0.363684    train-logloss:0.299723
## [92] val-logloss:0.363615    train-logloss:0.299458
```

```
#model prediction
xgbpred <- predict (xgb1,dtest)
xgbpred <- ifelse (xgbpred > 0.4,"1", "0")

y.test <- as.factor(y.test)
xgbpred <- as.factor(xgbpred)
y.test = test$dropout %>% unlist() %>% as.factor()

predict_xgboost <- predict(xgb1, dtest, type = 'response')
pred_xgboost <- prediction(predict_xgboost, test$dropout)
# Create a performance object for ROC curve
perf_xgboost <- performance(pred, "tpr", "fpr")
# Plot the first ROC curve (perf_log)
plot(perf_xgboost, colorize = TRUE, main = "ROC Curve")
```



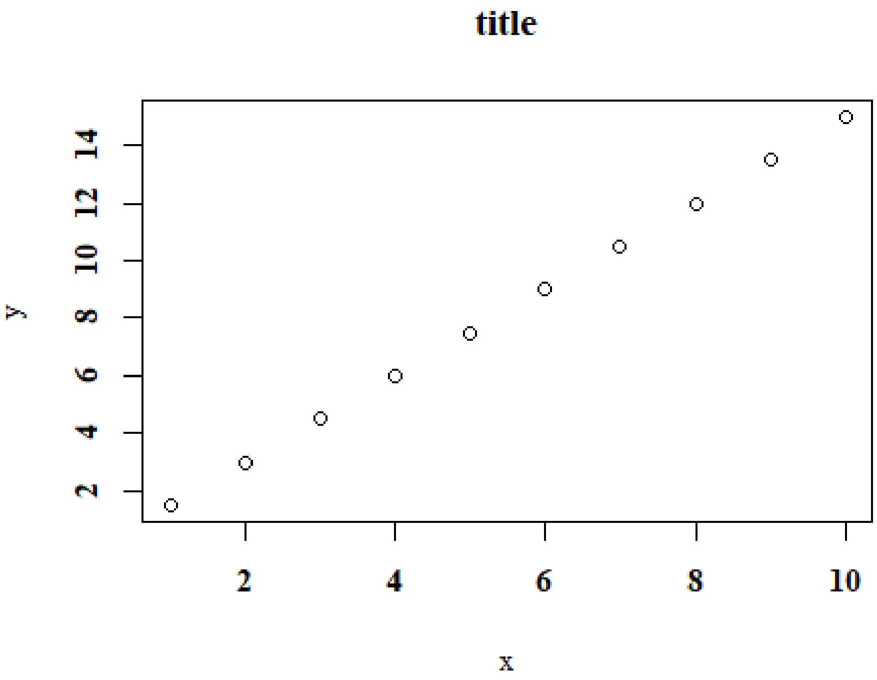
```
# Create confusion matrix
cm <- confusionMatrix(data = xgbpred, reference = y.test, positive = "1")
print(cm)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 73268   324
##          1 19405  2080
##
##                Accuracy : 0.7925
##                  95% CI : (0.7899, 0.7951)
##     No Information Rate : 0.9747
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1348
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.86522
##             Specificity : 0.79061
##          Pos Pred Value : 0.09681
##          Neg Pred Value : 0.99560
##              Prevalence : 0.02528
##          Detection Rate : 0.02188
##    Detection Prevalence : 0.22597
##       Balanced Accuracy : 0.82792
##
##        'Positive' Class : 1
##
```

```
auc <- performance(pred_xgboost, measure = "auc")
auc@y.values[[1]]
```

```
## [1] 0.9069999
```

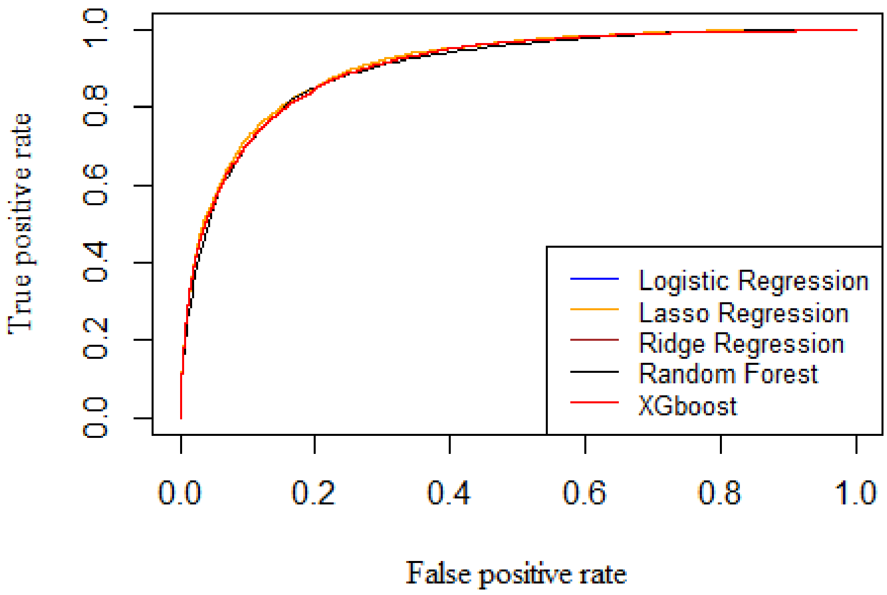
plotting undersampled ROC curves in one graph

```
x = seq(1,10,1)
y = 1.5*x
windowsFonts(A = windowsFont("Times New Roman"))
plot(x, y,
  family="A",
  main = "title",
  font=2)
```



```
windowsFonts(Times=windowsFont("Times New Roman"))
```

```
# Set the font to Times New Roman in the plots
plot(perf_log, colorize = FALSE, col = "blue", family = "Times New Roman")

# Add the second ROC curve for perf_lasso with a different color
plot(perf_lasso, colorize = FALSE, col = "orange", add = TRUE, family = "Times New Roman")
#plot(perf_ridge, colorize = FALSE, col = "brown", add = TRUE, family = "Times New Roman")
plot(perf_rf, colorize = FALSE, col = "black", add = TRUE, family = "Times New Roman")
plot(perf_xgboost, colorize = FALSE, col = "red", add = TRUE, family = "Times New Roman")

# Add a legend to the plot with Times New Roman font
legend("bottomright",  # Position of the legend (can change to topright, top, etc.)
      legend = c("Logistic Regression", "Lasso Regression", "Ridge Regression", "Random Forest",
"XGboost"),
      col = c("blue", "orange", "brown", "black", "red"),  # Colors of the curves
      lty = 1,  # Line type for the curves (solid line)
      cex = 0.8)  # Text size for the legend
```



```
   #  family = "Times New Roman")  # Font family for the legend
```

Figure B2: Code for research question 2

Data cleaning

```
#cleaning train data
train <- read.csv("D:/NCERDC_DATA/Alam/ML/Training sample/Data/trainingpanel.csv")
summary(train)
# str(train) this showed that almost no variables were factors
train <- train %>% mutate_if(is.integer, as.factor)
train = subset(train, select = -c(mastid) )
train <- train %>% as_tibble  %>% mutate(across(c(40:43), as.numeric))
str(train)

#cleaning test data
test <- read.csv("D:/NCERDC_DATA/Alam/ML/Testing sample/Data/testingpanel.csv")
test <- test %>% mutate_if(is.integer, as.factor)
test = subset(test, select = -c(mastid) )
train <- train %>% as_tibble  %>% mutate(across(c(40:43), as.numeric))
str(test)

write.csv(train,'train.csv', row.names=FALSE)
write.csv(test,'test.csv', row.names=FALSE)
```

ABROCA logistic regression

```
train <- read.csv("train.csv")
test <- read.csv("test.csv")

train$dropout <- as.factor(train$dropout)
test$dropout <- as.factor(test$dropout)

train <- train %>%
  mutate(across(c("female", "hispanic", "asian", "black", "white", "other_race", "eds", "lep_8",
"ever_lep", "swd_8", "ever_swd"), as.factor))

test <- test %>%
  mutate(across(c("female", "hispanic", "asian", "black", "white", "other_race", "eds", "lep_8",
"ever_lep", "swd_8", "ever_swd"), as.factor))


log1.m <- glm(dropout ~ ., data = subset(train, select = -c(female, hispanic, asian, black,
white,other_race, eds, lep_8, ever_lep, swd_8, ever_swd)), family='binomial')

test$pred = predict(log1.m, test, type = "response")

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
#LOOP for attributes where "0" is the majority (eds, ell, swd)
# Define a helper function to run ABROCA and print the result
run_abroca <- function(protected_attr, identifier) {
  result <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
                           protected_attr_col = protected_attr, majority_protected_attr_val = "0",
                           plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA
plots",
                           identifier = identifier)
  print(result)
}

# Run ABROCA for different protected attributes
run_abroca("female", "log reg female")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.00558827
```

```
run_abroca("eds", "log reg eds")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.04535414
```

```
run_abroca("lep_8", "log reg ell")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.08939902
```

```
run_abroca("swd_8", "log reg swd")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.05801772
```

```
#RACE ABROCA
abroca_logreg_white <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
            protected_attr_col = "white", majority_protected_attr_val = "1",
            plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA plots",
identifier="log reg white")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
print(abroca_logreg_white)
```

```
## [1] 0.03228699
```

Preparing for Lasso and Ridge

```
train <- read.csv("train.csv")
test <- read.csv("test.csv")

train = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )

y.train = train$dropout %>% unlist() %>% as.numeric()
y.test = test$dropout %>% unlist() %>% as.numeric()
x.train = model.matrix(dropout~., train)[,-1] #data should only be predictors
x.test = model.matrix(dropout~., test)[,-1]

dim(x.train)
dim(x.test)


write.csv(x.train,'x.train.csv', row.names=FALSE)
write.csv(x.test,'x.test.csv', row.names=FALSE)
write.csv(y.train,'y.train.csv', row.names=FALSE)
write.csv(y.test,'y.test.csv', row.names=FALSE)
```

ABROCA lasso regression

```
# Fit lasso regression model on training data
set.seed(2023)
cv.lasso <- cv.glmnet(x = x.train, y.train, alpha = 1, family='binomial')

#need to bring demographics back to test data
testdems <- read.csv("test.csv")
test <- testdems %>%
  mutate(across(c("female", "hispanic", "asian", "black", "white", "other_race", "eds", "lep_8",
"ever_lep", "swd_8", "ever_swd"), as.factor))
```

```
test$pred <- predict(cv.lasso, newx=x.test, s = "lambda.min", type="response")


#LOOP for attributes where "0" is the majority (eds, ell, swd)
# Define a helper function to run ABROCA and print the result
run_abroca <- function(protected_attr, identifier) {
  result <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
                           protected_attr_col = protected_attr, majority_protected_attr_val = "0",
                           plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA
plots",
                           identifier = identifier)
  print(result)
}

# Run ABROCA for different protected attributes
run_abroca("female", "lasso reg female")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.00448462
```

```
run_abroca("eds", "lasso reg eds")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.04490864
```

```
run_abroca("lep_8", "lasso reg ell")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.09048575
```

```
run_abroca("swd_8", "lasso reg swd")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.06055086
```

```
#RACE ABROCA
abroca_lassoreg_white <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
            protected_attr_col = "white", majority_protected_attr_val = "1",
            plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA plots",
identifier="lasso reg white")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
print(abroca_lassoreg_white)
```

```
## [1] 0.0371041
```

ABROCA ridge regression

```
set.seed(2023)
cv.ridge <- cv.glmnet(x = x.train, y.train, alpha = 0, family='binomial') # Fit lasso regression model on
training data


#need to bring demographics back to test data
testdems <- read.csv("test.csv")
test <- testdems %>%
  mutate(across(c("female", "hispanic", "asian", "black", "white", "other_race", "eds", "lep_8",
"ever_lep", "swd_8", "ever_swd"), as.factor))
test$pred <- predict(cv.ridge, newx=x.test, s = "lambda.min", type="response")


#LOOP for attributes where "0" is the majority (eds, ell, swd)
# Define a helper function to run ABROCA and print the result
run_abroca <- function(protected_attr, identifier) {
  result <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
                    protected_attr_col = protected_attr, majority_protected_attr_val = "0",
                    plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA
plots",
                    identifier = identifier)
  print(result)
}

# Run ABROCA for different protected attributes
run_abroca("female", "ridge reg female")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.004952419
```

```
run_abroca("eds", "ridge reg eds")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.04491096
```

```
run_abroca("lep_8", "ridge reg ell")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.09185735
```

```
run_abroca("swd_8", "ridge reg swd")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.05847756
```

```
#RACE ABROCA
abroca_ridgereg_white <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
              protected_attr_col = "white", majority_protected_attr_val = "1",
              plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA plots",
identifier="rige reg white")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
print(abroca_ridgereg_white)
```

```
## [1] 0.03696708
```

ABROCA random forest

```
train <- read.csv("train.csv")
test <- read.csv("test.csv")
train = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )
test <- test %>% mutate_if(is.factor, as.integer)
test$dropout <- as.factor(test$dropout)
train$dropout <- as.factor(train$dropout)


set.seed(2023)
RF.dropout <- randomForest(dropout ~ ., data = train, ntree = 100, importance = TRUE)
test$pred <- predict(RF.dropout, newdata = test, type = "prob")

#needno_math_proficiency_middle#need to bring demographics back to test data
testdems <- read.csv("test.csv")
test <- testdems %>%
  mutate(across(c("female", "hispanic", "asian", "black", "white", "other_race"), as.factor))
test$pred <- predict(RF.dropout, newdata = test, type = "prob")[,2]


#LOOP for attributes where "0" is the majority (eds, ell, swd)
# Define a helper function to run ABROCA and print the result
run_abroca <- function(protected_attr, identifier) {
  result <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
                          protected_attr_col = protected_attr, majority_protected_attr_val = "0",
                          plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA
plots",
                          identifier = identifier)
  print(result)
}

# Run ABROCA for different protected attributes
run_abroca("female", "rf female")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.02123401
```

```
run_abroca("eds", "rf eds")
```

```
## [WARNING] coercing column eds to factor
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.03441692
```

```
run_abroca("lep_8", "rf ell")
```

```
## [WARNING] coercing column lep_8 to factor

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.07710557
```

```
run_abroca("swd_8", "rf swd")
```

```
## [WARNING] coercing column swd_8 to factor

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.04108753
```

```
#RACE ABROCA
abroca_rf_white <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
            protected_attr_col = "white", majority_protected_attr_val = "1",
            plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA plots",
identifier="rf white")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
print(abroca_rf_white)
```

```
## [1] 0.01783665
```

Prepping for xgboost

```r
train <- read.csv("train.csv")
test <- read.csv("test.csv")

train = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )

y.train = train$dropout %>% unlist() %>% as.numeric()
y.test = test$dropout %>% unlist() %>% as.numeric()
x.train = model.matrix(dropout~., train)[,-1] #data should only be predictors
x.test = model.matrix(dropout~., test)[,-1]

# Data preparation
dtrain <- xgb.DMatrix(data = x.train, label = y.train)
dtest <- xgb.DMatrix(data = x.test, label = y.test)
ts_label <- test$dropout

# Initial parameter setup (if needed)
initial_params <- list(
  booster = "gbtree",
  objective = "binary:logistic",
  eval_metric = "logloss",
  eta = 0.3,
  max_depth = 6, gamma = 3
)

# Cross-validation to find optimal rounds of boosting
cv_results <- xgb.cv(
  params = initial_params,
  data = dtrain,
  nrounds = 100,
  nfold = 5,
  early_stopping_rounds = 20,
  verbose = 1
)

# Extract the Best Number of Rounds
best_nrounds <- cv_results$best_iteration

# Train the Final Model with Optimal Parameters
set.seed(2023)
final_model <- xgb.train(
  params = initial_params,
  data = dtrain,
  nrounds = best_nrounds
)

# Grid search for hyperparameter tuning
search_grid <- expand.grid(
  max_depth = c(3, 6),
  eta = c(0.01, 0.1),
  colsample_bytree = c(0.5, 0.7)
)

best_auc <- Inf  # Use Inf for minimization
best_params <- list()

for (i in 1:nrow(search_grid)) {
  params <- list(
    objective = "binary:logistic",
    eval_metric = "logloss",
    max_depth = search_grid$max_depth[i],
    eta = search_grid$eta[i],
    colsample_bytree = search_grid$colsample_bytree[i]
  )

  cv_results <- xgb.cv(
    params = params,
    data = dtrain,
    nfold = 5,
    nrounds = 100,
    early_stopping_rounds = 10,
    verbose = 1
  )

  mean_logloss <- min(cv_results$evaluation_log$test_logloss_mean)

  if (mean_logloss < best_auc) {
    best_auc <- mean_logloss
    best_params <- params
    best_nrounds <- cv_results$best_iteration
  }
}

# Train the final model with the best parameters
dtest <- xgb.DMatrix(data = x.test, label = y.test)
set.seed(2023)
xgb1 <- xgb.train (params = best_params, data = dtrain, watchlist = list(val=dtest,train=dtrain),
print_every_n = 10, nrounds = best_nrounds)
```

ABROCA xgboost

```
#need to bring demographics back to test data
testdems <- read.csv("test.csv")
test <- testdems %>%
  mutate(across(c("female", "hispanic", "asian", "black", "white", "other_race", "eds", "lep_8",
"ever_lep", "swd_8", "ever_swd"), as.factor))

test$pred <- predict(xgb1, dtest, type = 'response')
```

```
#LOOP for attributes where "0" is the majority (eds, ell, swd)
# Define a helper function to run ABROCA and print the result
run_abroca <- function(protected_attr, identifier) {
  result <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
                           protected_attr_col = protected_attr, majority_protected_attr_val = "0",
                           plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA
plots",
                           identifier = identifier)
  print(result)
}

# Run ABROCA for different protected attributes
run_abroca("female", "xgb female")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.005610438
```

```
run_abroca("eds", "xgb eds")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.04764772
```

```
run_abroca("lep_8", "xgb ell")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.09467474
```

```
run_abroca("swd_8", "xgb swd")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
## [1] 0.05967735
```

```
#RACE ABROCA
abroca_xgb_white <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
             protected_attr_col = "white", majority_protected_attr_val = "1",
             plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA plots",
identifier="xgb white")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
print(abroca_xgb_white)
```

```
## [1] 0.03376479
```

ABROCA SMOTE logistic regression

```
train <- read.csv("oversampletrain.csv")
test <- read.csv("test.csv")

train$dropout <- as.factor(train$dropout)
test$dropout <- as.factor(test$dropout)

train <- train %>%
  mutate(across(c("female", "hispanic", "asian", "black", "white", "other_race", "eds", "lep_8",
"ever_lep", "swd_8", "ever_swd"), as.factor))

test <- test %>%
  mutate(across(c("female", "hispanic", "asian", "black", "white", "other_race", "eds", "lep_8",
"ever_lep", "swd_8", "ever_swd"), as.factor))


log1.m <- glm(dropout ~ ., data = subset(train, select = -c(female, hispanic, asian, black,
white,other_race, eds, lep_8, ever_lep, swd_8, ever_swd)), family='binomial')

test$pred = predict(log1.m, test, type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
#LOOP for attributes where "0" is the majority (eds, ell, swd)
# Define a helper function to run ABROCA and print the result
run_abroca <- function(protected_attr, identifier) {
  result <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
                     protected_attr_col = protected_attr, majority_protected_attr_val = "0",
                     plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA
plots",
                    identifier = identifier)
  print(result)
}

# Run ABROCA for different protected attributes
run_abroca("female", "smote_log reg female")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.003953298
```

```
run_abroca("eds", "smote_log reg eds")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.04290329
```

```
run_abroca("lep_8", "smote_log reg ell")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.08240939
```

```
run_abroca("swd_8", "smote_log reg swd")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.05869391
```

```
#RACE ABROCA
smote_logreg_white <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
            protected_attr_col = "white", majority_protected_attr_val = "1",
            plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA plots",
identifier="smote_log reg white")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
print(smote_logreg_white)
```

```
## [1] 0.02947809
```

ABROCA SMOTE lasso regression

```
# Load data
train <- read.csv("oversampletrain.csv")
test <- read.csv("test.csv")

