A COMPARISON OF MODERN MACHINE LEARNING METHODS FOR APPLIED ATTRITION MODELING

by

RILEY ALYSSA HESS

(Under the Direction of Neal Outland)

ABSTRACT

The present work evaluates the effectiveness of various supervised machine learning (ML) methods for attrition modeling using real-world employee data, which includes self-reported, HRIS, and performance-related features. Seven algorithms—tree-based methods (CART, random forest), regularized regression (elastic net, LASSO, ridge regression), and a hybrid method incorporating decision trees and regularized regression (XGBoost)—are compared to a logistic regression model across two sample sizes (500, 1000) and 7 datasets. In summary, the two tree-based ensemble methods—XGBoost and random forest—demonstrated the best classification performance compared to their base methods across small (n = 500) and large (n = 1000) sample sizes. These methods relied on information from all three data sources to make predictions, with features related to tenure and pay being most important. Further exploratory analyses indicated that performance can be further refined by adjusting the prediction threshold below 50%. It is hoped that these results will inform practitioners in model selection and will guide additional research in this growing area of inquiry.

INDEX WORDS: Attrition modeling, Turnover research, Machine learning, Classification algorithms

A COMPARISON OF MODERN MACHINE LEARNING METHODS FOR APPLIED ATTRITION MODELING

by

RILEY ALYSSA HESS

B.A., The University of Kansas, 2017

M.S., The University of Georgia, 2022

A Dissertation Submitted to the Graduate Faculty of the University of Georgia in Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

ATHENS, GEORGIA

2025

A COMPARISON OF MODERN MACHINE LEARNING METHODS FOR APPLIED ATTRITION MODELING

by

RILEY ALYSSA HESS

Major Professor:	Neal Outland
Committee:	Frederick Maier
	Brina Hoffman

Electronic Version Approved:

Ron Walcott
Vice Provost for Graduate Education and Dean of the Graduate School
The University of Georgia
May 2025

DEDICATION

To my peers in the Ph.D. program, for the unwavering support, encouragement, and camaraderie we shared throughout this journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

Voluntary employee attrition—where employees choose to leave an organization—has long been a significant challenge across industries (Bolt et al., 2022). In recent years, this issue has become even more pressing. In 2021, 47.8 million workers quit their jobs, followed by over 50 million in 2022, and over 44 million in 2023 (Melhorn & Hoover, 2024). This phenomenon has become a mainstream topic, commonly referred to with colloquial terms like "The Great Resignation" (Melhorn & Hoover, 2024; Smet et al., 2021). High-attrition industries are disproportionately affected by this trend, as the cycle of attrition leads to understaffing, which in turn drives even more employees to leave due to overwork and burnout (Society for Human Resource Management [SHRM], 2024). As companies grapple with labor shortages and rising turnover costs, accurately predicting employee attrition is critical. By anticipating when employees are likely to leave, organizations can take proactive measures to mitigate the impact of attrition on their operations and overall productivity.

Over the past century, researchers in fields such as industrial/organizational psychology and organizational behavior have made significant strides in understanding the drivers of employee attrition (Hom et al., 2017). The resulting body of work, referred to as *turnover* research, is built upon explanatory models which collectively identify factors that influence attrition (Bolt et al., 2022; Rubenstein et al., 2018). As a result, organizations are better equipped to identify the relevant precursors of attrition and develop interventions aimed at reducing attrition. These contributions have undeniably improved our understanding of why employees leave.

Despite these advances, a critical gap remains. Traditional explanatory models, while excellent at identifying correlates of turnover, fall short when it comes to predicting future turnover. Traditional turnover models test theoretical relationships between antecedent variables (e.g., job satisfaction) and outcomes (i.e., turnover), using statistical methods (primarily structural-equation modeling) to evaluate how close the hypothesized model is to the actual data (Russell, 2013). This methodological approach has yielded valuable insights into the psychological mechanisms underlying attrition, but it does not provide evidence of predictive ability (Speer et al., 2019). Compared to explanatory approaches, predictive approaches are less constrained by theory, use different statistical techniques, make use of available data, and apply trained models to make predictions in independent samples.

The lesser-studied practice of *attrition modeling* may provide a path toward prediction of employee attrition. Attrition modeling is the process of using statistical techniques to forecast employee turnover by analyzing relevant organizational data (Speer et al., 2019). This predictive approach often relies on data from human resource information systems (HRIS), employee surveys, and performance metrics to identify patterns that signal oncoming attrition. Thus far, attrition modeling methodologies have received little attention in academic research, and even less is known about their use in applied settings (Allen et al., 2014). Notably, there are a few reports documenting applied attrition modeling research conducted by the U.S. military (e.g., Lucas et al., 2008; Strickland et al., 2005), but very little research is available outside of these works (Putka et al., 2018). This gap is indicative of the wider issue—predictive methodologies have not been fully integrated into organizational research or practice (Pargent et al., 2023; Yarkoni & Westfall, 2017).

At the same time, innovative predictive algorithms and methodologies have come out of the field of machine learning (ML), many of which can contribute to attrition modeling research and practice. Machine learning is a class of predictive modeling approaches in which *algorithms* are used to train *classifiers*, which learn complex patterns from existing data. Once trained, classifiers are tested on unseen data through a process known as *cross-validation*. Specifically, supervised machine learning (ML) algorithms, which predict outcomes based on patterns learned from labeled training data, are well suited for attrition modeling applications because they can be trained on datasets where the outcome (attrition vs. retention) is known. Importantly, ML algorithms are flexible and can incorporate a wide range of organizational data.

While turnover research has flourished in the organizational sciences, research on applied attrition modeling methodologies has mostly been investigated by researchers in the computer sciences, who have produced over 50 papers on the topic in the last 20 years (Akasheh et al., 2024). While this body of work provides valuable insights into the available attrition modeling methodologies, it has several important limitations, including its reliance on simulated data. Many challenges remain, including the lack of practical guidance on applying these techniques effectively in real-world organizational contexts (Landers et al., 2023; Putka et al., 2018).

In Chapter 2, I review the past and current state of turnover research, attrition research and statistical methodology in the organizational sciences. In Chapter 3, I discuss the merits of the ML approach to attrition modeling, providing an in-depth analysis of the advantages and disadvantages of the chosen algorithms, specifically: XGBoost, random forest, classification and regression trees (CART), elastic net regression, lasso regression, ridge regression, and logistic regression. In Chapter 4, I discuss the methodological approach in detail. In Chapter 5, I evaluate the effectiveness of the chosen ML algorithms compared to logistic regression using real-world

call center employee data. The classification performance of these algorithms is compared in three ways: *parsimony*, *technique,* and *sample size*. Parsimony is assessed by comparing the performance of ML algorithms to the current standard, logistic regression, and by comparing the performance of more complex models to their foundational models. Technique is assessed by comparing regularized regression-based algorithms with tree-based algorithms. To provide actionable recommendations for model selection in different organizational contexts, I also test algorithms on datasets with different sample sizes (n = 500, n = 1000).

Through this work, I aim to provide a balanced approach to predicting employee attrition, offering valuable insights for both researchers and practitioners. Findings from this study will serve as a guide for practitioners seeking guidance on attrition modeling analyses and processes, which is largely lacking from empirical literature thus far (Speer, 2024). In keeping with the principle of parsimony, I also seek to identify cases where more complex modeling is not needed. By identifying effective attrition modeling techniques, this study aims to contribute to the empirical literature on attrition modeling with ML and encourage future research on methodologies aimed at prediction.

CHAPTER 2

LITERATURE REVIEW

Turnover Research in the Organizational Sciences

Turnover research in the organizational sciences has historically emphasized explanation over prediction, keeping with broader trends in the psychological sciences (Yarkoni & Westerfall, 2017). Shmueli (2010) distinguishes between these two approaches. Explanatory research aims to minimize bias in each study to uncover generalizable causal mechanisms. Predictive research, on the other hand, prioritizes accuracy for forecasting future outcomes, often at the expense of theoretical fidelity. Turnover research, shaped by explanatory objectives, has primarily developed and tested theoretical models that describe why employees leave organizations (Hom et al., 2017). These models are evaluated using methods that align with explanatory goals, such as ordinary least squares regression and structural equation modeling (SEM), which focus on understanding associations rather than optimizing predictive accuracy. While this approach has broadened our understanding of the cognitive processes involved in voluntary turnover, it has also raised questions about the relevance of this research to practical applications aimed at predicting attrition in future datasets (Russel, 2013; Speer et al., 2019). The present chapter evaluates research on employee attrition in the organizational sciences (referred to as *turnover research*) from the lenses of explanation and prediction.

Theoretical Models

Given the large organizational impact of voluntary employee turnover, it comes as no surprise that turnover has been an active area of research since the early 1900s. During the first part of the 20th century, interest was mostly driven by high levels of turnover in US manufacturing and focused primarily on measuring individual differences (Bolt et al., 2022).

March and Simon (1958) are credited as having developed the first formalized model of voluntary employee turnover, which they published in their book, *Organizations*. Their theoretical model proposed that a combination of individual, organizational, and external factors influenced voluntary employee turnover. Specifically, they pointed to the *desire to leave* and the *ease of leaving* as key turnover antecedents. Their research laid the foundation for a long research tradition that focused on understanding the psychological processes leading up to employee attrition.

Mobley's (1977) Intermediate Linkages Model advanced the March and Simon tradition by theorizing a 10-step process through which job satisfaction influences turnover decisions. Job dissatisfaction was proposed to initiate a sequence of cognitive and behavioral steps (evaluation of one's existing job, job satisfaction or dissatisfaction, thoughts of quitting, attitude toward searching, attitude toward quitting, intention to search for alternatives, search for alternatives, evaluation of alternatives, comparison of alternatives to one's present job, and intention to quit) that result in either staying or quitting. By laying out these cognitive and behavioral steps, this work provided a theoretical taxonomy which identified both latent psychological variables and measurable behaviors as turnover antecedents.

Hom and Griffeth's (1991) work marked a pivotal moment in turnover research methodology. Their (1991) paper used *structural equation modeling* (SEM) to empirically test Mobley's (1977) Intermediate Linkages Model. SEM is a family of statistical approaches that combines factor analysis and linear regression for theory testing (Williams et al., 2009). Factor analysis constitutes the first step of testing a structural equation model. Factor analysis (e.g., confirmatory factor analysis, exploratory factor analysis, principal components analysis) is used to validate self-report survey scales by assessing the degree to which responses on survey items

fit together. Different "factors" are formed from the items, each representing a latent psychological construct that is being indirectly measured by responses to survey items. First, exploratory or confirmatory factor analysis (EFA, CFA) is used to validate the measurement model, assessing the degree to which responses to survey items accurately reflect the hypothesized latent constructs. Following this step, the *structural model* is evaluated, which tests the strength and direction of the hypothesized relationships with linear regression (Anderson & Gerbing, 1988).

SEM allowed researchers to answer questions about the mechanism and the underlying processes by which psychological predictors influenced attrition outcomes. SEM provided researchers with a formalized statistical framework to test the types of models they had theorized for years, including mediation and moderation hypotheses. Mediation models estimate the impact of an antecedent X variable on a consequent Y variable through an intermediate mediating variable M, answering the question of how an antecedent like job satisfaction impacts attrition outcomes (Hayes, 2013). Moderation models estimate the impact of an additional antecedent variable W in combination with the existing antecedent variable X on a consequent Y variable, answering the question of *when* or *under what circumstances* job satisfaction impacts attrition outcomes, for example. Moderation and mediation hypotheses are still frequently tested in attrition research (Bolt, 2022) given their ability to help explain causal processes and intervening factors.

Compared to regression, SEM puts less emphasis on explaining variance in the outcome with R^2 and puts more emphasis on model fit and the strength of paths between variables (Kline, 2012; Tanaka; 1993).While the measures Hom and Griffeth (1991) employed were developed in earlier work (Hom, Griffeth, & Sellaro, 1984), the 1991 study tested Mobley's Intermediate

Linkages Model using SEM. SEM, with its focus on causal relationships and goodness-of-fit indices, provided a framework for understanding the interrelations among turnover antecedents. By introducing SEM as a methodology, Hom and Griffeth's (1991) study influenced how organizational researchers tested turnover theories. Specifically, it demonstrated a method which does not evaluate the quality of a model based on whether or not it is capable of predicting future behaviors. Rather, the SEM approach, which uses goodness-of-fit indices, evaluates the quality of the model based on whether the size or direction of coefficients in the observed data match those that were implied by the theory. Hom et al. (2017) retrospectively noted that, shortly after structural equation modeling became popularized, "[...] SEM users became more interested in explaining *covariances* among exploratory constructs than variance in turnover" (Hom et al., 2017, p. 535).

In the years following Hom and Griffeth's (1991) influential paper, researchers continued to theorize the processes leading up to attrition. Lee and Mitchell (1994) introduced the Unfolding Model as a departure from the March and Simon (1958) framework, which had emphasized job dissatisfaction as the key driver of turnover. Instead, the Unfolding Model proposed that critical events, or "shocks," could independently prompt employees to reevaluate their employment situation. It offered four decision paths, only one of which involved job dissatisfaction, suggesting that elaborate cognitive deliberation might not always precede turnover.

Despite its innovative taxonomy, the Unfolding Model remained explanatory and retrospective, focusing on categorizing why employees left rather than predicting who would leave. Retrospective analyses, such as Lee et al.'s (1996) study of nurses who had recently quit, validated the model's paths and highlighted the prevalence of shocks. While this work provided

valuable insights into turnover mechanisms, its reliance on retrospective interviews offered limited applicability for forecasting turnover in future contexts. Moreover, the concept of a shock itself is inherently subjective and context-dependent, making it difficult to measure prospectively. A job offer might constitute a shock for one employee but not for another, and the unfolding model provides little guidance on how to operationalize such constructs in a way that generalizes across contexts.

Modern Turnover Research and Theory

The refinement and development of theoretical models persisted into the 21st century, with a growing emphasis on understanding why employees *stay* rather than why they leave (Hom et al., 2017). A prominent framework in this area is Job Embeddedness Theory, which posits that retention is influenced by an individual's embeddedness in their family, community, and occupation (Mitchell et al., 2001). The theory identifies three key dimensions: fit, the compatibility between an employee and their organization; links, the number of connections an individual has within their organization or community; and sacrifice, the perceived costs—both material and psychological—of leaving the job.

As with earlier theories, Job Embeddedness Theory expanded the scope of turnover research by introducing new psychological predictors (Bolt et al., 2022). However, many of these constructs, while theoretically compelling, are challenging to measure in practical contexts. For example, assessing an employee's community links would be both impractical and invasive. Even if these constructs were measurable, they are associated primarily with retention rather than attrition. From a predictive standpoint, retention is a low-priority outcome due to its high base rate, making it less useful for forecasting turnover trends or guiding workforce interventions.

The 21st century turnover literature has also been marked by a significant shift toward using turnover intentions rather than actual turnover as the primary outcome variable (Bolt et al., 2022). In their review of 100 years of turnover research, Bolt et al. (2022) found that 66% of studies published between 2001 and 2019 used turnover intentions as the dependent variable, compared to only 22% of studies published between 1937 and 2001. This shift has had significant implications for the field, as turnover intentions are not synonymous with actual turnover, and their correlation varies widely depending on the context and measurement. The meta-analytic correlation between turnover intentions and turnover behavior has been estimated to be $r = 0.50$ and $r = 0.35$ by Steel and Ovalle (1984) and Griffeth et al., (2000), respectively. Furthermore, focusing on an attitudinal variable rather than a behavioral outcome has entrenched turnover research further into the explanatory tradition, emphasizing theory development at the expense of predictive applicability.

Though it is unclear, one potential reason for the shift from turnover outcome to turnover intentions may have to do with how models with binary or continuous criteria are statistically evaluated. Up until the 1990s, researchers frequently used ordinary least squares (OLS) regression (linear regression) rather than logistic regression to model attrition as a binary outcome (Husleid & Day, 1991; Hom et al., 2017). This is problematic due to the lack of fit between the assumptions of OLS regression and the nature of employee attrition data. Attrition is a binary outcome, usually recorded as a 0 for stayers and a 1 for leavers. OLS regression is intended to predict a *continuous* outcome variable, as opposed to a *binary* outcome variable, because it assumes a normal distribution of the outcome. OLS regression is estimated as follows:

Equation 1. *Ordinary Least Squares Regression*

$$\hat{Y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots$$

Where $\hat{Y}$ is the predicted score for a given observation, β_0 is the intercept, which represents the value of Y when all predictors are equal to zero, and β_1 and β_2 represent the beta coefficients for the two predictors, x_1 and x_2 (Kline, 2013). Thus, OLS regression produces a *continuous* predicted score which is a linear combination of continuous predictors.

In contrast, logistic regression produces log odds, or the predicted probabilities of group membership in a binary category (0 or 1) based on the values of the independent variables. Unlike linear regression, it assumes a binary distribution of outcomes (Tansey et al., 1996). Its equation is as follows:

Equation 2. *Logistic Regression*

$$log\left(\frac{P(y=1)}{P(y=0)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots$$

Where $log\left(\frac{P(y=1)}{P(y=0)}\right)$ is the probability that the dependent variable (y) belongs to one of two categories ($y = 1$ or $y = 0$). Note that the rest of the equation remains the same as the OLS equation, but with a slightly different interpretation. β_0 is the intercept, and the beta coefficients which follow it, β_1, represent the regression coefficients indicating the change in the log-odds of Y for every one-unit change in the corresponding X predictor, x_1.

This slight difference in model assumptions produces meaningful differences. Huselid and Day (1991) reanalyzed Blau and Boal's (1987) study, which had used OLS (linear) regression to examine the interaction between organizational commitment and job involvement in predicting turnover. Blau and Boal reported significant interaction effects, but when Huselid and Day applied logistic regression—the appropriate method—the interaction effects disappeared. This finding demonstrated that OLS regression, despite its widespread use, was ill-suited for binary outcomes like turnover, as it relies on assumptions of continuity and normality

that are violated with binary data. However, OLS regression had long been favored for its simplicity, interpretability, and the convenience of using R^2—a measure of the variance in the outcome explained by the model coefficients—as a key evaluation criterion (Kutner et al., 2004). The use of turnover intentions, rather than binary turnover, allows researchers to continue to use OLS regression rather than logistic regression.

Increased use of turnover intentions as a dependent variable has both methodological and substantive consequences. Turnover intentions, while valuable as an explanatory construct, are inherently less actionable than actual turnover predictions for workforce planning. Their weaker and variable correlation with actual turnover further limits their utility for predicting behavioral outcomes. By prioritizing intentions over behavior, the field has reinforced its focus on explanation, sidelining opportunities to build predictive models that could forecast turnover with practical relevance.

Compared to OLS, logistic regression is better suited for prediction. Logistic regression produces expected attrition probabilities, which can be compared to actual attrition outcomes. The difference between the predictions made by the model and the outcomes observed constitute model performance. In contrast, an OLS regression model with turnover intentions as the dependent variable would produce expected values of the latent variable, turnover intentions.

Moving Forward: Applied Attrition Modeling

The historical reliance on explanatory methods like SEM, mediation, and moderation has shaped turnover research around understanding why employees leave rather than if they will leave. These methods have provided valuable insights into causal mechanisms, elucidating relationships among predictors, mediators, and outcomes. For example, SEM has advanced our understanding of latent constructs, while mediation and moderation models have helped clarify

how turnover antecedents interact with one another. This explanatory tradition has produced a wealth of knowledge and a robust library of predictors, offering a foundation for predictive analytics.

The practical challenge of predicting which employees will leave and when remains underexplored. In practice, people analytics teams in organizations are increasingly using predictive methodologies—referred to as *applied attrition modeling*—to address this challenge (Speer et al., 2019). These approaches prioritize forecasting accuracy and actionable insights, using statistical methodologies like logistic regression and modern machine learning (ML) algorithms. Speer et al. (2019, 2024) have significantly contributed to introducing machine learning (ML) approaches to attrition modeling in organizational research. Their 2019 publication outlined key considerations for applied attrition modeling and provided an overview of both traditional methods, such as logistic regression, and modern ML techniques. In a subsequent study, Speer (2024) demonstrated the application of random forests—a modern ML method discussed later—for attrition modeling, showing that it outperformed logistic regression. The 2024 study also focused on mitigating adverse impacts associated with ML-based attrition modeling. Despite the potential of these contributions, as of May 2024, Speer et al. (2019) had only 44 citations, with just two applying ML to attrition modeling: one a dissertation predicting attrition among federal STEM workers (Pasquarella, 2023) and the other a self-citation (Speer, 2024).

Meanwhile, computer sciences disciplines have begun leveraging our extensive turnover research library to develop predictive models, demonstrating the potential of machine learning in attrition modeling (Akasheh et al., 2024). Akasheh et al. (2024) conducted a systematic review of applied attrition modeling studies using machine learning (ML) published between 2012 and

May 2023, revealing that only 3 of the 52 reviewed papers appeared in organizational science journals: Human Resource Management Journal (Yuan et al., 2021), Management Research Review (Rombaut & Guerry, 2017), and Journal of Management Analytics (Wang & Zhi, 2021). These studies showcased a range of approaches, from Yuan et al.'s (2021) comparison of methods for modeling turnover intentions with grouped data, to Rombaut and Guerry's (2017) demonstration that archival features like age, sex, and seniority could predict attrition without survey data, and Wang and Zhi's (2021) evaluation of a custom ensemble algorithm on simulated datasets. Despite these contributions, most research on applied ML for attrition modeling originates from computer science disciplines, as Akasheh et al. highlighted in their review. These works typically compare ML algorithm performance on datasets comprising human resource information system (HRIS) data (e.g., attendance, salary), passive data (e.g., hours worked), and less frequently, self-reported survey data (e.g., job satisfaction, pay satisfaction).

In summary, research on applied attrition modeling with ML is disproportionately being conducted by researchers outside of the organizational sciences. Furthermore, most studies reviewed by Akasheh et al. (2024) (61%) used fictitious datasets like the IBM HR dataset (29%), raising concerns about the generalizability of findings. As of writing this dissertation, there are no published empirical studies which compare the performance of ML algorithms for applied attrition modeling with real-world HRIS, self-report, and performance data. The available studies use just one or two data types, which are often simulated.

It is now time to balance the field’s strong explanatory tradition with predictive methods. Explanatory methodologies and research findings have enabled practitioners to identify possible reasons why employees voluntarily exit the organization. However, there is an organizational question that has been largely under addressed: regardless of why, *how many* employees, and

which employees, are going to leave in a given period of time? This challenge is especially salient in high-turnover industries like customer service. Logistic regression provides an early bridge, accommodating the binary nature of turnover while prioritizing accuracy. Machine learning techniques build on this foundation, integrating diverse data sources and complex patterns to enhance predictive utility. By complementing explanatory research with applied attrition modeling, turnover research can evolve to address organizational needs, providing both theoretical depth and practical solutions for workforce planning.

CHAPTER 3

THE PRESENT STUDY

Overview: Transitioning from Explanation to Prediction

Predictive research differs from explanatory research in its goals, analytical tools, and approaches to data preparation (Shmueli, 2010). As previously noted, the research question drives the choice of analytical tools, and transitioning from explanation to prediction requires adopting statistical methodologies tailored to forecasting needs. The current chapter describes the ways in which machine learning (ML) approaches lend themselves to predictive endeavors.

Despite the growing interest in predictive methods, academic research comparing ML approaches to applied attrition modeling remains limited. While dozens of ML algorithms are available, studies in the computer sciences often test numerous methods without providing clear rationales for their selection or exploring the reasons behind performance differences. To address this gap, this study narrows the focus to a subset of algorithms which are well-suited for attrition modeling. These include regularized regression techniques (elastic net, lasso, and ridge regression) and tree-based methods (random forests and extreme gradient boosting, or XGBoost). Table 1 summarizes these methods, which are discussed in detail in this chapter.

Table 1. *Selected Modern Machine Learning and Traditional Methods*

Method	Technique	Ensemble Method	Base or Foundational Model(s)
Random Forest	Tree-based	Bagging	Decision Trees
Elastic Net Logistic Regression	Regularized regression	Hybrid model	Ridge regression Lasso regression Logistic regression

XGBoost	Regularized tree-based ensemble method	Boosting	Decision Trees, Ridge Regression
Logistic Regression	Regression	NA	NA

Methodological Rationale

Before selecting the methods for this study, I conducted a comprehensive literature review to examine which predictive modeling techniques have been used for applied attrition modeling and similar use cases. This review spanned the organizational sciences, psychological sciences, and computer science research. The goal was to identify methods which are well-suited for the types of data used in attrition models (e.g., categorical, continuous tabular data), methods which can be easily compared to one another, and methods which have demonstrated strong performance on similar tasks.

From the organizational sciences, key contributions include Landers et al. (2023), Putka et al. (2018), and Speer et al. (2019, 2024). Putka et al. (2018) provide an overview of several modern ML algorithms, including lasso regression (LARS), elastic net regression, decision trees (CART), support vector machines, and stochastic gradient boosted trees, which they compared to ordinary least squares regression and forward stepwise regression. Using a sample of biodata and performance data from the U.S. Army's Reserve Officer Training Corps, their study aimed to compare the effectiveness of ML methods to traditional methods in predicting job performance. Note that their outcome was a continuous variable rather than binary variable, which limits the generalizability of their findings to the present work. Overall, they found that all machine learning models yielded higher predictive validity than traditional OLS regression, forward stepwise regression, and decision trees. They also concluded that, for this particular use case,

linear regression methods were sufficient. Methods which accounted for more complex relationships such as nonlinearities and interactions did not substantially outperform linear methods like elastic net. Moreover, modern regression approaches (lasso and elastic net regression) outperformed stepwise regression most markedly at lower sample sizes.

Building on the work of Putka et al. (2018), Landers et al. (2023) conducted a simulation study to examine the use of several modern ML algorithms to predict job performance, focusing on adverse impact and the optimal number of features. Their overarching goal was to determine whether using individual items or scale composite scores would yield superior predictive validity across various ML methods. They compared elastic net regression, random forest, lasso regression, linear (OLS) regression, support vector matrices, XGBoost, deep neural networks, and k-nearest neighbors. Similarly to Putka et al. (2018), Landers et al. (2023) found that modern ML algorithms were superior to OLS regression across sample sizes and demonstrated the largest advantage in smaller sample sizes and when item wise prediction was used rather than scale wise prediction. Elastic net regression and lasso regression, which utilize regularization to drop uninformative predictors, performed especially well across conditions. Interestingly, random forest outperformed the other tree-based method, XGBoost, when the ratio of predictors to observations was high. Overall, performance prediction improved with ML in data-sparse conditions.

Speer and colleagues (2019) provided an in-depth guide on attrition modeling, effectively translating turnover research into practical steps for attrition modeling. This work did not test attrition models, but discussed several relevant methodologies, including logistic regression, decision trees, ensemble models, and survival analysis. Survival analysis is a class of methodologies first used in medical research to predict the amount of time it will take for an

outcome to occur, which has also been applied to turnover research but to a much lesser extent (Allen et al., 2014; Morita et al., 1989). Survival analysis, including Cox regression, produces estimates of the probability of departure at various future time points and regression weights indicating the relative impact of each feature on time to departure, with the assumption that all observations will depart at some point (Speer et al., 2019). However, applications of survival analysis in the turnover literature have been sparse. In their review of turnover research from 1958-2010, Allen et al. (2014) found that survival analysis accounted for only 11% of studies, with many of these using it for exploratory purposes. For instance, Mattox and Jinkerson (2005) applied it to quantify the relative effects of predictors on estimated survival times.

Speer et al. (2024) used a sample of call center employee data containing archival HRIS variables and performance variables to predict turnover over a 6-month period using random forests. They also tested logistic regression, elastic net regression, and a deep neural network at the request of the reviewers but note that random forest exhibited the best performance out of these methods. The outcome variable was multicategorical; algorithms predicted the probability of staying, voluntarily leaving, and involuntarily leaving. Overall, random forests were best at predicting involuntary attrition and performed well overall.

Outside of Speer's work, I found two works pertaining to applied attrition modeling which were published in conference proceedings. The first was Vahnove et al. (2023), who performed a meta-analysis on research which used supervised machine learning models to predict categorical human resource management outcomes (Vanhove et al., 2023). The complete meta-analysis was not yet published at the time I was writing this paper. However, the authors did summarize a few of the findings, which suggested that boosting and random forest algorithms consistently outperformed other algorithms (discriminant analysis, logistic regression,

Bayesian algorithms, K-nearest neighbors, neural networks, and support vector machines) across classification performance indices.

The second unpublished work was that of Shewach et al. (2024), published at the SIOP conference that year. Their simulation study compared the performance of logistic regression, elastic net logistic regression, and decision trees (C5.0) across varying degrees of class imbalance, different optimization criteria, and different sampling techniques. They note that they did not choose to include random forest and gradient boosting because the data simulation method did not incorporate complex feature relationships that would allow these models to outperform simpler methods. Their work did not compare the performance of these methods to one another; rather, they focused on comparing the relative performance of the methods across different conditions of the data using a vote-counting approach.

Finally, I evaluated results from a systematic literature review conducted on the past decade (2012-2022) of research on ML techniques for predicting employee turnover (Akasheh et al., 2024). In summary, the review found that supervised algorithms were used far more frequently than unsupervised algorithms (e.g., k-nearest neighbors), which were only used by 2 of the 52 papers included in the review. In terms of frequency, the review identified the following methodologies in order from most frequently tested to least frequently tested: support vector machines (n = 2), naïve bayes (n = 2), decision trees (n = 3), neural networks (n = 4), logistic regression (n = 4), "other" supervised classifiers (n =12), XGBoost (n = 5; one paper used a different boosting algorithm called CatBoost), and random forest (n =17). Classifiers in the "other" category included ad-hoc hybrid methods and methods which made slight adjustments to popular algorithms. In terms of predictive performance, the review suggests that

no single algorithm universally dominated, but tree-based ensembles like XGBoost and random forest were frequently among the strongest performers.

Given these findings from my review of the literature, I selected a set of algorithms that have demonstrated a high level of classification and predictive accuracy in similar applications, and which allow for meaningful comparisons across method types while also capturing a range of complexity in model structure. Logistic regression serves as a traditional benchmark as it is frequently used in turnover research, making it an essential point of comparison. Decision trees (CART) were included as a simple, interpretable tree-based method that forms the foundation for more complex ensemble models. Lasso regression and ridge regression were selected as base regularized regression methods, as they serve as key components of the chosen hybrid models, elastic net and XGBoost. Finally, I included three modern ML methods: XGBoost, elastic net, and random forest. All three of these methods were chosen for their superior performance and popularity across the reviewed studies. Moreover, XGBoost was selected as it integrates tree-based modeling with ridge-like regularization, elastic net was selected because it combines the benefits of lasso and ridge regression, and random forest was selected because it is an ensemble of decision trees.

To manage the scope of this work, several commonly used methods were intentionally excluded. Unsupervised methods, such as k-nearest neighbors, were excluded because they are not widely used in attrition modeling and tend to underperform on high-dimensional datasets. Neural networks (deep learning) were not included because they require much more tuning and computation compared to tree-based algorithms, such as XGBoost, and because neural networks significantly underperform such methods on tabular (i.e., Excel) data (Shwartz-Ziv & Armon, 2022). Survival analysis, while a valuable tool in attrition research, was excluded because it does

not produce probability estimates in a manner that allows for direct comparison to other ML models. Nevertheless, future research on the application of survival analysis to applied attrition modeling is needed, as it remains underutilized despite its potential (Morita et al., 1989; Somers & Birnbaum, 1999; Speer et al., 2019). Overall, the selected models build upon one another in a structured way, allowing for direct evaluation of how specific modeling choices (e.g., regularization, ensembling) impact predictive accuracy. Table 2 provides a summary of this section, some of which reflects the findings of Akasheh et al. (2024).

Table 2. *Methodologies Considered for Inclusion*

Method	Citation(s)	Included? Y/N	Rationale
Logistic regression	Rombaut and Guerry, (2017)*; Setiawan et al., (2020)*; Ozdemir et al., (2020)*; Najafi-Zangeneh et al., (2021)*, Vanhove et al. (2023); Speer et al. (2019; 2024); Shewach et al., (2024)	Y	Base method for comparison.
Lasso regression	Landers et al. (2023); Putka et al. (2018)	Y	Builds on logistic regression, base method for comparison to elastic net regression.
Ridge regression		Y	Builds on logistic regression, base method for comparison to elastic net regression.
Elastic net regression	Landers et al. (2023); Putka et al. (2018); Speer et al. (2019; 2024), Shewach et al., (2024).	Y	Builds on lasso and ridge regression; demonstrated high performance in similar applications (e.g., Landers et al., 2023; Putka et al., 2018).
Decision trees	Naz et al., (2022)*; Kang et al., (2021)*. Shewach et al., (2024)	Y	Base method for comparison to random forest and XGBoost.
XGBoost	Punnoose and Ajit (2016)*, Ain and Nayyar (2018)*, Zhao et al. (2018)*, Jhaver et al. (2019)*, Tharani and Raj (2020)*, Putka et a. (2018) (Stochastic Gradient Boosted Trees).	Y	Ensemble method which incorporates ridge regularization and tree-based classifiers. Commonly used in the attrition modeling literature. Demonstrated high performance in similar applications.
Random forest	Tama and Lim (2021)*; Bao et al. (2017)*, Alamsyah and Salma (2018)*, Sisodia et al. (2018)*, Gao et al. (2019)*, El-Rayes et al. (2020)*, Jain et al. (2020)*, Cai et	Y	Ensemble method which incorporates tree-based classifiers. Commonly used. Demonstrated high performance in similar applications.

	al. (2020)*, Hossen et al. (2021)*, Wild Ali (2021)*, Joseph et al. (2021)*, Hebbar et al. (2018)*, Wang and Zhi (2021)*, Jain and Jana, (2021)*, Krishna and Sidharth (2022)*, Raza et al., (2022)*, Alzate Vanegras et al. (2022)*, Landers et al. (2023); Speer et al. (2019, 2024)		
Neural networks	Srivastava and Nair, (2018)*; Meng et al., (2019)*; Teng et al., (2021)*; Al-Darraji et al., (2021)*; Srivastavaand Eachempati (2021)*; Alharbi et al., (2023)*, Landers et al. (2023), Speer et al. (2024).	N	Computationally intensive; underperforms tree-based methods on tabular data. Demonstrated to underperform random forest in applied attrition modeling context (Speer et al., 2024).
Support vector machines	Dolatabadi and Keynia, (2017)*; Yiğit and Shourabizadeh, (2017)*, Landers et al. (2023); Putka et al. (2018)	N	Performs poorly with high dimensional datasets (Yang et al., 2021)
k-Nearest neighbors	Fan et al. (2012)*, Avrahami et al. (2022)*, Landers et al. (2023)	N	Performs poorly with high-dimensional datasets (Saxena et al., 2017); Performed poorly across conditions in Landers et al. (2023)
Bayesian methods (naïve bayes)	Fallucchi et al., (2020)*; Thompson et al., (2022)*	N	Assumes that predictor features are independent (Jadhav & Channe, 2016).
Survival analysis	Speer et al., (2019)	N	Results are not easily comparable to ML results.
OLS regression	Landers et al. (2023); Putka et al. (2018)	N	Not suitable for binary data.

Note: Citations with an asterisk* were included in Akasheh's (2024) systematic review.

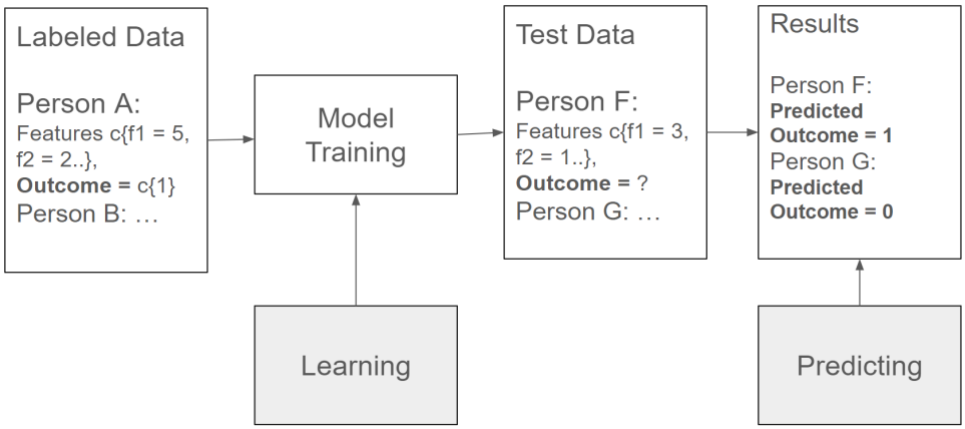
Supervised Classification Methods

Attrition modeling typically relies on *classification* algorithms (Speer et al., 2019). Classification algorithms estimate the probability that observation *i* belongs to the positive (1) or negative (0) class. There are two main categories of classification algorithms: supervised and unsupervised. At a high level, supervised ML algorithms work by learning patterns of X → Y relationships in a training dataset where the desired response Y is labeled. The model then forms predictions from learned patterns of X → Y relationships, which are applied to an out-of-sample dataset to make predictions on data where the outcome is unknown (Choudhary & Gianey,

2017). Based on learned patterns among predictors/independent variables, referred to as *features*, trained models (classifiers) predict group membership on unseen data. Figure 1 visualizes this process. Out-of-sample validation (cross validation) is typically not undergone in the turnover literature. However, it is a critical process which helps lend evidence for the model generalizability.

In contrast, unsupervised ML techniques make associations between features based on patterns in the data. Principal component analysis and factor analysis are examples of unsupervised techniques (Kuhn & Johnson, 2013). Almost all (96%) of computer science research on attrition modeling has been conducted using supervised ML algorithms, while the remaining 4% have used unsupervised ML algorithms (Akasheh et al., 2024). Compared to unsupervised ML algorithms, supervised ML algorithms are better suited for attrition modeling because they learn patterns from organizational data. Knowing the outcome in the training data provides important information that guides the algorithm in its predictions on the test data, giving it a competitive advantage over unsupervised methods. In addition, unsupervised methods are usually outcome-agnostic. A factor analysis, for example, is performed on theorized predictors, with the goal of understanding relationships between different factors rather than between factors and the outcome (Kuhn & Johnson, 2013).

Figure 1. *Training and Testing a Supervised Machine Learning Algorithm*



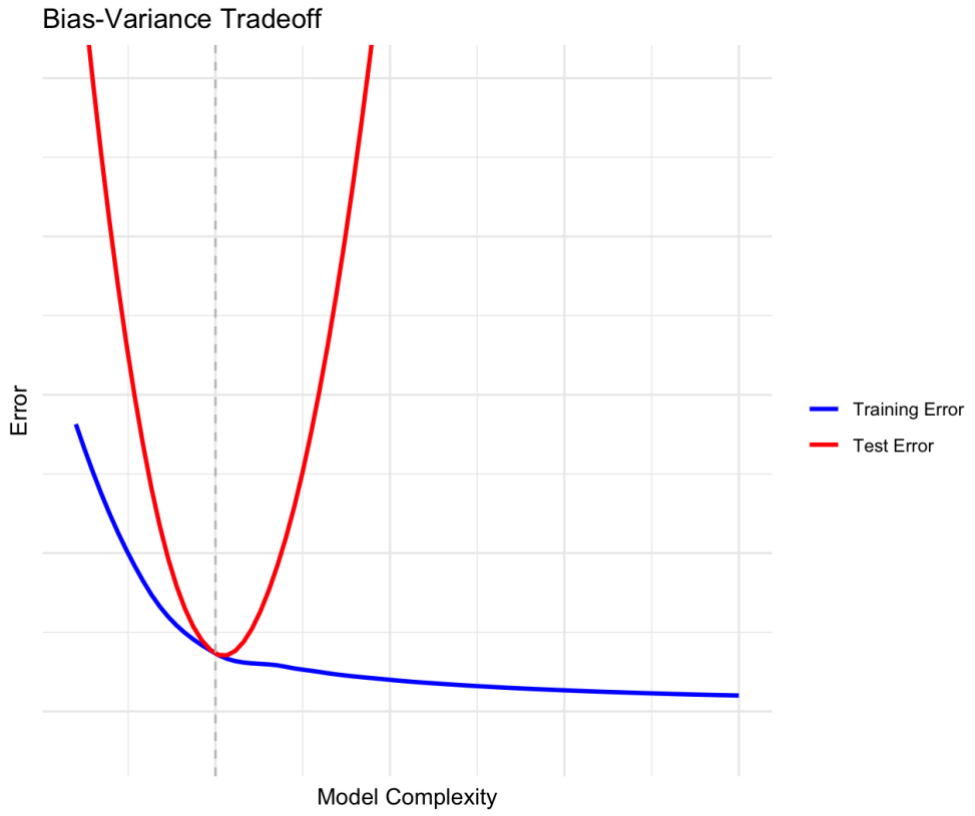
Logistic Regression

Though not always considered a ML algorithm, logistic regression serves as the foundational model for many ML algorithms. Despite being a traditional statistical approach, logistic regression can be deployed similarly to modern ML algorithms when enhanced with cross-validation. Cross-validation is an analytic method which applies a trained model to unseen data. A logistic regression model which has been "trained" or "run" on one dataset where the turnover outcome is known can be used to make predictions on another dataset in which the outcome is not known. This process essentially tests the validity of a model on unseen data and evaluates the model's quality based on the accuracy of its predictions, rather than the strength of its beta coefficients, using a confusion matrix, a 2 by 2 table displaying counts of true positive, true negative, false positive, and false negative predictions. Thus, one of the key issues with current applications of regression models in turnover research is their lack of cross-validation. Models are not tested; they are only trained and appear to work best when they are overfit. It is no surprise, therefore, that researchers in the psychological sciences often find it difficult to replicate other researchers' findings (Yarkoni & Westfall, 2017).

Modern ML algorithms also differ from logistic regression in how they balance *bias* and *variance* (Yarkoni & Westfall, 2017). Bias and variance both constitute sources of error in a model. A model with high bias is less flexible, applying more stringent rules about the form of the data. With too high of a bias, a model can miss meaningful patterns, or *underfit* the data. In contrast, a model with too high a variance will over capitalize on chance variations and outliers in the training data, resulting in poor generalization to the test data (Gupta et al., 2022). This is a phenomenon referred to as *overfitting* (Briscoe & Feldman, 2011). A model with too high variance is too flexible; it learns the idiosyncrasies of the data, not the underlying pattern, which becomes apparent when the model is applied to a different dataset.

Models with too many predictors, particularly irrelevant and redundant predictors, can overfit the data. Figure 2 provides a conceptual visualization of the bias-variance tradeoff, where model complexity, operationalized by the number of predictors in the model, is plotted against error. Low model complexity represents a high bias:variance ratio, and high model complexity represents a high variance:bias ratio. As model complexity increases, classification errors on the training data decrease as the data becomes overfit. The dotted line represents the optimal tradeoff, where classification error on the test data is the lowest.

Figure 2. *Visualizing the Bias-Variance Tradeoff*



Regularized Regression: A Solution to Overfitting

Logistic regression, as a relatively simple model, is characterized by high bias and low variance. However, high bias can also arise from challenges associated with the predictors in the model, such as collinearity and overfitting due to an excessive number of predictors. Collinearity, where predictors are strongly correlated with one another, inflates the standard errors of estimated coefficients, thereby reducing their reliability and interpretability. Overfitting, on the other hand, occurs when the model includes too many predictor variables, leading to unstable parameter estimates that can vary drastically with minor changes in the data, such as the removal of features or the addition of new observations. These issues collectively undermine the validity and generalizability of the logistic regression model.

Regularized regression is an advanced extension of logistic regression designed to address key challenges such as overfitting, multicollinearity, and high-dimensional data. As a class of algorithms designed to address overfitting, regularized regression introduces penalties to the model's loss function (which seeks to identify the model parameters that maximize the likelihood of the observed data) to constrain the size of the coefficients, thereby stabilizing estimates and improving generalizability. By introducing penalty terms to the logistic regression loss function, regularized regression methods reduce model complexity and optimize the bias-variance tradeoff. This balance ensures better generalization to unseen data. Three primary approaches are discussed: ridge regression (Hoerl & Kennard, 1970), lasso regression (Tibshirani, 1996), and elastic net regression (Zou & Hastie, 2005). These methods extend logistic regression while maintaining its foundational assumptions of binary outcomes and independence of observations. They differ from standard logistic regression by explicitly addressing multicollinearity and irrelevance among predictors, enhancing predictive accuracy and interpretability.

Ridge Regression

Ridge regression, first introduced by Hoerl and Kennard (1970) for linear regression, was one of the earliest methods in this class. Le Cessie and Van Houwelingen (1992) extended ridge regression to logistic regression, adapting it for binary outcome data. Ridge regression, also known as L2-regularized logistic regression, minimizes overfitting by shrinking coefficients of less predictive features towards zero without eliminating them entirely (Hoerl & Kennard, 1970). Ridge regression is useful in cases where there are collinear features by reducing the variance in coefficient estimates. This is a strong advantage over logistic regression, which is known to

produce unstable coefficient estimates when collinearity is present. Like logistic regression, ridge regression keeps all features in the model.

Lasso Regression

Lasso (least absolute shrinkage and selection operator) regression builds upon ridge regression by addressing one of its key limitations: the inability to perform variable selection (Tibshirani, 1996). While ridge regression shrinks all coefficients toward zero, it does not eliminate any, meaning irrelevant or redundant features remain in the model, albeit with smaller weights. Tibshirani (1996) introduced lasso regression to solve this problem. By replacing ridge's L2-norm penalty with an L1-norm penalty, lasso regression retains the ability to suppress coefficients of less predictive features while also setting some coefficients exactly to zero. This property makes lasso regression particularly well-suited for high-dimensional problems where many predictors may be irrelevant.

Both lasso and ridge regression have advantages for ML applications compared to logistic regression. Logistic regression produces coefficients that are tailored to the present sample, which is useful for understanding feature relationships within a one-shot analysis. However, these coefficients do not generalize well to unseen data (Rosenbusch et al., 2021). By reducing the number of predictors and/or the strength of model coefficients, lasso and ridge regression achieve better predictive accuracy on unseen data. In other words, algorithms can improve their performance on unseen data by reducing their performance on training data. This concept is central to the bias-variance tradeoff, which is present in all applications of ML (James et al., 2023).

Elastic Net Regression

The most recently developed regularization method – elastic net regression – (LARS – EN; Zou & Hastie, 2005) balances the L1 penalty used in lasso regression and the L2 penalty used in ridge regression (Putka et al., 2018). Additionally, elastic net has the capability to retain groups of highly correlated features (Zou & Hastie, 2005).

Consider a model with 10 features, three of which demonstrate a strong correlation with one another and with the outcome. Whereas the lasso regression algorithm would arbitrarily choose two of the three features to discard, the elastic net algorithm would group these features together as a factor. This allows for the capturing of a wider net of variance while still maintaining lower variance than ridge regression. Moreover, the elastic net helps reduce the complexity of the model by reducing a larger number of features to a single factor that represents them.

Elastic net regression with OLS estimation was developed by Zoe and Hastie (2005) and has since been adapted for use in logistic regression problems (Ren et al., 2022; Shiomi et al., 2022). The regularization term $\Pi(\beta)$ used in logistic elastic net regression is expressed as:

Equation 3. *Elastic Net Regularization Term*

$$\Pi(\beta) = \sum_{i=1}^{n_p} \quad \alpha|\beta_i| + \frac{1-a}{2}\beta^2{}_i$$

Where n_p represents the total number of features, and α is the hyperparameter that balances the λ_1 norm term $|\beta_i|$, and the λ_2 norm term $\beta^2{}_i$. A value of $\alpha = 1$ would produce a lasso regression model and a value of $\alpha = 0$ would produce a ridge regression model. Various values of α between 0-1 are tested during the hyperparameter tuning process. The λ_1 penalty shrinks coefficients of uninformative features to 0, while λ_2 distributes the weight of coefficients across features. This process reduces the risk of the model capturing spurious patterns in the data

and overfitting, thereby enhancing generalization of the model and out-of-sample performance (Sajaddian et al., 2021).

As a hybrid technique, elastic net regression provides an optimal balance between lasso regression and ridge regression. Elastic net can effectively handle data with correlated features, an important issue plaguing employee attrition data. Whereas ridge regression would choose only one of the correlated features arbitrarily, elastic net regression balances them by disturbing the L1 and L2 coefficient across the features (Zou & Hastie, 2005). This built-in feature balancing function makes elastic net a desirable algorithm for predicting employee attrition. In addition, elastic net's ability to handle groups of related variables, effectively creating factors, is highly desirable in contexts where several items from the same scale are used (Putka et al., 2018).

Hyperparameter Tuning

Hyperparameter tuning is the process of optimizing different settings in the algorithm to improve predictive performance (Hastie et al., 2009). This iterative process allows ML models to adapt flexibly to the data, balancing bias and variance for better generalization to unseen samples. Unlike traditional regression approaches, which focus on parameter estimation for theoretical validation, regularized regression seeks to find a model that best fits the data and makes accurate predictions.

Regularized regression models undergo hyperparameter tuning to determine the optimal lambda (L1 or L2) and mixture (alpha) values. The penalty parameter, lambda, controls the strength of regularization by shrinking the regression coefficients toward zero. Larger values of lambda increase the regularization effect, simplifying the model and reducing the risk of overfitting, while smaller values allow for more complex models with less regularization. For

lasso regression, the regularization is entirely L1-based, which can shrink some coefficients to exactly zero, effectively performing feature selection (selecting useful features and removing redundant or irrelevant features). In contrast, ridge regression uses L2 regularization, which shrinks coefficients without driving any to zero. During hyperparameter tuning, different values of lambda are tested to find the optimal level of regularization. The alpha parameter, used in elastic net regression, determines the balance between L1 and L2 penalties, where an alpha of 0 corresponds to ridge regression and an alpha of 1 corresponds to lasso regression. Tuning alpha allows elastic net models to achieve an optimized balance of L1 and L2 regularization for a specific dataset.

Regularized regression has not received attention in the computer science attrition literature and was not included in Akasheh et al.'s (2024) review. However, it has received attention in two publications both of which used regularized regression methods to explain variance in future job performance from psychometric tests. Putka et al. (2018) used modern ML algorithms and OLS regression to explain variance in performance from biodata in a sample of U.S. Army Reserve Trainees, comparing the ML algorithms to each other and to OLS regression. They compared results (R^2) at the item- and measure-level, finding that modern ML methods consistently outperformed OLS regression. Similarly, Landers et al. (2023) also found that modern ML algorithms outperformed OLS regression in explaining variance in job performance from psychometric tests. Both papers note that the significant performance gains from regularized regression were due to its ability to exclude irrelevant features.

A notable difference between the studies by Putka et al. (2018) and Landers et al. (2023) and the present study is that they evaluated algorithmic classification performance in terms of variance explained (R^2) rather than predictive accuracy. Nonetheless, investigating regularized

regression for predictive applications is worthwhile due to its relevance in attrition modeling, where models often include numerous potential features.

Tree-Based Algorithms: Random Forest and Decision Trees

While regularized regression techniques provide solutions to mitigate bias in logistic regression by reducing overfitting and improving model generalization, they do not address some of the inherent structural limitations of the model, namely, assumptions of *linearity* and *independence of predictors*. Logistic regression assumes a log-linear relationship between predictors and the outcome, which may not hold in cases where interactions or non-linear dependencies exist in the data. The assumption that features independently contribute to the log-odds further constrains the model's flexibility in capturing complex patterns. Decision trees and other tree-based algorithms offer an alternative to the shortcomings of logistic regression.

Decision Trees: The Foundational Model

The decision tree, being a tree-based algorithm itself, is also the foundational model upon which other tree-based models are built. Decision trees (Breiman, 1984) work by breaking observations into smaller and smaller groups that are as homogenous as possible with respect to the outcome, resulting in a set of probabilities representing their likelihood of belonging to a particular group (Speer et al., 2019). Unlike logistic regression, decision trees do not rely on a pre-specified form, making them well-suited for capturing non-linear relationships and interactions without feature engineering (Breiman et al., 1984).

The most popular decision tree algorithm, *classification and regression trees* (CART; Breiman, 1984), uses binary recursive partitioning to repeatedly split groups of observations with similar standings on features. Decision trees are made up of nodes, branches, and leaves. The strongest predictor is used as the root node (Kotsiantis, 2007), which is the first feature by which

observations are split. Figure 2 visualizes a decision tree with tenure as the root node. The red branches represent critical values predictive of outcome variable with a value of 0, attrition, and green branches represent critical values of an outcome variable with a value of 1, retention. In other words, the tree creates branches based on critical values of tenure (less than one year or greater than one year, for example), but is not yet able to make a determination. From there, it forms two interior nodes, "pay satisfaction" and "average hours worked per week", and branches based on values of these two features. Thus, the algorithm identifies points at which a feature predicts an outcome and results in the purest nodes and splits the group into child nodes based on these points. The decision tree will stop at a predetermined number of samples in each node, maximum tree depth, or minimal improvement in node purity (Putka et al., 2018). The final nodes are referred to as *leaves,* which represent the outcomes or predictions for the target variable based on the paths taken through the tree from the root to leaf during training (Provost & Fawcett, 2013).

At each split, the algorithm selects the feature and threshold that maximize a splitting criterion, such as minimizing impurity. Common measures of impurity include the Gini index used in classification tasks and mean squared error (MSE) for regression. For classification, the impurity at a node is typically calculated using Gini Impurity:

Equation 3. *Gini Impurity*

$$Gini\ Impurity\ =\ 1\ -\ \sum_{k=1}^{K}\ \ {p^2}_k$$

Where p_k is the proportion of observation in class k at the node. Thus, if the split results in all observations in a node belonging to the same class, $p_k = 1$ for one class and $p_k = 0$ for another class, the sum of the squared proportions is 100%. Subtract this from 1 to get Gini

impurity of 0. Splits are chosen to reduce impurity in the node as much as possible, resulting in child nodes that are more homogenous than the parent (Breiman, 1984).