# Convert demographic columns to factors

# Prepare data for modeling
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race, eds, lep_8, ever_lep,
swd_8, ever_swd))
y.train = train$dropout %>% unlist() %>% as.numeric()
y.test = test$dropout %>% unlist() %>% as.numeric()
x.test = model.matrix(dropout~., test)[,-1]
x.train = model.matrix(dropout~., train)[,-1]

dim(x.train)
```

```
## [1] 175021     47
```

```
dim(x.test)
```

```
## [1] 95077    36
```

```
# Set seed for reproducibility
set.seed(2023)

# Lasso regression
cv.lasso <- cv.glmnet(x = subset(x.train, select = -c(female, hispanic, asian, black, white, other_race, eds,
lep_8, ever_lep, swd_8, ever_swd)), y.train, alpha = 1, family='binomial') #

# Reload demographic data for test set
testdems <- read.csv("test.csv")
test <- testdems %>%
  mutate(across(c("female", "hispanic", "asian", "black", "white", "other_race", "eds", "lep_8",
"ever_lep", "swd_8", "ever_swd"), as.factor))
test$predlasso <- predict(cv.lasso, newx=x.test, s = "lambda.min", type="response")


run_abroca <- function(protected_attr, identifier) {
  result <- compute_abroca(test, pred_col = "predlasso", label_col = "dropout",
                           protected_attr_col = protected_attr, majority_protected_attr_val = "0",
                           plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA
plots",
                           identifier = identifier)
  print(result)
}
```

```
# Run ABROCA for Lasso and Ridge models with different protected attributes
run_abroca("female", "smote_lasso reg female")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.003708448
```

```
run_abroca("eds", "smote_lasso reg eds")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.04281845
```

```
run_abroca("lep_8", "smote_lasso reg ell")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.08001103
```

```
run_abroca("swd_8", "smote_lasso reg swd")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.06088197
```

```
#RACE ABROCA
smote_lasso_white <- compute_abroca(test, pred_col = "predlasso", label_col = "dropout",
            protected_attr_col = "white", majority_protected_attr_val = "1",
            plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA plots",
identifier="smote_lasso reg white")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
print(smote_lasso_white)
```

```
## [1] 0.02880066
```

ABROCA SMOTE ridge regression

```
# Ridge regression
cv.ridge <- cv.glmnet(x = subset(x.train, select = -c(female, hispanic, asian, black, white, other_race, eds,
lep_8, ever_lep, swd_8, ever_swd)),
                  y.train, alpha = 0, family='binomial') # Ridge regression
test$predridge <- predict(cv.ridge, newx=x.test, s = "lambda.min", type="response")

run_abroca <- function(protected_attr, identifier) {
  result <- compute_abroca(test, pred_col = "predridge", label_col = "dropout",
                      protected_attr_col = protected_attr, majority_protected_attr_val = "0",
                      plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA
plots",
                      identifier = identifier)
  print(result)
}


run_abroca("female", "smote_ridge reg female")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.004925568
```

```
run_abroca("eds", "smote_ridge reg eds")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.04434673
```

```
run_abroca("lep_8", "smote_ridge reg ell")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.0864775
```

```
run_abroca("swd_8", "smote_ridge reg swd")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.05766656
```

```
# Race-specific ABROCA for Ridge regression
smote_ridge_white <- compute_abroca(test, pred_col = "predridge", label_col = "dropout",
                              protected_attr_col = "white", majority_protected_attr_val = "1",
                              plot_slices = TRUE,
image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA plots",
                              identifier="smote_ridge reg white")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
print(smote_ridge_white)
```

```
## [1] 0.03225496
```

ABROCA SMOTE RF

```
train <- read.csv("oversampletrain.csv")
test <- read.csv("test.csv")
train = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )
test <- test %>% mutate_if(is.factor, as.integer)
test$dropout <- as.factor(test$dropout)
train$dropout <- as.factor(train$dropout)
```

```
set.seed(2023)
RF.dropout <- randomForest(dropout ~ ., data = train, ntree = 100, importance = TRUE)
test$pred <- predict(RF.dropout, newdata = test, type = "prob")

#needno_math_proficiency_middle#need to bring demographics back to test data
testdems <- read.csv("test.csv")
test <- testdems %>%
  mutate(across(c("female", "hispanic", "asian", "black", "white", "other_race"), as.factor))
test$pred <- predict(RF.dropout, newdata = test, type = "prob")[,2]


#LOOP for attributes where "0" is the majority (eds, ell, swd)
# Define a helper function to run ABROCA and print the result
run_abroca <- function(protected_attr, identifier) {
  result <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
                           protected_attr_col = protected_attr, majority_protected_attr_val = "0",
                           plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA
plots",
                           identifier = identifier)
  print(result)
}

# Run ABROCA for different protected attributes
run_abroca("female", "rf smote female")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.01724477
```

```
run_abroca("eds", "rf smote eds")
```

```
## [WARNING] coercing column eds to factor

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.03761579
```

```
run_abroca("lep_8", "rf smote ell")
```

```
## [WARNING] coercing column lep_8 to factor
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.09039756
```

```
run_abroca("swd_8", "rf smote swd")
```

```
## [WARNING] coercing column swd_8 to factor
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.04336146
```

```
#RACE ABROCA
abroca_rfsmote_white <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
            protected_attr_col = "white", majority_protected_attr_val = "1",
            plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA plots",
identifier=" rf smote white")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
print(abroca_rfsmote_white)
```

```
## [1] 0.02631874
```

Prep for SMOTE xgboost

```
train <- read.csv("oversampletrain.csv")
test <- read.csv("test.csv")
str(train)
str(test)

train = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )

y.train = train$dropout %>% unlist() %>% as.numeric()
y.test = test$dropout %>% unlist() %>% as.numeric()
x.train = model.matrix(dropout~., train)[,-1] #data should only be predictors
x.test = model.matrix(dropout~., test)[,-1]
```

```r
# Data preparation
dtrain <- xgb.DMatrix(data = x.train, label = y.train)
dtest <- xgb.DMatrix(data = x.test, label = y.test)
ts_label <- test$dropout


# Initial parameter setup (if needed)
initial_params <- list(
  booster = "gbtree",
  objective = "binary:logistic",
  eval_metric = "logloss",
  eta = 0.3,
  max_depth = 6, gamma = 3
)

# Cross-validation to find optimal rounds of boosting
cv_results <- xgb.cv(
  params = initial_params,
  data = dtrain,
  nrounds = 100,
  nfold = 5,
  early_stopping_rounds = 20,
  verbose = 1
)

# Extract the Best Number of Rounds
best_nrounds <- cv_results$best_iteration

# Train the Final Model with Optimal Parameters
set.seed(2023)
final_model <- xgb.train(
  params = initial_params,
  data = dtrain,
  nrounds = best_nrounds
)

# Grid search for hyperparameter tuning
search_grid <- expand.grid(
  max_depth = c(3, 6),
  eta = c(0.01, 0.1),
  colsample_bytree = c(0.5, 0.7)
)

best_auc <- Inf  # Use Inf for minimization
best_params <- list()

for (i in 1:nrow(search_grid)) {
  params <- list(
    objective = "binary:logistic",
    eval_metric = "logloss",
    max_depth = search_grid$max_depth[i],
    eta = search_grid$eta[i],
    colsample_bytree = search_grid$colsample_bytree[i]
  )

  cv_results <- xgb.cv(
    params = params,
    data = dtrain,
    nfold = 5,
    nrounds = 100,
    early_stopping_rounds = 10,
    verbose = 1
  )

  mean_logloss <- min(cv_results$evaluation_log$test_logloss_mean)

  if (mean_logloss < best_auc) {
    best_auc <- mean_logloss
    best_params <- params
    best_nrounds <- cv_results$best_iteration
  }
}

# Train the final model with the best parameters
dtest <- xgb.DMatrix(data = x.test, label = y.test)
set.seed(2023)
xgb1 <- xgb.train (params = best_params, data = dtrain, watchlist = list(val=dtest,train=dtrain),
print_every_n = 10, nrounds = best_nrounds)
```

ABROCA SMOTE xgboost

```
#need to bring demographics back to test data
testdems <- read.csv("test.csv")
test <- testdems %>%
  mutate(across(c("female", "hispanic", "asian", "black", "white", "other_race", "eds", "lep_8",
"ever_lep", "swd_8", "ever_swd"), as.factor))

test$pred <- predict(xgb1, dtest, type = 'response')


#LOOP for attributes where "0" is the majority (eds, ell, swd)
# Define a helper function to run ABROCA and print the result
run_abroca <- function(protected_attr, identifier) {
  result <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
                           protected_attr_col = protected_attr, majority_protected_attr_val = "0",
                           plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA
plots",
                    identifier = identifier)
  print(result)
}

# Run ABROCA for different protected attributes
run_abroca("female", "smote_xgb female")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.006328533
```

```
run_abroca("eds", "smote_xgb eds")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.03154205
```

```
run_abroca("lep_8", "smote_xgb ell")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.1003346
```

```
run_abroca("swd_8", "smote_xgb swd")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.06308864
```

```
#RACE ABROCA
smote_xgb_white <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
           protected_attr_col = "white", majority_protected_attr_val = "1",
           plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA plots",
identifier="smote_xgb white")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
print(smote_xgb_white)
```

```
## [1] 0.03473689
```

ABROCA US log reg

```
train <- read.csv("D:/NCERDC_DATA/Alam/ML/undersampletrain.csv")
test <- read.csv("test.csv")

log1.m <- glm(dropout ~ ., data = subset(train, select = -c(female, hispanic, asian, black, white,
other_race)), family = 'binomial')

test$pred = predict(log1.m, test, type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
#LOOP for attributes where "0" is the majority (eds, ell, swd)
# Define a helper function to run ABROCA and print the result
run_abroca <- function(protected_attr, identifier) {
  result <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
                           protected_attr_col = protected_attr, majority_protected_attr_val = "0",
                           plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA
plots",
                           identifier = identifier)
  print(result)
}

# Run ABROCA for different protected attributes
run_abroca("female", "US_log reg female")
```

```
## [WARNING] coercing column female to factor
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.005373849
```

```
run_abroca("eds", "US_log reg eds")
```

```
## [WARNING] coercing column eds to factor
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.04300543
```

```
run_abroca("lep_8", "US_log reg ell")
```

```
## [WARNING] coercing column lep_8 to factor

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.09075302
```

```
run_abroca("swd_8", "US_log reg swd")
```

```
## [WARNING] coercing column swd_8 to factor

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.05623001
```

```
#RACE ABROCA
US_logreg_white <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
              protected_attr_col = "white", majority_protected_attr_val = "1",
              plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA plots",
identifier="US_log reg white")
```

```
## [WARNING] coercing column white to factor

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
print(US_logreg_white)
```

```
## [1] 0.04132319
```

ABROCA US lasso regression

```
# Load data
train <- read.csv("D:/NCERDC_DATA/Alam/ML/undersampletrain.csv")
test <- read.csv("test.csv")

# Convert demographic columns to factors

# Prepare data for modeling
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race, eds, lep_8, ever_lep,
swd_8, ever_swd))
y.train = train$dropout %>% unlist() %>% as.numeric()
y.test = test$dropout %>% unlist() %>% as.numeric()
x.test = model.matrix(dropout~., test)[,-1]
gc()
```

```
##           used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells  3183149 170.0    5829654  311.4   5829654  311.4
## Vcells 28920979 220.7  318582535 2430.6 398216737 3038.2
```

```
x.train = model.matrix(dropout~., train)[,-1]

dim(x.train)
```

```
## [1] 3102   47
```

```
dim(x.test)
```

```
## [1] 95077    36
```

```
# Set seed for reproducibility
set.seed(2023)

# Lasso regression
cv.lasso <- cv.glmnet(x = subset(x.train, select = -c(female, hispanic, asian, black, white, other_race, eds,
lep_8, ever_lep, swd_8, ever_swd)),
                      y.train, alpha = 1, family='binomial') # Lasso regression
test$predlasso <- predict(cv.lasso, newx=x.test, s = "lambda.min", type="response")

# Reload demographic data for test set
testdems <- read.csv("test.csv")
test <- testdems %>%
  mutate(across(c("female", "hispanic", "asian", "black", "white", "other_race", "eds", "lep_8",
"ever_lep", "swd_8", "ever_swd"), as.factor))
test$predlasso <- predict(cv.lasso, newx=x.test, s = "lambda.min", type="response")



run_abroca <- function(protected_attr, identifier) {
  result <- compute_abroca(test, pred_col = "predlasso", label_col = "dropout",
                           protected_attr_col = protected_attr, majority_protected_attr_val = "0",
                           plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA
plots",
                          identifier = identifier)
  print(result)
}


# Run ABROCA for Lasso and Ridge models with different protected attributes
run_abroca("female", "smote_lasso reg female")

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.004281276

run_abroca("eds", "US_lasso reg eds")

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.04333633

run_abroca("lep_8", "US_lasso reg ell")

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.08555474

run_abroca("swd_8", "US_lasso reg swd")

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] 0.05609477

#RACE ABROCA
US_lasso_white <- compute_abroca(test, pred_col = "predlasso", label_col = "dropout",
              protected_attr_col = "white", majority_protected_attr_val = "1",
               plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA plots",
identifier="US_lasso reg white")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
print(US_lasso_white)
```

```
## [1] 0.03080395
```

ABROCA US ridge regression

```
#CV to estimate best lambda
set.seed(2023)
cv.ridge <- cv.glmnet(x.train, y.train, alpha = 0, family='binomial') # Fit ridge regression model on
training data
cv.ridge <- cv.glmnet(x = subset(x.train, select = -c(female, hispanic, asian, black, white, other_race, eds,
lep_8, ever_lep, swd_8, ever_swd)),
                      y.train, alpha = 0, family='binomial') # Ridge regression
test$predridge <- predict(cv.ridge, newx=x.test, s = "lambda.min", type="response")

run_abroca <- function(protected_attr, identifier) {
  result <- compute_abroca(test, pred_col = "predridge", label_col = "dropout",
                           protected_attr_col = protected_attr, majority_protected_attr_val = "0",
                           plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA
plots",
                           identifier = identifier)
  print(result)
}


run_abroca("female", "US_ridge reg female")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.00567008
```

```
run_abroca("eds", "US_ridge reg eds")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.04602865
```

```
run_abroca("lep_8", "US_ridge reg ell")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.09220061
```

```
run_abroca("swd_8", "US_ridge reg swd")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.05361092
```

```
# Race-specific ABROCA for Ridge regression
US_ridgereg_white <- compute_abroca(test, pred_col = "predridge", label_col = "dropout",
                                    protected_attr_col = "white", majority_protected_attr_val = "1",
                                    plot_slices = TRUE,
image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA plots",
                                    identifier="US_ridge reg white")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
print(US_ridgereg_white)
```

```
## [1] 0.0337102
```

ABROCA US random forest

```
train <- read.csv("undersampletrain.csv")
test <- read.csv("test.csv")
train = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )
test <- test %>% mutate_if(is.factor, as.integer)
test$dropout <- as.factor(test$dropout)
train$dropout <- as.factor(train$dropout)
```

```
set.seed(2023)
RF.dropout <- randomForest(dropout ~ ., data = train, ntree = 100, importance = TRUE)
test$pred <- predict(RF.dropout, newdata = test, type = "prob")

#needno_math_proficiency_middle#need to bring demographics back to test data
testdems <- read.csv("test.csv")
test <- testdems %>%
  mutate(across(c("female", "hispanic", "asian", "black", "white", "other_race"), as.factor))
test$pred <- predict(RF.dropout, newdata = test, type = "prob")[,2]


#LOOP for attributes where "0" is the majority (eds, ell, swd)
# Define a helper function to run ABROCA and print the result
run_abroca <- function(protected_attr, identifier) {
  result <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
                           protected_attr_col = protected_attr, majority_protected_attr_val = "0",
                           plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA
plots",
                           identifier = identifier)
  print(result)
}

# Run ABROCA for different protected attributes
run_abroca("female", "US_rf female")
```

```
## [1] 0.01252986
```

```
run_abroca("eds", "US_rf eds")
```

```
## [WARNING] coercing column eds to factor
```

```
## [1] 0.04422393
```

```
run_abroca("lep_8", "US_rf ell")
```

```
## [WARNING] coercing column lep_8 to factor
```

```
## [1] 0.09303252
```

```
run_abroca("swd_8", "US_rf swd")
```

```
## [WARNING] coercing column swd_8 to factor
```

```
## [1] 0.05869447
```

```
#RACE ABROCA
US_rf_white <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
               protected_attr_col = "white", majority_protected_attr_val = "1",
               plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA plots",
identifier="US_rf white")
print(US_rf_white)
```

```
## [1] 0.03285044
```

Prep for US xgboost

```
train <- read.csv("undersampletrain.csv")
test <- read.csv("test.csv")
str(train)
str(test)

train = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )

y.train = train$dropout %>% unlist() %>% as.numeric()
y.test = test$dropout %>% unlist() %>% as.numeric()
x.train = model.matrix(dropout~., train)[,-1] #data should only be predictors
x.test = model.matrix(dropout~., test)[,-1]


# Data preparation
dtrain <- xgb.DMatrix(data = x.train, label = y.train)
dtest <- xgb.DMatrix(data = x.test, label = y.test)
ts_label <- test$dropout


# Initial parameter setup (if needed)
initial_params <- list(
  booster = "gbtree",
  objective = "binary:logistic",
  eval_metric = "logloss",
  eta = 0.3,
  max_depth = 6, gamma = 3
)

# Cross-validation to find optimal rounds of boosting
cv_results <- xgb.cv(
  params = initial_params,
  data = dtrain,
  nrounds = 100,
  nfold = 5,
  early_stopping_rounds = 20,
  verbose = 1
)

# Extract the Best Number of Rounds
best_nrounds <- cv_results$best_iteration
```

```
# Train the Final Model with Optimal Parameters
set.seed(2023)
final_model <- xgb.train(
  params = initial_params,
  data = dtrain,
  nrounds = best_nrounds
)

# Grid search for hyperparameter tuning
search_grid <- expand.grid(
  max_depth = c(3, 6),
  eta = c(0.01, 0.1),
  colsample_bytree = c(0.5, 0.7)
)

best_auc <- Inf  # Use Inf for minimization
best_params <- list()

for (i in 1:nrow(search_grid)) {
  params <- list(
    objective = "binary:logistic",
    eval_metric = "logloss",
    max_depth = search_grid$max_depth[i],
    eta = search_grid$eta[i],
    colsample_bytree = search_grid$colsample_bytree[i]
  )

  cv_results <- xgb.cv(
    params = params,
    data = dtrain,
    nfold = 5,
    nrounds = 100,
    early_stopping_rounds = 10,
    verbose = 1
  )

  mean_logloss <- min(cv_results$evaluation_log$test_logloss_mean)

  if (mean_logloss < best_auc) {
    best_auc <- mean_logloss
    best_params <- params
    best_nrounds <- cv_results$best_iteration
  }
}

# Train the final model with the best parameters
dtest <- xgb.DMatrix(data = x.test, label = y.test)
set.seed(2023)
```

```
xgb1 <- xgb.train (params = best_params, data = dtrain, watchlist = list(val=dtest,train=dtrain),
print_every_n = 10, nrounds = best_nrounds)
```

ABROCA US xgboost

```
# Train the final model with the best parameters
dtest <- xgb.DMatrix(data = x.test, label = y.test)
set.seed(2023)
xgb1 <- xgb.train (params = best_params, data = dtrain, watchlist = list(val=dtest,train=dtrain),
print_every_n = 10, nrounds = best_nrounds)
```

```
## [1]  val-logloss:0.646212    train-logloss:0.647242
## [11] val-logloss:0.460663    train-logloss:0.441257
## [21] val-logloss:0.411490    train-logloss:0.370018
## [31] val-logloss:0.392529    train-logloss:0.340938
## [41] val-logloss:0.376640    train-logloss:0.325796
## [51] val-logloss:0.371436    train-logloss:0.317605
## [61] val-logloss:0.367045    train-logloss:0.311442
## [71] val-logloss:0.365427    train-logloss:0.306871
## [81] val-logloss:0.364583    train-logloss:0.303276
## [91] val-logloss:0.363684    train-logloss:0.299723
## [92] val-logloss:0.363615    train-logloss:0.299458
```

```
#need to bring demographics back to test data
testdems <- read.csv("test.csv")
test <- testdems %>%
  mutate(across(c("female", "hispanic", "asian", "black", "white", "other_race", "eds", "lep_8",
"ever_lep", "swd_8", "ever_swd"), as.factor))

test$pred <- predict(xgb1, dtest, type = 'response')


#LOOP for attributes where "0" is the majority (eds, ell, swd)
# Define a helper function to run ABROCA and print the result
run_abroca <- function(protected_attr, identifier) {
  result <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
                           protected_attr_col = protected_attr, majority_protected_attr_val = "0",
                           plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA
plots",
                           identifier = identifier)
  print(result)
}

# Run ABROCA for different protected attributes
run_abroca("female", "xgb female")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.006529927
```

```
run_abroca("eds", "xgb eds")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.04823376
```

```
run_abroca("lep_8", "xgb ell")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.1000961
```

```
run_abroca("swd_8", "xgb swd")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## [1] 0.0619064
```

```
#RACE ABROCA
abroca_xgb_white <- compute_abroca(test, pred_col = "pred", label_col = "dropout",
              protected_attr_col = "white", majority_protected_attr_val = "1",
              plot_slices = TRUE, image_dir="D:/NCERDC_DATA/Alam/ML/Analysis/ABROCA plots",
identifier="xgb white")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
print(abroca_xgb_white)
```

```
## [1] 0.03826571
```

Kruskal wallis tests

```
gender = list(0.006, 0.005, 0.006, 0.021, 0.006, 0.004, 0.004, 0.004, 0.017, 0.006, 0.003, 0.005, 0.007,
0.017, 0.006)
kruskal.test(gender)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  gender
## Kruskal-Wallis chi-squared = 14, df = 14, p-value = 0.4497
```

```
ell = list(0.090, 0.087, 0.089, 0.077, 0.095, 0.082, 0.080, 0.080, 0.090, 0.100, 0.086, 0.092, 0.101, 0.090,
0.10)
kruskal.test(ell)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  ell
## Kruskal-Wallis chi-squared = 14, df = 14, p-value = 0.4497
```

```
disability = list(0.058, 0.06, 0.058, 0.041, 0.06, 0.059, 0.061, 0.061, 0.043, 0.063, 0.056, 0.056, 0.046,
0.043, 0.063)
kruskal.test(disability)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  disability
## Kruskal-Wallis chi-squared = 14, df = 14, p-value = 0.4497
```

```
econdis = list(0.045, 0.045, 0.045, 0.034, 0.048, 0.042, 0.042, 0.043, 0.038, 0.032, 0.042, 0.043, 0.043,
0.038, 0.032)
kruskal.test(econdis)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  econdis
## Kruskal-Wallis chi-squared = 14, df = 14, p-value = 0.4497
```

```
econdis = list(0.045, 0.045, 0.045, 0.034, 0.048, 0.042, 0.042, 0.043, 0.038, 0.032, 0.042, 0.043, 0.043,
0.038, 0.032)
kruskal.test(econdis)
```

```
##
##  Kruskal-Wallis rank sum test
##
```

```
## data:  econdis
## Kruskal-Wallis chi-squared = 14, df = 14, p-value = 0.4497
```

```
race = list(0.033, 0.031, 0.032, 0.018, 0.034, 0.029, 0.023, 0.029, 0.027, 0.035, 0.030, 0.04, 0.041, 0.027,
0.035)
kruskal.test(race)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  race
## Kruskal-Wallis chi-squared = 14, df = 14, p-value = 0.4497
```

```
total = list(gender, ell, disability, econdis, race)
kruskal.test(total)
```

```
## Warning in kruskal.test.default(total): some elements of 'x' are not numeric
## and will be coerced to numeric
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  total
## Kruskal-Wallis chi-squared = 68.385, df = 4, p-value = 4.975e-14
```

Equalized opportunity for US XGB

```
xgbpred <- predict (xgb1,dtest)
xgbpred <- ifelse (xgbpred >= 0.4,"1", "0")


test <- read.csv("test.csv")
test <- test %>%
  mutate(across(c("female", "hispanic", "asian", "black", "white", "other_race", "eds", "lep_8",
"ever_lep", "swd_8", "ever_swd"), as.factor))

test$pred <- predict(xgb1, dtest, type = 'response')



#devtools::install_github('kozodoi/fairness')
library(fairness)


## function for a cutoff of 0.4 for all but race
get_metric <- function(group_name) {
  result <- equal_odds(data         = test,
                       outcome      = 'dropout',
                       outcome_base = '0',
                       group        = group_name,
                     probs       = 'pred',
                     cutoff      = 0.4,
                     base        = '0')
  return(result$Metric)
}

female_eq_metric <- get_metric('female')
ell_eq_metric <- get_metric('lep_8')
swd_eq_metric <- get_metric('swd_8')
eds_eq_metric <- get_metric('eds')
## function for race at 0.4 cutoff
get_metric <- function(group_name) {
  result <- equal_odds(data        = test,
                     outcome      = 'dropout',
                     outcome_base = '0',
                     group        = group_name,
                     probs        = 'pred',
                     cutoff       = 0.4,
                     base         = '1')
  return(result$Metric)
}

white_eq_metric <- get_metric('white')

# Results
white_eq_metric
```

```
##                           1            0
## Sensitivity    8.551089e-01 8.752066e-01
## Equalized odds 1.000000e+00 1.023503e+00
## Group size     4.976300e+04 4.531400e+04
```

```
female_eq_metric
```

```
##                           0            1
## Sensitivity    8.896277e-01 8.244444e-01
## Equalized odds 1.000000e+00 9.267298e-01
## Group size     4.818000e+04 4.689700e+04
```

```
ell_eq_metric
```

```
##                           0            1
## Sensitivity    8.703103e-01    0.8204082
## Equalized odds 1.000000e+00    0.9426616
## Group size     9.097900e+04 4098.0000000
```

```
swd_eq_metric
```

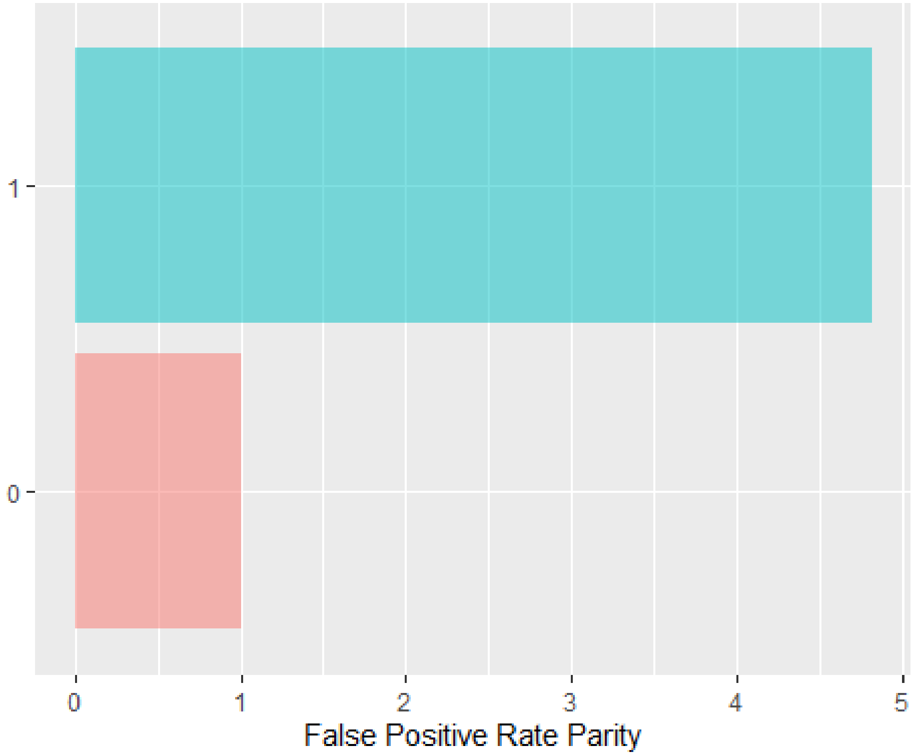
```
##                           0            1
## Sensitivity    8.387471e-01 9.323529e-01
## Equalized odds 1.000000e+00 1.111602e+00
## Group size     8.337800e+04 1.169900e+04
```

```
eds_eq_metric
```

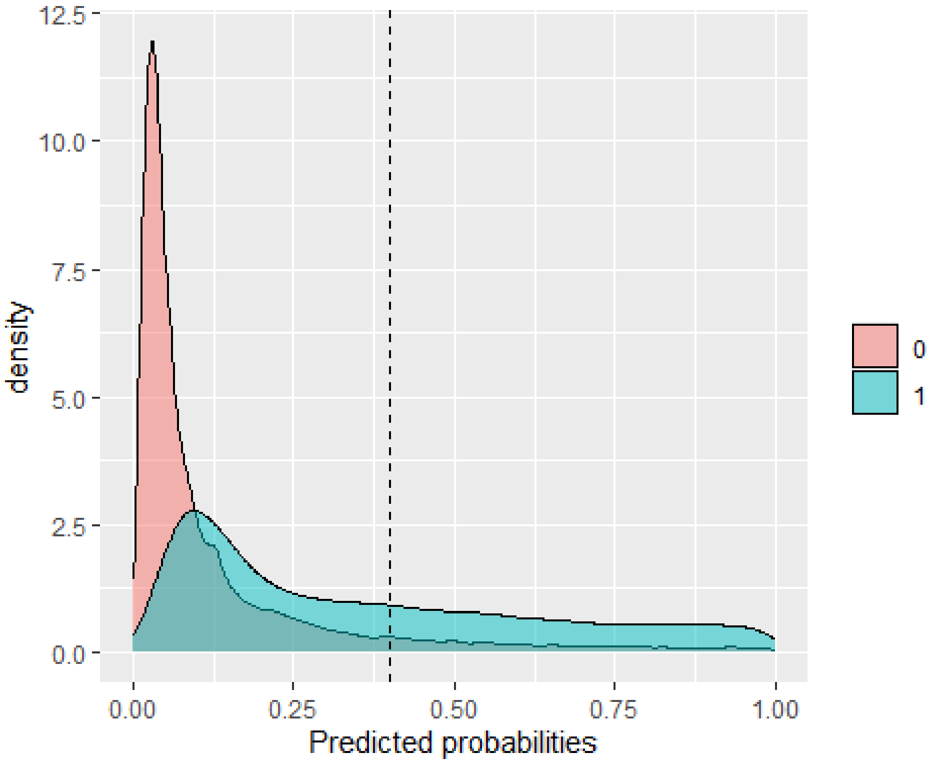
```
##                          0            1
## Sensitivity    7.177419e-01 9.035639e-01
## Equalized odds 1.000000e+00 1.258898e+00
## Group size     4.967400e+04 4.540300e+04
```