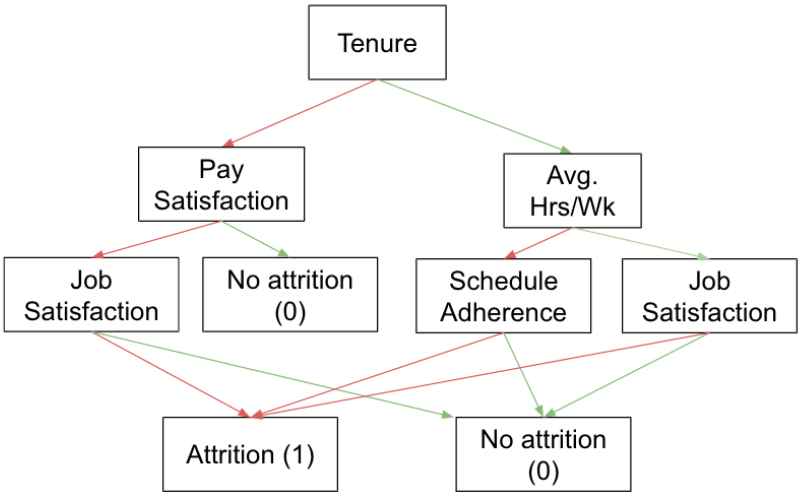
Notably, decision trees can handle different types of data, such as categorical and continuous data, in the same analysis (Louppe, 2015), making them suitable for attrition modeling. For example, categorical variables like team membership and continuous variables like job satisfaction can be included in the same model. Another aspect of decision trees that makes them valuable for attrition modeling is that they automatically implement feature selection to some degree. They do so through the determination of feature importance, which indicates the extent to which each feature contributes to node purity (Louppe, 2015). Thus, if a predictor does not contribute to the splitting of nodes, it is left out of the tree. In addition, decision trees are more robust to outliers compared to parametric tests like regression (Pargent et al., 2023). Decision trees can model nonlinear relationships by splitting the same feature many times, effectively addressing a key limitation of regression models (Putka et al., 2018). Additionally, interactions are captured natively by decision trees. In the example provided in Figure 2, the first split is based on the value of Tenure, and the second split is based on the value of Pay Satisfaction. By producing these splits, the tree is modeling the interaction between Tenure and Pay Satisfaction because the importance of Pay Satisfaction is conditional on the value of Tenure being below a given threshold (Scottifer, 2017). This effectively answers the question addressed in a moderation hypothesis: under what circumstance does Tenure influence Pay Satisfaction? Additionally, the model answers the question of mediation. We can see through the structure of the tree that Pay Satisfaction mediates the effect of Tenure on Job Satisfaction. These relationships are observed empirically through the modeling process, rather than being

hypothesized. Rather than fitting the data to a predefined structural model, the data *create* the structural model.

For CART decision trees, three key hyperparameters are tuned: cost complexity (cp), min_n, and tree depth (max_depth). The cost complexity parameter (cp) controls pruning by penalizing splits that do not sufficiently reduce the overall error. Higher values of cp result in simpler trees with fewer splits, reducing overfitting, while smaller values allow for more complex trees. The min_n parameter specifies the minimum number of samples required in a node for it to be split. Larger values of min_n prevent small, overly specific splits, promoting model generalization. The tree depth (max_depth) parameter limits the maximum number of levels in the tree. Shallow trees are less likely to overfit but may underfit the data, whereas deeper trees allow more granular splits at the risk of capturing noise.

Decision trees are non-parametric models that do not assume specific data distributions or predefined relationships between features and outcomes. They assume that predictors are related to the target variable, an assumption which is evident in the calculation of metrics like Gini impurity, which assess the quality of splits based on how well features separate classes. Decision trees rely on the presence of predictive features. If a feature lacks discriminatory power, the decision tree cannot use it to determine group membership, leading to poor classification performance. Moreover, unlike some regression-based models, decision trees cannot extrapolate to predict classes they have not encountered during training. For instance, a tree trained exclusively on positive class instances will fail to recognize or accurately predict negative class instances, as it lacks the necessary exposure to learn distinguishing patterns.

Figure 3. *Example Decision Tree for Attrition Prediction*



Random Forest: An Ensemble of Trees

Random forests are an ensemble method which combines the predictions of multiple decision trees. They use an ensemble technique called bagging, where multiple independent classifiers (trees) are trained, and the predictions from the independent classifiers are combined (Oshiro et al., 2012). The core principle of ensemble methods like random forests is that the performance of a set of many weak classifiers is usually better than the performance of a single classifier given the same training data (Sirikulviriya & Sinthupinyo, 2011). Thus, random forests combine multiple trees, effectively canceling out inefficiencies or biases of each individual tree to improve prediction and reduce the risk of overfitting (Berk, 2006).

First, trees are trained on random bootstrapped samples with replacement from the dataset. Next, features are randomly sampled, such that individual trees are tasked with splitting on a random subset of features. Finally, a vote is taken across trees to produce the final trained model. Through this process, the randomness introduced into the process provides a trained model that is more stable, more predictive, and more robust to overfitting compared to a single tree (Louppe, 2015; Pargent et al., 2023).

Random forests also inherently provide an out of sample error estimate during training. Because each tree is trained on a bootstrap sample, the samples not included in training can act as a test set for evaluating the performance of the model. This provides an unbiased estimate of the model's generalization error without requiring a separate validation set (Breiman, 1996; Breiman, 2001).

Thus far, the most applied method for the study of employee attrition in the computer science literature has been random forests (Breiman, 2001), which comprise 32% of papers reviewed by Akasheh et al., (2024). There are several reasons why random forests are well-suited for applied attrition modeling. Like decision trees, random forests also do not assume linear relationships between features and outcomes (Louppe, 2015). This makes random forest and tree-based methods appropriate for attrition modeling, given that relationships between antecedents and outcomes may not always be linear. For example, there is often a curvilinear relationship between job performance and attrition, such that very high and very low performers are more likely to leave the organization (Sturman et al., 2012). This lack of assumptions makes algorithms such as random forest powerful tools in predicting attrition, where there may be many complex interrelations between variables with each other and with the outcome.

Through careful hyperparameter tuning, such as adjusting the number of trees, the number of features to consider at each split, or the maximum depth of each tree, random forests can be further optimized. Number of trees specifies the number of trees in the first. More trees generally improve performance because they reduce variance by averaging predictors over more independent trees. However, beyond a certain point, adding more trees results in diminishing returns in accuracy and increases computational costs (Oshiro et al., 2012). The number of features considered for splitting determines the number of features which are evaluated to

determine a split. Smaller values reduce the correlation between trees, increasing diversity of the forest, but too small a value may reduce predictive ability (Breiman, 2001). Finally, maximum tree depth limits the "depth" of individual trees in the forest, or the number of times the tree can produce child nodes. Deeper trees capture more details about the data, but are more likely to overfit (Probst et al., 2019).

Extreme Gradient Boosting (XGBoost): An Ensemble Method with Regularization

XGBoost is a tree-based ensemble method that incorporates advanced regularization techniques to improve predictive accuracy and control overfitting (Chen & Guestrin, 2016). Unlike random forests, which use bagging to train trees independently, XGBoost employs a boosting framework. In boosting, classifiers are trained iteratively, with each successive tree focusing on correcting the errors made by the previous ones (Dong et al., 2020). Gradient boosting, the foundation of XGBoost, works by "learning" from mistakes at each step, iteratively adjusting weights assigned to observations to prioritize those that were misclassified (Bentéjac et al., 2021). Initially, all observations are given equal weights. After each iteration, weights are increased for incorrectly classified observations and decreased for those correctly classified, ensuring the model focuses on the hardest-to-predict cases (Hastie et al., 2009).

The iterative nature of gradient boosting allows the model to progressively improve performance until it reaches a predetermined number of iterations or when further improvements plateau (González et al., 2020). Building on traditional gradient boosted machines, XGBoost incorporates both L1 (lasso) and L2 (ridge) regularization in its objective function, penalizing overly complex models and reducing the risk of overfitting (Chen & Guestrin, 2016).

XGBoost relies on gradient descent as the optimization method to minimize a specific loss function, such as mean squared error (MSE) or log loss, at each iteration (Bentéjac et al.,

2021). Gradient descent is a first-order optimization algorithm that iteratively updates model parameters in the direction of the negative gradient of the loss function with respect to those parameters. This ensures that the model progressively moves toward minimizing prediction errors (Hastie et al., 2009).

The boosting process begins with an initial prediction (e.g., the mean of the target variable for regression or the log odds for classification). For each observation, the gradient of the loss function is computed, representing the error for the current prediction. A new decision tree is then trained to predict these gradients, effectively modeling the adjustments needed to minimize the loss. The model's predictions are updated iteratively by adding a scaled version of the tree's output to the current predictions. The scaling factor, known as the learning rate (eta), dampens the updates, preventing large jumps that could lead to overfitting (Hastie et al., 2009). By focusing on the largest errors at each step, gradient descent ensures that the ensemble becomes progressively stronger, combining the outputs of all trees to create a robust final model.

XGBoost is tuned with several hyperparameters: trees, tree depth (max_depth), learning rate (eta), loss reduction (gamma), sample proportion (subsample), and minimum n (min_child_weight). The trees parameter determines the number of boosting rounds (i.e., the number of trees in the ensemble). As with random forest, more trees generally improve performance but increase computational cost and risk of overfitting. The tree depth (max_depth) limits the maximum depth of each tree; shallow trees control overfitting, while deeper trees capture more complex patterns in the data. The learning rate (eta) scales the contribution of each tree to the final prediction, with smaller values requiring more boosting rounds but often improving generalization. The loss reduction (gamma) parameter specifies the minimum reduction in loss required to make a split, effectively regularizing the model by preventing splits

that do not significantly reduce error. The sample proportion (subsample) determines the fraction of data randomly sampled for each boosting iteration, promoting model diversity and reducing overfitting. Finally, the min_child_weight parameter specifies the minimum sum of weights in a leaf node; higher values prevent small, unreliable splits, further regularizing the model. Tuning these parameters allows XGBoost to balance flexibility, computational efficiency, and predictive performance.

XGBoost has been used for applied attrition modeling in the computer sciences. Akasheh et al. (2024) conclude from their systematic review of the attrition modeling literature that random forests and XGBoost (a tree-based boosting algorithm) have received the most research attention (constituting 34% and 11% of papers, respectively), and may demonstrate the best classification performance compared to the other methods.

Ensemble and Hybrid Methods Compared to Base Methods

The methods discussed in this section were chosen because they facilitate comparisons of different techniques, namely, tree-based methods and regularized regression methods. Among them, XGBoost combines elements of both: it is both tree-based and incorporates regularization. This method is also more complex in that it has more hyperparameters to tune. While tree-based methods including decision trees, random forests, and XGBoost have been widely used for applied attrition modeling, regularized regression methods (lasso, ridge, and elastic net) have not received research attention (Akasheh et al., 2024). By including regularization methods, tree-based methods, and a hybrid method (XGBoost), comparisons can be made between these methodologies based on their shared and unshared techniques.

Ensemble and hybrid methods like elastic net, random forests, and XGBoost were built to mitigate high levels of bias and overfitting present in their base models. Elastic net builds upon

ridge regression and lasso regression. It balances the L1 and L2 regularization parameters, and has the added capability of grouping related features. Random forests enhance the decision tree methodology by introducing randomness in training and by aggregating the predictions of multiple trees, thereby producing more stable estimates that are less prone to overfitting. XGBoost, another ensemble method which utilizes both trees and regularized regression, iteratively corrects errors of previous trees through gradient boosting, further refined with regularization to curb overfitting. These advancements constitute an important evolution from a single tree to a more robust ensemble method. In light of these advancements, I expect hybrid and ensemble models to outperform their base models.

Hypothesis 1: *The classification performance of elastic net regression on out-of-sample data will surpass that of lasso regression and ridge regression.*

Hypothesis 2: *The classification performance of random forests on out-of-sample attrition data will surpass that of decision trees.*

Hypothesis 3: *The classification performance of XGBoost on out-of-sample attrition data will surpass that of decision trees, random forest, lasso regression, ridge regression, and elastic net regression.*

Additionally, the ML methods presented in this chapter have many advantages over logistic regression. Models with high bias tend to underfit the data, missing important patterns, while those with high variance overfit, capturing noise and reducing generalizability. Regularization techniques, such as ridge and lasso regression, mitigate these issues by penalizing model complexity.

Modern ML algorithms, such as random forests and XGBoost, extend beyond the capabilities of logistic regression by incorporating advanced techniques to handle nonlinearity,

multicollinearity, and complex interactions among predictors. These algorithms also integrate hyperparameter tuning, an iterative process that optimizes model settings to improve performance across diverse datasets. Given these advancements, I expect ML algorithms to outperform logistic regression:

Hypothesis 4: *The classification performance of the ML algorithms (lasso, ridge, elastic net regression; decision trees, random forest, and XGBoost) on out-of-sample attrition data will surpass that of logistic regression.*

If supported, these findings would suggest that researchers and practitioners could benefit from the use of more complex and laborious methodologies over simpler approaches.

Choice of Predictors

The previous section explored the ways in which explanatory and predictive research differ in their statistical approach. Beyond methodology, these approaches also differ in their choice and use of data. Turnover research has historically used validated self-report surveys to measure theorized predictors (Bolt et al., 2022, Hom et al., 2017). As an explanatory endeavor, turnover research prioritizes with theoretical fidelity, using validated surveys to reduce unnecessary noise and sources of error, effectively isolating the variables of interest. Structural equation modeling methods, which have long been favored, may have further constrained the types of data used by necessitating clean and well-structured input data (e.g., psychometrically validated measures). In contrast, ML approaches offer greater flexibility, handling diverse data types, including categorical, ordinal, and continuous variables, within a single analysis.

In their review of the turnover literature, Bolt et al. (2022) found that 77% of studies used survey data, while only 9% incorporated organizational records. Although validated psychometric scales from self-report surveys offer valuable insights, their use in applied contexts

is often limited by time and resource constraints (Speer et al., 2019). Moreover, explanatory models' reliance on such measures has historically restricted progress in identifying broader predictors of voluntary attrition. Despite decades of research, explanatory studies have struggled to account for more than 10-15% of the variance in voluntary turnover (Lee & Mitchell, 1994; Holton et al., 2008; Russel, 2013).

Unlike explanatory research, which prioritizes theoretical fidelity, predictive research emphasizes practicality and empirical predictive power. Predictor variables are chosen not only for their theoretical relevance but also for their data availability, completeness, and quality (Shmueli, 2010). This approach necessitates the use of diverse datasets that combine self-reported affective factors, such as job satisfaction, with objective organizational data, including absences, tenure, pay, and performance metrics. These performance metrics may be collected passively through software or actively via structured performance evaluations. The integration of these data types has the potential to enhance the predictive power of models, particularly when leveraging ML methods.

By integrating self-reported data with archival HRIS records, predictive research can address the limitations of traditional approaches. Self-report measures capture subjective aspects of employee attitudes, such as satisfaction and engagement, while archival data provide consistent, objective insights into behaviors and organizational factors. Rubenstein et al. (2018) highlighted that variables such as tenure, rewards, and job alternatives significantly correlate with turnover, underscoring the importance of integrating multiple data sources for more comprehensive models. However, the unique contributions of these data sources and their combined predictive potential remain understudied, particularly within machine learning contexts. To explore this further, the following research question is posed:

Research Question 1: *What data type (HRIS, performance, or self-report) is most predictive of attrition?*

Sample Size

In explanatory research, sample sizes are typically optimized to achieve sufficient statistical power, ensuring the detection of meaningful relationships between variables. Researchers calculate the required number of observations (n) based on the expected effect size, desired significance level (α), and statistical power (1−β), often targeting a power of 0.80 or higher. This approach ensures that findings are robust and reproducible, with minimal risk of Type II errors (failing to detect a true effect). Predictive research, including applied attrition modeling, operates under different constraints, as sample sizes are often determined by the size of the organization or the available dataset. This limitation can lead to smaller sample sizes, especially in smaller organizations, which may challenge the generalizability and performance of predictive models. To explore the robustness of various modeling methods, this study compares their performance at two distinct sample sizes, simulating conditions that reflect both small and large organizations. This comparison will aid in selecting methods that perform consistently across different sample size contexts, offering insights into the scalability of predictive approaches in applied settings.

Conventional wisdom holds that, in general, larger samples are better. Larger samples get us closer to the true population, which should produce more generalizable estimates. There are two aspects of sample size that are important: *n*, the number of observations, and *k*, the number of features. In a study comparing item- and scale-wise prediction of job performance, Putka et al. (2018) found that ML algorithms outperform OLS regression most markedly at smaller *n:k* ratios. In sample sizes between 50-100, the performance gap between ML and OLS regression

was the greatest, gradually decreasing up until 1500. Specifically, elastic net outperformed stepwise regression and lasso regression, and random forest outperformed decision trees (CART). Putka et al. (2018) interprets this finding to indicate that ensemble ML methods are better equipped to handle the bias-variance tradeoff in smaller samples compared to their base models and compared to OLS regression to an even greater extent. Although ML methods outperformed OLS regression at small sample sizes, ML methods still performed best at larger sample sizes.

In a similar study evaluating the use of psychometrically validated scales to predict job performance, Landers et al. (2023) also found that ML algorithms outperformed OLS when the ratio of *n:k* was low. Similar to findings from Putka et al. (2018), Landers et al. (2023) still found that the overall R^2 for elastic net regression, random forest, and linear regression was highest when the *n:k* ratio or overall sample size was high.

Like Putka et al. (2018) and Landers et al. (2023), Zou and Hastie (2003) found that elastic net outperformed lasso regression in a sample of genetics data with a low *n:k* ratio. They conclude that the ability of elastic net to retain correlated features results in superior performance with a smaller *n*, whereas lasso regression would retain only one of the correlated features, missing out on potentially meaningful relationships between the dropped features and the outcome.

In summary, while ML algorithms may have some advantage over traditional methods in samples with few observations on each feature, they still perform better with more observations. However, they may be more useful than traditional methods in small sample sizes. I propose the following hypothesis:

Hypothesis 5: *Random Forest, Elastic Net, and XGBoost will outperform their base*

models most markedly at small sample sizes.

CHAPTER 4

METHODS

Sample

The current study sought to test and compare ML approaches to applied attrition modeling with a sample of archival contact center employee data. Data spanning from January to December of 2023 was obtained from an anonymous organization specializing in customer service and sales. Only employees who successfully completed the selection and training process and initiated paid work with the organization were included in the analysis. All employees were at the same level in the organization and belonged to one of 35 different teams. Demographic data is not available for this sample, but all employees were all US citizens who were 18 years and older.

Defining Attrition

The focus of this study is voluntary attrition, which occurs when employees choose to leave an organization for reasons such as job dissatisfaction, career change, or better opportunities elsewhere. Voluntary attrition is distinct from other forms of attrition like retirement and involuntary attrition (Feldman, 1994). In the present study, only cases of voluntary attrition were classified as attrition events. Counting employees who leave under circumstances other than voluntary attrition as positive cases could introduce unnecessary noise into the analysis. Precursors to different types of attrition are distinct both theoretically and empirically, and should be modeled separately (Adams & Beehr, 1998; Speer et al., 2019).

In each dataset, individuals were assigned one of three values under the “attrition” variable: 0, 1, or NA. A value of “0” indicated no attrition or involuntary attrition for that month. A value of “1” indicated voluntary attrition for that month. NA values indicate that an individual

was not present for that month. These NA observations were removed prior to running the models.

Attrition was tracked on a monthly basis. In this organization, employees often leave without providing prior notice, making it challenging to identify their exact departure day or week in real time. For example, an employee might fail to show up for their shifts during the last few days of month 1. If the employee returns to work in month 2, they are not considered to have voluntarily left in month 1. However, if they do not return in month 2, their departure is recorded as having occurred in month 1. Consequently, data from the past month (month 1) is used to predict attrition in the following month (month 3) while we are still in the present month (month 2). Observations are only counted as having turned over in month 3 if they were present in month 2.

Forecasting attrition at the monthly level is more practical for this organization compared to daily, weekly, or quarterly forecasts. A one-month interval provides sufficient time for the organization to respond proactively, such as initiating recruitment efforts to address anticipated workforce gaps.

Table 3. *Organizational Data*

Performance Data	
Feature Name	Description
Performance*	Other-rated performance score given weekly. Percent of points earned out of available points.
Attendance*$^{\Delta}$	Number of scheduled hours that were worked.
Number of Phone Calls*	Number of phone calls
Self-Report Data	
Feature Name	Description
Client Satisfaction**	Overall satisfaction with the client being served.
Pay Satisfaction**	Satisfaction with rate of compensation.
Recommendation Intentions**	How likely the employee would be to recommend the company to a friend or colleague.
Job Resources – 1**	Perceived completeness of job resources.

Job Resources – 2**	Perceived accuracy of job resources.
Job Resources – 3**	Perceived ease of use of job resources.
Schedule Availability Satisfaction**	Satisfaction with the number of hours available to schedule each week.
Scheduling Satisfaction Overall**	Overall satisfaction with the scheduling process.
Manager Satisfaction – Overall**	Overall satisfaction with the management team.
Manager Satisfaction – Communications**	Satisfaction with communications from the management team.
Manager Satisfaction – Helpfulness**	Satisfaction with the helpfulness of the management team.
Manager Satisfaction – Responsiveness**	Satisfaction with the responsiveness of the management team.
Manager Satisfaction – Respect**	Satisfaction with the respectfulness of the management team.
Manager Satisfaction – Professionalism**	Satisfaction with the professionalism of the management team.
Manager Satisfaction – Feedback**	Satisfaction with the receptiveness to feedback of the management team.
Manager Satisfaction – Knowledge**	Satisfaction with the knowledge of the management team.
Manager Satisfaction – Kindness**	Satisfaction with the kindness of the management team.
HRIS Data	
Feature Name	Description
Invoice Total*$^{\Delta}$	Total amount earned in one week.
Invoice Other*	Other hours the employee was paid for (training, misc. events)
Total Hours*	Total hours worked in one week.

Note. Features with a single asterisk* were used in the final dataset. Features with a double asterisk** were used as a rolling average feature in the final dataset. Features with a delta$^{\Delta}$ were included as a % change feature. Note that features are all individual items, not scale composites.

Table 4. *Engineered Features for Attrition Modeling*

Engineered Features		
Feature Name	Description	Use
Number of Satisfaction Surveys Taken*	Count of the number of satisfaction surveys taken during employees' tenure.	Counted monthly.
Tenure* in days (M = 260.24, SD = 284.02)	Number of days since the first day of work.	Counted monthly.
Number of Weeks Skipped* (M = 0.64, SD = 284.24)	Number of weeks gone without working in a month.	Counted monthly.
MoM % Change	% change from last month to the present month.	Calculated for Performance and HRIS variables.
Rolling Average	Rolling average over all months, including the present month.	Calculated for Performance, HRIS, and Self-Report variables.
Attrition – Outcome*	Whether or not the employee left the company voluntarily.	Outcome.

Features and Feature Engineering

Three types of predictor variables (features) are included in this study: performance, human resources information system (HRIS), and self-report features. Performance and HRIS features were captured passively by the software used by the organization and from employee evaluations performed by managers, and self-report data came from satisfaction surveys which were distributed monthly. Survey responses were recorded on a Likert-type 1-5 scale. See Table 2 for a description of these features.

Feature Engineering. Feature engineering is a method used to either generate new data from existing data or reformat data in a way that the model can better process (Vasquez et al., 2024). Feature engineering is particularly important when working with time-series data with ML. Most ML algorithms cannot be implemented with true time-series data, where all

observations for all months are included in a single dataset. Instead, information must be transformed into a feature which represents changes that occur over time (Verdonck et al., 2024).

Engineered features include the number of weeks gone without scheduling in the current month, rolling averages of self-reported and performance variables, cumulative counts of self-reported surveys taken, month-over-month percentage changes in performance, and the attrition outcome. See Table 3 for a description of all engineered features. Number of weeks gone without scheduling was created based on organizational feedback. Individuals at the organization noted that this metric was predictive of attrition in years past. Percent change, rolling averages, and cumulative counts are discussed below.

Percent Change. The percentage change in performance variables was determined using the following formula:

$$Percent\ Change = \frac{Current\ Month - Previous\ Month}{Previous\ Month} \times 100$$

Rolling Averages. As is common in organizational research, data missingness was a concern due to low survey response rates. To minimize the amount of missing data for each observation, a rolling average was generated for each self-report survey item. Despite the creation of rolling averages, the proportion of missing self-report data across months January through October was still high, averaging 100%, 83%, 65%, 58%, 56%, 53%, 47%, 42%, 46%, 47%, respectively. Overall response rates were low, and high levels of attrition followed by increased mid-year hiring efforts resulted in a dip in valid response counts in July.

In addition to creating rolling averages, I also created a feature representing the count of the number of employee satisfaction surveys taken. This feature represents the information (survey frequency) which was lost by averaging scores monthly, since some employees opted to respond more than once per month. Secondly, this feature was instrumental in determining

whether the data were missing at random. Notably, a significant correlation was observed between the number of surveys completed and employee attrition over time ($r = -.15$, $p < .05$), such that employees who completed fewer surveys were more likely to leave. Consistent with Ding and Simonoff's (2010) recommendations, I included this indicator in each model to ensure that the relationship between missingness and the outcome variable, attrition, is adequately captured by the model.

Feature Selection

Feature selection is the process of identifying and removing irrelevant or redundant features from the dataset to improve model performance, reduce computational complexity, and enhance interpretability (Kotsiantis et al., 2006). Some machine learning methods, such as decision trees, random forests, XGBoost, lasso regression, and elastic net regression, perform feature selection inherently as part of their algorithmic structure. These methods are often referred to as "embedded feature selection" because the selection occurs during model training. However, when comparing models across multiple datasets or time periods, relying solely on embedded methods can lead to inconsistent feature selection, as each algorithm would select different features depending on the specific training data.

To ensure consistent feature selection across months and facilitate clear comparisons, I applied a filter method to remove redundant or weakly correlated features from all datasets. Filter methods are advantageous for preprocessing as they operate independently of specific algorithms, making them computationally efficient and more robust to overfitting compared to wrapper or embedded methods (Saeys et al., 2007; Chandrashekar & Sahin, 2014). A correlation-based filter method was particularly suited for this study, as it allowed the consistent

removal of irrelevant and redundant features across months, enabling meaningful comparisons of predictive performance.

The filter method was applied to a random subset of 1,000 observations from April, as the proportion of missing data stabilized at approximately 50% after this month (83% missing in February, 65% in March, and 58% in April). Using a single dataset (April) for feature selection ensured uniformity in the features selected across months, minimizing potential biases introduced by temporal variability in the data. Within the April subset, missing data were addressed using *k*-Nearest Neighbors (*k*NN) imputation, testing *k* values from 3 to 15. The Kolmogorov-Smirnov test showed no significant differences between the original and imputed datasets ($p > 0.05$), so imputation proceeded with k=5, balancing imputation accuracy and computational efficiency.

Once missing data were imputed, correlation matrices were generated to identify features that were either redundant (highly correlated with other features in the dataset, $r > .80$) or irrelevant (weakly or not significantly correlated with the outcome variable at $\alpha = .05$ level). Redundant features were removed to prevent multicollinearity, which can inflate variance and lead to unstable models, particularly in algorithms sensitive to correlated inputs (Dormann et al., 2013). Irrelevant features were removed to improve model generalization and reduce the dimensionality of the dataset, aligning with research that demonstrates the negative impact of irrelevant features on ML model performance (Guyon & Elisseeff, 2003). This process ensures that the features retained are both relevant to the prediction task and non-redundant, enabling consistent feature selection across datasets while preserving meaningful variability.

Correlation results for the month of April are presented in Appendix A (rolling averages of self-reported features), Appendix B (HRIS features), Appendix C (performance features), and

Appendix D (engineered features). For self-reported features, only one item, prior experience, was removed due to its low correlation with the outcome. Among the HRIS features, several redundant variables were eliminated. Specifically, Invoiced Rate was removed because it was a linear combination of Invoiced Other and Invoiced Total. Similarly, the rolling average (RA) of Invoiced Total was removed due to its high correlation ($r = 0.92$) with Invoiced Total. Total Hours was retained over Total Hours RA and Total Hours percent change (%Δ), as Total Hours %Δ correlated perfectly ($r = 1.0$) with Invoiced Total %Δ, and Total Hours RA had a high correlation ($r = 0.90$) with Invoiced Total RA while being less proximal to the outcome variable. Regarding performance features, "# of Phone Calls RA" was removed because it exhibited a low correlation with the outcome and a high correlation with "# of Phone Calls." Similarly, "# of Phone Calls %Δ" was removed for its low correlation with the outcome. Among performance metrics, "Performance" was retained, while "Performance RA" and "Performance %Δ" were removed due to their high correlations with "Performance" and their low correlations with the outcome. Lastly, "Attendance RA" was removed because of its strong correlation with "Attendance." All four engineered features (tenure, number of satisfaction surveys taken, weeks skipped, and attrition) were retained.

This research-oriented methodology differs from the approach typically taken in practice, where feature selection would be performed on each dataset. By employing this systematic filtering approach, I ensured that all datasets used in model training and evaluation shared the same feature set, facilitating fair comparisons of predictive performance across months.

Dummy-Coded Features. I transformed two categorical features, employee department and direct supervisor, into dummy variables, where a series of 0's and 1's across columns indicate the level of the categorical feature (i.e., manager1 = 1, 0, 0, 0; manager2 = 0, 1, 0, 0;

etc.). Ultimately, I excluded these columns from analysis because observations were too sparse for the model to run. Specifically, there were several instances of zero variance (all values of 0) within folds, which prevented the models from running.

Procedure

A total of seven methods were tested and compared: logistic regression, decision trees using the "classification and regression trees" (CART) algorithm (Breiman, 1984), least absolute shrinkage and selection operator logistic regression (lasso/LARS; Tibshirani, 1996), ridge logistic regression (Hoerl & Kennard, 1970), elastic net logistic regression (LARS-EN; Zou & Hastie, 2005), random forest (Breiman, 2001), and XGBoost (Chen & Guestrin, 2016). I initially sought to compare CART to C5.0, another decision tree algorithm, but was unable to generate a model with C5.0 which effectively produced splits.

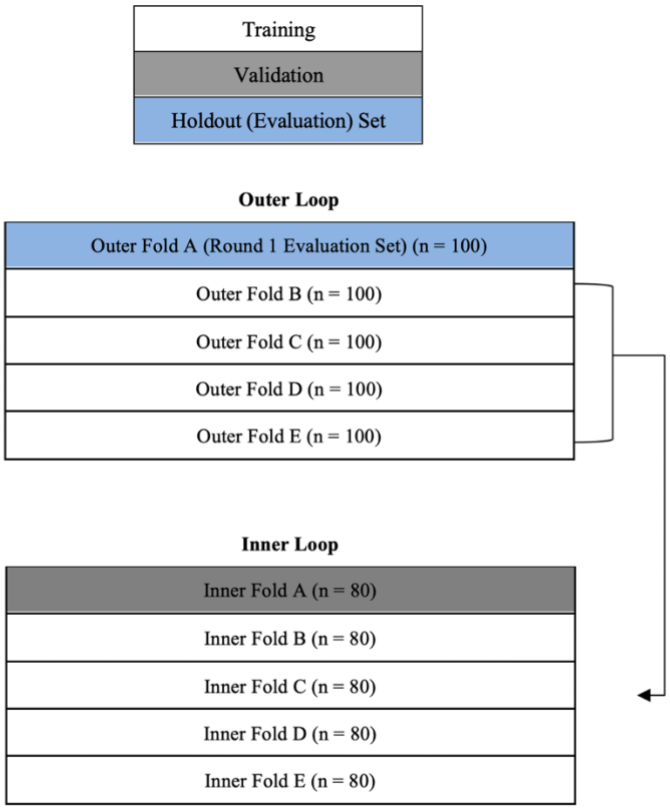
I used the R package tidymodels (Kuhn & Wickham, 2020) to specify the models and model parameters. The tidymodels package, which provides workflows for deploying various ML algorithms, integrates with several other R packages to implement the specified algorithms. Specifically, I used packages "glmnet" for elastic net regression, lasso regression, ridge regression, and logistic regression (Friedman & Hastie, 2010), "ranger" for random forest (Ziegler, 2017), "rpart" for CART (Therneau & Atkinson, 2023), and "xgboost" for XGBoost (Chen et al. 2024) within the tidymodels framework. I chose to use the tidymodels package because it allows for the development of standardized workflows for data preprocessing and cross-validation, which can be applied across various algorithms and datasets.

I used 5-fold nested cross-validation to train and evaluate models. This process is visualized in Figure 4. First, data were partitioned into equally sized groups, called *folds*, to form the "outer loop". In the first round, folds B, C, D, and E are split into 5 equally sized folds to

form the inner loop. The inner loop is responsible for model training and hyperparameter selection. In this loop, each fold takes a turn as the validation set, while the remaining folds are used for training. This process is repeated 5 times. After completing inner loop cross-validation, the best-performing classifier (based on inner validation results) is selected and then evaluated on the held-out outer fold A, which serves as the test set for this round. This process is repeated 5 times, with each outer fold taking its turn as the test set, while the remaining outer folds are used to form the inner loop for training and validation. This process provides results for 5 subsets of data. Final model parameters, performance metrics, and variable importance scores or variable coefficients were recorded for each cross-validation round.

Prior to running models, I used the set.seed() function to generate a starting point for R's random number generator, which allows the random numbers to follow a sequence. This function allows for results to be reproduced and ensures that folds are generated the same way across models (Lantz, 2019). I chose the seed value 123.

Figure 4. *5-fold Nested Cross Validation Process*



Data Imputation

Data Imputation. Missing data is inevitable in organizational datasets, and most ML methods cannot handle missing values. Several approaches – including mean and median imputation, listwise deletion, and k-nearest neighbor (*k*NN) imputation – have been utilized in the ML literature to address the issue of missing data. *k*NN imputation estimates missing values based on the average of observations with similar response patterns (Peterson, 2009). I selected this method because it has been found to produce more reasonable imputed values compared to listwise deletion or mean and median imputation (Batista & Monard, 2003; Jadhav et al., 2019).

To determine the optimal *k*, I performed an exhaustive test of all values ranging from 3-15 to find the value which minimized the difference in variable distribution between the original and imputed datasets. A Kolmogorov-Smirnov test was performed to estimate statistical

significance between variable distributions before and after imputation. The average *p*-value for differences between original and imputed datasets was 1 across all conditions, suggesting that the *k*NN method retained the original variable distributions. Ultimately, I proceeded with the commonly used value, $k = 5$ (Jadhav et al., 2019).

Imputation was performed within-folds, with the outcome variable removed prior to imputation to prevent data leakage (Sajjadian et al., 2021). Additionally, features were centered and scaled within folds. Centering aids in model interpretation, and scaling ensures that features on larger scales (e.g., 1-100) do not dominate those on smaller scales (e.g., 1-5) (Vasquez et al., 2024).

Sample Size Conditions

Two sample size conditions, $n = 1000$ and $n = 500$, were tested to assess the impact of sample size on model performance. The larger sample size condition, $n = 1000$, was selected because it represents the maximum number of observations available in the monthly datasets. Note that three months were shy of 1000 observations. Datasets for August, September, and October had 945, 830, and 848 observations. The large sample size condition constitutes the study's best-case scenario, as larger samples generally improve the stability and reliability of ML models by providing more representative data distributions (Kuhn & Johnson, 2013). The smaller sample size condition, $n = 500$, was chosen for its alignment with the constraints of the 5-fold nested cross-validation process. In this condition, each fold includes a minimum of 80 observations for inner loop validation. This ensures that the cross-validation process is adequately powered while avoiding excessively small validations sets that could lead to unstable performance metrics (Varma & Simon, 2006). Testing this smaller sample size allows for the

evaluation of model performance in smaller organizations. Samples for the small dataset condition were taken as a random sample from the large sample size condition.

Addressing Class Imbalance

Monthly attrition rates can be found in Table 5. Across datasets, the average attrition rate was 11%, indicating a high degree of *class imbalance*. Most attrition datasets face class imbalance, where the number of positive cases in the outcome (attrition) is disproportionately low compared to the number of negative cases (retention). In situations of class imbalance, the algorithm often exhibits a majority class bias, wherein it underpredicts the occurrence of the minority class to maintain overall high classification performance (Kotsiantis et al., 2006). This bias exists because the algorithm's overall performance suffers less from misclassifying minority cases, which constitute a smaller proportion of total observations (Chawla et al., 2004). As a result of this bias, class imbalance can harm prediction, particularly in small datasets and in datasets where the relationship between features and the outcome is complex (Japkowicz & Stephen, 2002). Across base and ensemble algorithms, all algorithms perform better with class balanced data (Garcia et al., 2022). Thus, it is best practice to use a method of class-balancing, which evens out the number of positive and negative observations.

I considered three class-balancing techniques: over-sampling, under-sampling, and the Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002; Provost, 2000; Elkan, 2001). Over-sampling involves randomly re-sampling the underrepresented class, while under-sampling reduces the number of observations in the overrepresented class. Both methods come with notable drawbacks. Over-sampling retains the dataset's original size but can lead to overfitting due to the duplication of observations in the training data (Garcia et al., 2022). On the other hand, under-sampling reduces the dataset's overall size, which risks discarding potentially

important cases that may compromise the model's performance (Kotsiantis et al., 2006; Garcia et al., 2022).

SMOTE blends aspects of over-sampling and under-sampling, and is widely regarded as an effective method for addressing class imbalance (Kotsiantis et al., 2006). Unlike random over-sampling, which duplicates minority class instances, SMOTE generates synthetic samples by interpolating between existing minority class observations. Specifically, for a given minority class instance, synthetic samples are generated based on the values of other similar instances (Chawla et al., 2002). This process preserves the feature space's structure and mitigates overfitting associated with other oversampling methods, such as random oversampling or random under sampling (Batista et al., 2004). Given its ability to balance the dataset while preserving its underlying characteristics, I ultimately decided to use SMOTE for this analysis.

To maintain the integrity of the dataset during cross-validation, SMOTE was applied separately to each fold of the inner loop during cross-validation, ensuring that synthetic samples were generated only within the training folds and not carried over into the validation fold. Note that class imbalance is only used on training folds. The goal of model training is to produce a model that can make predictions on data to predict real outcomes, and therefore the test or validation set should remain untouched.

Table 5. *Monthly Voluntary Attrition Rates*

Month	Sample Size	Attrition Rate
April	500	9.6%
April	1000	9.6%
May	500	12.0%
May	1000	10.5%
June	500	12.4%
June	1000	12.7%
July	500	12.4%
July	1000	12.2%

August	500	7.4%
August	1000	9.6%
September	500	11.8%
September	1000	11.6%
October	500	12.8%
October	1000	11.4%

Note. Attrition is determined by whether the employee returned the following month.

Hyperparameter Tuning

Grid search with a specified search space was used in the inner loops to select the set of hyperparameters which maximized algorithmic performance. Grid search is a systematic method for hyperparameter optimization that tests different parameter combinations within a provided search space to identify the set of parameters that yields the best performance for a given ML model (Liashchynskyi & Liashchynskyi, 2019). Grid search was performed automatically and individually for each algorithm.

Note that each additional value of hyperparameters tested increases the number of models that are fit multiplicatively. For example, a single inner loop fold with three values of three hyperparameters to test will produce $3 \times 3 \times 3$ fits. The best hyperparameters will be chosen from this fold, and the process will repeat four more times, producing 135 calculations per outer fold and 675 total calculations for the classifier. Not only is this process computationally expensive and time consuming; testing too many possible hyperparameter values can result in overfitted models. For this reason, algorithms with more hyperparameters to test (i.e., XGBoost) were restricted to fewer values per hyperparameter. The number of distinct values tested for each hyperparameters were as follows: 2 for XGBoost, 2 for CART, 5 for random forest and elastic net, and 10 for ridge and lasso regression. The tuned hyperparameters, as well as the hyperparameter values chosen for use in the outer loop, can be found in Appendix E.

Evaluating Model Performance

Classification performance of the chosen algorithms and logistic regression is evaluated by the predictions made on the validation fold during 5-fold nested cross-validation. Classification algorithms produce probabilities of group membership for each observation in the dataset. With a 0 or 1 outcome, where an outcome of 1 is considered positive, a probability assigned to an observation that is greater than 50% is labeled positive, and probabilities less than 50% are labeled negative. The confusion matrix (*see* Table 6) is a table which contains the number of false positive, false negative, true positive, and true negative classifications made by the classifier. The foundational metrics typically used to evaluate classifiers are sensitivity/recall (Equation 1), specificity (Equation 2), accuracy (Equation 3), and precision (Equation 4).

Table 6. *Confusion Matrix*

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Equation 4. *Sensitivity/Recall (True Positive Rate)*

$$Sensitivity = \frac{TP}{TP + FN}$$

Equation 5. *Specificity*

$$Specificity = \frac{TN}{TN + FP}$$

Equation 6. *Accuracy*

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Equation 7. *Precision*

$$Precision = \frac{TP}{TP + FP}$$

Performance Metric Evaluation with Class Imbalance

Even when class imbalance is addressed during model training, class imbalance in the test data impacts the usefulness of certain evaluation metrics, especially accuracy. Consider a sample of *n*=100 with a 20% attrition rate and a trained classifier which predicts that only one of the 100 cases are positive and 99 of the cases are negative. Although the model's performance on the positive class appears poor, it would still achieve a relatively high accuracy score of 80%:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{1 + 80}{1 + 80 + 0 + 19} = 80\%$$

Accuracy is a poor metric to use in cases of class imbalance because it overemphasizes performance on the majority class. Similarly, specificity and precision are not useful in this example. In contrast, recall strongly penalizes the model for missing true positive cases.

$$Recall = \frac{TP}{TP + FN} = \frac{1}{1 + 19} = 5\%$$

$$Specificity = \frac{TN}{TN + FP} = \frac{80}{80 + 0} = 100\%$$

$$Precision = \frac{TP}{TP + FP} = \frac{1}{1 + 0} = 100\%$$

For this reason, I do not evaluate classifiers based on accuracy, specificity, or precision alone. Instead, I use balanced metrics – balanced accuracy (BA), BA-Recall, and BA-Specificity – along with recall and AUC. Balanced accuracy (BA) is the average of recall and specificity and gives equal importance to correctly classifying both positive and negative classes.

Equation 8. *Balanced Accuracy*

$$Balanced\ Accuracy = \frac{Recall + Specificity}{2}$$

Following from the previous example, the classifiers' BA would be calculated as follows:

$$Balanced\ Accuracy = \frac{Recall + Specificity}{2} = \frac{5 + 100}{2} = 52.5\%$$

Balanced accuracy represents the degree to which the model effectively balances recall and specificity, which are inversely correlated. A model with higher recall will also have a higher false positive rate, because more positive instances must be predicted overall in order for true positives to increase. On the other hand, a model with higher specificity will have a lower false positive rate because more negative predictions will need to be made to have a high true negative rate. Similarly, BA-Recall weights recall at 75% and specificity at 25%, and BA-Specificity weights specificity at 75% and recall at 25%. Some organizations may be risk-averse in one direction – perhaps an organization has a low budget and cannot afford to over hire. Such an organization would prefer the model with a higher BA-Specificity score over a model with a high BA or BA-Recall score. Balanced accuracy metrics, and weighted balanced accuracy metrics, provide an index of performance that balances risk in the desired direction, allowing the user to determine which approach fits best with their organization's needs (Shewach et al., 2024).

Lastly, I evaluate classifier performance using area under the curve (AUC). Rather than evaluating performance based on the confusion matrix, AUC evaluates the reflects the ability of a classifier to distinguish between positive and negative classes across a range of thresholds. It is calculated using the predicted probabilities of class membership, rather than directly relying on the confusion matrix. This distinction is important because AUC evaluates a classifier's ability to rank observations correctly—distinguishing between positive and negative classes—independent of any specific classification threshold. The predicted probabilities are used to plot the Receiver Operating Characteristic (ROC) curve, which visualizes the trade-off between the true positive

rate (sensitivity) and the false positive rate at various thresholds. By considering all possible thresholds, AUC provides a single, summary measure of a model's overall discriminatory power.

In contrast, metrics like accuracy, sensitivity, and specificity are threshold-dependent and tied directly to the confusion matrix, which represents outcomes based on a fixed decision threshold (e.g., 50%). Because AUC uses probabilities rather than hard classifications, it is threshold agnostic. A classifier with higher AUC scores is better at ranking positive instances higher than negative ones, regardless of where the threshold is set. A low AUC score suggests poor ranking performance, even if threshold-dependent metrics like accuracy appear high. This feature makes AUC useful when evaluating models under imbalanced data conditions, where reliance on confusion-matrix-based metrics may overstate a model's true performance.

Evaluating Feature Importance

Regression-based methods (logistic, lasso, ridge, and elastic net regression) produce beta coefficients for each feature. The magnitude of these coefficients represents the importance of each feature, with larger absolute values indicating that the feature was useful in generating predictions (Saarela & Jauhiainen, 2021). This is especially true in regularized regression models, where uninformative coefficients are reduced (as in ridge and elastic net regression) or shrunk to zero (as in lasso regression) (Shiomi et al., 2022).

In contrast, decision trees and random forest estimate feature importance through Gini importance, which identifies the most influential predictors in the dataset (Breiman, 2001; Louppe, 2015; Strobl et al., 2008). Gini importance is formulated as:

Equation 6: *Gini Importance.*

$$Gini = p_1(1 - p_1) + p_2(1 - p_2)$$

where p_1 and p_2 are the probabilities of class 1 and class 2, respectively (Saarela & Jauhiainen, 2021). Gini importance is calculated after the entire model has run and estimates the extent to which a feature improves the homogeneity of the resulting child node compared to the parent node. Each tree begins with a root node (Tenure in Figure 2, for example) with a given impurity where X% of observations belong to class 1 and class 2. If we then split on another variable, we get two child nodes. Gini importance compares the purity of the two child nodes resulting from the split to the root node that preceded it. A higher Gini importance score indicates that a feature provides a strong ability to differentiate between classes (Nembrini et al., 2016).

XGBoost does not use Gini impurity to split nodes like decision trees and random forests. As previously discussed, XGBoost measures model improvement based on the log loss. Feature importance is measured retrospectively using Gain, which quantifies improvements in the loss function.

Comparing Classification Performance Across Models

A statistical test is needed to determine whether differences in performance metrics among ML classifiers (e.g., specificity, recall, etc.) are meaningful. With 5-fold nested cross-validation, a total of 5 classifiers are produced for each dataset, each with their own performance metrics. Across 7 algorithms, 7 months, and 2 sample sizes, there are a total of 490 classifiers to evaluate.

As of writing this dissertation, there is a lack of consensus within the ML literature on the best method for cross-algorithm comparison. Researchers have used a variety of methods including paired t-tests, corrected t-tests, counting wins and losses, ANOVA, the Friedman test (a non-parametric ANOVA), and the Wilcoxon signed-ranks test (*see* Demšar et al., 2006; Nadeau & Bengio, 2003). Of these methods, the Friedman test and the Friedman aligned rank

test appears to be the most reasonable options for comparing classifier performance across and within sampled months.

The Friedman Test

The Friedman test, a non-parametric test similar to ANOVA, can be used to compute the rank performance of several classifiers across different datasets (Demšar et al., 2006). It ranks classifier performance within blocks (datasets) and tests whether the rank performance of classifiers are significantly different (García et al., 2010). The Friedman test is preferred when comparing the relative performance of ML classifiers because, unlike parametric tests like t-tests and ANOVA, it does not assume a normal distribution of parameters, homogeneity of variance, or independence of datasets. As an omnibus test, the Friedman test analyzes the performance of classification algorithms separately on different datasets and calculates a statistic (F) indicating whether there were significant differences (Santafe et al. 2015).

To calculate the test statistic, I used the Inman-Davenport (1980) corrected F statistic, F_{ID}, a variation of the Friedman test designed to address small-sample biases in the original Friedman test statistic. This correction makes the test more robust, particularly in scenarios with a limited number of datasets or classifiers. If the null hypothesis of no differences between classifier ranks is rejected, the post-hoc Nemenyi test is applied to perform pairwise comparisons between algorithms.

Friedman Aligned Rank Test

The Friedman aligned rank test, like the standard Friedman test, is used to compare the relative performance of classifiers. However, it is particularly suited for situations where the same dataset is used across comparisons (García et al., 2010; Santafé et al., 2015). Given that

datasets within months overlap, I applied the Friedman aligned rank test to each matrix to account for this dependency.

Averaging Across Folds

Guidance on handling multiple folds when evaluating classification algorithms is limited in the literature. Using metric scores (i.e., BA, BA-Recall, BA-Specificity, Recall, AUC) from all 5 folds for each month would violate the Friedman test's assumption of independence between rows. To address this, I chose to average the scores across folds, ensuring that comparisons across months adhered to the test's assumptions while maintaining the integrity of the results.

CHAPTER 5

RESULTS

Comparing Performance Metrics Between and Across Sample Size Conditions

To compare the overall rank performance of algorithms across different sample size conditions, I conducted Inman-Davenport corrected Friedman tests on five matrices, each corresponding to a performance metric (true positive rate, balanced accuracy, balanced accuracy-sensitivity weighted, balanced accuracy-recall weighted, and area under the curve). The performance scores were averaged across cross-validation folds. Each month was represented by two rows in the matrices, one for each sample size (n = 500 and n = 1000). Both the Inman-Davenport corrected Friedman test and the Friedman aligned ranks test were applied. Significant differences indicate that a given method outperformed another method more frequently. Since significant results were consistent across both omnibus tests, only the results from the Inman-Davenport corrected Friedman test (χ^2) are reported for simplicity. When significant differences were found, the Nemenyi post-hoc test was used to determine the critical difference (CD) values and generate a matrix of paired difference scores. Critical difference represents the value at which differences are statistically significant. CD plots were created to visually represent the magnitude of differences between methods and their statistical significance.

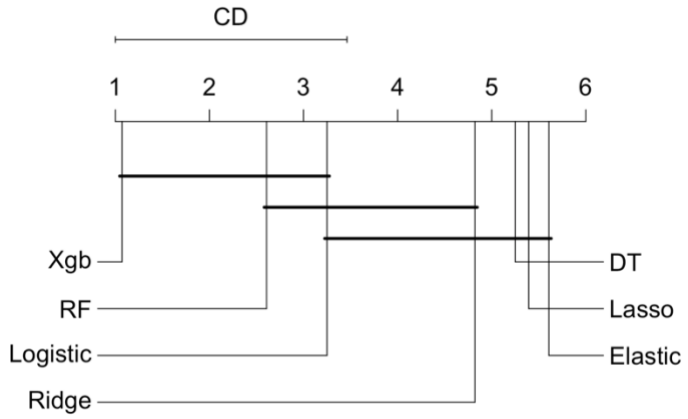
To evaluate Hypothesis 5, which predicted that hybrid and ensemble methods would outperform their base methods most noticeably at larger sample sizes, I also performed an Inman-Davenport corrected Friedman test on matrices with two columns for each method, one for each sample size. This way, methods can be evaluated across different sample sizes.

Recall

Recall Across Sample Sizes. First, I compared each method on recall, also known as sensitivity or true positive. Recall is a metric which indicates how well a classifier identifies positive cases. The omnibus Iman-Davenport corrected Friedman's rank sum test indicated a statistically significant difference between methods on their recall rankings (χ^2 (6,78) = 22.824, p < 0.001). Based on the critical difference value produced by the Nemenyi post-hoc test (CD (7,91) = 2.462, a = 0.05), the following pairwise differences from the Nemenyi difference matrix were statistically significant, with the first listed method outranking the second listed method: XGBoost (Xgb) vs. decision trees (DT) (p < 0.001), XGBoost vs. elastic net (EN) (p < 0.001), XGBoost vs. lasso regression (Lasso) (p < 0.001), XGBoost vs. ridge regression (Ridge) (p < 0.001), random forest (RF) vs. decision trees (p = 0.002), random forest vs. lasso regression (p = 0.002), and random forest vs. decision trees (p < 0.001). The CD plot in Figure 5 visualizes these pairwise differences, with bold horizontal lines connecting classifiers with statistically similar rankings at the p = 0.05 level. Note that lower rankings (e.g., 1, 2) indicate better performance.

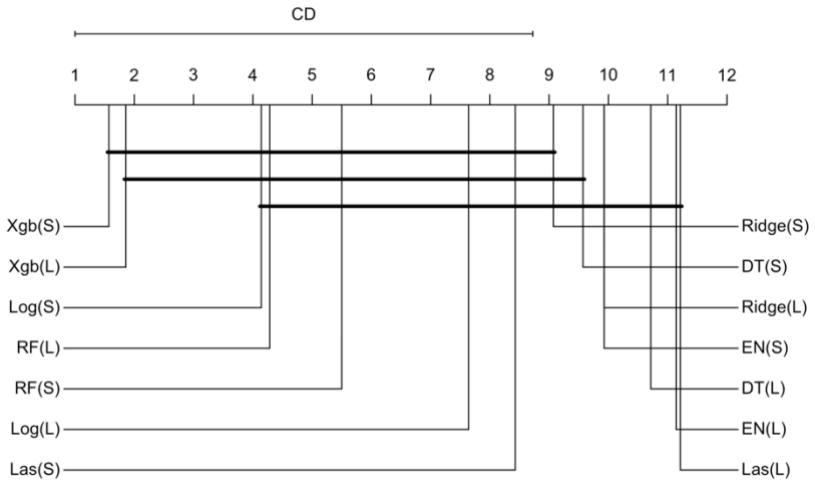
Interestingly, decision trees exhibited a higher recall in April (47%), the month during which feature selection was performed, compared to other months, where recall ranged between 25%-27% (*see* Table 7). This suggests that decision trees may benefit from monthly variable selection. In contrast, other methods did not display such a substantial difference in recall between April and other months, indicating that their performance was less influenced by the timing of feature selection.

Figure 5. *Critical Difference Plot for Recall Across Sample Sizes*



Recall Between Sample Sizes. The Iman Davenport corrected Friedman's rank sum test revealed a statistically significant difference between method-sample size pairs in recall score rankings (χ^2 (13,78) = 11.484, $p < 0.001$). However, the Nemenyi post-hoc test did not indicate significant differences between most pairs of base methods and ensemble/hybrid methods at small sample sizes (see Fig. 6 for pairwise comparisons). Specifically, random forest (S) did not outrank decision trees (S), and elastic net (S) did not outrank lasso regression (S) or ridge regression (S). However, XGBoost (S) did outrank decision trees (S). These mixed null results indicate that base methods are as capable as ensemble and hybrid methods at identifying true positives in small (n = 500) sample sizes, apart from XGBoost, which significantly outperformed decision trees.