```
#FPR
groups <- c('eds', 'female', 'lep_8', 'swd_8')
for (group in groups) {
  fpr <- fpr_parity(data = test, outcome = 'dropout', group = group,
                      probs = 'pred', cutoff = 0.4, base = '0')
  print(fpr)
}
```

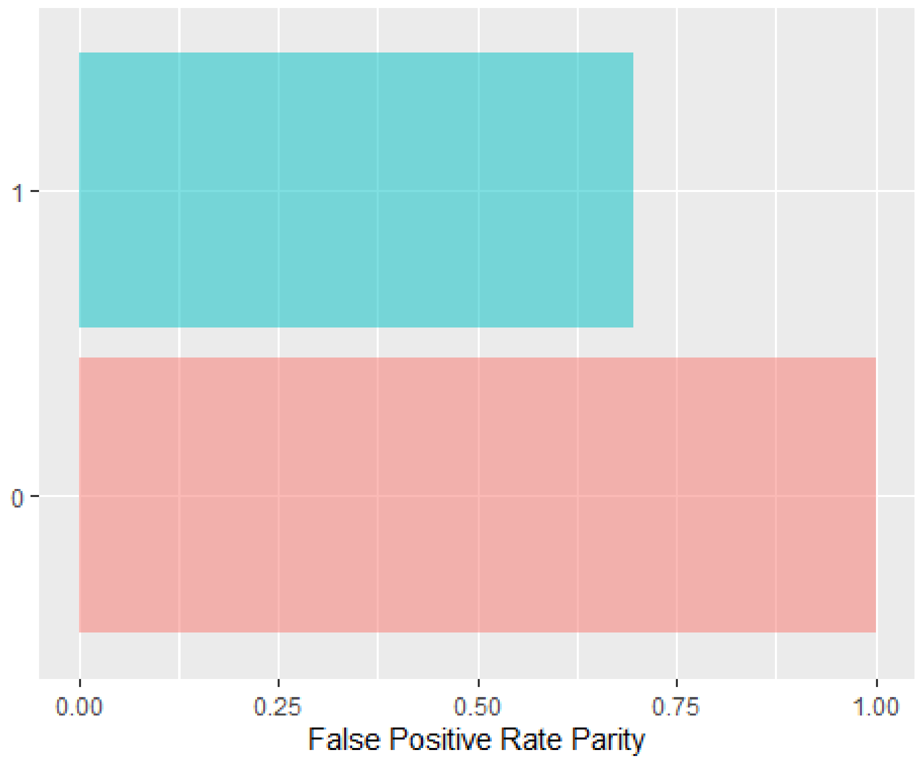
```
## $Metric
##                     0            1
## FPR        7.503355e-02 3.613059e-01
## FPR Parity 1.000000e+00 4.815258e+00
## Group size 4.967400e+04 4.540300e+04
##
## $Metric_plot
```



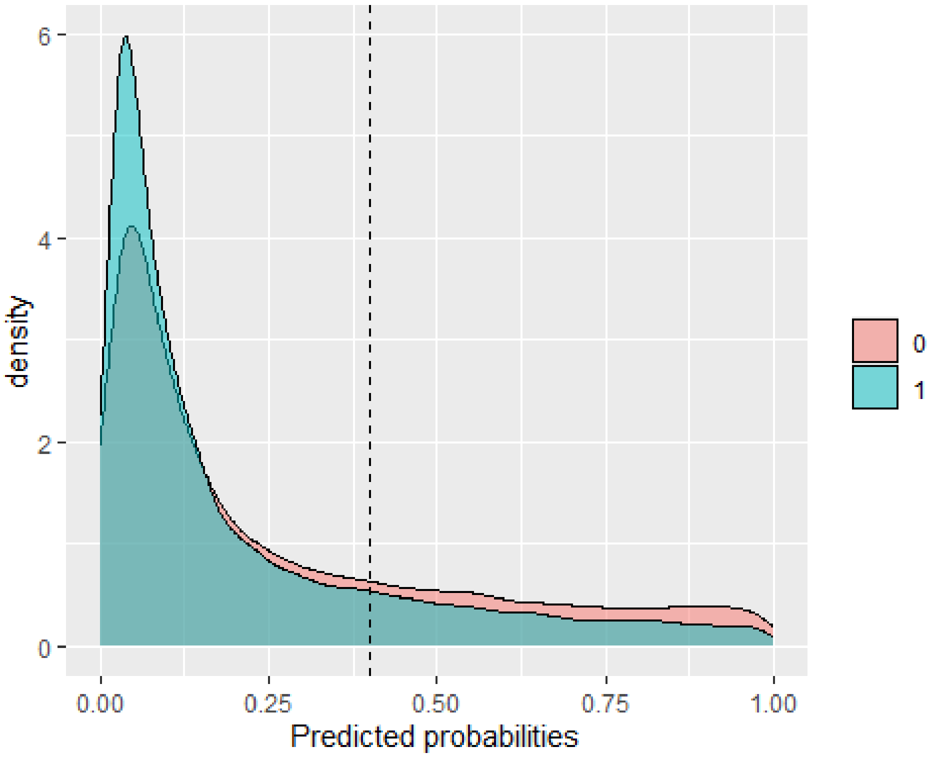
```
##
## $Probability_plot
```



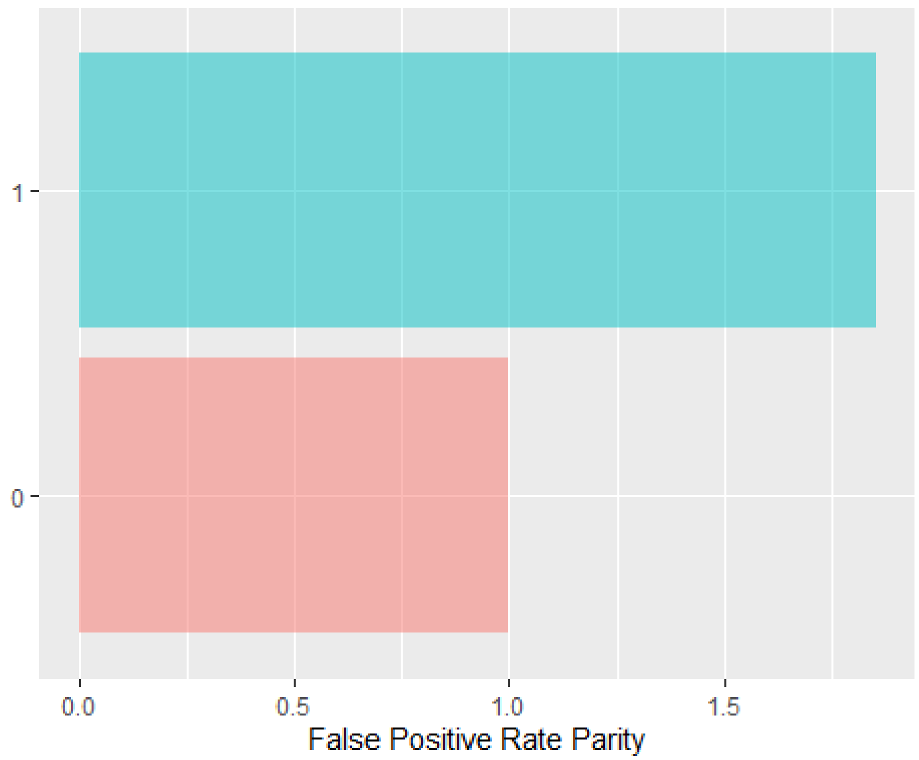
```
##
## $Metric
##                     0            1
## FPR        2.464436e-01 1.717938e-01
## FPR Parity 1.000000e+00 6.970919e-01
## Group size 4.818000e+04 4.689700e+04
##
## $Metric_plot
```



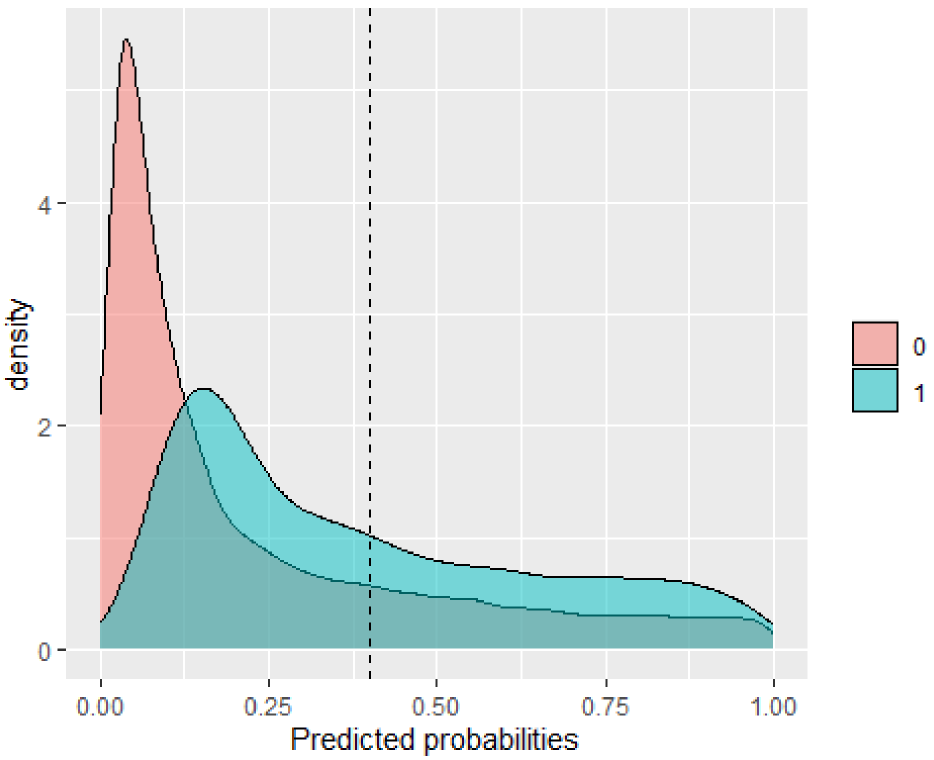
```
##
## $Probability_plot
```



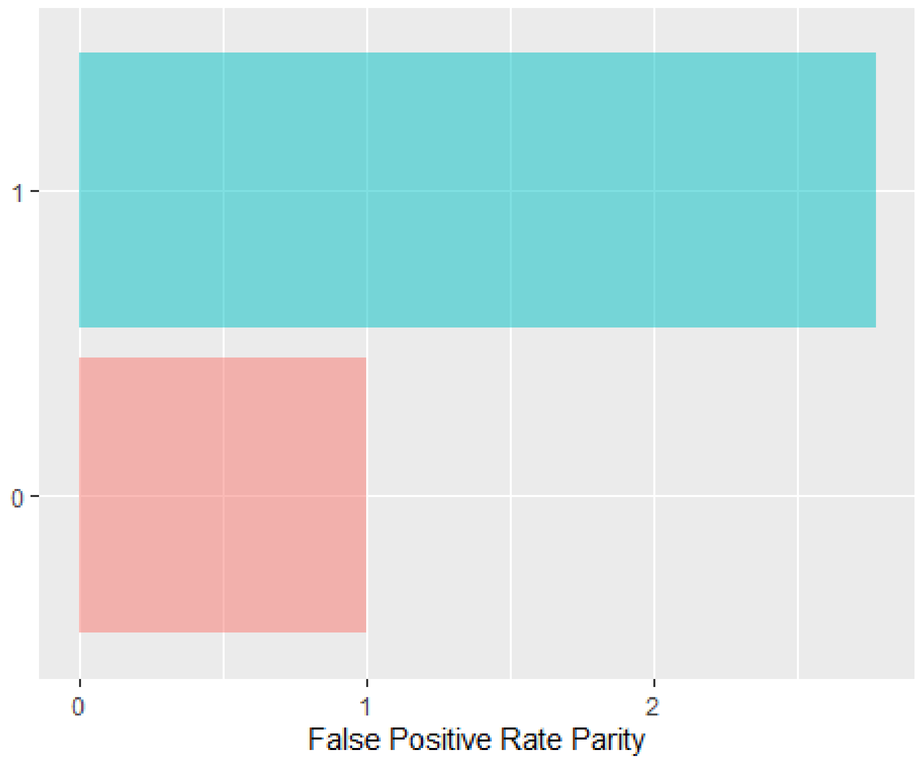
```
##
## $Metric
##                    0            1
## FPR        2.022292e-01    0.3745134
## FPR Parity 1.000000e+00    1.8519250
## Group size 9.097900e+04 4098.0000000
##
## $Metric_plot
```



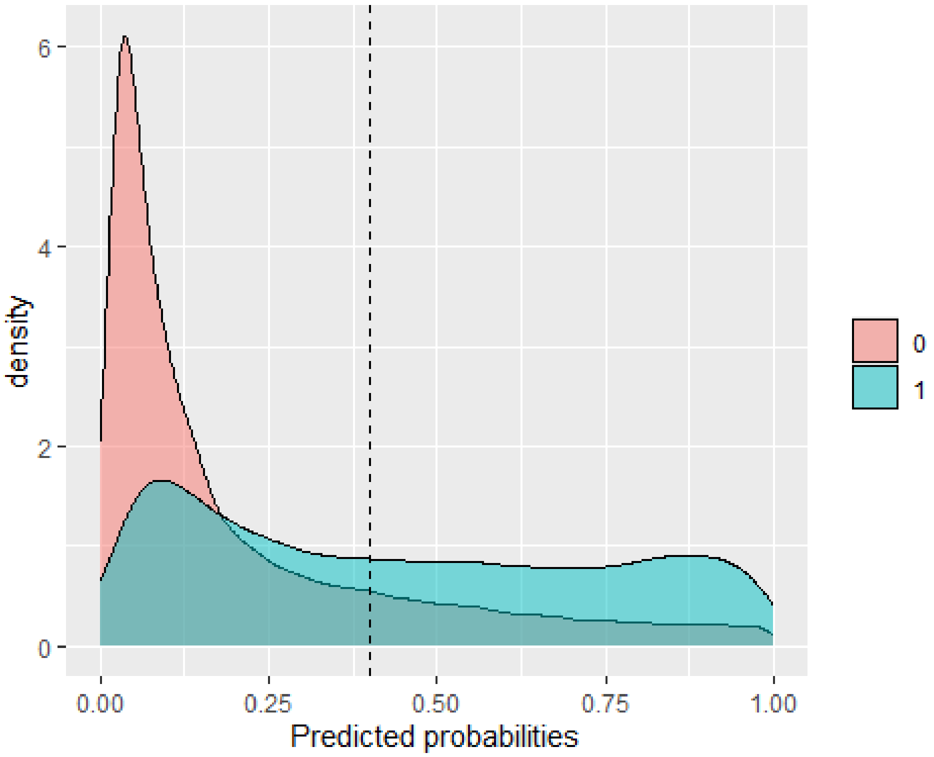
```
##
## $Probability_plot
```



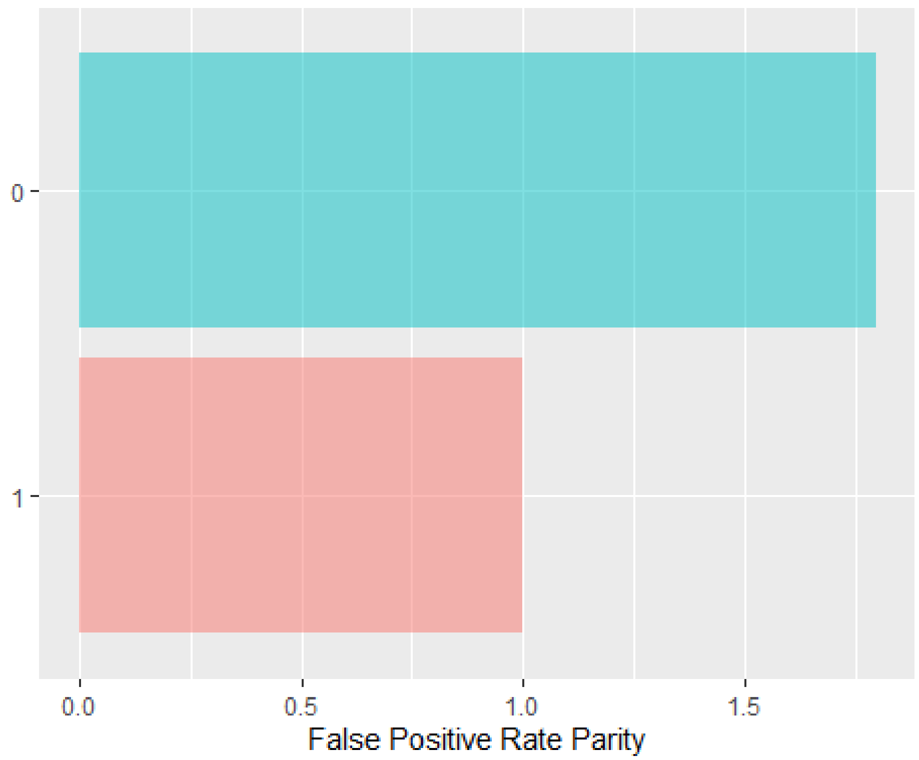
```
##
## $Metric
##                       0            1
## FPR        1.729493e-01 4.794446e-01
## FPR Parity 1.000000e+00 2.772169e+00
## Group size 8.337800e+04 1.169900e+04
##
## $Metric_plot
```



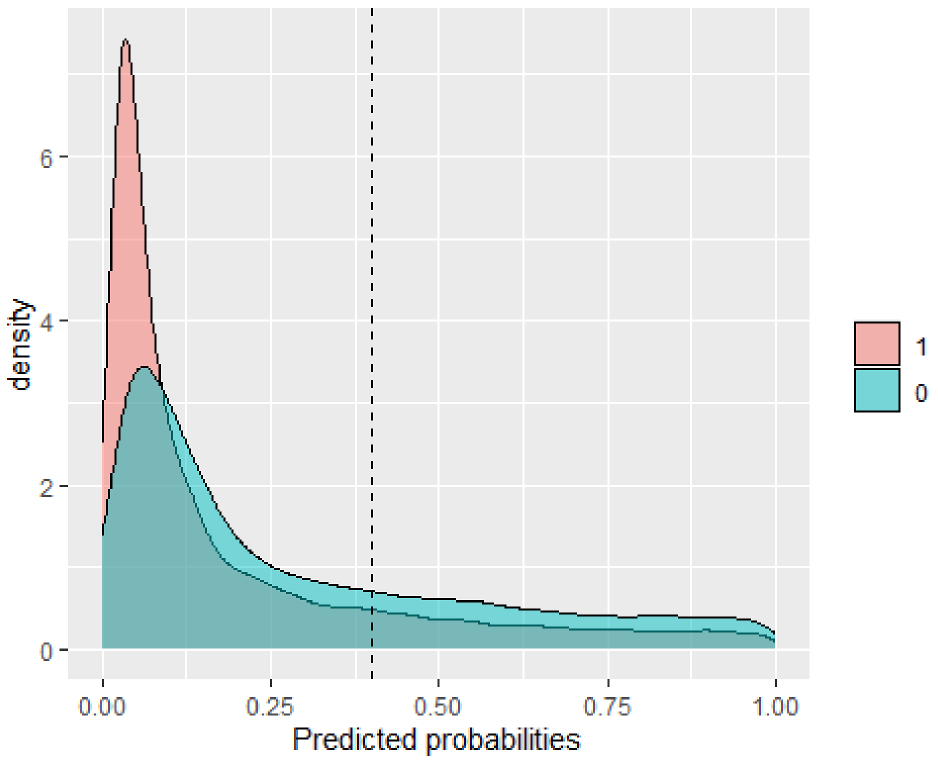
```
##
## $Probability_plot
```



```
fpr_white <- fpr_parity(data = test, outcome = 'dropout', group = 'white',
probs = 'pred', cutoff = 0.4, base = '1')
fpr_white

## $Metric
##                    1           0
## FPR        1.520105e-01     0.272583
## FPR Parity 1.000000e+00     1.793185
## Group size 4.976300e+04 45314.000000
##
## $Metric_plot
```



```
##
## $Probability_plot
```



Equalized opportunity for US LGR

```
train <- read.csv("D:/NCERDC_DATA/Alam/ML/undersampletrain.csv")
test <- read.csv("test.csv")

log1.m <- glm(dropout ~ ., data = subset(train, select = -c(female, hispanic, asian, black, white,other_race
, eds, lep_8, ever_lep, swd_8, ever_swd)), family='binomial')

predict_log <- predict(log1.m, test[,-1], type = 'response')

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

# Create a prediction object
pred <- prediction(predict_log, test$dropout)
test$pred = predict(log1.m, test, type = "response")

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

library(fairness)
## function for a cutoff of 0.1 for all but race
get_metric <- function(group_name) {
  result <- equal_odds(data         = test,
                       outcome      = 'dropout',
                       outcome_base = '0',
                   group        = group_name,
                   probs        = 'pred',
                   cutoff       = 0.1,
                   base         = '0')
  return(result$Metric)
}

female_eq_metric <- get_metric('female')
ell_eq_metric <- get_metric('lep_8')
swd_eq_metric <- get_metric('swd_8')
eds_eq_metric <- get_metric('eds')
## function for race at 0.1 cutoff
get_metric <- function(group_name) {
  result <- equal_odds(data         = test,
                   outcome      = 'dropout',
                   outcome_base = '0',
                   group        = group_name,
                   probs        = 'pred',
                   cutoff       = 0.41,
                   base         = '1')
  return(result$Metric)
}

white_eq_metric <- get_metric('white')

# Results
white_eq_metric

##                          1            0
## Sensitivity    6.616415e-01 6.900826e-01
## Equalized odds 1.000000e+00 1.042986e+00
## Group size     4.976300e+04 4.531400e+04

female_eq_metric

##                          0            1
## Sensitivity    9.335106e-01 8.888889e-01
## Equalized odds 1.000000e+00 9.522001e-01
## Group size     4.818000e+04 4.689700e+04

ell_eq_metric

##                          0            1
## Sensitivity    9.198703e-01    0.8897959
## Equalized odds 1.000000e+00    0.9673058
## Group size     9.097900e+04 4098.0000000

swd_eq_metric
```

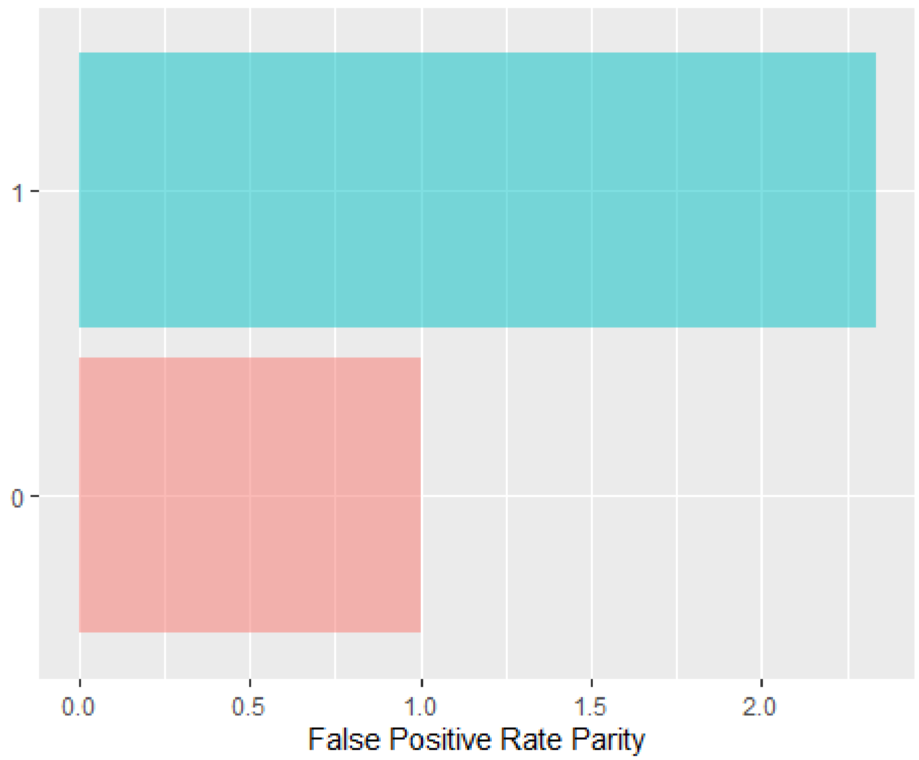
```
##                          0            1
## Sensitivity    8.944316e-01 9.735294e-01
## Equalized odds 1.000000e+00 1.088434e+00
## Group size     8.337800e+04 1.169900e+04
```

```
eds_eq_metric
```

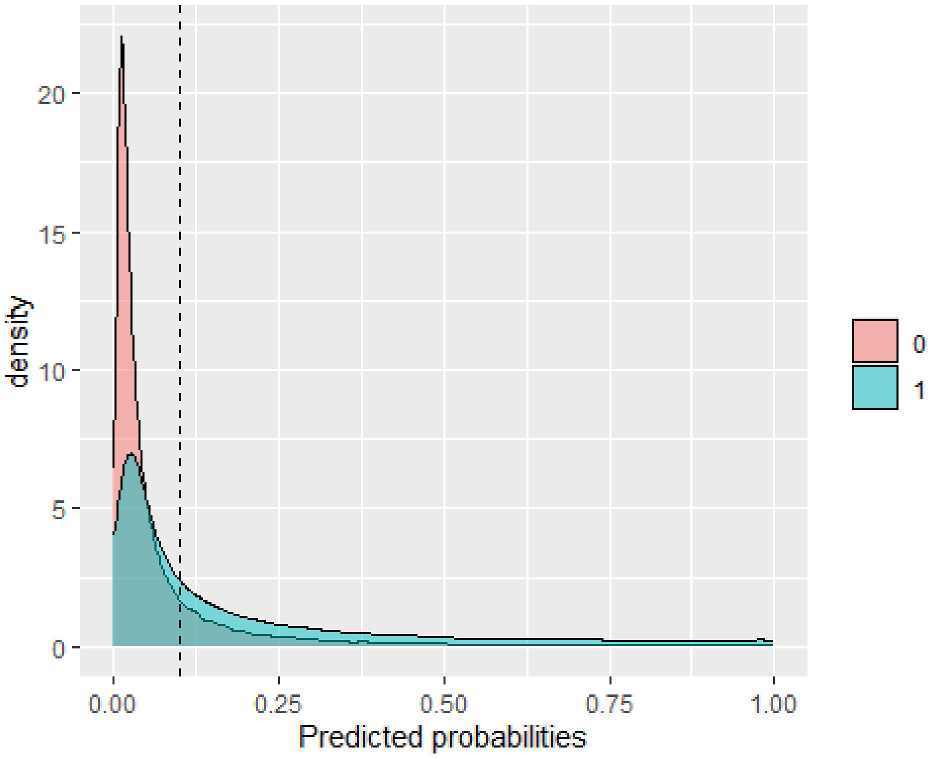
```
##                          0            1
## Sensitivity    8.427419e-01 9.360587e-01
## Equalized odds 1.000000e+00 1.110730e+00
## Group size     4.967400e+04 4.540300e+04
```