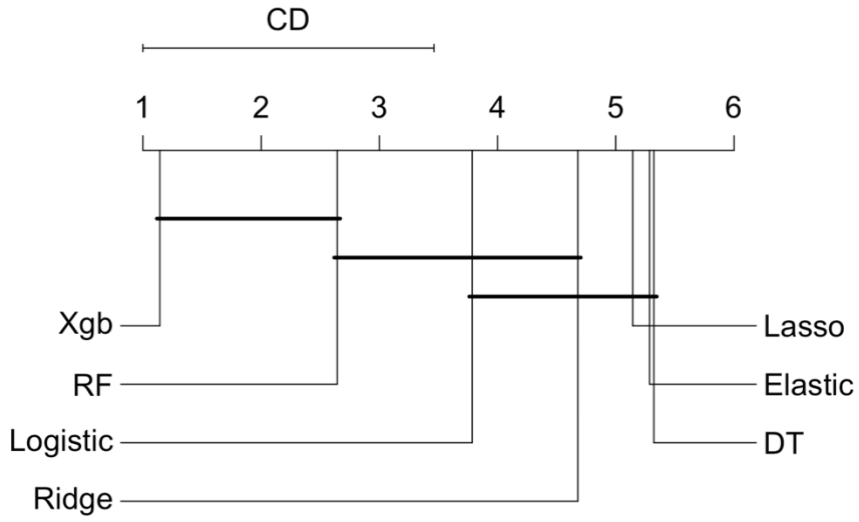
Figure 6. *Critical Difference Plot for Recall Between Sample Sizes*



Balanced Accuracy

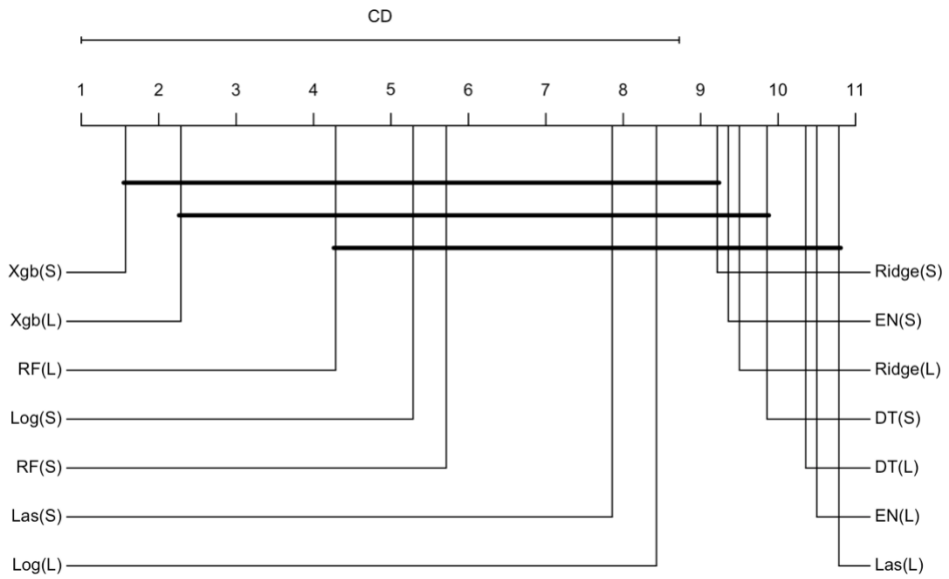
BA Across Sample Sizes. An Inman-Davenport correction of Friedman's rank sum test revealed a statistically significant difference between scores on balanced accuracy across methods, $\chi^2(6,78) = 15.475$, $p < .001$. The post-hoc Nemenyi test produced a critical difference score of 2.462 ($k = 7$, $df = 91$). Based on the difference matrix, XGBoost ranked significantly higher on BA compared to all other methods except for random forest. Additionally, random forest performed similarly to logistic regression and ridge regression, and outperformed lasso regression, elastic net, and decision trees. The CD plot shown in Figure 8 visualizes pairwise differences on BA.

Figure 7. *Critical Difference (CD) Plot for Balanced Accuracy across Sample Sizes*



BA Between Sample Sizes. Inman-Davenport correction of Friedman's rank sum test indicated a significant difference between method-sample size pairs in balanced accuracy scores, $\chi^2(13,78) = 7.402$, $p < 0.001$. The Nemenyi post-hoc test indicated a critical difference of 7.723 (k =14, df = 84, a = 0.05). I did not find full support for Hypothesis 2, which predicted ensemble and hybrid methods to outperform base methods at small sample sizes. Contrary to expectations, XGBoost with a small sample size did not outperform ridge regression used with a small sample size, but it did outperform ridge regression used with a large sample size. XGBoost (S) did not outperform ridge regression (S), but it did outperform ridge regression (L). However, XGBoost (S) did outperform decision trees (S). No other significant differences were found for BA between sample sizes.

Figure 8. *Critical Difference (CD) Plot for Balanced Accuracy Between Sample Sizes*



Balanced Accuracy – Specificity Weighted

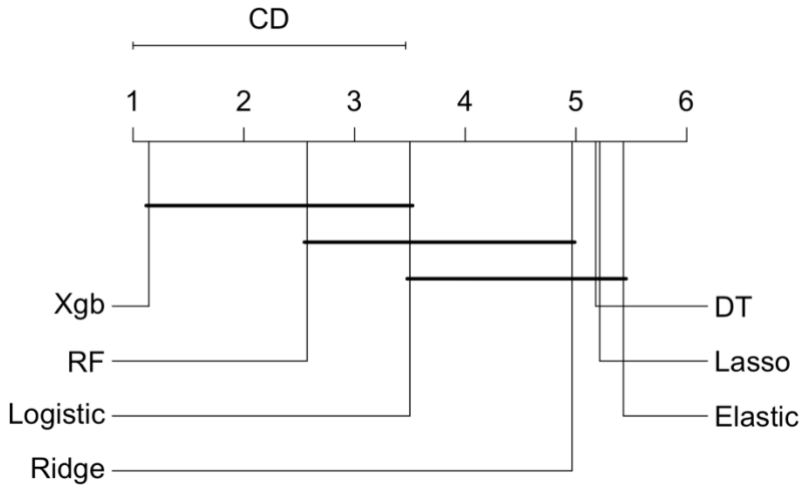
BA-Specificity across Sample Sizes. Overall, methods did not vary substantially in their scores on specificity, which limits the utility of the specificity-weighted balanced accuracy metric. Specificity is a metric used to estimate how well a classifier can identify negative instances. Because there were so many negative instances in the data, classifiers did not have to try very hard to find them. Thus, all methods scored relatively well on specificity, and on BA-Specificity by extension. An Inman-Davenport correction of Friedman's rank sum test revealed a statistically nonsignificant difference in scores on BA-Specificity across methods, $\chi^2(6,78) = 0.431$, $p = .855$. There were no statistically significant differences between scores on BA-Specificity among methods. Mean BA-Specificity scores can be found in Table 9.

BA-Specificity between Sample Sizes. Similarly, the Inman-Davenport correction of Friedman's rank sum test indicated a nonsignificant difference between method-sample pairs on BA-Specificity scores, $\chi^2(13,78) = 0.271$, $p = 0.994$.

Balanced Accuracy – Recall Weighted

BA-Recall across Sample Sizes. An Inman-Davenport correction of Friedman's rank sum test revealed a statistically significant difference in scores on BA-Recall across methods, χ^2 (6,78) = 18.08, $p < 0.001$. The Nemenyi post-hoc test yielded a critical difference score of 2.462 ($k = 7$, $df = 91$, $a = 0.05$). Based on the difference matrix, statistically significant differences exist between XGBoost vs. ridge regression, XGBoost vs. decision trees, XGBoost vs. lasso regression, XGBoost vs. elastic net, random forest vs. decision trees, random forest vs. lasso regression, and random forest vs. elastic net (see Figure 9).

Figure 9. *Critical Difference (CD) Plot for BA-Recall across Sample Sizes*



BA-Recall Between Sample Sizes. An Inman-Davenport correction of Friedman's rank sum test indicated a significant difference between pairs on BA-Recall scores $\chi^2(13,78) = 9.807$, $p < 0.001$. The Nemenyi post-hoc test indicated a critical difference of 7.723 ($k = 14$, $df = 84$, $a = 0.05$). There were statistically significant differences between several pairs, as can be seen in Figure 10. XGBoost (S) significantly outranked ridge regression (S) and decision trees (S). See Table 10 for all BA-Recall means across methods and sample sizes.

Table 7. *Mean Recall across Months and Sample Sizes.*

	DT(S)	EN(S)	Las(S)	Log(S)	RF(S)	Ridge(S)	Xgb(S)	DT(L)	EN(L)	Las(L)	Log(L)	RF(L)	Ridge(L)	Xgb(L)
April	0.31	0.00	0.14	0.09	0.21	0.00	0.27	0.03	0.01	0.01	0.01	0.04	0.01	0.23
May	0.00	0.00	0.00	0.02	0.01	0.00	0.23	0.00	0.00	0.00	0.00	0.17	0.00	0.22
June	0.00	0.03	0.03	0.08	0.10	0.00	0.26	0.00	0.00	0.01	0.01	0.07	0.01	0.19
July	0.00	0.04	0.06	0.18	0.00	0.08	0.24	0.00	0.01	0.00	0.04	0.08	0.01	0.27
August	0.03	0.00	0.00	0.11	0.04	0.04	0.18	0.02	0.00	0.00	0.04	0.06	0.01	0.15
September	0.00	0.00	0.00	0.10	0.12	0.00	0.16	0.00	0.00	0.00	0.01	0.06	0.00	0.15
October	0.00	0.03	0.03	0.12	0.15	0.05	0.16	0.00	0.00	0.00	0.10	0.20	0.02	0.22

Table 8. *Mean BA across Months and Sample Sizes*

	DT(S)	EN(S)	Las(S)	Log(S)	RF(S)	Ridge(S)	Xgb(S)	DT(L)	EN(L)	Las(L)	Log(L)	RF(L)	Ridge(L)	Xgb(L)
April	0.64	0.50	0.55	0.53	0.59	0.50	0.61	0.50	0.50	0.50	0.50	0.51	0.50	0.58
May	0.50	0.50	0.50	0.55	0.52	0.52	0.57	0.50	0.50	0.50	0.52	0.52	0.51	0.55
June	0.50	0.51	0.53	0.57	0.50	0.53	0.58	0.50	0.50	0.50	0.51	0.52	0.50	0.59
July	0.50	0.51	0.51	0.52	0.54	0.50	0.58	0.50	0.50	0.50	0.50	0.53	0.50	0.56
August	0.50	0.50	0.50	0.49	0.50	0.50	0.57	0.50	0.50	0.50	0.50	0.55	0.50	0.57
September	0.50	0.51	0.51	0.55	0.54	0.52	0.56	0.50	0.50	0.50	0.54	0.58	0.51	0.56
October	0.50	0.50	0.50	0.54	0.53	0.50	0.54	0.50	0.50	0.50	0.50	0.52	0.50	0.54

Table 9. *Mean BA-Specificity Across Months and Sample Sizes*

	DT(S)	EN(S)	Las(S)	Log(S)	RF(S)	Ridge(S)	Xgb(S)	DT(L)	EN(L)	Las(L)	Log(L)	RF(L)	Ridge(L)	Xgb(L)
April	0.80	0.75	0.75	0.75	0.79	0.75	0.78	0.74	0.75	0.75	0.75	0.75	0.75	0.76
May	0.74	0.75	0.75	0.73	0.75	0.75	0.74	0.75	0.75	0.75	0.75	0.75	0.75	0.75
June	0.75	0.75	0.75	0.73	0.76	0.75	0.74	0.75	0.75	0.75	0.74	0.76	0.75	0.75
July	0.75	0.75	0.76	0.76	0.75	0.76	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.76
August	0.73	0.75	0.75	0.77	0.76	0.76	0.76	0.74	0.75	0.75	0.76	0.75	0.75	0.75
September	0.75	0.75	0.75	0.76	0.74	0.75	0.74	0.75	0.75	0.75	0.75	0.75	0.75	0.73
October	0.75	0.76	0.76	0.76	0.74	0.76	0.75	0.75	0.75	0.75	0.76	0.76	0.75	0.72

Table 10. *Mean BA-Recall Across Months and Sample Sizes*

	DT(S)	EN(S)	Las(S)	Log(S)	RF(S)	Ridge(S)	Xgb(S)	DT(L)	EN(L)	Las(L)	Log(L)	RF(L)	Ridge(L)	Xgb(L)
April	0.47	0.25	0.34	0.31	0.40	0.25	0.44	0.27	0.26	0.26	0.26	0.27	0.26	0.41
May	0.27	0.25	0.25	0.33	0.28	0.28	0.37	0.26	0.25	0.25	0.28	0.29	0.26	0.35
June	0.25	0.28	0.29	0.38	0.25	0.30	0.41	0.25	0.26	0.25	0.28	0.30	0.26	0.43
July	0.25	0.27	0.27	0.30	0.32	0.25	0.42	0.25	0.25	0.25	0.26	0.30	0.26	0.38
August	0.25	0.25	0.25	0.26	0.26	0.25	0.40	0.25	0.25	0.25	0.25	0.36	0.25	0.40
September	0.25	0.27	0.27	0.33	0.34	0.29	0.36	0.25	0.25	0.25	0.32	0.39	0.26	0.39
October	0.25	0.25	0.25	0.32	0.32	0.25	0.35	0.25	0.25	0.25	0.25	0.29	0.25	0.34

Table 11*. Mean Area Under the Curve (AUC) Values Across Months and Sample Sizes*

	DT(S)	EN(S)	Las(S)	Log(S)	RF(S)	Ridge(S)	Xgb(S)	DT(L)	EN(L)	Las(L)	Log(L)	RF(L)	Ridge(L)	Xgb(L)
April	0.31	0.26	0.37	0.30	0.72	0.25	0.66	0.50	0.32	0.33	0.23	0.72	0.22	0.71
May	0.46	0.41	0.41	0.26	0.68	0.26	0.67	0.50	0.33	0.34	0.30	0.61	0.31	0.67
June	0.50	0.29	0.25	0.24	0.73	0.23	0.68	0.50	0.21	0.23	0.23	0.76	0.21	0.74
July	0.50	0.22	0.24	0.25	0.82	0.20	0.74	0.50	0.22	0.32	0.24	0.74	0.24	0.69
August	0.51	0.37	0.50	0.31	0.68	0.29	0.63	0.50	0.39	0.47	0.31	0.66	0.30	0.64
September	0.50	0.40	0.46	0.32	0.67	0.25	0.66	0.50	0.36	0.37	0.30	0.66	0.25	0.64
October	0.50	0.23	0.31	0.25	0.67	0.24	0.69	0.50	0.24	0.24	0.24	0.67	0.23	0.64

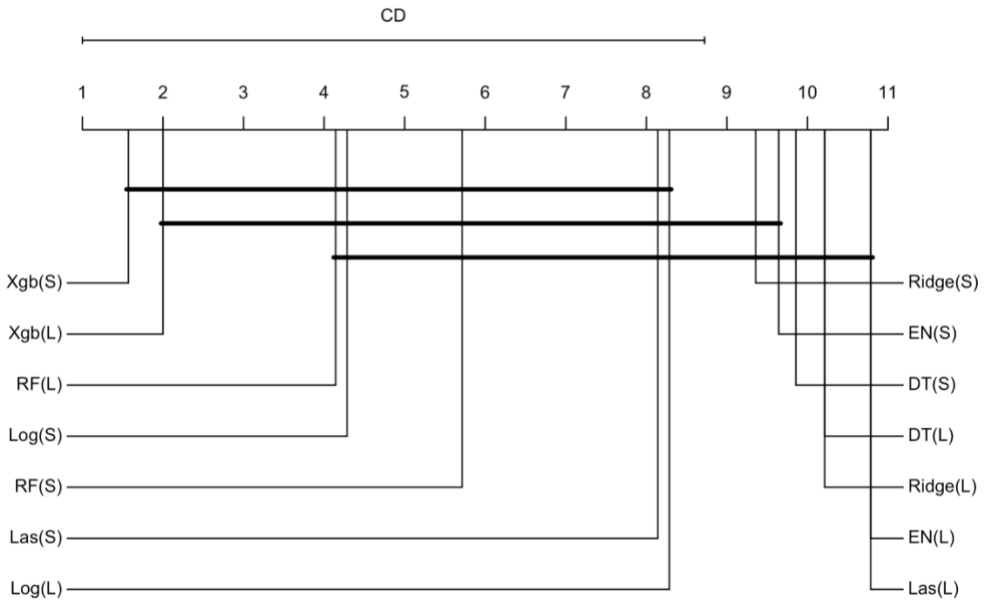
Table 12. *Mean Performance Metrics for Classifiers Across Months, Folds and Sample Sizes*

Classifier	Precision	Recall	Specificity	Accuracy	F1	AUC
Decision Tree	0.05	0.03	0.99	0.88	0.03	0.48
Elastic Net	0.05	0.01	1.00	0.89	0.01	0.30
Lasso Regression	0.06	0.02	0.99	0.89	0.03	0.35
Logistic Regression	0.23	0.06	0.98	0.88	0.09	0.27
Random Forest	0.19	0.09	0.97	0.87	0.11	0.70
Ridge Regression	0.10	0.02	1.00	0.89	0.02	0.25
XGBoost	0.28	0.21	0.93	0.85	0.22	0.68

Table 13*. Mean Performance Metrics for Classifiers Across Months and Folds by Sample Size*

Classifier	Sample Size	Precision	Recall	Specificity	Accuracy	F1	AUC
Decision Tree	500	0.09	0.05	0.99	0.88	0.06	0.47
Decision Tree	1000	0.01	0.01	0.99	0.89	0.01	0.50
Elastic Net	500	0.04	0.01	1.00	0.89	0.02	0.31
Elastic Net	1000	0.06	0.00	1.00	0.89	0.00	0.29
Lasso Regression	500	0.09	0.04	0.99	0.88	0.05	0.36
Lasso Regression	1000	0.03	0.00	1.00	0.89	0.00	0.33
Logistic Regression	500	0.25	0.10	0.97	0.87	0.13	0.28
Logistic Regression	1000	0.22	0.03	0.99	0.89	0.05	0.26
Random Forest	500	0.21	0.09	0.98	0.88	0.10	0.71
Random Forest	1000	0.17	0.10	0.97	0.87	0.11	0.69
Ridge Regression	500	0.08	0.02	1.00	0.89	0.03	0.25
Ridge Regression	1000	0.13	0.01	1.00	0.89	0.01	0.25
XGBoost	500	0.28	0.21	0.93	0.85	0.22	0.68
XGBoost	1000	0.28	0.20	0.93	0.85	0.22	0.68

Figure 10. *Critical Difference (CD) Plot for BA-Recall Between Sample Sizes*

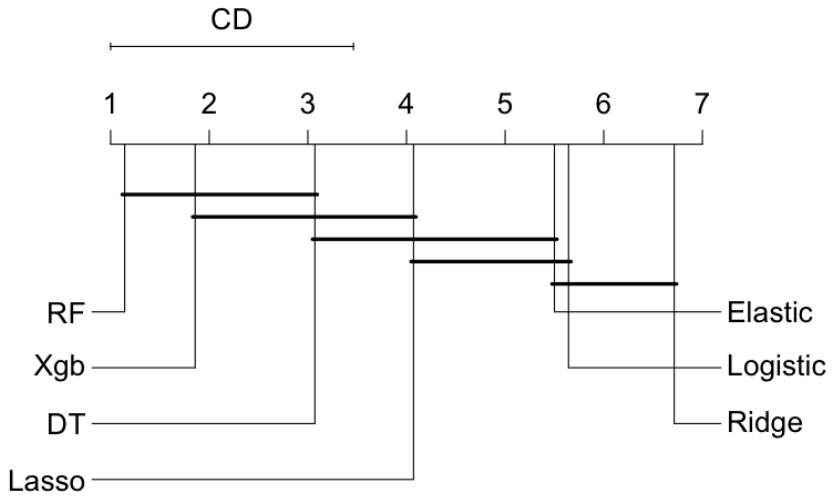


AUC

AUC Across Sample Sizes. An Inman-Davenport correction of Friedman's rank sum test indicated a significant difference between pairs on AUC scores $\chi^2(6,78) = 163.59$, $p < 0.001$. The Nemenyi post-hoc test determined a critical difference of 2.462 ($k = 7$, $df = 91$, $a = 0.05$). While the ranking order varied slightly, the significant differences followed a consistent pattern: random forest outperformed lasso regression, elastic net, logistic regression, and ridge regression. Similarly, XGBoost outperformed elastic net, logistic regression, and ridge regression. Random forest and XGBoost achieved AUC values of 70% and 68%, respectively. These values represent the likelihood that these classifiers would correctly rank a randomly chosen positive instance higher than a randomly chosen negative instance. Thus, AUC values below 50% indicate that a classifier is not practically useful. Notably, decision trees outperformed logistic regression, a result that, while statistically significant, lacks practical

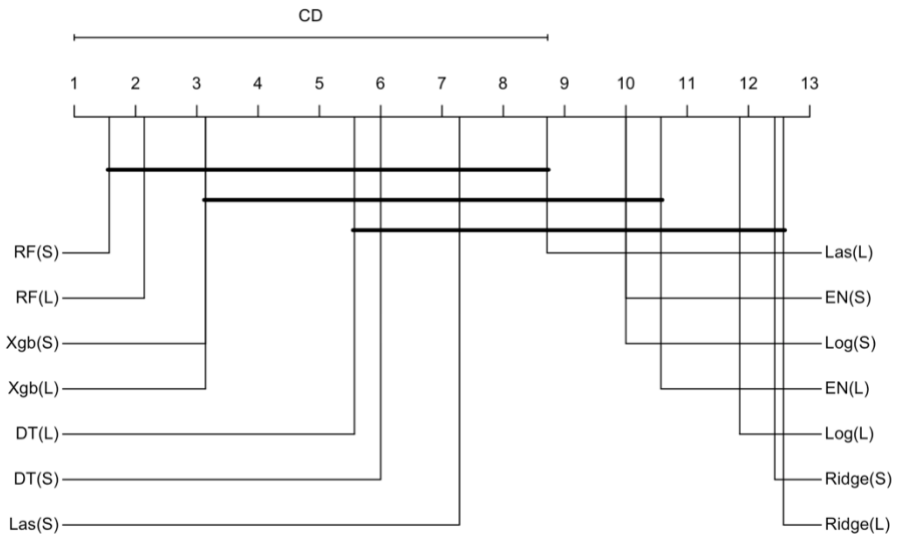
relevance. Decision trees averaged an AUC of 48% across folds, which is worse than random guessing.

Figure 11. *Critical Difference (CD) Plot for AUC Across Sample Sizes*



AUC between Sample Sizes. The Inman-Davenport correction of Friedman's rank sum test indicated a significant difference between method-sample pairs on AUC scores $\chi^2(13,78) = 44.064$, $p < 0.001$. The Nemenyi post-hoc test indicated a critical difference of 7.7233 (k =14, df = 84, a = 0.05). As with previous analyses, there were no significant differences between base-hybrid/ensemble pairs with different sample sizes aside from XGBoost (S), which significantly outranked ridge regression (S).

Figure 12. *Critical Difference (CD) Plot for AUC Between Sample Sizes*



Summary: Performance Metric Rankings

In summary, XGBoost and random forest classifiers consistently outranked elastic net, lasso regression, ridge regression, decision trees, and logistic regression in terms on recall, balanced accuracy, BA-Recall, and AUC. XGBoost and random forest were in the top ranks across all these metrics. Decision trees appeared in the top ranks for AUC, and logistic regression also appeared in the top ranks for recall and BA-Recall.

Regularized Regression

Hypothesis 1 posited that elastic net would outperform its base methods, lasso regression and ridge regression. This hypothesis was not supported by the data, as elastic net, lasso regression, and ridge regression did not show significantly different performance rankings on BA, BA-Recall, or BA-Specificity. Though not hypothesized, lasso regression significantly outranked ridge regression on AUC. However, because both lasso regression (AUC = 35%) and ridge regression (AUC = 25%) had AUC values below 50%, this finding is not practically meaningful.

Overall, regularized regression methods performed poorly compared to tree-based methods. Interestingly, logistic regression frequently ranked similarly to ridge regression, lasso regression, and elastic net regression, suggesting that regularization does not introduce any added benefit above and beyond logistic regression in applied attrition modeling applications. Findings suggest that regression-based methods are less suited for applied attrition modeling compared to tree-based methods.

Tree-Based Methods

Hypothesis 2 proposed that random forest would outperform its base method, decision trees. This hypothesis was largely supported by the data. Random forest outranked decision trees based across all performance metrics except for AUC, where there was not a significant difference between the two methods. Although the higher rank of random forest compared to decision trees on AUC was not statistically significant, it was practically meaningful. Random forest had an AUC greater than 50% (AUC =70%), whereas decision trees scored below 50% (48%).

Hypothesis 3, which proposed the XGBoost would outperform all other methods, was partially supported. Based on the critical difference plots, random forest performed as well as XGBoost across all metrics. However, XGBoost outperformed random forest on recall substantially. Across all folds and all months, XGBoost successfully identified 20% of positive cases, whereas random forest only correctly identified 9%. Due to the degree of class imbalance in the data, correctly identifying positive instances is more challenging than correctly identifying negative instances, so attention should be paid to recall and BA-recall. Although the difference in performance between XGBoost and random forest finding is not statistically significant based on the Nemenyi post-hoc test, it is practically meaningful.

There is an inherent tradeoff between recall and specificity. When more positive cases are predicted, fewer negative cases are predicted. This is evident in XGBoost and random forests' specificity scores, where XGBoost correctly identified 93% of negative instances compared to 97% for random forest. Table 14 presents information from the confusion matrix, which provides the total count of true and false positive and negative predictions across all months and folds by each method. XGBoost has substantially higher counts of false positives, but it also is closer to predicting the correct absolute number of leavers and stayers.

Although random forest and XGBoost did not rank significantly higher on AUC than decision trees, this discrepancy is likely attributable to decision trees' high AUC score in April—the month utilized for feature selection. Moreover, although the difference is not statistically significant, it is practically significant. AUC measures a classifier's ability to distinguish between positive and negative cases across various classification thresholds. A test demonstrating an AUC value less than .51 would never be used because it does not outperform chance. In summary, Hypothesis 3, which proposed that XGBoost would outperform all other methods, was partially supported. Random forest also performs strongly, but favors the majority class, suggesting that it does not handle class imbalance as effectively as XGBoost. Additional mean performance scores across sample sizes and months are available in Table 12 and 13.

ML vs. Logistic Regression

Hypothesis 4 predicted that all ML algorithms included in this study (lasso, ridge, and elastic net regression, decision trees, random forests, and XGBoost) would demonstrate superior classification performance on out-of-sample attrition data compared to logistic regression. Surprisingly, this hypothesis was not supported by the Friedman tests or by the post-hoc Nemenyi tests. Logistic regression's score on recall or BA-Recall across months was not

statistically significantly different from that of random forest or XGBoost. However, logistic regression had a substantially lower average recall (6%) compared to random forest (9%) and especially compared to XGBoost (21%). While the difference in rankings is not statistically significant, there was a practically meaningful difference in average performance. While logistic regression underperformed relative to random forest and XGBoost, its overall performance was comparable to other regression-based methods.

Modern ML vs. Base Methods

Hypothesis 5 predicted that modern machine learning methods (XGBoost, elastic net, and random forest) would outperform their base methods (decision trees, ridge regression, lasso regression, and logistic regression) at smaller sample sizes. There was no support for this hypothesis from random forest or from elastic net regression, as no significant differences between sample size and the rank performance of base-hybrid/ensemble method pairs were observed for these classifiers. It is likely that larger differences in sample sizes are required to detect any meaningful performance variations among these classifiers. However, XGBoost demonstrated superior rankings compared to decision trees on BA and recall; compared to ridge regression on AUC; and compared to ridge regression and decision trees on BA-Recall. Thus, this hypothesis was partially supported.

Additional Exploratory Analysis: Results Across Thresholds

Table 14 contains a summary of model predictions. TP, TN, FP, and FN refer to the true and false positive and negative predictions made by each model. Net refers to the difference between the number of stayers predicted by the model and the actual number of stayers. Across methods, a surplus of stayers was predicted.

As previously discussed, classification algorithms produce predicted probabilities, which are typically converted into positive or negative predictions using a 50% threshold. However, thresholds can be adjusted to modify the tradeoff between identifying positive instances (recall/sensitivity) and identifying negative instances (specificity). Figure 13 illustrates the ROC curves for each method, which is calculated using the predicted probability and actual outcome for each individual observation (a total of 10123 predicted probabilities for each method). The ROC curve highlights the tradeoff between recall and specificity at various threshold settings. Using predicted probabilities generated by the best performing method, XGBoost, across folds, months, and sample sizes, performance metrics at different thresholds were calculated and are presented in Table 14 to demonstrate how changing the threshold impacts performance metric scores.

As the threshold lowers and the number of true positives increases, the number of false positives also rises, reflecting the tradeoff inherent in adjusting thresholds. Although XGBoost demonstrated strong overall performance, its precision remains limited. By reducing the threshold to 35%, the model can be optimized to provide a more accurate count of employees likely to leave the organization, a valuable tool for workforce planning. However, at this threshold, precision is only 25%, meaning that just 25% of the individuals predicted to leave were correctly identified. While XGBoost performs well with this dataset at an aggregate level and offers promise for applied attrition modeling, its individual-level predictions remain imprecise.

Table 14. *XGBoost Performance Metrics Across Different Thresholds*

Threshold	TP	TN	FP	FN	Precision	Recall	Specificity	Accuracy	F1 Score	Actual Positive	Predicted Positive	Predicted Minus Actual
10%	96	1435	363	131	0.21	0.42	0.80	0.76	0.28	227	459	232
15%	85	1492	306	142	0.22	0.37	0.83	0.78	0.28	227	391	164
20%	76	1537	261	151	0.23	0.33	0.85	0.80	0.27	227	337	110
25%	66	1574	224	161	0.23	0.29	0.88	0.81	0.26	227	290	63
30%	62	1606	192	165	0.24	0.27	0.89	0.82	0.26	227	254	27
35%	**57**	**1624**	**174**	**170**	**0.25**	**0.25**	**0.90**	**0.83**	**0.25**	**227**	**231**	**4**
40%	54	1648	150	173	0.26	0.24	0.92	0.84	0.25	227	204	-23
45%	47	1663	135	180	0.26	0.21	0.92	0.84	0.23	227	182	-45
50%	43	1677	121	184	0.26	0.19	0.93	0.85	0.22	227	164	-63
55%	39	1685	113	188	0.26	0.17	0.94	0.85	0.21	227	152	-75
60%	36	1699	99	191	0.27	0.16	0.94	0.86	0.20	227	135	-92
65%	30	1708	90	197	0.25	0.13	0.95	0.86	0.17	227	120	-107
70%	25	1714	84	202	0.23	0.11	0.95	0.86	0.15	227	109	-118
75%	22	1724	74	205	0.23	0.10	0.96	0.86	0.14	227	96	-131
80%	17	1734	64	210	0.21	0.07	0.96	0.86	0.11	227	81	-146
85%	11	1747	51	216	0.18	0.05	0.97	0.87	0.08	227	62	-165
90%	9	1755	43	218	0.17	0.04	0.98	0.87	0.06	227	52	-175

Figure 13. *ROC by Method Across all Months and Sample Sizes*

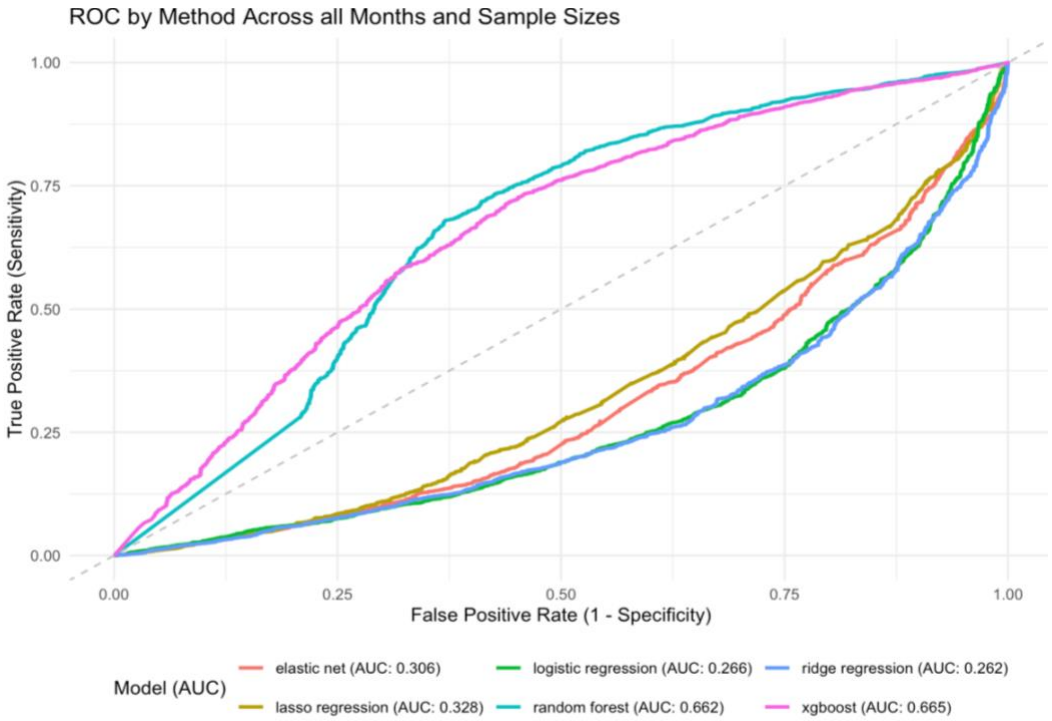
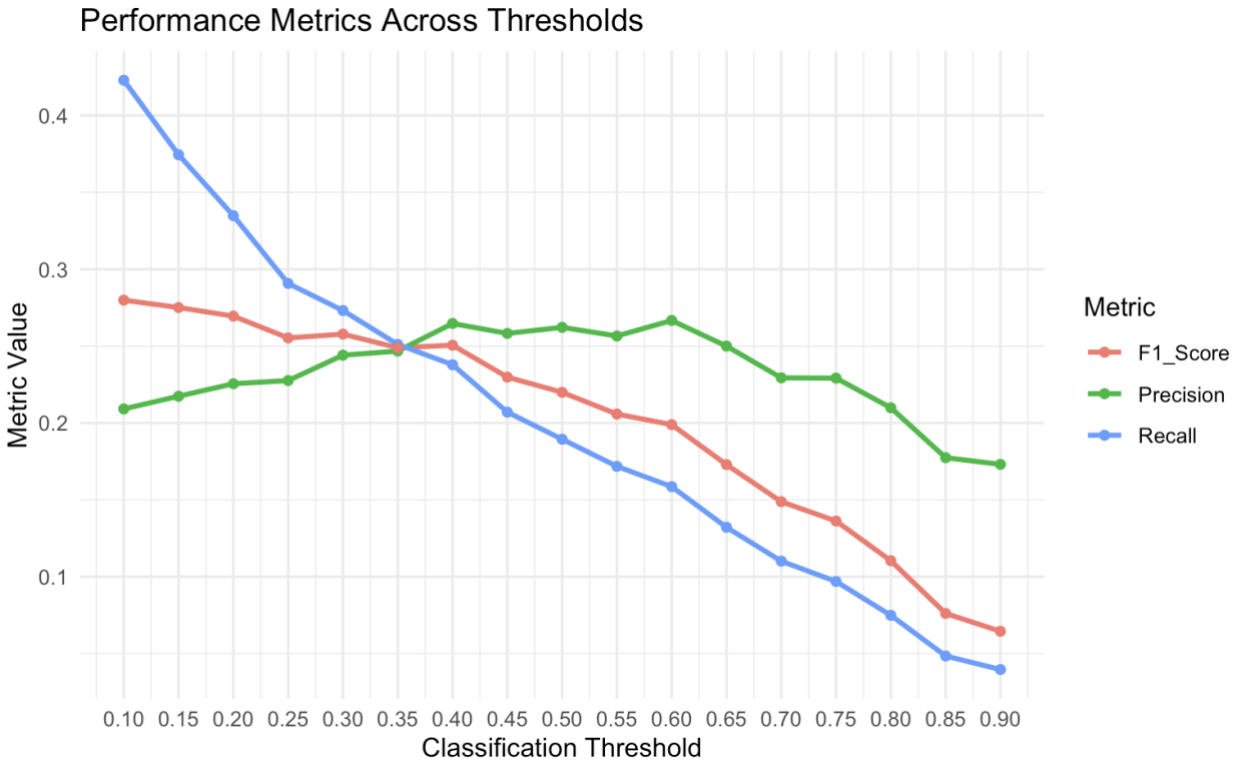


Figure 14. *XGBoost Performance Metrics Across Thresholds*

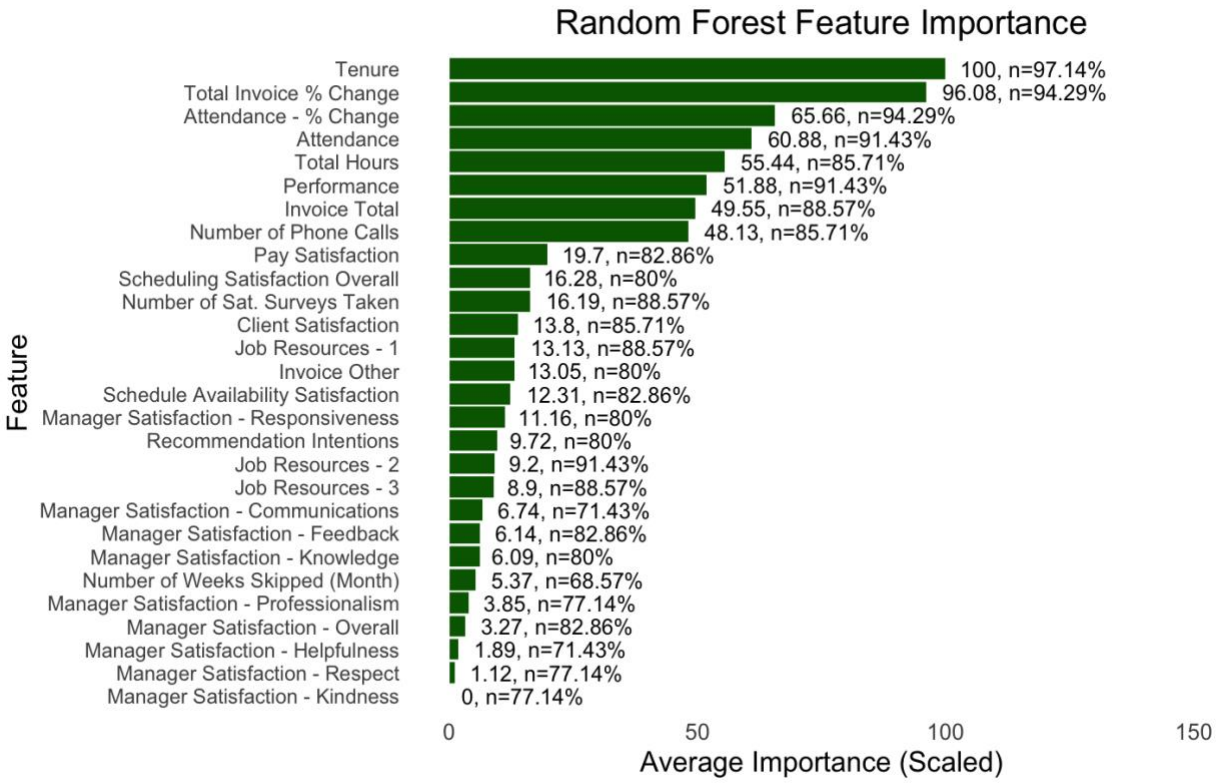


Feature Importance Results

Feature Importance – Tree-Based Methods

An exploratory research question was posed to examine how feature importance scores vary across methods. Random forest, XGBoost, and decision trees produce feature importance scores which indicate the extent to which each feature contributes to the quality of its splits. For XGBoost, three metrics were generated to assess feature importance: gain, cover, and frequency. Gain measures each feature's contribution to the model's predictive ability via informing splits, cover reflects the number of observations associated with a feature across the trees, and frequency indicates the proportion of trees where a feature appears (Chen et al., 2024). Of these metrics, gain was selected for interpretation as it provides a closer comparison to random forest's feature importance scores. Moreover, the results from gain were similar to those of cover and frequency. Using results from n = 1000 samples, I normalized feature importance scores on a scale of 0-100 and summarized them in Figures 15-16. Features are listed on the Y axis, and average importance score is represented on the X axis.

Figure 15. *Random Forest Feature Importance Across Months*



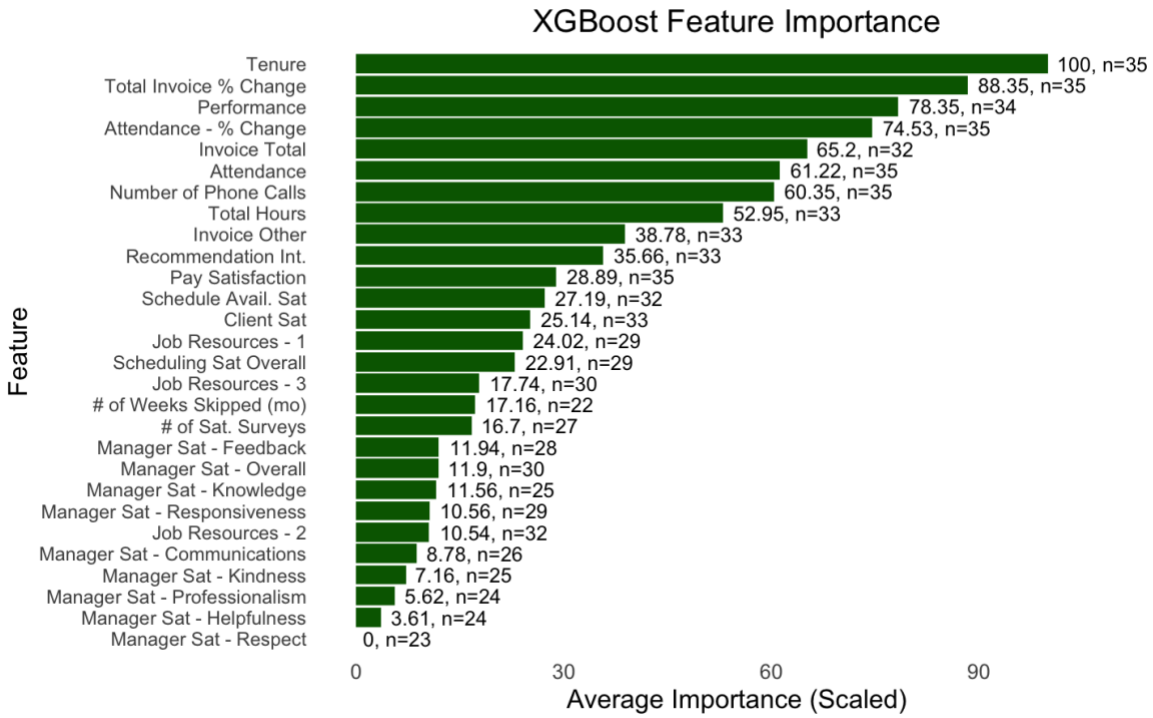
Note. Feature importance values have been scaled 0-100. Numbers next to the bars represent the mean, and n = represents that percent of time they were used (not dropped).

Decision Trees. In the case of decision trees, not all folds resulted in fully developed branches, which led to inconsistent feature importance rankings. While random forests produced 70 importance rankings for each feature, decision trees yielded only one or two rankings. Due to this inconsistency and the limited interpretability, feature importance rankings from decision trees were excluded from further analysis.

XGBoost vs. Random Forest. To compare the significance of feature importance rankings, I performed a paired Wilcox signed ranks test on the average rank value (for $n = 1000$, rankings within month, normalized on a 0-100 scale) of each feature between the two methods. Overall, feature rankings were similar between random forest and XGBoost. With a Bonferroni

correction for multiple comparisons, there were no significant differences in average rankings (p = 1.00).

Figure 16. *XGBoost Variable Importance Across Months*

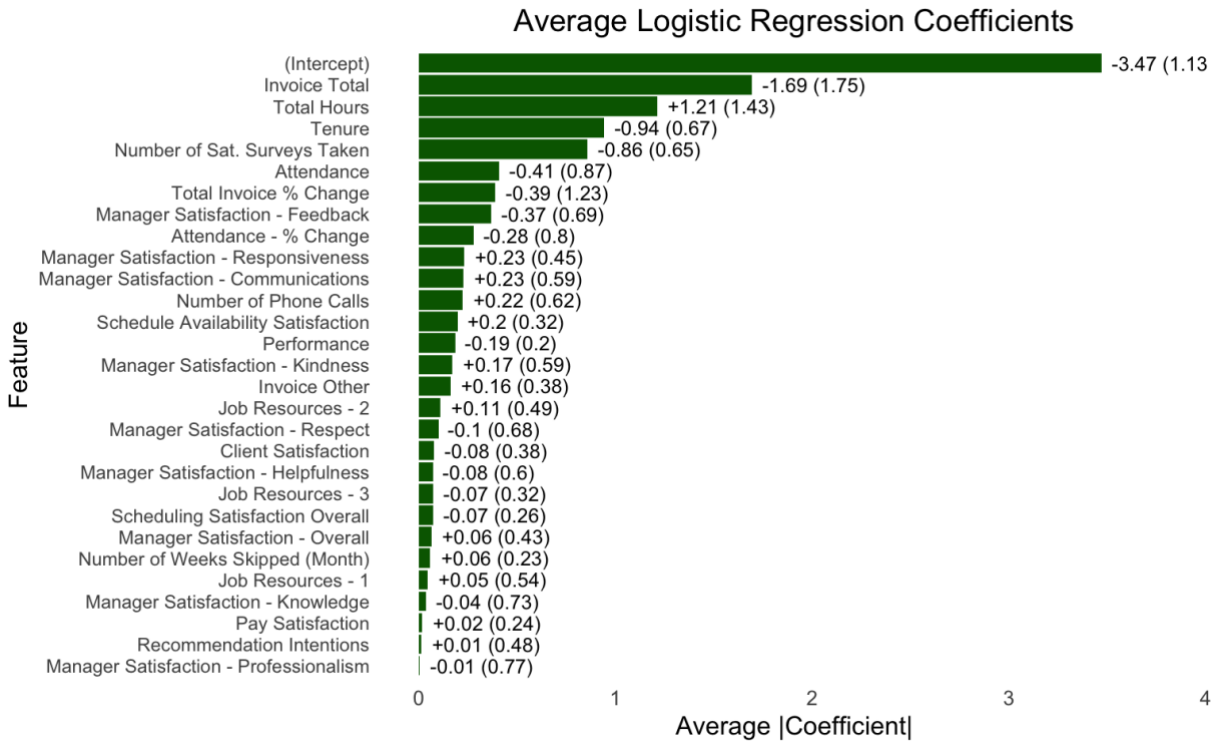


Feature Importance –Regression

Regularized regression and logistic regression do not have a direct comparison for variable importance. Instead, I report means and standard deviations of regression coefficients (β) across models. Note that absolute values are used with a + or – sign to indicate directionality.

Logistic Regression Model Coefficients. Figure 17 visualizes the mean and standard deviation of each feature coefficient across logistic regression models. Overall, many of the features with the strongest coefficients are HRIS features. However, the only feature with a β approaching a significant effect at an a = 0.05 confidence level Number of Satisfaction Surveys Taken, (M = 0.86, SD = 0.65, p = 0.07).

Figure 17*. Logistic Regression Model Coefficients*



Ridge, Lasso, and Elastic Net Regression Model Coefficients. Figures 18, 19, and 20 plot the average regression coefficients for ridge, lasso, and elastic net regression across all models with n = 1000 sample sizes. The first number to the right of the bar indicates the average coefficient for that respective feature. The number in parentheses represents the standard deviation and the percentage indicates the percent of the time they were included in the model. Unlike logistic regression, regularized regression tests do not provide an explicit *p*-value; rather, significance is determined by the size of the coefficient and/or whether it was dropped.

Intercepts are not plotted, but were equal to -2.24, -2.36, and -2.35 for elastic net, lasso, and ridge regression, respectively. Based on these intercepts and the average β coefficients, we could calculate the probability of the positive event occurring based on the value of the feature,

holding all else constant. However, these probabilities would be more meaningful coming from a model with better predictive accuracy. Due to the poor performance of these models, I refrain from interpreting their β coefficients.

Across methods, coefficients are similar in rank order. Logistic regression has overall higher coefficients with wider standard deviations overall, as is to be expected since it does not incorporate regularization and experiences more instability in coefficient values compared to regularized methods. In practice, more care should be taken in paring down the list of features when using logistic regression, particularly those with strong correlations.

Figure 18. *Average Ridge Regression Model Coefficients*

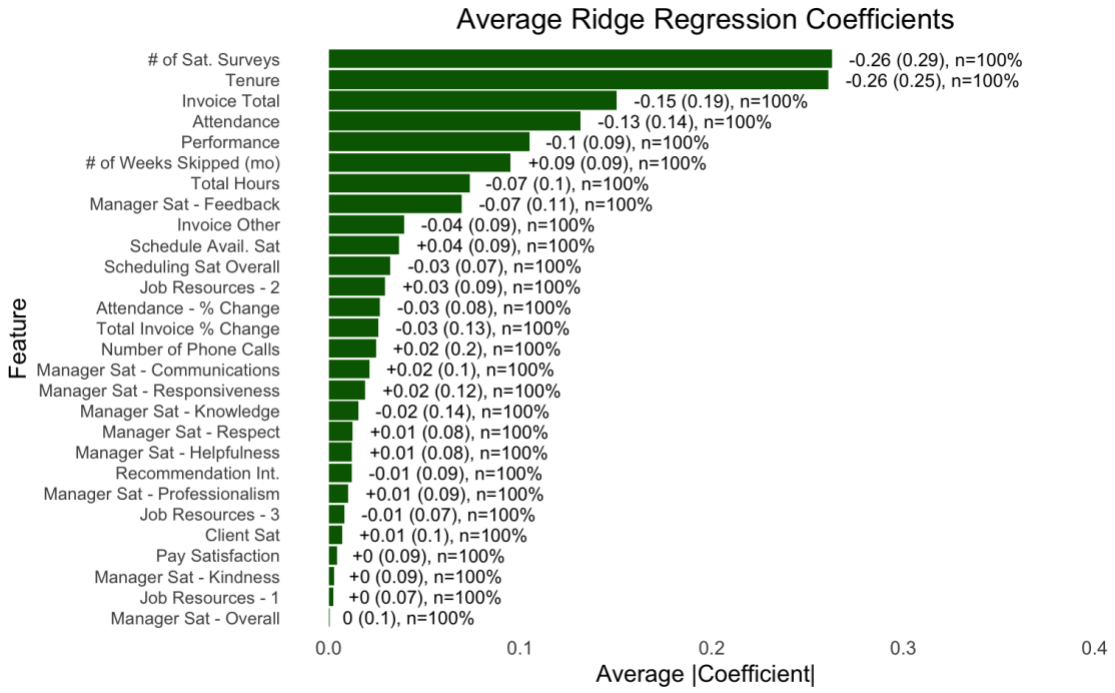


Figure 19. *Average Lasso Regression Model Coefficients*

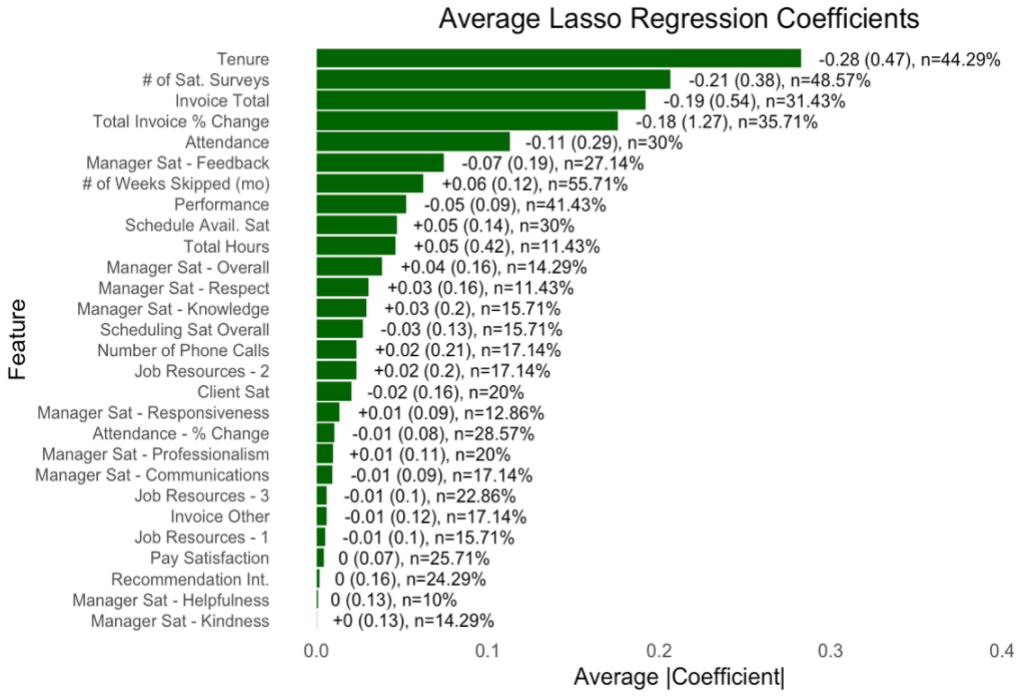
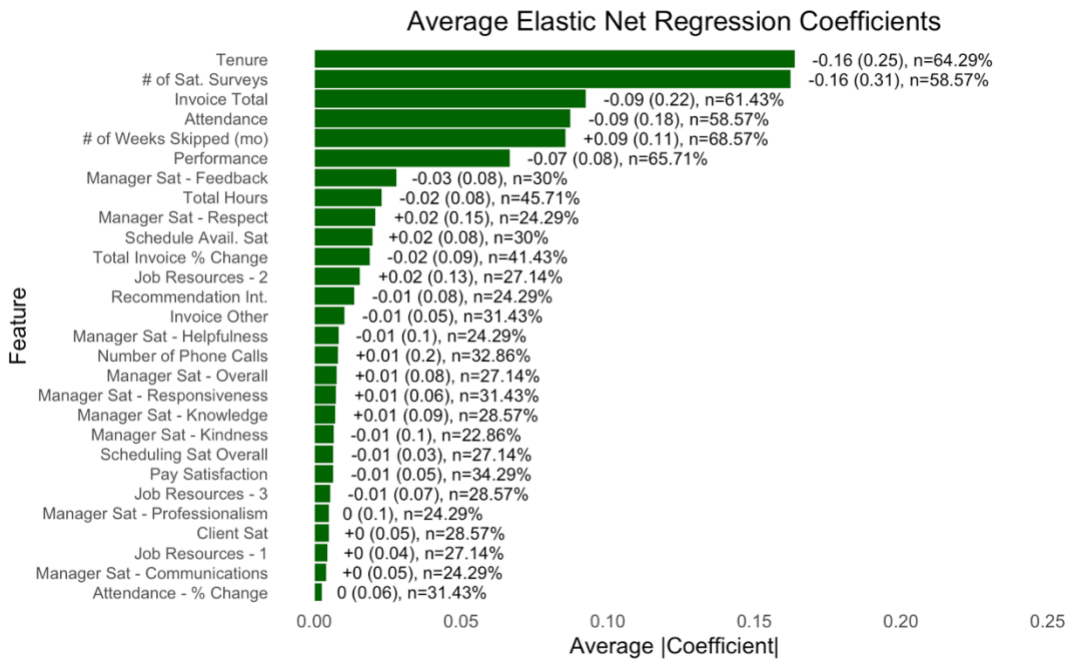


Figure 20. *Average Elastic Net Regression Coefficients*



CHAPTER 6

DISCUSSION

This dissertation evaluates the comparative performance of two prominent machine learning approaches: tree-based methods and regularized regression methods. Both base methods (logistic regression, lasso regression, ridge regression, and decision trees) and hybrid/ensemble techniques (random forest, elastic net, XGBoost) were evaluated. Across various metrics, XGBoost and random forest consistently outperformed the incumbent method, logistic regression, as well as decision trees, ridge regression, and lasso regression. First, these findings suggest that researchers and practitioners may benefit from using newer, more complex ML algorithms over their base methods or logistic regression. Second, findings suggest that ensemble methods are better suited for applied attrition modeling compared to base methods for datasets with a similar *k* (~28 features) and n *(*n = 500-1000). Third, tree-based ensemble methods significantly outperformed regularized regression methods, suggesting that tree-based methods are better suited for applied attrition modeling.