```
#FPR
groups <- c('eds', 'female', 'lep_8', 'swd_8')
for (group in groups) {
  fpr <- fpr_parity(data = test, outcome = 'dropout', group = group,
                    probs = 'pred', cutoff = 0.1, base = '0')
  print(fpr)
}
```

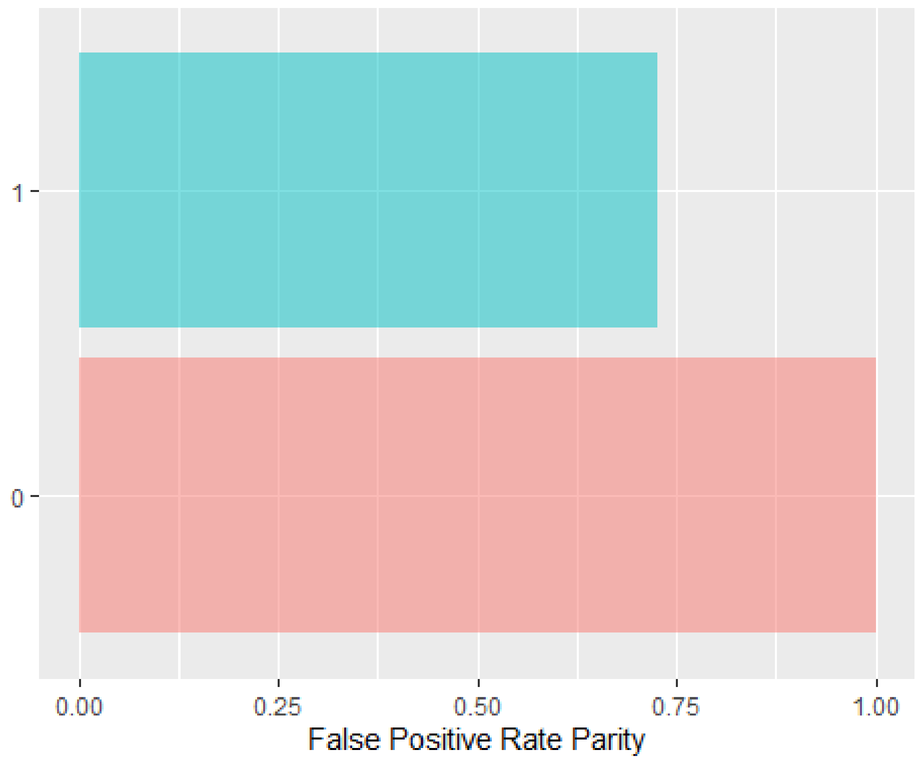
```
## $Metric
##                     0            1
## FPR        1.864858e-01 4.350385e-01
## FPR Parity 1.000000e+00 2.332823e+00
## Group size 4.967400e+04 4.540300e+04
##
## $Metric_plot
```



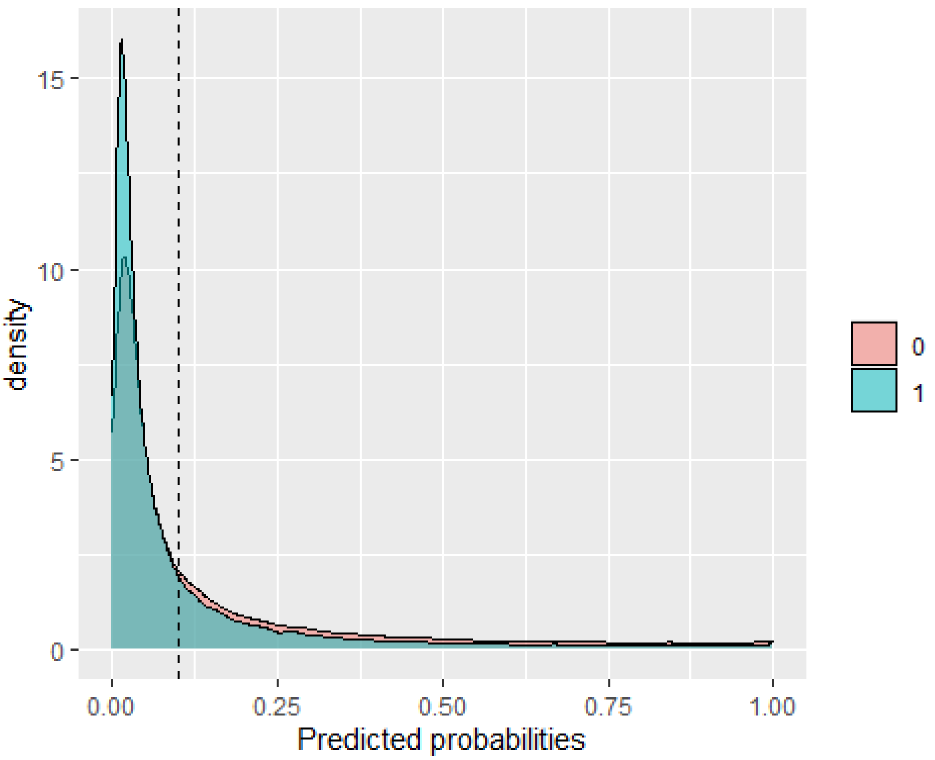
```
##
## $Probability_plot
```



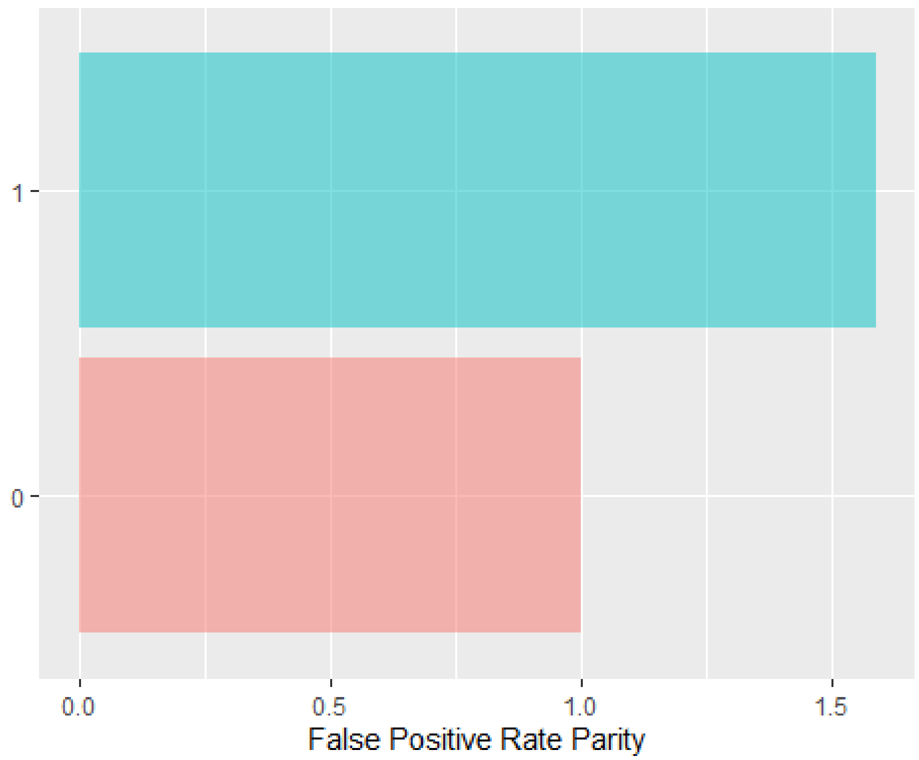
```
##
## $Metric
##                      0            1
## FPR        3.50737e-01 2.548427e-01
## FPR Parity 1.00000e+00 7.265920e-01
## Group size 4.81800e+04 4.689700e+04
##
## $Metric_plot
```



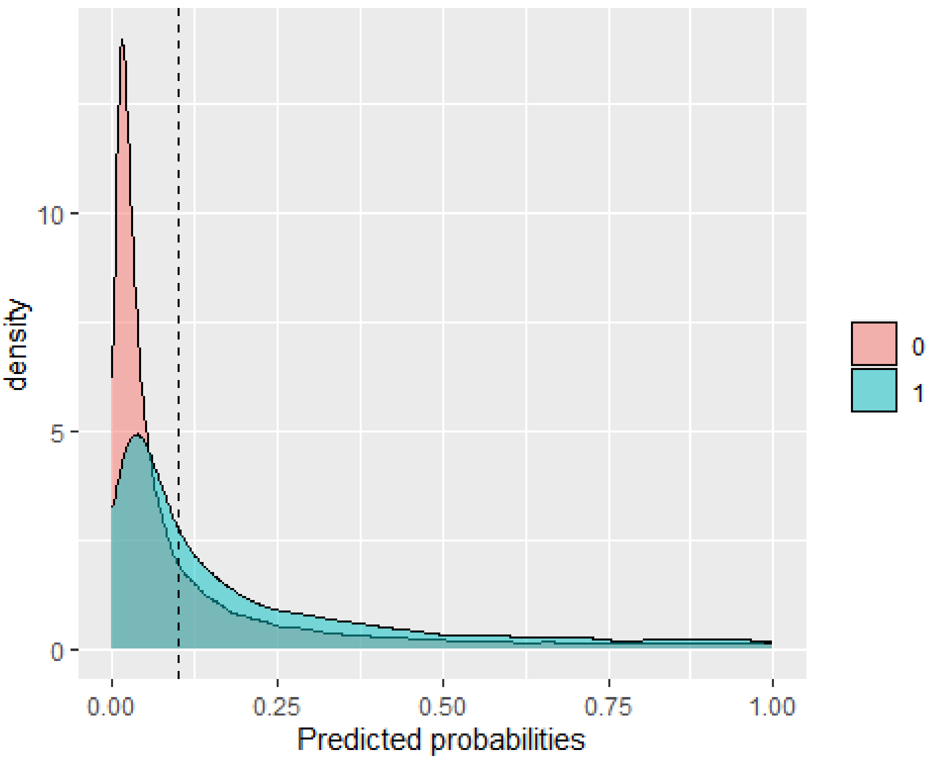
```
##
## $Probability_plot
```



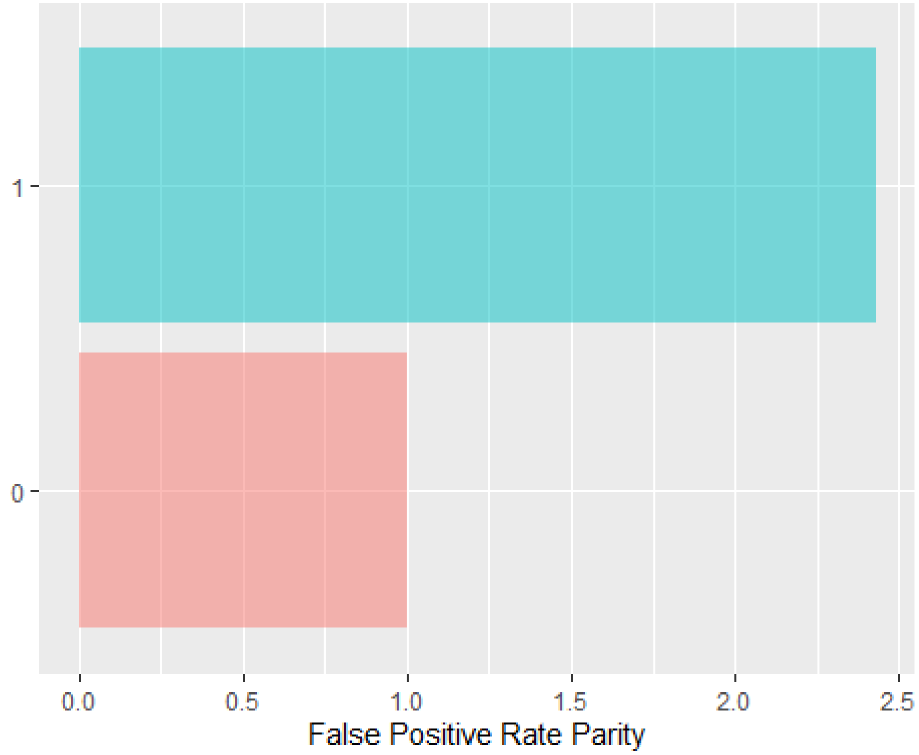
```
##
## $Metric
##                          0              1
## FPR          2.959356e-01    0.4692447
## FPR Parity 1.000000e+00    1.5856313
## Group size 9.097900e+04 4098.0000000
##
## $Metric_plot
```



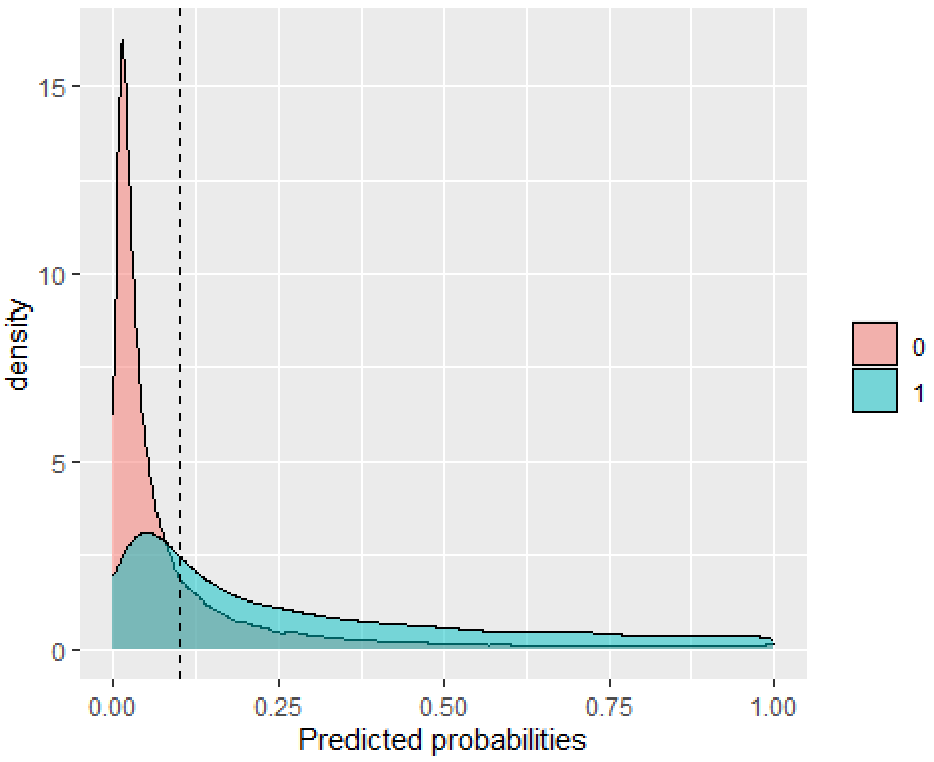
```
##
## $Probability_plot
```



```
##
## $Metric
##                    0            1
## FPR        2.591545e-01 6.290952e-01
## FPR Parity 1.000000e+00 2.427491e+00
## Group size 8.337800e+04 1.169900e+04
##
## $Metric_plot
```

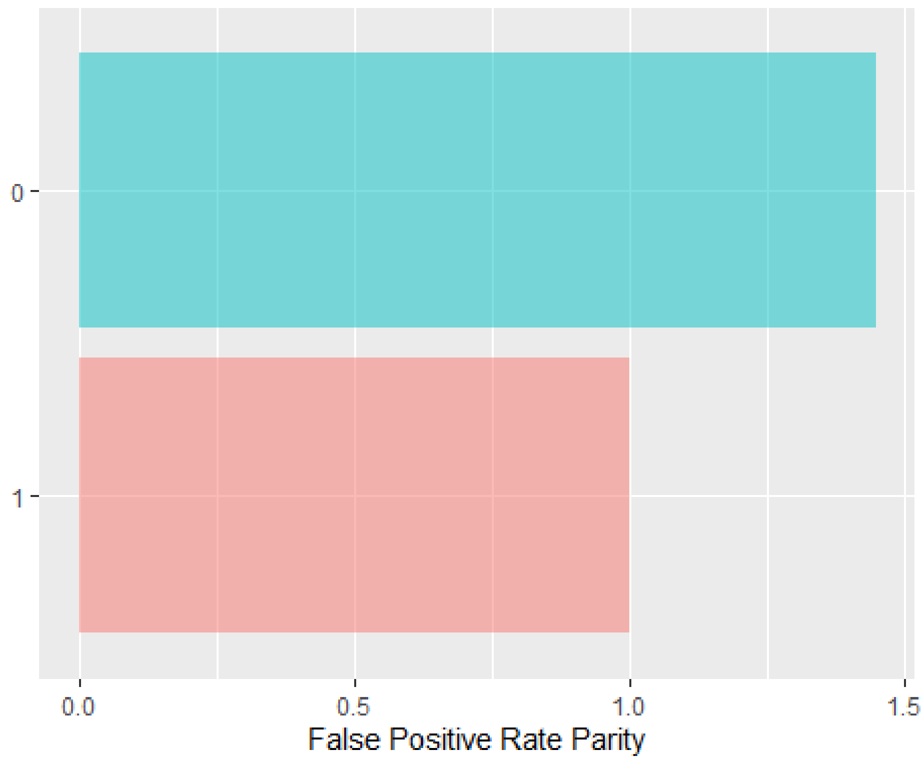


```
##
## $Probability_plot
```



```
fpr_white <- fpr_parity(data = test, outcome = 'dropout', group = 'white',
probs = 'pred', cutoff = 0.1, base = '1')
fpr_white
```

```
## $Metric
##                     1            0
## FPR         2.499743e-01 3.616905e-01
## FPR Parity 1.000000e+00 1.446911e+00
## Group size 4.976300e+04 4.531400e+04
##
## $Metric_plot
```



```
##
## $Probability_plot
```

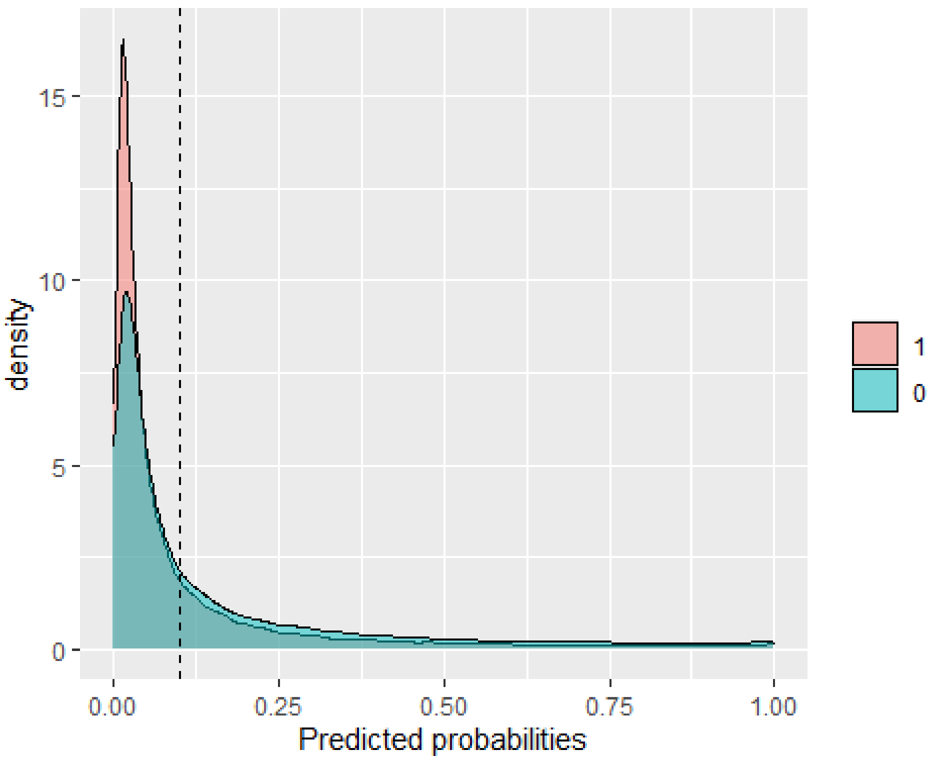

15
10
5
0
density
0.00
0.25
0.50
0.75
1.00
Predicted probabilities
1
0

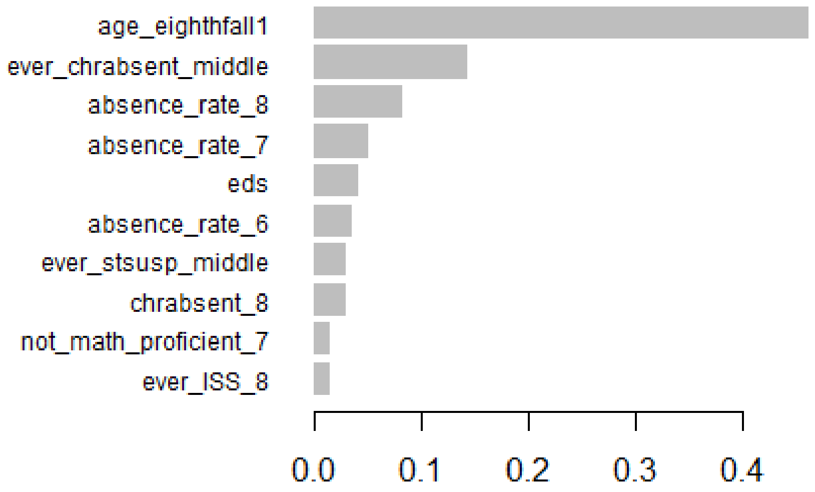
Figure B3: Code for research question 3

Preparing and running the undersampled XGboost model

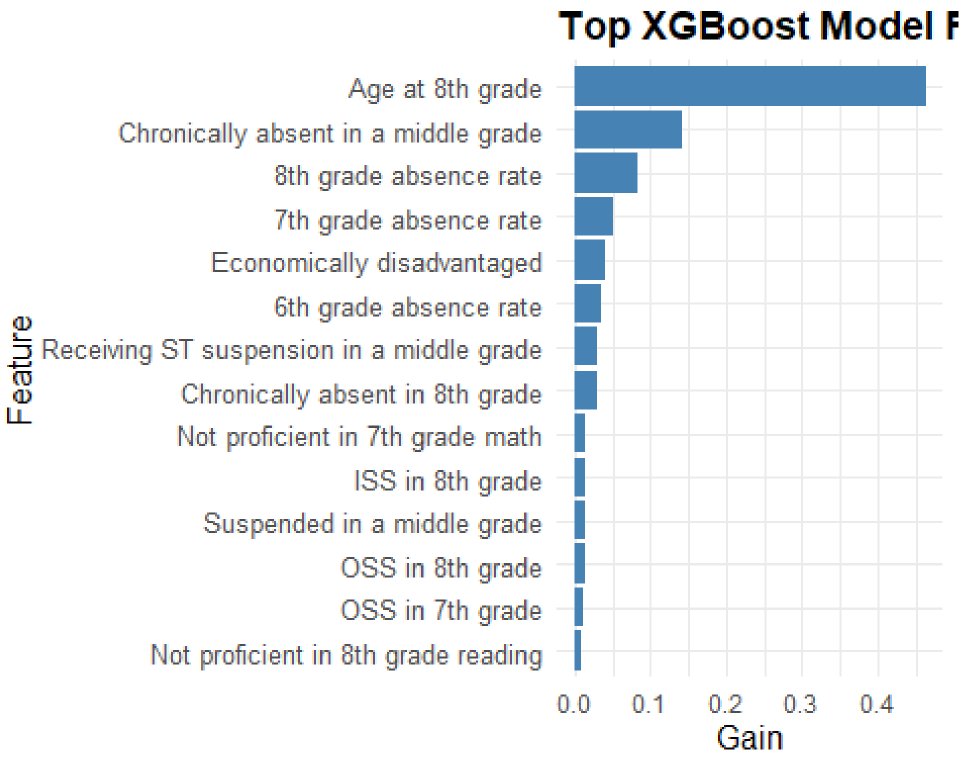
```
train <- read.csv("D:/NCERDC_DATA/Alam/ML/undersampletrain.csv")
test <- read.csv("test.csv")

train = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )

y.train = train$dropout %>% unlist() %>% as.numeric()
y.test = test$dropout %>% unlist() %>% as.numeric()
x.train = model.matrix(dropout~., train)[,-1] #data should only be predictors
x.test = model.matrix(dropout~., test)[,-1]