Practical Implications

Although a resampling method (SMOTE) was employed during the cross-validation process to address class imbalance in training folds, all models still underpredicted the number of positive cases (employee attrition). This highlights a key area for improvement, suggesting that a lower classification threshold should be tested via model calibration, which is the process of testing various cutoff thresholds. As demonstrated with XGBoost, lowering the threshold from 50% would increase sensitivity by capturing more true positive cases, albeit at the cost of reduced specificity. In practice, researchers and industry professionals can use thresholds to adjust the model based on the organization's tolerance for false positives versus false negatives.

An organization most concerned about over hiring would opt for a higher threshold than an organization concerned about under hiring, as missing potential positive cases would lead to insufficient replacement hiring and increased workload for existing staff. If avoiding false positives is critical, for example, when organizations face budget constraints that prevent over-hiring, a lower threshold may be optimal.

Compared to other methods included in this study, XGBoost was the most effective at correctly identifying true positive instances across different sample sizes and months, as evidenced by its mean true positive rate. This makes XGBoost a strong choice for organizations aiming to identify as many at-risk employees as possible. However, the higher recall exhibited by XGBoost comes at the expense of lower specificity because more instances are classified as positive overall (see Table 11). Random forest also offers the advantages of simplicity, interpretability, and computational efficiency, making it a practical choice when these factors are prioritized alongside balanced performance across metrics.

Even the best performing models, XGBoost and random forest, are not yet full optimized for deployment in an applied organizational setting. While XGBoost outperforms a random guess, or an educated guess based on how many employees left the previous month—the gains, though meaningful, are incremental. For instance, in a sample size of 1000 employees, XGBoost predicted that 89 employees would leave in October, while 97 actually left. Compared to an educated guess based on the 73 leavers in September, XGBoost would leave us at -8 and versus -24. However, precision remains a concern, as only 30% of the predicted positive cases are correct based on average precision scores. Further refinements are needed to produce models that are ready for deployment in organizations. To improve the operational utility of these models, further hyperparameter tuning would be needed. The hyperparameters outlined in Appendix A

offer a foundational starting point for these efforts, providing a guide for future iterations and refinements of the model.

Different sample sizes were investigated to compare conditions that represent practical limitations faced by organizations. Hypothesis 5 stated that random forest, elastic net, and XGBoost would outperform their base models most markedly at small sample sizes. This hypothesis was partially supported; no significant relationships between sample size and rank performance of base-hybrid/ensemble method pairs were observed aside from XGBoost, which outranked decision trees on recall, BA, and BA-Recall, and outranked ridge regression on BA-Recall and AUC in small sample size conditions. This finding is not surprising, given that XGBoost emerged as the top performer and that there were only slight differences between the rank performance of other methods. Moreover, this finding is in line with previous research, which suggests that modern methods outperform base methods in small samples (Landers et al., 2024).

Interestingly, there was a slight (but nonsignificant) performance difference among base methods at small and large sample sizes, favoring smaller samples. For example, based on AUC, every method except for decision trees performs slightly better in small sample sizes. Based on precision, decision trees, lasso, logistic and random forest perform better at small sample sizes. However, XGBoost performs equally well across sample sizes, with just a one-point differences on recall (recall = 0.21 when n = 500; recall 0.20 when n = 1000). These results indicate that XGBoost performs more robustly on small samples compared to other methods, suggesting that XGBoost effectively mitigates the main drivers of performance inconsistency, including random variance, overfitting, and class imbalance. This is likely due to its boosting mechanics rather than

regularization or tree-based structure, given that the other methods with these characteristics did not perform as effectively.

Feature Importance: Comparisons to Prior Work and Theoretical Significance

Results explored feature importance rankings across ML methods, highlighting the most influential predictors of voluntary attrition. To my knowledge, no prior research has integrated performance data, HRIS data, and self-reported data to train attrition models using ML techniques. The results of this study indicate that feature importance rankings were largely consistent between random forest and XGBoost, reinforcing the reliability of these predictors. Features related to tenure, performance, and pay consistently emerged as the strongest indicators of attrition risk, aligning with previous research on voluntary turnover.

Among the top-ranked predictors, tenure demonstrated the highest importance in both random forest and XGBoost models, with a negative correlation with attrition (r = -.19, $p <$ 0.01), suggesting that longer-tenured employees were less likely to leave. This aligns loosely with findings from Rubenstein et al.'s (2018) meta-analysis, which reported a meta-analytic point biserial correlation of r = 0.20 (r = -0.27 when excluding an outlier). The consistency between this study and prior research reinforces the well-documented empirical and theoretical relationship between tenure and turnover, where longer-tenured employees accumulate firm-specific knowledge, develop stronger workplace ties, and face higher opportunity costs associated with leaving.

Another key finding concerns the relationship between manager satisfaction and voluntary attrition, which was notably weaker than prior research might suggest. In this study, point-biserial correlations between attrition and manager satisfaction were small and counterintuitive, averaging around r = 0.08. By contrast, Rubenstein et al. (2018) found that job

satisfaction, which includes managerial satisfaction as a component, had a much stronger meta-analytic correlation with attrition (r = -0.25). This discrepancy raises important questions about the role of job attitudes in high-turnover roles such as customer service. One possible explanation is that employees in these roles do not enter them expecting high levels of managerial support or intrinsic job satisfaction. Unlike workers in other sectors with higher average tenure, customer service employees may be less motivated by their enjoyment of the work itself.

This study also revealed important findings related to pay and scheduling stability, which further differentiate voluntary attrition patterns in customer service jobs from those in more stable professions. Pay satisfaction exhibited a weaker (and counterintuitive) relationship with attrition (r = .08, $p < 0.01$) compared to Rubenstein et al.'s meta-analytic estimate of r = -.17, again suggesting that financial considerations in this dataset were more complex than a simple dissatisfaction-to-turnover pathway. Instead, behavioral indicators such as Total Invoice % Change emerged as stronger predictors based both on feature importance scores and the bivariate correlation between Total Invoice % Change and attrition (r = -.19, $p < 0.01$). Specifically, employees who experienced a decrease in their total invoiced hours from month to month were slightly more likely to leave, a pattern consistent with research on the negative effects of schedule instability in frontline service jobs (Choper et al., 2021). This suggests that the unpredictability of earnings – rather than absolute pay satisfaction – was a primary driver of attrition, reinforcing findings that financial stability is often a stronger predictor of retention than static pay levels in hourly workforces (Henly & Lambert, 2010).

Another interesting finding relates to attendance and total hours worked, both of which ranked highly in feature importance. Attendance, representing the extent to which employees

showed up to work within their scheduled time blocks, was among the most predictive features based on feature importance and based on its strong bivariate correlation with attrition (r = -0.19, $p < 0.01$). This negative bivariate correlation suggests that employees with higher levels of attendance (the percent of time that they worked their scheduled shifts) were less likely to attrit. This finding is consistent with turnover research demonstrating that withdrawal behaviors often precede voluntary attrition (Hom et al., 2017). Overall, many of the most important features were indicative of how much an employee had been present in the previous month. This finding is aligned with research on pre-quitting behaviors, defined as "behavioral changes reflecting progression through the turnover process that (a) observers can notice and (b) are associated with future turnover behavior" (Gardner et al., 2018, p. 3224). Essentially, pre-quitting behaviors are behavioral indicators that an employee may have already decided to quit. Thus, the algorithms likely identified individuals who have already made the conscious decision to leave the organization via their disengagement or "pre-quit" behaviors.

In contrast, employees who consistently worked a greater number of hours exhibited lower attrition rates (Total Hours, r = -.21, $p < 0.01$), likely reflecting a stronger attachment to the organization and their intention to continue work. This aligns with job embeddedness theory (Mitchell et al., 2001), which posits that employees who are more integrated into their work environment through stable schedules, greater work commitments, and stronger financial resilience, are less likely to quit. When compared to meta-analytic findings, these findings suggest that turnover in customer service roles follows a distinct pattern from turnover in higher-tenured professions.

Cross-Model Comparisons

Employee turnover research has long theorized and researched the complex cognitive, behavioral, and situation-dependent processes that ultimately result in voluntary employee attrition. The attrition decision process is long and complex, and relationships between predictor variables may be moderated and mediated by one another and/or may be nonlinear. The results of this dissertation lend support for the complexity of this process. Across various performance evaluation metrics, XGBoost and random forest algorithms, both tree-based methods, consistently outperformed the incumbent method, logistic regression, and the other regression-based methods. Regression-based models, including logistic regression, elastic net regression, lasso regression, and ridge regression significantly underperformed random forest and XGBoost. Methods which incorporate regularization were expected to outperform logistic regression, but it appears that regularization did not play a significant role in model performance. One interpretation of this null finding could be that the feature selection employed before running models could have removed the advantages of regularized regression. However, the feature selection used was very conservative, retaining feature correlations as high as $r = .79$. Given that regression-based methods performed poorly overall, the most likely explanation is that these methods could not adequately account for the complex relationships between features with other features and with the outcome.

Ethical Implications

There are several importance ethical and legal considerations stemming from the implementation of an attrition modeling algorithm, depending on how it is used. The purpose of the attrition modeling methodologies presented here is to arrive at an estimated number of total attrits with the highest possible certainty. However, the use of such models in areas like promotions and compensation introduces the potential for adverse impact, even if demographic

data is not explicitly included as a feature. For example, a model that results in higher turnover probabilities for women could lead to unintentional biases in pay or promotion decisions, even in the absence of gender as a variable (Castille & Castille, 2019). This is a significant limitation to the present work; demographic data was not available. Researchers interested in implementing attrition modeling for such uses ought to consider the potential legal consequences of doing so when designing and evaluating attrition models and must evaluate their outcomes for adverse impact. Speer (2024) outlines procedures for practitioners to test their attrition models for adverse impact and provides recommendations for reducing adverse impact if it is indeed found.

Another critical ethical consideration is the use of employee surveillance in data collection for attrition modeling. While electronic monitoring may provide valuable insights, research suggests that excessive surveillance can have negative consequences for employee morale and organizational culture. Thiel et al. (2022) found that electronic monitoring undermines the positive influence of leaders on employees, eroding trust and diminishing engagement. If employees perceive that they are being excessively monitored, it may foster feelings of distrust and reduce job satisfaction, potentially exacerbating. Organizations must weigh the benefits of monitoring against these risks and ensure that data collection methods respect employees' privacy and autonomy.

Limitations and Directions for Future Research

The largest limitation to the current work is the scope. Other methods of hyperparameter tuning (random hyperparameter search), resampling (under-sampling, over-sampling), and model optimization (cost-sensitive learning with weights, optimizing for Kappa or ROC) are available, but could not all be tested in this work. Specifically, I encourage future researchers to integrate cost-sensitive learning into their models. Cost-sensitive learning is a method of addressing the

majority-class bias demonstrated by classifiers in the case of class imbalance. It works by applying a high performance "cost" to misclassifying the minority class (Provost, 2000; Elkan, 2001). Cost-sensitive learning can be applied in tandem with oversampling techniques, providing a more robust solution to class imbalance than resampling alone (Shewach et al., 2024).

This work tested only a small subset of the available ML algorithms. Other promising methods include ad-hoc ensemble models, where researchers can combine multiple types of classifiers to create a custom model. Ensemble models have been found to be highly effective and have won SIOP's machine learning competition in the past. Indeed, ensemble models were the best-performing methods in this study. Additionally, researchers can build on this work by comparing additional tree-based boosting methods to random forest and XGBoost. I recommend future researchers investigate uses of LightGBM. LightGBM was designed as a highly efficient alternative to XGBoost which can demonstrate superior speed and predictive accuracy in high dimensional, large datasets (Ke et al., 2017). Similarly to XGBoost, it LightGBM is a tree-based boosting algorithm which incorporates L2 regularization. However, its implementation Is optimized for speed and memory on large datasets, making it a great candidate for use with large attrition datasets. Another recent development is CatBoost, which is a tree-based gradient boosting algorithm similar to XGBoost but is optimized for datasets with categorical features, making it a strong option for datasets with categorical features like department, manager, etc. (Prokhorenkova et al., 2018). Both CatBoost and LightGBM are faster and more memory-efficient than XGBoost, making them great options in organizational contexts when resources are limited. Another interesting and novel boosting algorithm is SnapBoost, which varies the type of base learner at each implementation (Parnell et al., 2020). Specifically, SnapBoost randomly chooses between a decision tree or a linear model at each boosting step, allowing it to

combine the strengths of trees and linear models within a single ensemble (Parnell et al., 2020). This novel approach may prove highly useful in applied attrition modeling, as some feature-outcome relationships may demonstrate linear associations. In summary, no single algorithm will work well across all use cases (Wolpert & Macready, 1997), so I encourage future researchers to explore various options for applied attrition modeling.

Several limitations of the present study pertain to the dataset. As noted in the Ethical Implications section, demographic data was not available, which limits the ability to examine how attrition patterns may vary across different groups. Additionally, many of the predictive features in the dataset are inherently tied to an individual's tenure at the organization and their presence in the previous month. Prior research has consistently shown that newer employees are at a higher risk of attrition (Hom et al., 2017), and the models in this study largely capitalize on this relationship. The most influential features—tenure, total invoice percent change, attendance, attendance percent change, and total hours—serve as indicators of whether an employee was active in the preceding month. As a result, these models may be less effective in identifying attrition risk for longer-tenured employees or for cases where attrition is driven by factors unrelated to recent attendance patterns. Future work should explore the inclusion of more static features, such as job role or department, to improve predictive performance across a broader range of employees.

Another key direction for future research involves improving the usability of organizational data for ML models. The process of identifying and extracting useful data is the most time-consuming part of attrition modeling. The data typically used in organizations often requires significant data engineering, from addressing missing values to consolidating records from multiple sources. Simulated datasets used in academic research bypass many of these

challenges, but they may fail to capture the complexities of real-world organizational data or produce overly optimistic correlations between predictors and outcomes. Future studies could focus on developing techniques that enhance the quality and usability of organizational datasets. Moreover, using methods which identify messy data (redundancies, multicollinearity, mixed data types) with user supervision would expedite the process of getting from model ideation to model creation and exploration.

Future researchers may wish to investigate different prediction windows. For instance, attrition could be predicted over a 6-month period rather than over a 1-month period. Using a larger window would drastically improve the degree of class imbalance in the outcome, as more people attrit over a longer period. However, there are additional challenges associated with using a larger window, such as handling missing data from employees who start work during that time.

There are several additional temporal factors that may warrant consideration in applied attrition modeling depending. For example, patterns of attrition may change significantly under different economic circumstances. The presence of viable job alternatives, for example, has been found to positively predict attrition (Rosenbusch et al., 2018). Beyond economic conditions, industry-specific cycles can influence turnover rates, necessitating the inclusion of domain knowledge when constructing predictive models. For instance, academic institutions may experience increased faculty turnover at the end of academic years, whereas industries like retail or hospitality may experience higher seasonal turnover post-holiday or during off-peak seasons. Similarly, performance review cycles, fiscal year transitions, and organizational restructuring events can introduce attrition spikes that would be overlooked in a purely static model.

To account for these effects, categorical variables representing economic trends, seasonality, and industry-specific events can be introduced into attrition models. Researchers

must ensure that a long enough window is available to capture longitudinal patterns. Incremental learning can also help. Incremental learning is a machine learning approach where a model continuously updates itself as new data becomes available rather than being retrained from scratch. Incremental learning is especially useful in attrition modeling because workforce dynamics, economic conditions, and organizational trends change over time, making it impractical to rely solely on static models trained on historical data. By enabling models to adapt to new information without forgetting previously learned patterns**,** incremental learning ensures that seasonal effects, economic shifts, and industry-specific trends are incorporated in real time**.** At the time of writing this dissertation, the available ML packages in R do not support incremental learning. Researchers may opt for other platforms for model building such as Python.

Implications for Turnover Theory: Integrating ML-Driven Insights into Theoretical Models

Finally, the present work was primarily focused on methodology rather than theoretical contribution. However, the two do not have to be opposed (Shumeli, 2013). The emergence of predictive modeling presents an opportunity to refine and challenge existing turnover frameworks, including the Unfolding Model (Lee & Mitchell, 1994) and Job Embeddedness Theory (Mitchell et al., 2001). These frameworks emphasize the complex decision-making process that leads up to voluntary attrition. ML algorithms, particularly tree-based algorithms, can capture complex, nonlinear relationships in real-time workforce data, allowing researchers to better understand the relationships between predictors and “steps” in the decision-making process. Such methods may also allow researchers to uncover relations among features not captured by more rigid methods. Current methodologies, like SEM, produce insights which are

model-driven. In contrast, ML methods are largely data-driven, allowing for unexplored or unexpected feature relationships to emerge.

Conclusions

Over the past decade, there has been a growing movement toward applying predictive models in psychology, with researchers like Rosenbusch et al. (2021) and Pargent et al. (2023) providing accessible guidelines for using supervised ML algorithms. These models not only offer the potential for organizations to develop more effective interventions but can also uncover previously unexplored psychological constructs. Integrating both predictive and explanatory models, as advocated by Yarkoni & Westfall (2017), would allow organizational researchers to advance both theory and practice, addressing real-world problems while contributing to the broader understanding of employee behavior.

This dissertation demonstrates the application of predictive techniques to address a longstanding research topic in the organizational sciences. Findings suggest that the use of ML algorithms, and XGBoost and Random Forest in particular, can improve predictive accuracy over logistic regression and over their base methods. Results also suggest that these algorithms utilized information from all three data sources – HRIS records, performance evaluations, and self-report surveys – to arrive at predictions. However, further refinement of these models, particularly through hyperparameter tuning and threshold optimization, is necessary to make them fully applicable in practice. While the incremental improvements over traditional methods are clear, organizations must weigh these gains against the practical and computational resources required to implement ML solutions effectively.

REFERENCES

Adams, G., & Beehr, T. (1998). Turnover and Retirement: A Comparison of Their Similarities and Differences. *Personnel Psychology*, *51*, 643–665. https://doi.org/10.1111/j.1744-6570.1998.tb00255.x

Akasheh, M. A., Malik, E. F., Hujran, O., & Zaki, N. (2024). A decade of research on machine learning techniques for predicting employee turnover: A systematic literature review. *Expert Systems with Applications*, *238*, 121794. https://doi.org/10.1016/j.eswa.2023.121794

Allen, D. G., Hancock, J. I., Vardaman, J. M., & Mckee, D. L. N. (2014). Analytical mindsets in turnover research. *Journal of Organizational Behavior, 35*(S1), S61-S86. https://doi.org/10.1002/job.1912

Anderson, J. C., & Gerbing, D. W. (1988). Structural equation modeling in practice: A review and recommended two-step approach. *Psychological Bulletin, 103*(3), 411-423. https://doi.org/10.1037/0033-2909.103.3.411

Aquino, K., Griffeth, R. W., Allen, D. G., & Hom, P. W. (1997). Integrating justice constructs into the turnover process: A test of a referent cognitions model. *Academy of Management Journal, 40*(5), 1208-1227. https://doi.org/10.2307/256933

Batista, G. E. A. P. A., & Monard, M. C. (2003). An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, *17*(5–6), 519–533. https://doi.org/10.1080/713827181

Becker, W., & Cropanzano, R. (2011). Dynamic aspects of voluntary turnover: An integrated approach to curvilinearity in the performance-turnover relationship. *Journal of Applied Psychology, 96*, 233–246. https://doi.org/10.1037/a0022041

Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of XGBoost. *Artificial Intelligence Review*, *54*(3), 1937–1967. https://doi.org/10.1007/s10462-020-09896-5

Berk, R. A. (2006). An introduction to ensemble methods for data analysis. *Sociological Methods & Research*, *34*(3), 263–295. https://doi.org/10.1177/0049124105283119

Bolt, E. E. T., Winterton, J., & Cafferkey, K. (2022). A century of labour turnover research: A systematic literature review. *International Journal of Management Reviews*, *24*(4), 555–576. https://doi.org/10.1111/ijmr.12294

Breaugh, J. A. (2014). Breaugh, J. A. (2014). Predicting voluntary turnover from job applicant biodata and other applicant information. *International Journal of Selection and Assessment, 22*(3), 321–332. https://doi.org/10.1111/ijsa.12080

Breiman, L. (1984). *Classification and regression trees*. Routledge. https://doi.org/10.1201/9781315139470

Breiman, L. (2001). Random forests. *Machine Learning, 45,* 3-32. https://doi.org/10.1023/A:1010933404324

Briscoe, E., & Feldman, J. (2011). Conceptual complexity and the bias/variance tradeoff. *Cognition*, *118*(1), 2–16. https://doi.org/10.1016/j.cognition.2010.10.004

Castille, C. M., & Castille, A. M. R. (2019). Disparate treatment and adverse impact in applied attrition modeling. *Industrial and Organizational Psychology*, *12*(3), 310-313.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. Computers & electrical engineering, 40(1), 16-28.

Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, *6*(1), 1–6.

https://doi.org/10.1145/1007730.1007733

Chen, T., & Guestrin, C. (2016). XGBoost: a scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchen, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y., Li, Y., & Yuan, J. (2024). Xgboost: Extreme gradient boosting. R package version 1.7.8.1.

Choper, J., Schneider, D., & Harknett, K. (2022). Uncertain time: Precarious schedules and job turnover in the US Service Sector. Industrial & Labor Relations Review, 75(5), 1099–1132. https://doi.org/10.1177/00197939211048484

Choudhary, R., & Gianey, H. K. (2017). Comprehensive review on supervised machine learning algorithms. *2017 International Conference on Machine Learning and Data Science (MLDS)*, 37–43. https://doi.org/10.1109/MLDS.2017.11

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, *7*, 1-30.

Ding, Y., & Simonoff, J. S. (2010). An investigation of missing data methods for classification trees applied to binary response data. Journal of Machine Learning Research, 11, 131–170.

Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2020). A survey on ensemble learning. *Frontiers of Computer Science*, *14*(2), 241–258. https://doi.org/10.1007/s11704-019-8208-z

Dormann, C. F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., ... & Lautenbach, S. (2013). Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. Ecography, 36(1), 27-46.

Feldman, P. H. (1994). ‘Dead end' work or motivating job? Prospects for frontline paraprofessional workers in LTC. *Generations: Journal of the American Society on Aging, 18*(3), 5–10

García, S., Fernández, A., Luengo, J., & Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, *180*(10), 2044–2064. https://doi.org/10.1016/j.ins.2009.12.010

Garcia, N., Strzoda, R., Lucca, G., & Borges, E. (2022). A performance analysis of classifiers on imbalanced data: Proceedings of the 24th International Conference on Enterprise Information Systems, 602–609. https://doi.org/10.5220/0011089100003179

Gardner, T. M., Van Iddekinge, C. H., & Hom, P. W. (2018). If you’ve got leavin’ on your mind: The identification and validation of pre-quitting behaviors. Journal of Management, 44(8), 3231–3257. https://doi.org/10.1177/0149206316665462

González, S., García, S., Del Ser, J., Rokach, L., & Herrera, F. (2020). A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. *Information Fusion,* 64, 205-237.

Griffeth, R. W., Hom, P. W., & Gaertner, S. (2000). A meta-analysis of antecedents and correlates of employee turnover: Update, moderator tests, and research implications for the next millennium. *Journal of Management*, *26*(3), 463-488.

Gupta, N., Smith, J., Adlam, B., & Mariet, Z. E. (2022). Ensembles of classifiers: a bias-variance perspective. *Transactions on Machine Learning Research*.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data*

mining, inference, and prediction (2nd ed.). Springer. https://doi.org/10.1007/978-0-387-84858-7

Hayes, A. F. (2013). Mediation, moderation, and conditional process analysis. Introduction to mediation, moderation, and conditional process analysis: A regression-based approach, 1(6), 12-20.