# Data preparation
dtrain <- xgb.DMatrix(data = x.train, label = y.train)
dtest <- xgb.DMatrix(data = x.test, label = y.test)
ts_label <- test$dropout


# Initial parameter setup (if needed)
initial_params <- list(
  booster = "gbtree",
  objective = "binary:logistic",
  eval_metric = "logloss",
  eta = 0.3,
  max_depth = 6, gamma = 3
)

# Cross-validation to find optimal rounds of boosting
cv_results <- xgb.cv(
  params = initial_params,
  data = dtrain,
  nrounds = 100,
  nfold = 5,
  early_stopping_rounds = 20,
  verbose = 1
)

# Extract the Best Number of Rounds
best_nrounds <- cv_results$best_iteration

# Train the Final Model with Optimal Parameters
set.seed(2023)
final_model <- xgb.train(
  params = initial_params,
  data = dtrain,
  nrounds = best_nrounds
)

# Grid search for hyperparameter tuning
search_grid <- expand.grid(
  max_depth = c(3, 6),
  eta = c(0.01, 0.1),
  colsample_bytree = c(0.5, 0.7)
)

best_auc <- Inf  # Use Inf for minimization
best_params <- list()

for (i in 1:nrow(search_grid)) {
  params <- list(
    objective = "binary:logistic",
    eval_metric = "logloss",
    max_depth = search_grid$max_depth[i],
    eta = search_grid$eta[i],
    colsample_bytree = search_grid$colsample_bytree[i]
  )

  cv_results <- xgb.cv(
    params = params,
    data = dtrain,
    nfold = 5,
    nrounds = 100,
    early_stopping_rounds = 10,
    verbose = 1
  )

  mean_logloss <- min(cv_results$evaluation_log$test_logloss_mean)

  if (mean_logloss < best_auc) {
    best_auc <- mean_logloss
    best_params <- params
    best_nrounds <- cv_results$best_iteration
  }
}

# Train the final model with the best parameters
set.seed(2023)
xgb1 <- xgb.train (params = best_params, data = dtrain, watchlist = list(val=dtest,train=dtrain), print_every_n = 10, nrounds = best_nrounds)
```

Interpretting the undersampled XGboost model

```
mat <- xgb.importance (feature_names = colnames(x.train),model = xgb1)
xgb.plot.importance (importance_matrix = mat[1:10])
```



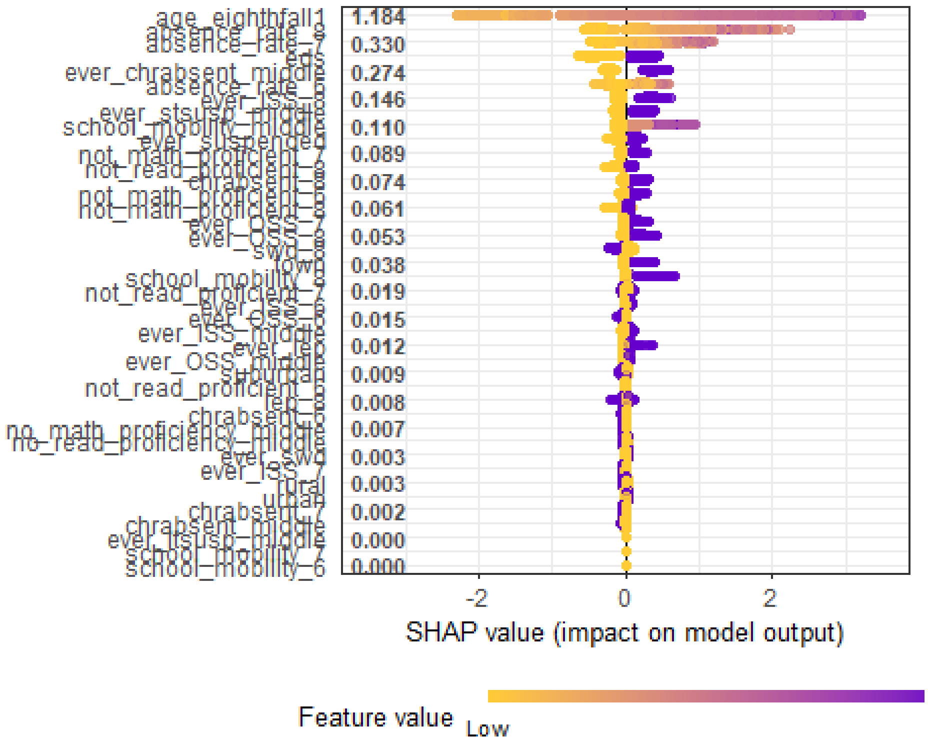
```
# Convert the importance matrix to a data frame for ggplot2
importance_df <- as.data.frame(mat)

# Define a mapping of old feature names to new feature names
name_mapping <- c(
 "age_eighthfall1" = "Age at 8th grade",
 "ever_chrabsent_middle" = "Chronically absent in a middle grade",
  "chrabsent_8" = "Chronically absent in 8th grade",
 "ever_ISS_8" = "ISS in 8th grade",
 "ever_OSS_8" = "OSS in 8th grade",
 "ever_OSS_7" = "OSS in 7th grade",
 "school_mobility_middle" = "School mobility in middle grades",
 "not_read_proficient_7" = "Not proficient in 7th grade reading",
  "not_math_proficient_6" = "Not proficient in 6th grade math",
   "not_math_proficient_8" = "Not proficient in 8th grade math",
 "ever_ISS_middle" = "ISS in 8th grade",
 "town" = "School is classified as town",
 "absence_rate_8" = "8th grade absence rate",
 "absence_rate_7" = "7th grade absence rate",
  "absence_rate_6" = "6th grade absence rate",
 "eds" = "Economically disadvantaged",
 "ever_stsusp_middle" = "Receiving ST suspension in a middle grade",
 "not_math_proficient_8" = "Not proficient in 8th grade math",
 "ever_suspended" = "Suspended in a middle grade",
 "not_read_proficient_8" = "Not proficient in 8th grade reading",
  "not_math_proficient_7" = "Not proficient in 7th grade math"
)

# Replace old feature names with new feature names
importance_df$Feature <- ifelse(importance_df$Feature %in% names(name_mapping),
                                name_mapping[importance_df$Feature],
                                importance_df$Feature)

#extrafont::loadfonts(device="win")

base_fig <- ggplot(importance_df[1:14, ], aes(x = reorder(Feature, Gain), y = Gain)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(title = "Top XGBoost Model Features by Gain",
       x = "Feature",
       y = "Gain") +
  theme_minimal() +
  theme(axis.text.y = element_text(size = 10),
        axis.title = element_text(size = 12),
        plot.title = element_text(size = 14, face = "bold"),
        legend.position = "none", windowsFonts(Times=windowsFont("TT Times New Roman"))

)
base_fig +
  theme(text = element_text(family = "Times New Roman"))
```



```
library(iml)
predictor <- Predictor$new(model = xgb1, data = train, y = train$dropout)

# Compute Shapley values for a single instance (e.g., the first row of X)

library(caret)
library(kernelshap)
library(shapviz)


#devtools::install_github("liuyanguu/SHAPforxgboost")
library("SHAPforxgboost")


# To return the SHAP values and ranked features by mean|SHAP|
shap_values <- shap.values(xgb_model = xgb1, x.train)

# The ranked features by mean |SHAP|
shap_values$mean_shap_score
```

```
##        age_eighthfall1         absence_rate_8
##            1.183715129            0.430753728
##         absence_rate_7                    eds
##            0.330072428            0.295450338
##   ever_chrabsent_middle         absence_rate_6
##            0.274359854            0.208614580
##              ever_ISS_8    ever_stsusp_middle
##            0.145924016            0.134095743
##  school_mobility_middle         ever_suspended
##            0.110083195            0.104912609
##   not_math_proficient_7  not_read_proficient_8
##            0.088865684            0.080230655
##             chrabsent_8  not_math_proficient_6
##            0.074105589            0.069114520
##   not_math_proficient_8              ever_OSS_7
##            0.060925299            0.053809065
##              ever_OSS_8                  swd_8
##            0.053234336            0.043047057
##                    town      school_mobility_8
##            0.038472891            0.020358813
##   not_read_proficient_7              ever_ISS_6
##            0.018984973            0.017594534
##              ever_OSS_6        ever_ISS_middle
##            0.015166483            0.013973461
##                ever_lep        ever_OSS_middle
##            0.012489851            0.011727669
##                suburban  not_read_proficient_6
##            0.009328411            0.008317310
##                   lep_8            chrabsent_6
##               0.007829356                0.007380275
## no_math_proficiency_middle no_read_proficiency_middle
##               0.007072286                0.004742620
##                  ever_swd                 ever_ISS_7
##               0.003348294                0.003040845
##                     rural                      urban
##               0.002562293                0.001954877
##               chrabsent_7           chrabsent_middle
##               0.001918539                0.001610911
##        ever_ltsusp_middle          school_mobility_7
##               0.000000000                0.000000000
##         school_mobility_6
##               0.000000000
```

```
shap_long <- shap.prep(xgb_model = xgb1, X_train = x.train)
shapplot <- shap.plot.summary(shap_long)
shapplot
```



Preparing for Lasso and Ridge regressions

```
train <- read.csv("D:/NCERDC_DATA/Alam/ML/undersampletrain.csv")
test <- read.csv("test.csv")
str(train)
str(test)
```

```
train = subset(train, select = -c(female, hispanic, asian, black, white, other_race) )
test = subset(test, select = -c(female, hispanic, asian, black, white, other_race) )

y.train = train$dropout %>% unlist() %>% as.numeric()
x.train = model.matrix(dropout~., train)[,-1] #data should only be predictors

dim(x.train)
dim(x.test)


write.csv(x.train,'x.train.csv', row.names=FALSE)
write.csv(y.train,'y.train.csv', row.names=FALSE)
```

Running the undersampled lasso and ridge regressions

```
## LASSO
set.seed(2023)
cv.lasso <- cv.glmnet(x.train, y.train, alpha = 1, family='binomial') # Fit lasso regression model on training data

lasso.coefs <- coef(cv.lasso, s = "lambda.min")  # or use lambda.1se for a more regularized solution

# To view the coefficients in a more readable format (as a dataframe):
lasso.coefs_df <- as.data.frame(as.matrix(lasso.coefs))
lasso.coefs_df <- lasso.coefs_df %>%
  arrange(desc(s1))
print(lasso.coefs_df)
```

```
##                                 s1
## absence_rate_8         9.367399688
## absence_rate_7         6.471641131
## absence_rate_6         3.150566658
## age_eighthfall1        1.919511878
## school_mobility_8      0.762946853
## eds                    0.711248725
## ever_lep               0.699603902
## not_math_proficient_7  0.552257508
## ever_ISS_8             0.531284949
## not_math_proficient_6  0.509319947
## ever_chrabsent_middle  0.446116657
## ever_stsusp_middle     0.397142100
## school_mobility_middle 0.363591181
## not_math_proficient_8  0.305526867
## town                   0.270230335
## ever_ISS_6             0.245130525
## not_read_proficient_8  0.222794540
## ever_OSS_7             0.218125213
## ever_OSS_8             0.115798068
## school_mobility_6      0.095530800
## ever_ltsusp_middle          0.041073734
## ever_suspended              0.031405622
## urban                       0.005775141
## ever_OSS_middle             0.000000000
## ever_ISS_middle             0.000000000
## not_read_proficient_6       0.000000000
## no_read_proficiency_middle  0.000000000
## ever_swd                    0.000000000
## chrabsent_8                 0.000000000
## chrabsent_middle            0.000000000
## school_mobility_7           0.000000000
## rural                       0.000000000
## ever_ISS_7                 -0.008853850
## suburban                   -0.090386220
## not_read_proficient_7      -0.101199956
## swd_8                      -0.196432972
## chrabsent_6                -0.197540613
## ever_OSS_6                 -0.240537592
## chrabsent_7                -0.394023042
## no_math_proficiency_middle -0.687711542
## lep_8                      -0.692154631
## (Intercept)               -30.906457591
```

```
write.csv(lasso.coefs_df, "lasso.smote.coefs.csv", row.names = TRUE)

## RIDGE
set.seed(2023)
cv.ridge <- cv.glmnet(x.train, y.train, alpha = 0, family='binomial') # Fit ridge regression model on training data

# Extract the coefficients at the best lambda (lambda.min or lambda.1se)
ridge.coefs <- coef(cv.ridge, s = "lambda.min")  # or use lambda.1se for a more regularized solution

# View the coefficients
ridge.coefs_df <- as.data.frame(as.matrix(ridge.coefs))
ridge.coefs_df <- ridge.coefs_df %>%
  arrange(desc(s1))
print(ridge.coefs_df)
```

```
##                                      s1
## absence_rate_8              4.019117804
## absence_rate_7              3.793818048
## absence_rate_6              3.078968714
## age_eighthfall1             1.340412716
## school_mobility_8           0.684564638
## eds                         0.589478997
## ever_chrabsent_middle       0.450851208
## ever_ltsusp_middle          0.434884287
## chrabsent_8                 0.406745204
## not_math_proficient_7        0.379000722
## ever_ISS_8                   0.376820712
## ever_lep                     0.357693708
## not_math_proficient_6        0.333044826
## not_math_proficient_8        0.324102371
## school_mobility_middle       0.280548309
## school_mobility_6            0.255287010
## ever_OSS_7                   0.243330189
## town                         0.238160809
## not_read_proficient_8        0.232533659
## ever_OSS_8                   0.192493260
## ever_OSS_middle              0.174942398
## ever_stsusp_middle           0.173030003
## ever_ISS_6                   0.172409246
## not_read_proficient_6        0.114398942
## ever_swd                     0.104089649
## ever_suspended               0.100830121
## ever_ISS_middle              0.093267749
## chrabsent_7                  0.014119038
## urban                        0.008271267
## chrabsent_middle            -0.020081707
## chrabsent_6                 -0.029856129
## rural                       -0.034946480
## ever_ISS_7                  -0.038168593
## not_read_proficient_7       -0.046164099
## no_read_proficiency_middle  -0.052569972
## school_mobility_7           -0.052874225
## suburban                    -0.088514173
## swd_8                       -0.113696765
## ever_OSS_6                  -0.161602334
## lep_8                       -0.258671378
## no_math_proficiency_middle  -0.304471721
## (Intercept)                -22.368243059
```

```
write.csv(ridge.coefs_df, "ridge.smote.coefs.csv", row.names = TRUE)
```

Running the undersampled Random Forest

```
# Running RF
set.seed(2023)
RF.dropout <- randomForest(dropout ~ ., data = train, ntree = 100, importance = TRUE)
```

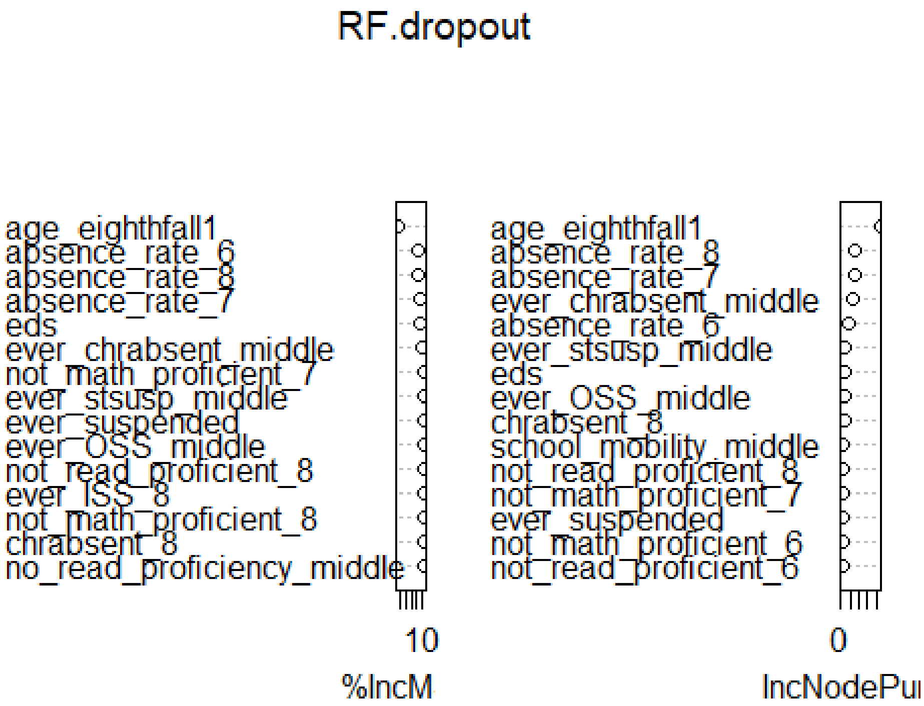
```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values.  Are you sure you want to do regression?
```

```
print(RF.dropout)
```

```
## 
## Call:
```

```
##  randomForest(formula = dropout ~ ., data = train, ntree = 100,      importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 100
## No. of variables tried at each split: 13
##
##           Mean of squared residuals: 0.1111992
##                     % Var explained: 55.52
```

```
varImpPlot(RF.dropout,n.var=min(15, nrow(RF.dropout$importance)), type=NULL, class=NULL,
scale=TRUE)
```



```
importance(RF.dropout)
```

```
##                              %IncMSE IncNodePurity
## ever_stsusp_middle         7.357801552   23.03567126
## ever_ltsusp_middle         0.000000000    0.02645886
## ever_OSS_6               3.011104964    3.04166274
## ever_OSS_7               4.885170317    5.27615004
## ever_OSS_8               4.372833450    6.22498879
## ever_OSS_middle            6.822605874   15.83789509
## ever_ISS_middle           5.238630202    5.07931276
## ever_ISS_6              2.843352651    4.45979083
## ever_ISS_7              3.923779699    5.20951590
## ever_ISS_8              6.041411416    6.28624668
## not_math_proficient_6      4.374896357    8.02647551
## not_math_proficient_7       7.641696211   9.02598334
## not_math_proficient_8       5.619537707   6.73327242
## no_math_proficiency_middle  0.597667263   3.07844779
## not_read_proficient_6       5.361775300   7.83358303
## not_read_proficient_7       5.126090763   7.56060598
## not_read_proficient_8       6.595685135   9.03686084
## no_read_proficiency_middle  5.453730668   4.62100027
## eds                        12.353052044  17.68302315
## age_eighthfall1            55.308823205 221.80200689
## ever_swd                   -0.314064360   4.60125657
## swd_8                      -2.017342949   4.57285834
## ever_lep                    5.037565831   4.17305636
## lep_8                       2.472650218   4.54003225
## absence_rate_6             16.821171955  44.21367875
## absence_rate_7             15.134229519  71.86635305
## absence_rate_8             16.501284923  80.22064527
## chrabsent_6                 2.077164161   2.22124475
## chrabsent_7                 1.669718593   2.02149604
## chrabsent_8                 5.604564115  12.27762659
## ever_chrabsent_middle       9.695786815  65.62614747
## chrabsent_middle           -2.062052328   0.73220428
## school_mobility_middle      3.831014208  10.29501623
## school_mobility_8           5.195899455   1.67606163
## school_mobility_7           0.310168666   0.78588163
## school_mobility_6           0.447620569   1.37300890
## urban                       0.183004385   6.13505405
## suburban                    1.633137098   6.44602829
## town                       -0.951965923   5.13384612
## rural                       0.005143379   6.79848137
## ever_suspended              6.953041375   8.45884022
```

Running the undersampled Logistic regression

```
train <- read.csv("D:/NCERDC_DATA/Alam/ML/undersampletrain.csv")

# Fit the logistic regression model
log1.m <- glm(dropout ~ ., data = subset(train, select = -c(female, hispanic, asian, black, white, other_rac
e)), family = 'binomial')
summary(log1.m)
```

```
## 
## Call:
## glm(formula = dropout ~ ., family = "binomial", data = subset(train, 
##     select = -c(female, hispanic, asian, black, white, other_race)))
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -3.6099  -0.4793  -0.0425   0.3986   2.8545  
## 
```

```
## Coefficients: (2 not defined because of singularities)
##                             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                -31.98031    1.79009 -17.865  < 2e-16 ***
## ever_stsusp_middle           0.42797    0.32696   1.309 0.190561    
## ever_ltsusp_middle           0.95081    1.58589   0.600 0.548810    
## ever_OSS_6                  -0.40687    0.22344  -1.821 0.068610 .  
## ever_OSS_7                   0.24676    0.22251   1.109 0.267430    
## ever_OSS_8                   0.11029    0.22510   0.490 0.624163    
## ever_OSS_middle                   NA         NA      NA       NA    
## ever_ISS_middle             -0.05401    0.29571  -0.183 0.855075    
## ever_ISS_6                   0.34776    0.23926   1.453 0.146087    
## ever_ISS_7                  -0.09817    0.18527  -0.530 0.596205    
## ever_ISS_8                   0.59844    0.17122   3.495 0.000474 ***
## not_math_proficient_6        0.69017    0.21998   3.137 0.001704 ** 
## not_math_proficient_7        0.75114    0.21309   3.525 0.000424 ***
## not_math_proficient_8        0.35258    0.17201   2.050 0.040386 *  
## no_math_proficiency_middle  -1.02751    0.31315  -3.281 0.001033 ** 
## not_read_proficient_6       -0.04529    0.24213  -0.187 0.851623    
## not_read_proficient_7       -0.25915    0.17853  -1.452 0.146613    
## not_read_proficient_8        0.26872    0.15985   1.681 0.092750 .  
## no_read_proficiency_middle   0.11145    0.29008   0.384 0.700832    
## eds                          0.75157    0.12353   6.084 1.17e-09 ***
## age_eighthfall1              1.97810    0.10924  18.108  < 2e-16 ***
## ever_swd                     0.27628    0.34247   0.807 0.419821    
## swd_8                       -0.53111    0.36159  -1.469 0.141881    
## ever_lep                     1.38435    0.52323   2.646 0.008150 ** 
## lep_8                       -1.44213    0.56131  -2.569 0.010193 *  
## absence_rate_6               4.07835    1.79960   2.266 0.023436 *  
## absence_rate_7               7.59319    1.89950   3.997 6.40e-05 ***
## absence_rate_8              10.23255    1.82863   5.596 2.20e-08 ***
## chrabsent_6                 -0.62118    0.34927  -1.778 0.075323 .  
## chrabsent_7                 -0.81284    0.32747  -2.482 0.013057 *  
## chrabsent_8                 -0.34491    0.34974  -0.986 0.324041    
## ever_chrabsent_middle        0.84825    0.36509   2.323 0.020156 *  
## chrabsent_middle             0.31961    0.58433   0.547 0.584404    
## school_mobility_middle       0.39525    0.12521   3.157 0.001595 ** 
## school_mobility_8            0.87265    0.50490   1.728 0.083923 .  
## school_mobility_7           -0.22066    0.58291  -0.379 0.705015    
## school_mobility_6            0.19069    0.53461   0.357 0.721325    
## urban                        0.04086    0.14124   0.289 0.772356    
## suburban                    -0.11058    0.14925  -0.741 0.458752    
## town                         0.34076    0.20092   1.696 0.089882 .  
## rural                             NA         NA      NA       NA    
## ever_suspended               0.06706    0.27385   0.245 0.806536    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
```

```
##     Null deviance: 4300.3  on 3101  degrees of freedom
## Residual deviance: 2039.7  on 3062  degrees of freedom
## AIC: 2119.7
##
## Number of Fisher Scoring iterations: 7
```

BIBLIOGRAPHY

Agasisti, T., & Bowers, A. J. (2017). Data analytics and decision making in education: towards the educational data scientist as a key actor in schools and higher education institutions. In *Handbook of contemporary education economics* (pp. 184-210). Edward Elgar Publishing.

Ali, A., Shamsuddin, S. M., & Ralescu, A. L. (2013). Classification with class imbalance problem. *Int. J. Advance Soft Compu. Appl*, *5*(3), 176-204.

Allensworth, E. M., & Easton, J. Q. (2007). What Matters for Staying On-Track and Graduating in Chicago Public High Schools: A Close Look at Course Grades, Failures, and Attendance in the Freshman Year. Research Report. *Consortium on Chicago School Research*.

Allensworth, E. M. (2013). The use of ninth grade early warning indicators to improve Chicago schools. *Journal of Education for Students Placed at Risk*, 18(1), 68–83.

Allensworth E., Gwynne J., Moore P., de la Torre M. (2014). *Looking forward to high school and college: Middle grade indicators of readiness in Chicago Public Schools*. Chicago, IL: University of Chicago Consortium on Chicago School Research.

Allensworth, E. M., Nagaoka, J., & Johnson, D. W. (2018). High School Graduation and College Readiness Indicator Systems: What We Know, What We Need to Know. Concept Paper for Research and Practice. *University of Chicago Consortium on School Research*.

Allensworth, E. M., & Clark, K. (2019). Are GPAs An Inconsistent Measure Of College Readiness Across High Schools? Examining Assumptions About Grades Versus Standardized Test Scores. University Of Chicago Consortium On School Research.

Anderson, H., Boodhwani, A., & Baker, R. S. (2019). Assessing the Fairness of Graduation Predictions. In EDM.

Asselman, A., Khaldi, M., & Aammou, S. (2023). Enhancing the prediction of student performance based on the machine learning XGBoost algorithm. *Interactive Learning Environments*, *31*(6), 3360-3379.

Babar, V. S., & Ade, R. (2015). A review on imbalanced learning methods. *Int. J. Comput. Appl*, *975*(2), 23-27.

Baker, R. S., & Hawn, A. (2022). Algorithmic bias in education. International Journal of *Artificial Intelligence in Education, 32*.

Baker, R. S., Esbenshade, L., Vitale, J., & Karumbaiah, S. (2023a). Using Demographic Data as Predictor Variables: A Questionable Choice. *Journal of Educational Data Mining*, *15*(2), 22-52.

Baker, R., Hawn, M.A., Lee, S. (2023b). Algorithmic bias: the state of the situation and policy recommendations, in *OECD Digital Education Outlook 2023: Towards an Effective Digital Education Ecosystem*, OECD Publishing, Paris.

Baker, R.S. (in press) Algorithmic Bias in Education and Steps Towards Fairness. In A.S. Wells, E.N. Walker (Eds.) *Learning and Thriving Across the Lifespan: The 100-Year Intellectual Legacy of Professor Edmund Gordon*.

Bala, N. (2019). The Danger of Facial Recognition in Our Children's Classrooms. *Duke L. & Tech. Rev.*, *18*, 249.

Balfanz, R., Herzog, L., & Mac Iver, D. J. (2007). Preventing Student Disengagement And Keeping Students On The Graduation Path In Urban Middle-Grades Schools: Early Identification And Effective Interventions. Educational Psychologist, 42(4), 223-235.

Balfanz, R. (2009). Putting middle grades students on the graduation path. *Policy and practice brief*.

Balfanz, R., & Byrnes, V. (2012). The importance of being there: A report on absenteeism in the nation’s public schools. Baltimore, MD: Baltimore: Johns Hopkins University Center for Social Organization of Schools.

Balfanz, R., Byrnes, V., & Fox, J. H. (2014). Sent home and put off track: The antecedents, disproportionalities, and consequences of being suspended in the ninth grade. Journal of Applied Research on Children, 5.

Balfanz, R., Byrnes, V., & Fox, J. H. (2015). Sent home and put off track. *Closing the school discipline gap: Equitable remedies for excessive exclusion*, 17-30.

Balfanz, R., & Byrnes, V. (2018). Using data and the human touch: Evaluating the NYC interagency campaign to reduce chronic absenteeism. Journal of Education for Students Placed at Risk (JESPAR), 23, 107-121.

Balfanz, R. (2016). Missing school matters. *Phi Delta Kappan*, *98*(2), 8-13.

Balu, R. & Ehrlich, S. B. (2018) Making sense out of incentives: A framework for considering the design, use, and implementation of incentives to improve attendance, Journal of Education for Students Placed at Risk, 23, 93-106.

Balu, R., Porter, K., & Gunton, B. (2016). Can informing parents help high school students show up for school. Policy Brief. New York, NY: MDRC.

Belfield, C. R., & Levin, H. M. (Eds.). (2007). The price we pay: Economic and social consequences of inadequate education. Brookings Institution Press.

Bishop, C.M., Bishop, H., & Cham, S. (2023). Deep Learning: Foundations and Concepts. Hardback. ISBN 978-3031454677.

Bowers, A. (2009), "Reconsidering grades as data for decision making: more than just academic knowledge", Journal of Educational Administration, Vol. 47/5.

Bowers, A. J. (2010). Grades And Graduation: A Longitudinal Risk Perspective To Identify Student Dropouts. The Journal Of Educational Research, 103 (3), 191–207.

Bowers, A.J., Sprott, R. (2012a) Why Tenth Graders Fail to Finish High School: A Dropout Typology Latent Class Analysis. The Journal of Education for Students Placed at Risk (JESPAR), 17(3), 129-148.

Bowers, A.J., Sprott, R. (2012b) Examining the Multiple Trajectories Associated with Dropping Out of High School: A Growth Mixture Model Analysis. The Journal of Educational Research, 105(3), 176-195.

Bowers, A. J., Sprott, R., & Taff, S. A. (2013). Do we know who will drop out?: A review of the predictors of dropping out of high school: Precision, sensitivity, and specificity. *The High School Journal*, *96*(2), 77-100.

Bowers, A. J. (2019). Analyzing the longitudinal K-12 grading histories of entire cohorts of students: Grades, data driven decision making, dropping out and hierarchical cluster analysis. *Practical Assessment, Research, and Evaluation*, *15*(1), 7.

Bowers, A. J., & Zhou, X. (2019). Receiver operating characteristic (ROC) area under the curve (AUC): A diagnostic measure for evaluating the accuracy of predictors of education outcomes. *Journal of Education for Students Placed at Risk (JESPAR)*, *24*(1), 20-46.

Bowers, A.J. (2021) Early Warning Systems and Indicators of Dropping Out of Upper Secondary School: The Emerging Role of Digital Technologies. OECD Digital Education Outlook 2021: Pushing the Frontiers with Artificial Intelligence, Blockchain and Robots, Chapter 9, p.173-194. OECD Publishing, Paris.

Bowers, A. J., & Choi, Y. (2023). Building school data equity, infrastructure, and capacity through FAIR data standards: Findable, Accessible, Interoperable, and Reusable. *Educational Researcher*, *52*(7), 450-458.

Belfield, C. R., & Levin, H. M. (Eds.). (2007). *The price we pay: Economic and social consequences of inadequate education*. Brookings Institution Press.

Borman, G. D., Rozek, C. S., Pyne, J., & Hanselman, P. (2019). Reappraising academic and social adversity improves middle school students' academic achievement, behavior, and well-being. *Proceedings of the National Academy of Sciences*, *116*(33), 16286-16291.

Burke, A. (2015). Early Identification Of High School Graduation Outcomes In Oregon Leadership Network Schools. Rel 2015-079. Regional Educational Laboratory Northwest.

Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2012). DBSMOTE: density-based synthetic minority over-sampling technique. *Applied Intelligence*, *36*, 664-684.

Butler, M. A. (1990). Rural-urban continuum codes for metro and nonmetro counties. US Department of Agriculture, Economic Research Service, Agriculture and Rural Economy Division.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5-32.

Brookhart, S., T. Guskey, A. Bowers, J. McMillan, J. Smith, L. Smith, M. Stevens and M. Welsh (2016), “A Century of Grading Research”. Review of Educational Research, Vol. 86/4.

Brown, G. (2017). Ensemble Learning. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning and Data Mining. Springer, Boston, MA.

Canbolat, Y. (2024). Early Warning for Whom? Regression Discontinuity Evidence From the Effect of Early Warning System on Student Absence. *Educational Evaluation and Policy Analysis*, 01623737231221503.

Cannistrà, M., Masci, C., Ieva, F., Agasisti, T., & Paganoni, A. M. (2022). Early-predicting dropout of university students: an application of innovative multilevel machine learning and statistical techniques. *Studies in Higher Education, 47*(9), 1935-1956.

Casillas, A., Robbins, S., Allen, J., Kuo, Y.-L., Hanson, M. A., & Schmeiser, C. (2012). Predicting early academic failure in high school from prior academic achievement, psychosocial characteristics, and behavior. *Journal of Educational Psychology, 104*(2), 407–420.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research, 16*, 321-357.

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

Chung, J. Y., & Lee, S. (2019). Dropout early warning systems for high school students using

machine learning. Children and Youth Services Review, 96, 346-353.

Colak Oz, H., Güven, Ç., & Nápoles, G. (2023). School dropout prediction and feature importance exploration in Malawi using household panel data: machine learning approach. Journal of Computational Social Science, 6(1), 245-287.

Cook, J., & Ramadas, V. (2020). When to consult precision-recall curves. The Stata Journal, 20(1), 131-148.

Coleman, C., Baker, R. S., & Stephenson, S. (2019). A Better Cold-Start for Early Prediction of Student At-Risk Status in New School Districts. *International Educational Data Mining Society*.

Coleman, C. J. (2021). Exploring A Generalizable Machine Learned Solution For Early Prediction Of Student At-Risk Status. Columbia University.

Cook, J., & Ramadas, V. (2020). When to consult precision-recall curves. *The Stata Journal*, *20*(1), 131-148.

Corbett-Davies, S., Gaebler, J., Nilforoshan, H., Shroff, R., & Goel, S. (2023). The measure and mismeasure of fairness. *Journal of Machine Learning Research*, 24(312), 1–117.

Cortes, K. E., & Goodman, J. S. (2014). Ability-tracking, instructional time, and better pedagogy: The effect of double-dose algebra on student achievement. *American Economic Review*, *104*(5), 400-405.

Crofton, M., & Neild, R. C. (2018). Getting on Track to Graduation: Ninth Graders' Credit Accumulation in the School District of Philadelphia, 2015-2017. Starting Strong: A Research Series on the Transition to High School. *Philadelphia Education Research Consortium*.

Cutler, A., Cutler, D. R., & Stevens, J. R. (2012). Random forests. *Ensemble machine learning: Methods and applications*, 157-175.

Daggett, L. M. (2020). Female Student Patient" Privacy" at Campus Health Clinics: Realities and Consequences. *U. Balt. L. Rev.*, *50*, 77.

Dalton, B., Glennie, E., & Ingles, S. J. (2009). Late high school dropouts: Characteristics, experiences, and changes across cohorts. (NCES 2009–307). Washington, DC: National Center for Education Statistics, Institute of Education Sciences, U.S. Department of Education.

Davis, L. P., & Museus, S. D. (2019). What is deficit thinking? An analysis of conceptualizations of deficit thinking and implications for scholarly research. *NCID Currents*, *1*(1).

Dee, T. S. (2004). Are there civic returns to education? Journal of public economics, 88(9-10), 1697-1720.

Dee, T. S. (2023). Where the kids went: Nonpublic schooling and demographic change during the pandemic exodus from public schools. *Teachers College Record*, *125*(6), 119-129.

Doll, J. J., Eslami, Z., & Walters, L. (2013). Understanding why students drop out of high school, according to their own reports: Are they pushed or pulled, or do they fall out? A comparative analysis of seven nationally representative studies. *Sage Open*, *3*(4).

Dunkelau, J., & Duong, M. K. (2022). Towards equalised odds as fairness metric in academic performance prediction. *arXiv preprint arXiv:2209.14670*.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness through awareness. In Proceedings of the 3rd innovations in theoretical computer science conference (pp. 214–226).

Easton, J. Q., Johnson, E., & Sartain, L. (2017). The predictive power of ninth-grade GPA. *Chicago, IL: University of Chicago Consortium on School Research*, 2018-10.

Fahrmeir, L., Kneib, T., Lang, S., & Marx, B. D. (2022). Mixed Models. In *Regression: Models, Methods and Applications* (pp. 367-430). Berlin, Heidelberg: Springer Berlin Heidelberg.

Faria, A. M., Sorensen, N., Heppen, J., Bowdon, J., Taylor, S., Eisner, R., & Foster, S. (2017). Getting Students on Track for Graduation: Impacts of the Early Warning Intervention and Monitoring System after One Year. REL 2017-272. *Regional Educational Laboratory Midwest*.

Fassett, K. T., Wolcott, M. D., Harpe, S. E., & McLaughlin, J. E. (2022). Considerations for writing and including demographic variables in education research. *Currents in Pharmacy Teaching and Learning*, *14*(8), 1068-1078.

Feathers, T. (2023a, April 27). *False alarm: How Wisconsin uses race and income to label students high risk*. The Markup. https://themarkup.org/machine-learning/2023/04/27/false-alarm-how-wisconsin-uses-race-and-income-to-label-students-high-risk

Feathers, T. (2023b, May 11). *Takeaways from our investigation into Wisconsin's racially inequitable dropout algorithm*. The Markup. https://themarkup.org/the-breakdown/2023/04/27/takeaways-from-our-investigation-into-wisconsins-racially-inequitable-dropout-algorithm

Feng, W., Huang, W., & Ren, J. (2018). Class imbalance ensemble learning based on the margin theory. *Applied Sciences*, *8*(5), 815.

Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, *61*, 863-905.

Flach, P., & Kull, M. (2015). Precision-recall-gain curves: PR analysis done right. *Advances in neural information processing systems*, *28*.

Frazelle, S., & Nagel, A. (2015). A Practitioner's Guide to Implementing Early Warning Systems. REL 2015-056. *Regional Educational Laboratory Northwest*.

Fredricks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School Engagement: Potential of the Concept, State of the Evidence. *Review of Educational Research*, *74*(1), 59–109.

Freeman, J., & Simonsen, B. (2015). Examining the impact of policy and practice interventions on high school dropout and school completion rates: A systematic review of the literature. *Review of educational research*, *85*(2), 205-248.

Friedman, J., Hastie, T., Tibshirani, R., Narasimhan, B., Tay, K., Simon, N., ... & Yang, J. (2023). glmnet: Lasso and elastic-net regularized generalized linear models. *Astrophysics Source Code Library*, ascl-2308.

Galligan, C., Rosenfeld, H., Kleinman, M., & Parthasarathy, S. (2020). *Cameras in the classroom: Facial recognition technology in schools*.

Geverdt, D., & Nixon, L. (2018). Sidestepping The Box: Designing A Supplemental Poverty Indicator For School Neighborhoods (Nces 2017-039). Us Department Of Education. Washington, Dc: National Center For Education Statistics. National Center For Education Statistics, Washington, DC.

Geverdt, D. (2017). Education Demographic and Geographic Estimates (EDGE) Geocodes: Public Schools and Local Education Agencies, 2015-2016 (NCES 2017-041). U.S. Department of Education. Washington, DC: National Center for Education Statistics.

Gong, X., Hu, M., & Zhao, L. (2018). Big data toolsets to pharmacometrics: application of machine learning for time-to-event analysis. *Clinical and translational science, 11(3)*, 305-311.

Goodman, J., Cortes, K., & Nomi, T. (2013). *A Double Dose of Algebra* (No. 95911).

Gottfried, M. A. (2014). Chronic absenteeism and its effects on students' academic and socioemotional outcomes. Journal of Education for Students Placed at Risk (JESPAR), 19(2), 53-75.

Gottfried, M. A. (2015). Chronic absenteeism in the classroom context: Effects on achievement. Urban Education, 1-32.

Gottfried, M. A. (2017). Linking getting to school with going to school. Educational Evaluation and Policy Analysis, 39, 571-592.

Gubbels, J., van der Put, C. E., & Assink, M. (2019). Risk factors for school absenteeism and dropout: A meta-analytic review. *Journal of youth and adolescence, 48*, 1637-1667.

Gutierrez-Pachas, D. A., Garcia-Zanabria, G., Cuadros-Vargas, E., Camara-Chavez, G., & Gomez-Nieto, E. (2023). Supporting Decision-Making Process on Higher Education Dropout by Analyzing Academic, Socioeconomic, and Equity Factors through Machine Learning and Survival Analysis Methods in the Latin American Context. Education Sciences, 13(2), 154.

Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. *International conference on intelligent computing* (pp. 878-887). Berlin, Heidelberg: Springer Berlin Heidelberg.

Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. *Advances in neural information processing systems, 29.*

Harwell, M., & Lebeau, B. (2010). Student Eligibility For A Free Lunch As SES Measure In Education Research. Educational Researcher, 39 (2), 120–131.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Tibshirani, R., & Friedman, J. (2009a). Random forests. *The elements of statistical learning: Data mining, inference, and prediction*, 587-604.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Tibshirani, R., & Friedman, J. (2009b). Ensemble learning. *The elements of statistical learning: data mining, inference, and prediction*, 605-624.

Hastie, T., Tibshirani, R., & Wainwright, M. (2015). Statistical learning with sparsity. *Monographs on statistics and applied probability*, *143*(143), 8.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)* (pp. 1322-1328). IEEE.

He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55-67.

Huang, Y., Li, J., Li, M., & Aparasu, R. R. (2023). Application of machine learning in predicting survival outcomes involving real-world data: a scoping review. *BMC medical research methodology*, *23*(1), 268.

Humm Patnode, A., Gibbons, K., & Edmunds, R. R. (2018). Attendance and Chronic Absenteeism: Literature Review. Saint Paul, MN: University of Minnesota, College of Education and Human Development, Center for Applied Research and Educational Improvement.

Intellispark. (n.d.). *From early warning to early action*. Intellispark. https://www.intellispark.com/blog/from-early-warning-to-early-action

Jackson, C. K. (2010). The effects of an incentive-based high-school intervention on college outcomes (No. w15722). National Bureau of Economic Research.

Jackson C. K. (2018). What do test scores miss? The importance of teacher effects on non-test score outcomes. Journal of Political Economy, 126, 2072–2107.

Jackson, C. K. (2018). What do test scores miss? The importance of teacher effects on non–test score outcomes. *Journal of Political Economy*, *126*(5), 2072-2107.

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). Statistical learning. In An Introduction to Statistical Learning: with Applications in Python (pp. 15-67). Cham: Springer International Publishing.

Jens, C., Page, T. B., & Reeder III, J. C. (2022). Controlling for group-level heterogeneity in causal forest.

Jiang, W., & Pardos, Z. A. (2021). Towards equity and algorithmic fairness in student grade prediction. In Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society (pp. 608–617).

Kahlenberg, R. D. (2004). All Together Now: Creating Middle-Class Schools Through Public School Choice. Rowman & Littlefield.

Kasem, A., Ammar Ghaibeh, A., & Moriguchi, H. (2017). Empirical study of sampling methods for classification in imbalanced clinical datasets. In *Computational Intelligence in Information Systems: Proceedings of the Computational Intelligence in Information Systems Conference (CIIS 2016)* (pp. 152-162). Springer International Publishing.

Khan, A. A., Chaudhari, O., & Chandra, R. (2024). A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation. *Expert Systems with Applications*, *244*, 122778.

Kieffer M.J., Marinell W.H., Stephenson N.S. (2011). *The middle grades student transitions study: Navigating the middle grades and preparing students for high school graduation*. New York, NY: New York University, Steinhardt School of Education, The Research Alliance for New York City Schools.

Knowles, J. E. (2015). Of Needles and Haystacks: Building an Accurate Statewide Dropout Early Warning System in Wisconsin. *Journal of Educational Data Mining*, *7*(3), 18-67.

Kolasseri, A. E. (2024). Comparative study of machine learning and statistical survival models for enhancing cervical cancer prognosis and risk factor assessment using SEER data. *Scientific Reports, 14*(1), 22203.

Kotsiantis, S. B., Pierrakeas, C. J., & Pintelas, P. E. (2003). Preventing student dropout in distance learning using machine learning techniques. In Knowledge-Based Intelligent Information and Engineering Systems: 7th International Conference, KES 2003, Oxford, UK, September 2003. Proceedings, Part II 7 (pp. 267-274). Springer Berlin Heidelberg.

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* (Vol. 26, p. 13). New York: Springer.

Kroese, D. P., Botev, Z., Taimre, T., & Vaisman, R. (2019). Data Science And Machine Learning: Mathematical And Statistical Methods. CRC Press.

Kruger, J. G. C., De Souza Britto Jr, A., & Barddal, J. P. (2023). An Explainable Machine Learning Approach For Student Dropout Prediction. Expert Systems With Applications, 233, 120933.

Kunapuli, G. (2023). *Ensemble methods for machine learning*. Simon and Schuster.

Lee, S., & Chung, J. Y. (2019). The Machine Learning-Based Dropout Early Warning System For Improving The Performance Of Dropout Prediction. Applied Sciences, 9 (15), 3093.

Lee, H., & Kizilcec, R. F. (2020). Evaluation of fairness trade-offs in predicting student success. *arXiv preprint arXiv:2007.00088*.

Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., & Seliya, N. (2018). A Survey On Addressing High-Class Imbalance In Big Data. Journal Of Big Data, 5 (1), 1–30.

Lin, W. C., Tsai, C. F., Hu, Y. H., & Jhang, J. S. (2017). Clustering-based undersampling in class-imbalanced data. *Information Sciences*, *409*, 17-26.

Losen, D., Orfield, G., & Balfanz, R. (2006). Confronting The Graduation Rate Crisis In Texas. Civil Rights Project At Harvard University.

Loukina, A., Madnani, N., & Zechner, K. (2019). The many dimensions of algorithmic fairness in educa tional applications. In Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Build ing Educational Applications (pp. 1–10).

Lunardon, N., Menardi, G., and Torelli, N. (2014). ROSE: a Package for Binary Imbalanced Learning. R Jorunal, 6:82–92.

Lundberg., S., & Lee, S.I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, *30*, 4765-4774.

Mani, I., & Zhang, I. (2003). kNN approach to unbalanced data distributions: a case study involving information extraction. *Proceedings of workshop on learning from imbalanced datasets* (Vol. 126, No. 1, pp. 1-7). ICML.

Mac Iver, M. A. (2013). Early warning indicators of high school outcomes. *Journal of Education for Students Placed at Risk (JESPAR)*, *18*(1), 1-6.

Mac Iver, M. A., & Messel, M. (2013). The ABCs of keeping on track to graduation: Research findings from Baltimore. *Journal of Education for Students Placed at Risk (JESPAR)*, *18*(1), 50-67.

Mac Iver, M. A., Stein, M. L., Davis, M. H., Balfanz, R. W., & Fox, J. H. (2019). An efficacy study of a ninth-grade early warning indicator intervention. *Journal of Research on Educational Effectiveness*, *12*(3), 363-390.

McFarland, J., Cui, J., Rathbun, A., & Holmes, J. (2018). Trends in High School Dropout and Completion Rates in the United States: 2018. Compendium Report. NCES 2019-117. *National Center for Education Statistics*.

McLaughlin, J. E., McLaughlin, G. W., McLaughlin, J. S., & White, C. Y. (2016). Using Simpson’s diversity index to examine multidimensional models of diversity in health professions education. *International journal of medical education*, *7*, 1.

Mduma, N., Kalegele, K., & Machuve, D. (2019). A survey of machine learning approaches and techniques for student dropout prediction. doi: 5334/dsj-2019-014

Menardi, G., & Torelli, N. (2014). Training and assessing classification rules with imbalanced data. *Data mining and knowledge discovery*, *28*, 92-122.

Mienye, I. D., & Sun, Y. (2022). A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access*, *10*, 99129-99149.

Nakas, C., Bantis, L., & Gatsonis, C. (2023). *ROC Analysis for Classification and Prediction in Practice*. CRC Press.

National Center for Education Statistics. (2017). *The condition of education 2017* (NCES 2017-017). U.S. Department of Education, Institute of Education Sciences.

National Center for Education Statistics. (2022). Common Core Of Data Public Elementary/Secondary School Universe Survey.

National Center for Education Statistics. (2024). High School Graduation Rates. *Condition of Education*. U.S. Department of Education, Institute of Education Sciences. Retrieved from https://nces.ed.gov/programs/coe/indicator/coi.Nembrini, S., König, I. R., & Wright, M. N. (2018). The revival of the Gini importance?. Bioinformatics, 34(21), 3711-3718.

Neild, R. C., Stoner-Eby, S., & Furstenberg, F. F. (2008). Connecting entrance and departure: The transition to ninth grade and high school dropout. *Education and Urban Society, 40*, 543–569.

Neild, R. C. (2009). Falling off track during the transition to high school: What we know and what can be done. *The Future of Children*, 53-76.

Nomi, T., & Allensworth, E. M. (2013). Sorting and supporting: Why double-dose algebra led to better test scores but more course failures. *American Educational Research Journal*, *50*(4), 756-788.

Nomi, T., Raudenbush, S. W., & Smith, J. J. (2021). Effects of double-dose algebra on college persistence and degree attainment. *Proceedings of the National Academy of Sciences*, *118*(27), e2019030118.

Norbury, H., Wong, M., Wan, Y., Reese, K., Dhillon, S., & Gerdeman, R. D. (2012). Using the Freshman On-Track Indicator to Predict Graduation in Two Urban Districts in the Midwest Region. Issues & Answers. REL 2012-No. 134. *Regional Educational Laboratory Midwest*.

North Carolina General Assembly. (n.d.). *Chapter 115C, Article 7: The North Carolina School Improvement and Accountability Act*. https://www.ncleg.gov/EnactedLegislation/Statutes/PDF/ByArticle/Chapter_115c/Article_7.pdf

Nussberger, A. M., Luo, L., Celis, L. E., & Crockett, M. J. (2022). Public attitudes value interpretability but prioritize accuracy in Artificial Intelligence. *Nature communications, 13*(1), 5821.

Ogresta, J., Rezo, I., Kožljan, P., Paré, M. H., & Ajduković, M. (2021). Why do we drop out? Typology of dropping out of high school. *Youth & society, 53*(6), 934-954.

Okoye, K., & Hosseini, S. (2024). Mann–Whitney U Test and Kruskal–Wallis H Test Statistics in R. In *R programming: Statistical data analysis in research* (pp. 225-246). Singapore: Springer Nature Singapore.

Paquette, L., Ocumpaugh, J., Li, Z., Andres, J.M.A.L., Baker, R.S. (2020) Who's Learning? Using Demographics in EDM Research. *Journal of Educational Data Mining, 12* (3), 1-30.

Parsons, E., Koedel, C., & Tan, L. (2019). Accounting For Student Disadvantage In Value-Added Models. Journal Of Educational And Behavioral Statistics, 44 (2), 144–179.

Perdomo, J. C., Britton, T., Hardt, M., & Abebe, R. (2023). Difficult lessons on social prediction from Wisconsin Public Schools. *arXiv preprint arXiv:2304.06205*.

Pérez Fernández, S., Martínez Camblor, P., Filzmoser, P., & Corral Blanco, N. O. (2018). nsROC: an R package for non-standard ROC curve analysis. The R Journal, 10 (2).

Peters, N. R. (2021). The Golem in the Machine: FERPA, Dirty Data, and Digital Distortion in the Education Record. *Wash. & Lee L. Rev., 78,* 1991.

Polikar, R. (2012). Ensemble learning. Ensemble machine learning: Methods and applications. *Cham: Springer*.

Purcell, Z. A., Dong, M., Nussberger, A. M., Köbis, N., & Jakesch, M. (2024). People have different expectations for their own versus others' use of AI-mediated communication tools. *British Journal of Psychology*.

Reardon, S. F., & Bischoff, K. (2011). Income Inequality And Income Segregation. American Journal Of Sociology, 116 (4), 1092–1153.

Rickles, J., Heppen, J. B., Allensworth, E., Sorensen, N., & Walters, K. (2018). Online credit recovery and the path to on-time high school graduation. *Educational Researcher*, *47*(8), 481-491.

Rumberger, R. W., & Lim, S. A. (2008). Why students drop out of school: A review of 25 years of research.

Rumberger, R. W. (2011). Dropping out: Why students drop out of high school and what can be done about it.

Rumberger, R. W., & Rotermund, S. (2012). The relationship between engagement and high school dropout. In *Handbook of research on student engagement* (pp. 491-513). Boston, MA: Springer US.

Rumberger, R. W., Addis, H., Allensworth, E., Balfanz, R., Bruch, J., Dillon, E., ... & Tuttle, C. (2017). Preventing Dropout in Secondary Schools. Educator's Practice Guide. What Works Clearinghouse. NCEE 2017-4028. *What Works Clearinghouse*.

Rumberger, R. W. (2020). The economics of high school dropouts. *The economics of education*, 149-158.

Sansone, D. (2019). Beyond Early Warning Indicators: High School Dropout And Machine Learning. Oxford Bulletin Of Economics And Statistics, 81 (2), 456–485.

Sara, N. B., Halland, R., Igel, C., & Alstrup, S. (2015). High-School Dropout Prediction Using Machine Learning: A Danish Large-scale Study. In ESANN (Vol. 2015, p. 23rd).

Sara, N. B., Halland, R., Igel, C., & Alstrup, S. (2015). High-School Dropout Prediction Using Machine Learning: A Danish Large-scale Study. In *ESANN* (Vol. 2015, p. 23rd).

Schober, P., & Vetter, T. R. (2018). Survival analysis and interpretation of time-to-event data: the tortoise and the hare. *Anesthesia & Analgesia*, *127*(3), 792-798.

Seeskin, A., Massion, T., & Usher, A. (2022). Elementary On-Track: Elementary School Students' Grades, Attendance, and Future Outcomes. Research Report. *University of Chicago Consortium on School Research*.

Sha, L., Rakovic, M., Whitelock-Wainwright, A., Carroll, D., Yew, V. M., Gasevic, D., & Chen, G. (2021). Assessing algorithmic fairness in automatic classifiers of educational forum posts. In *Artificial Intelligence in Education: 22nd International Conference, AIED 2021, Utrecht, The Netherlands, June 14–18, 2021, Proceedings, Part I 22* (pp. 381-394). Springer International Publishing.

Sha, L., Raković, M., Das, A., Gašević, D., & Chen, G. (2022). Leveraging class balancing techniques to alleviate algorithmic bias for predictive tasks in education. *IEEE Transactions on Learning Technologies, 15*(4), 481-492.

Sha, L., Gašević, D., & Chen, G. (2023). Lessons from debiasing data for fair and accurate predictive modeling in education. *Expert Systems with Applications*, *228,* 120323.

Shapley, L. S. (1953). A value for n-person games. *Contribution to the Theory of Games, 2*.

Silver, D., Saunders, M., & Zarate, E. (2008). What factors predict high school graduation in the Los Angeles Unified School District California Dropout Research Project. Santa Barbara, CA: University of California Santa Barbara.

Smith, H. (2020). Algorithmic bias: should students pay the price? AI & SOCIETY, 35 (4), 1077– 1078.

Siriseriwan, W. (2024) *smotefamily: A Collection of Oversampling Techniques for Class Imbalance Problem Based on SMOTE*. 2019. R package version 1.4.1

Siriseriwan, W. (2019). *A collection of oversampling techniques for class imbalance problem based on SMOTE*.

Snyder, T. (2022). *With so many kids struggling in school, experts call for revamping early warning systems*. Education Week. https://www.edweek.org/leadership/with-so-many-kids-struggling-in-school-experts-call-for-revamping-early-warning-systems/2022/05

Sorensen, L. C. (2019). "Big Data" In Educational Administration: An Application For Predicting School Dropout Risk. Educational Administration Quarterly, 55 (3), 404–446.

Spooner, A., Chen, E., Sowmya, A., Sachdev, P., Kochan, N. A., Trollor, J., & Brodaty, H. (2020). A comparison of machine learning methods for survival analysis of high-dimensional clinical data for dementia prediction. *Scientific reports*, *10*(1), 20410.

Srujana, B., Verma, D., & Naqvi, S. (2024). Machine learning vs. survival analysis models: a study on right censored heart failure data. *Communications in Statistics-Simulation and Computation*, *53*(4), 1899-1916.

Streiner, D. L., & Cairney, J. (2007). What's under the ROC? An introduction to receiver operating characteristics curves. *The Canadian Journal of Psychiatry*, *52*(2), 121-128.

Štrumbelj, E., & Kononenko, I. (2010). An efficient explanation of individual classifications using game theory. *The Journal of Machine Learning Research*, *11*, 1-18.

Stuit, D., O'Cummings, M., Norbury, H., Heppen, J., Dhillon, S., Lindsay, J., & Zhu, B. (2016). Identifying Early Warning Indicators in Three Ohio School Districts. REL 2016-118. *Regional Educational Laboratory Midwest*.

Su, M., Olson, L. A., Jarratt, D. C., Varma, S., Konstan, J. A., Keller, R. J., & Chen, B. (2022, June). Re-envisioning a K-12 Early Warning System with School Climate Factors. In *Proceedings of the Ninth ACM Conference on Learning@ Scale* (pp. 405-408).

Swets, J. A. (1988). Measuring the accuracy of diagnostic systems. *Science*, *240*(4857), 1285-1293.

Swets, J. A., Dawes, R. M., & Monahan, J. (2000). Better decisions through science. *Scientific American*, *283*(4), 82-87.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *58*(1), 267-288.

Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: data mining, inference, and prediction*. Springer.

Tomek, I. (1976) Two Modifications of CNN. IEEE Transactions on Systems Man and Communications, 6, 769-772.

Valles-Coral, M. A., Salazar-Ramírez, L., Injante, R., Hernandez-Torres, E. A., Juárez-Díaz, J., Navarro-Cabrera, J. R., ... & Vidaurre-Rojas, P. (2022). Density-Based Unsupervised Learning Algorithm to Categorize College Students into Dropout Risk Levels. *Data*, 7(11), 165.

Vance, A., & Waughn, C. (2020). Student privacy's history of unintended consequences. *Seton Hall Legis. J.*, *44*, 515.

Wang, C., Wang, K., Bian, A., Islam, R., Keya, K. N., Foulds, J., & Pan, S. (2022). Do Humans Prefer Debiased AI Algorithms? A Case Study in Career Recommendation. In 27th International Conference on Intelligent User Interfaces (pp. 134–147).

Weissman, A. (2022). Friend Or Foe? The Role Of Machine Learning In Education Policy Research [Doctoral Dissertation].

Wu, T., & Weiland, C. (2024). Leveraging Modern Machine Learning to Improve Early Warning Systems and Reduce Chronic Absenteeism in Early Childhood. EdWorkingPaper No. 24-1081. *Annenberg Institute for School Reform at Brown University*.

Yu, R., Li, Q., Fischer, C., Doroudi, S., & Xu, D. (2020). Towards Accurate and Fair Prediction of Col lege Success: Evaluating Different Sources of Student Data. *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, 292–301.

Zaff, J. F., Donlan, A., Gunning, A., Anderson, S. E., Mcdermott, E., & Sedaca, M. (2017). Factors That Promote High School Graduation: A Review Of The Literature. *Educational Psychology Review, 29,* 447–476.

Zhou, Z., & Hooker, G. (2021). Unbiased measurement of feature importance in tree-based methods. *ACM Transactions on Knowledge Discovery from Data (TKDD), 15*(2), 1-21.