Henly, J. R., & Lambert, S. J. (2014). *Unpredictable work timing in retail jobs: Implications for employee work-life conflict*. University of Chicago, Employment Instability, Family Well-being, and Social Policy Network. https://cpb-us-w2.wpmucdn.com/voices.uchicago.edu/dist/3/1174/files/2018/06/univ_of_chicago_work_scheduling_manager_report_6_25_0-1gq8rxc.pdf

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics, 12(1),* 55-67.https://doi.org/10.1080/00401706.2000.10485983

Hom, P. W., Lee, T. W., Shaw, J. D., & Hausknecht, J. P. (2017). One hundred years of employee turnover theory and research. *Journal of Applied Psychology*, *102*(3), 530–545. https://doi.org/10.1037/apl0000103

Huselid, M. A., & Day, N. E. (1991). Organizational commitment, job involvement, and turnover: A substantive and methodological analysis. *Journal of Applied Psychology, 76*(3), 380–391. https://doi.org/10.1037/0021-9010.76.3.380

Jadhav, A., Pramod, D., & Ramanathan, K. (2019). Comparison of performance of data imputation methods for numeric dataset. *Applied Artificial Intelligence*, *33*(10), 913–933. https://doi.org/10.1080/08839514.2019.1637138

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). Support vector machines. In G. James, D. Witten, T. Hastie, R. Tibshirani, & J. Taylor, *An introduction to*

statistical learning (pp. 367–398). Springer International Publishing. https://doi.org/10.1007/978-3-031-38747-0_9

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, *6*(5), 429–449. https://doi.org/10.3233/IDA-2002-6504

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 1-9.

Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised Learning. *International Journal of Computer Science, 1*(1).

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160(1), 3-24.

Kristof-Brown, A., & Guay, R. P. (2011). Person–environment fit. In S. Zedeck (Ed.), APA handbook of industrial and organizational psychology, Vol. 3. Maintaining, expanding, and contracting the organization (pp. 3–50). American Psychological Association. https://doi.org/10.1037/12171-001

Kuhn, M., & Wickham, H. (2020). *Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles.*. https://www.tidymodels.org.

Landers, R. N., Auer, E. M., Dunk, L., Langer, M., & Tran, K. N. (2023). A simulation of the impacts of machine learning to combine psychometric employee selection system predictors on performance prediction, adverse impact, and number of dropped predictors. *Personnel Psychology*, *76*(4), 1037–1060. https://doi.org/10.1111/peps.12587

Lantz, B. (2019). *Machine learning with R: expert techniques for predictive modeling*. Packt

publishing ltd.

Lee, T. W., & Mitchell, T. R. (1994). An alternative approach: The unfolding model of voluntary employee turnover. *Academy of Management Review*, *19*(1), 51-89. https://doi.org/10.5465/amr.1994.9410122008

Lee, T. W., Mitchell, T. R., Holtom, B. C., McDaneil, L. S., & Hill, J. W. (1999). The unfolding model of voluntary turnover: A replication and extension. *Academy of Management Journal*, *42*(4), 450–462. https://doi.org/10.2307/257015

Liashchynskyi, P., & Liashchynskyi, P. (2019). Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS. arXiv preprint arXiv:1912.06059

Liu, L., Gao, J., Beasley, G., & Jung, S.-H. (2023). LASSO and Elastic Net Tend to Over-Select Features. *Mathematics*, *11*(17), 3738. https://doi.org/10.3390/math11173738

Louppe, G. (2015). *Understanding Random Forests: From Theory to Practice* (arXiv:1407.7502). arXiv. http://arxiv.org/abs/1407.7502

Lucas, J. W., Whitestone, Y., Segal, D. R., Segal, M. W., White, M. A., Mottern, J. A., & Harris, R. N. (2008). The role of social support in first-term sailors' attrition from recruit training. *Millington, TN: Navy Personnel Research, Studies, and Technology Division, Bureau of Naval Personnel NPRST/BUPERS-1*.

March, J. G. and Simon, H. A. (1958). Organizations. New York: Wiley.

Melhorn, F. S., & Hoover, M. (2024, October 15). *Understanding America's labor shortage: The most impacted industries.* U.S. Chamber of Commerce. https://www.uschamber.com/workforce/understanding-americas-labor-shortage

Mitchell, T. R., B. C. Holtom, T. W. Lee, C. J. Sablynski, & M. Erez (2001). Why people stay: Using job embeddedness to predict voluntary turnover. *Academy of Management Journal*

44: 1102-1121. https://doi.org/10.2307/3069391

Mobley, W. H. (1977). Intermediate linkages in the relationship between job satisfaction and employee turnover. *Journal of Applied Psychology, 62*(2), 237–240. https://doi.org/10.1037/0021-9010.62.2.237

Morita, J. G., Lee, T. W., & Mowday, R. T. (1989). Introducing survival analysis to organizational researchers: A selected application to turnover research. *Journal of Applied Psychology*, *74*(2), 280. https://doi.org/10.1037/0021-9010.74.2.280

Nadeau, C., Bengio, Y. Inference for the Generalization Error. Machine Learning 52, 239–281 (2003). https://doi.org/10.1023/A:1024068626366

Oshiro, T., Perez, P., & Baranauskas, J. (2012). How many trees in a random forest? In *Lecture Notes in Computer Science (Vol. 7376)* (pp. 154–168). https://doi.org/10.1007/978-3-642-31537-4_13

Pargent, F., Schoedel, R., & Stachl, C. (2023). Best practices in supervised machine learning: A tutorial for psychologists. *Advances in Methods and Practices in Psychological Science*, *6*(3). https://doi.org/10.1177/25152459231162559

Parnell, T., Anghel, A., Łazuka, M., Ioannou, N., Kurella, S., Agarwal, P., ... & Pozidis, H. (2020). Snapboost: A heterogeneous boosting machine. Advances in Neural Information Processing Systems, 33, 1-12.

Pasquarella, A. C. (2023). A Machine Learning Approach to Predicting Federal STEM Workforce Attrition (Doctoral dissertation, The George Washington University).

Peterson, L. (2009). K-nearest neighbor. *Scholarpedia*, *4*(2). https://doi.org/10.4249/scholarpedia.1883

Pohlmann, J. T., & Leitner, D. W. (2003). A comparison of ordinary least squares and logistic

regression (1). *The Ohio journal of science*, *103*(5), 118-126.Provost, F. (2000). *Machine Learning from Imbalanced Data Sets 10*.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. Advances in neural information processing systems, 1-11.

Putka, D. J., Beatty, A. S., & Reeder, M. C. (2018). Modern Prediction Methods: New Perspectives on a Common Problem. *Organizational Research Methods*, *21*(3), 689–732. https://doi.org/10.1177/1094428117697041

Ren, Y., Tang, G., Li, X., & Chen, X. (2023). A study of multifactor quantitative stock-selection strategies incorporating knockoff and elastic net-logistic regression. *Mathematics, 11*(16), 3502. https://doi.org/10.3390/math11163502

Rombaut, E., & Guerry, M.-A. (2018). Predicting voluntary turnover through human resources database analysis. *Management Research Review*, *41*(1), 96–112. https://doi.org/10.1108/MRR-04-2017-0098

Rosenbusch, H., Soldner, F., Evans, A. M., & Zeelenberg, M. (2021). Supervised machine learning methods in psychology: A practical introduction with annotated R code. *Social and Personality Psychology Compass*, *15*(2), e12579. https://doi.org/10.1111/spc3.12579

Rubenstein, A. L., Eberly, M. B., Lee, T. W., & Mitchell, T. R. (2018). Surveying the forest: A meta-analysis, moderator investigation, and future-oriented discussion of the antecedents of voluntary employee turnover. *Personnel Psychology*, *71*(1), 23–65. https://doi.org/10.1111/peps.12226

Russell, C. J. (2013). Is it time to voluntarily turn over theories of voluntary turnover? *Industrial and Organizational Psychology*, *6*(2), 156–173. https://doi.org/10.1111/iops.12028

Sajjadian, M., Lam, R. W., Milev, R., Rotzinger, S., Frey, B. N., Soares, C. N., Parikh, S. V., Foster, J. A., Turecki, G., Müller, D. J., Strother, S. C., Farzan, F., Kennedy, S. H., & Uher, R. (2021). Machine learning in the prediction of depression treatment outcomes: A systematic review and meta-analysis. *Psychological Medicine*, *51*(16), 2742–2751. https://doi.org/10.1017/S0033291721003871

Sajjadiani, S., Sojourner, A. J., Kammeyer-Mueller, J. D., & Mykerezi, E. (2019). Using machine learning to translate applicant work history into predictors of performance and turnover. *Journal of Applied Psychology*, *104*(10), 1207–1225. https://doi.org/10.1037/apl0000405

Santafé, G., Inza, I., & Lozano, J. (2015). Dealing with the evaluation of supervised classification algorithms. *Artificial Intelligence Review*, *44*. https://doi.org/10.1007/s10462-015-9433-y

Saeys, Y., Inza, I., & Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. Bioinformatics, 23(19), 2507-2517.

Schneider, B. (1987). The people make the place. *Personnel Psychology, 40*(3), 437–453. https://doi.org/10.1111/j.1744-6570.1987.tb00609.x

Scottifer. (2017). Modelling Interactions with Decision Trees. RPubs. https://rpubs.com/scottifer8/296739

Shewach, O. R., Ingels, D., Dahlke, J. A., Putka, D. J., & Ingerick, M. (2024, April). Evaluating machine learning methods to predict turnover: Modeling imbalanced criteria [Paper presentation]. 39th annual conference of the Society for Industrial and Organizational Psychology, Chicago, IL.

Shiomi, Y., Toriumi, A., & Nakamura, H. (2022). International analysis on social and personal

determinants of traffic violations and accidents employing logistic regression with elastic net regularization. *IATSS Research*, *46*(1), 36–45. https://doi.org/10.1016/j.iatssr.2021.12.004

Shwartz-Ziv, R., & Armon, A. (2022). Tabular data: Deep learning is not all you need. *Information Fusion*, *81*, 84-90.

Singh, A., & Pandey, B. (2016, August). An Euclidean distance-based KNN computational method for assessing degree of liver damage. In *2016 International Conference on Inventive Computation Technologies (ICICT)* (Vol. 1, pp. 1–4). IEEE. https://doi.org/10.1109/ICICT.2016.7830040

Sirikulviriya, N., & Sinthupinyo, S. (2011, May). Integration of rules from a random forest. In *International Conference on Information and Electronics Engineering* (Vol. 6, pp. 194–198).

Smet, A. D., Dowling, B., Mugayar-Baldocchi, M., & Schaninger, B. (2021, September 8). *'Great attrition' or 'great attraction'? The choice is yours.* McKinsey & Company. https://www.mckinsey.com/capabilities/people-and-organizational-performance/our-insights/great-attrition-or-great-attraction-the-choice-is-yours

Society for Human Resource Management (SHRM). (2024). *2023-2024 state of the workplace report*. https://www.shrm.org/content/dam/en/shrm/research/2023-2024-State-of-the-Workplace-Report.pdf

Somers, M. J., & Birnbaum, D. (1999). Survival versus traditional methodologies for studying employee turnover: Differences, divergences and directions for future research. *Journal of Organizational Behavior*, *20*(2), 273–284. https://doi.org/10.1002/(SICI)1099-1379(199903)20:2<273::AID-JOB959>3.0.CO;2-X

Speer, A. B. (2024). Empirical attrition modelling and discrimination: Balancing validity and group differences. *Human Resource Management Journal*, *34*(1), 1–19. https://doi.org/10.1111/1748-8583.12355

Speer, A. B., Dutta, S., Chen, M., & Trussell, G. (2019). Here to stay or go? Connecting turnover research to applied attrition modeling. *Industrial and Organizational Psychology*, *12*(3), 277–301. https://doi.org/10.1017/iop.2019.22

Steel, R. P., & Ovalle, N. K. (1984). A review and meta-analysis of research on the relationship between behavioral intentions and employee turnover. *Journal of Applied Psychology*, *69*(4), 673-686. https://doi.org/10.1037/0021-9010.69.4.673

Strickland, W. J. (Ed.). (2005). A longitudinal examination of first term attrition and reenlistment among FY1999 enlisted accessions (Technical Report 1172). Arlington, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. Retrieved from https://apps.dtic.mil/dtic/tr/fulltext/u2/a440522.pdf

Sturman, M. C., Shao, L., & Katz, J. H. (2012). The effect of culture on the curvilinear relationship between performance and turnover. *Journal of Applied Psychology, 97*(1), 46–62. https://doi.org/10.1037/a0024868

Tansey, R., White, M., Long, R. G., & Smith, M. (1996). A comparison of loglinear modeling and logistic regression in management research. *Journal of Management, 22*(2), 339–358. https://doi.org/10.1016/S0149-2063(96)90037-1

Therneau, T., & Atkinson, B. (2023). Rpart: Recursive partitioning and regression trees. *R Package Version 4.1.23*. https://cran.r-project.org/web/packages/rpart

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology, 58(1),* 267-288.

https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Vanhove, A. J., Graham, B. Z., Titareva, T., & Udomvisawakul, A. (2023). Classification performance of supervised machine learning to predict HRM outcomes: A meta-analysis. *Academy of Management Proceedings, 2023*(1), 15366. https://doi.org/10.5465/AMPROC.2023.300bp

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, *10*(5), 988–999. https://doi.org/10.1109/72.788640

Wang, X., & Zhi, J. (2021). A machine learning-based analytical framework for employee turnover prediction. *Journal of Management Analytics*, *8*(3), 351–370. https://doi.org/10.1080/23270012.2021.1961318

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82. https://doi.org/10.1109/4235.585893

Yuan, S., Kroon, B., & Kramer, A. (2021). Building prediction models with grouped data: A case study on the prediction of turnover intention. *Human Resource Management Journal, 34*(1), 20–38. https://doi.org/10.1111/1748-8583.12396

Ziegler, W. M. A., (2017). Ranger: A fast implementation of random forest for high dimensional data in C++ and R. Journal of Statistical Software, 77(1), 1-17. doi:10.18637/jss.v077.i01.

Zou, H., & Hastie, T. (2005). Regularization and Variable Selection Via the Elastic Net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *67*(2), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x

APPENDIX A

Means, Standard Deviations, and Correlations with Confidence Intervals for Self-Report Features (Rolling Averages, April)

Variable	*M*	*SD*	1	2	3	4	5	6	7	8	9	10
1. Attrition	0.10	0.30										
2. Client Sat.	3.73	0.67	.07* [.01, .12]									
3. Pay Sat.	3.28	0.93	.08** [.02, .13]	.47** [.43, .51]								
4. Rec. Int.	9.43	1.42	.09** [.04, .15]	.73** [.70, .75]	.50** [.45, .54]							
5. Job Res. 1	4.19	0.61	-.06* [-.11, -.00]	.23** [.17, .28]	.21** [.16, .26]	.24** [.19, .29]						
6. Job Res. 2	4.75	0.62	.10** [.05, .16]	.57** [.53, .60]	.39** [.34, .43]	.58** [.54, .61]	.54** [.50, .58]					
7. Job Res. 3	4.78	0.60	.09** [.03, .14]	.57** [.53, .60]	.34** [.29, .39]	.50** [.46, .54]	.41** [.36, .45]	.68** [.65, .71]				
8. Sched. Avail. Sat.	4.64	0.87	.07* [.01, .12]	.45** [.40, .49]	.24** [.19, .30]	.44** [.40, .49]	.06* [.01, .12]	.37** [.32, .42]	.40** [.35, .45]			
9. Sched. Sat. Overall	4.70	0.73	.07* [.01, .12]	.52** [.48, .56]	.32** [.27, .37]	.52** [.48, .56]	.11** [.06, .17]	.42** [.37, .46]	.44** [.40, .49]	.70** [.67, .73]		
10. Mgr. Sat. Ov.	4.81	0.55	.09** [.03, .14]	.63** [.60, .66]	.43** [.38, .47]	.70** [.67, .73]	.24** [.19, .29]	.52** [.48, .56]	.47** [.43, .51]	.38** [.33, .43]	.46** [.42, .50]	

Note. $n = 1000$. Data from April were used for descriptive statistics. *M* and *SD* are used to represent mean and standard deviation, respectively. Values in square brackets indicate the 95% confidence interval for each correlation. The confidence interval is a plausible range of population correlations that could have caused the sample correlation (Cumming, 2014). Features with * were removed for nonsignificant correlation with attrition.

Variable	M	SD	1	2	3	4	5	6	7	8	9	10
11. Mgr. Sat. Com.	4.81	0.59	.09** [.03, .14]	.53** [.49, .57]	.34** [.29, .38]	.57** [.53, .61]	.21** [.15, .26]	.48** [.44, .52]	.43** [.39, .48]	.27** [.22, .32]	.36** [.31, .40]	.74** [.71, .76]
12. Mgr. Sat. Help	4.83	0.58	.08** [.02, .13]	.50** [.46, .54]	.33** [.28, .38]	.55** [.51, .58]	.16** [.10, .21]	.41** [.36, .45]	.39** [.34, .44]	.27** [.22, .32]	.33** [.28, .38]	.74** [.72, .77]
13. Mgr. Sat. Responsiveness	4.80	0.61	.09** [.03, .14]	.55** [.51, .59]	.39** [.34, .43]	.61** [.57, .64]	.22** [.17, .27]	.49** [.44, .53]	.44** [.39, .48]	.32** [.27, .36]	.37** [.33, .42]	.74** [.72, .77]
14. Mgr. Sat. Respect	4.85	0.55	.08** [.02, .13]	.46** [.42, .51]	.31** [.26, .36]	.49** [.45, .53]	.14** [.09, .20]	.42** [.37, .46]	.35** [.30, .40]	.26** [.21, .31]	.30** [.25, .35]	.69** [.66, .72]
15. Mgr. Sat. Prof.	4.84	0.55	.08** [.02, .13]	.50** [.45, .54]	.32** [.26, .36]	.52** [.48, .56]	.15** [.09, .20]	.43** [.38, .47]	.35** [.30, .40]	.26** [.21, .31]	.31** [.26, .36]	.68** [.65, .71]
16. Mgr. Sat. Feedback	4.80	0.61	.08** [.03, .14]	.55** [.51, .59]	.38** [.33, .43]	.59** [.55, .62]	.18** [.13, .24]	.47** [.43, .52]	.45** [.41, .50]	.30** [.24, .35]	.38** [.33, .42]	.75** [.72, .77]
17. Mgr. Sat. Knowledge	4.83	0.56	.08** [.02, .13]	.52** [.47, .56]	.36** [.31, .40]	.56** [.52, .60]	.22** [.17, .28]	.49** [.44, .53]	.42** [.37, .47]	.27** [.22, .32]	.34** [.29, .38]	.73** [.70, .76]
18. Mgr. Sat. Kindness	4.84	0.54	.08** [.03, .14]	.48** [.44, .52]	.33** [.28, .38]	.54** [.50, .58]	.16** [.10, .21]	.43** [.38, .47]	.39** [.34, .44]	.28** [.22, .33]	.34** [.29, .38]	.71** [.68, .73]
19. Prior Experience*	4.56	1.37	.03 [-.03, .08]	.21** [.15, .26]	.13** [.07, .18]	.18** [.13, .23]	-.07* [-.13, -.02]	.19** [.13, .24]	.18** [.12, .23]	.20** [.15, .25]	.23** [.17, .28]	.11** [.05, .16]
20. # Surveys Taken	0.60	0.86	-.15** [-.20, -.09]	-.41** [-.46, -.36]	-.20** [-.25, -.14]	-.40** [-.44, -.35]	.18** [.12, .23]	-.41** [-.46, -.36]	-.37** [-.42, -.33]	-.39** [-.44, -.35]	-.39** [-.44, -.35]	-.35** [-.39, -.30]

Variable	*M*	*SD*	11	12	13	14	15	16	17	18	19
12. Mgr. Sat. Help	4.83	0.58	.87** [.86, .89]								
13. Mgr. Sat. Responsiveness	4.80	0.61	.86** [.84, .87]	.88** [.86, .89]							
14. Mgr. Sat. Respect	4.85	0.55	.86** [.84, .87]	.85** [.84, .87]	.81** [.79, .83]						
15. Mgr. Sat. Prof.	4.84	0.55	.86** [.84, .87]	.87** [.85, .88]	.82** [.80, .84]	.92** [.91, .93]					
16. Mgr. Sat. Feedback	4.80	0.61	.89** [.87, .90]	.85** [.84, .87]	.88** [.87, .89]	.84** [.82, .85]	.84** [.83, .86]				
17. Mgr. Sat. Knowledge	4.83	0.56	.88** [.87, .89]	.88** [.87, .89]	.87** [.86, .89]	.81** [.79, .83]	.81** [.79, .83]	.88** [.86, .89]			
18. Mgr. Sat. Kindness	4.84	0.54	.86** [.84, .87]	.89** [.88, .90]	.85** [.83, .86]	.94** [.93, .94]	.91** [.90, .92]	.87** [.85, .88]	.84** [.83, .86]		
19. Prior Experience	4.56	1.37	.10** [.05, .16]	.07* [.01, .12]	.11** [.06, .17]	.05 [-.00, .11]	.06* [.00, .11]	.10** [.04, .15]	.06* [.01, .12]	.07* [.02, .13]	
20. # Surveys Taken	0.60	0.86	-.33** [-.38, -.28]	-.30** [-.34, -.24]	-.31** [-.36, -.26]	-.28** [-.33, -.23]	-.29** [-.34, -.24]	-.33** [-.38, -.28]	-.30** [-.35, -.25]	-.29** [-.34, -.23]	-.31** [-.36, -.26]

APPENDIX B

Means, Standard Deviations, and Correlations with Confidence Intervals for HRIS Features

Feature	*M*	*SD*	1	2	3	4	5	6	7	8
1. **Attrition**	0.10	0.30								
2. Invoice Rate	10.41	2.12	.01 [-.04, .07]							
3. **Invoiced Other**	14.24	29.69	-.08** [-.13, -.02]	-.01 [-.06, .05]						
4. **Invoiced Total**	221.85	160.78	-.21** [-.27, -.16]	.03 [-.02, .09]	.61** [.57, .64]					
5. Invoiced Total RA	224.30	149.23	-.22** [-.28, -.17]	.01 [-.05, .06]	.54** [.50, .58]	.92** [.91, .92]				
6. **Invoiced Total %Δ**	5422.21	191166.71	.08** [.03, .14]	-.00 [-.06, .05]	.05 [-.01, .10]	.06* [.01, .12]	.00 [-.05, .06]			
7. **Total Hours**	18.66	13.32	-.21** [-.26, -.16]	.02 [-.03, .08]	.32** [.27, .37]	.91** [.90, .92]	.82** [.80, .84]	.07* [.01, .12]		
8. Total Hours RA	18.20	11.98	-.23** [-.28, -.17]	.01 [-.04, .07]	.27** [.22, .32]	.83** [.82, .85]	.90** [.89, .91]	.00 [-.05, .06]	.91** [.90, .92]	
9. Total Hours %Δ	4767.86	168104.54	.08** [.03, .14]	-.00 [-.06, .05]	.05 [-.01, .10]	.06* [.01, .12]	.00 [-.05, .06]	1.00** [1.00, 1.00]	.07* [.01, .12]	.00 [-.05, .06]

Note. n = 1000. Data from April were used for descriptive statistics. *M* and *SD* are used to represent mean and standard deviation, respectively. Values in square brackets indicate the 95% confidence interval for each correlation. The confidence interval is a plausible range of population correlations that could have caused the sample correlation (Cumming, 2014). * indicates $p < .05$. ** indicates $p < .01$. **Bolded** features were used in the final analyses. "Invoice Rate" was removed for its weak correlation with the outcome. Percent Change (%Δ) and Rolling Average (RA) were not calculated for Invoice Rate because it changes very infrequently. Total Hours was retained over Total Hours RA and Total Hours %Δ because Total Hours %Δ correlates 1.0 with Invoiced Total %Δ and Total Hours RA correlates .90 with Invoiced Total RA, and is less proximal to the outcome.

APPENDIX C

Means, Standard Deviations, and Correlations with Confidence Intervals for Performance Features

Feature	*M*	*SD*	1	2	3	4	5	6	7	8	9
1. **Attrition**	0.10	0.30									
2. **Performance**	90.72	12.25	-.15** [-.21, -.10]								
3. Performance RA	89.77	11.71	-.14** [-.20, -.09]	**.93**** [.92, .93]							
4. Performance %Δ	1.41	8.67	.06* [.00, .11]	.22** [.17, .27]	-.03 [-.08, .03]						
5. **Attendance**	19.31	12.52	-.19** [-.24, -.13]	.01 [-.04, .07]	-.01 [-.06, .05]	.04 [-.02, .09]					
6. Attendance RA	18.44	11.78	-.20** [-.25, -.15]	.07* [.01, .12]	.05 [-.01, .10]	.02 [-.03, .08]	.92** [.91, .93]				
7. **Attendance %Δ**	7.98	94.71	-.09** [-.14, -.04]	-.03 [-.09, .02]	-.06* [-.12, -.01]	.10** [.04, .15]	.17** [.12, .23]	.05 [-.01, .10]			
8. **# Phone Calls**	62.63	52.23	-.12** [-.17, -.06]	.18** [.13, .24]	.19** [.14, .25]	.02 [-.04, .07]	.43** [.38, .47]	.37** [.32, .42]	.06* [.01, .12]		
9. # Phone Calls RA	57.05	42.92	-.17** [-.23, -.12]	.18** [.13, .23]	.18** [.12, .23]	.03 [-.02, .09]	.55** [.51, .59]	.56** [.52, .59]	.05 [-.01, .10]	**.85**** [.84, .87]	
10. # Phone Calls %Δ	14.53	70.60	-.03 [-.08, .03]	-.01 [-.07, .04]	-.00 [-.06, .05]	.01 [-.04, .07]	-.01 [-.06, .05]	-.12** [-.17, -.06]	.56** [.52, .60]	.03 [-.03, .08]	-.07** [-.13, -.02]

Note. $n = 1000$. Data from April were used for descriptive statistics.*M* and *SD* are used to represent mean and standard deviation, respectively. Values in square brackets indicate the 95% confidence interval for each correlation. The confidence interval is a plausible range of population correlations that could have caused the sample correlation (Cumming, 2014). * indicates $p < .05$. ** indicates $p < .01$. **Bolded** features were included in the final analyses. "# of Phone Calls RA" was removed for its low correlation with the outcome and high correlation with "# of Phone Calls". "# of Phone Calls %Δ" was removed for its low correlation with the outcome. "Performance" was retained, and "Performance RA" and "Performance %Δ" were removed for their high correlations with "Performance" and low correlations with the outcome. "Attendance RA" was removed for its strong correlation with "Attendance".

APPENDIX D

Means, Standard Deviations, and Correlations with Confidence Intervals for Engineered Features

Feature	*M*	*SD*	1	2	3
1. **Attrition**	0.10	0.30			
2. **# Sat Surveys Taken**	0.60	0.86	-.15** [-.20, -.09]		
3. **Tenure**	208.45	264.94	-.19** [-.24, -.13]	.34** [.30, .39]	
4. **Weeks Skipped**	0.83	1.33	.30** [.25, .35]	-.25** [-.30, -.19]	-.29** [-.34, -.24]

Note. $n = 1000$. Data from April were used for descriptive statistics. *M* and *SD* are used to represent mean and standard deviation, respectively. Values in square brackets indicate the 95% confidence interval for each correlation. The confidence interval is a plausible range of population correlations that could have caused the sample correlation (Cumming, 2014). * indicates $p < .05$. ** indicates $p < .01$.

APPENDIX E

Summary of Hyperparameter Tuning Results Across Classifiers

Method	Parameters	Values Used in Outer Loop	Best Parameters (n=500)	Best Parameters (n=1000)	% (n=500)	% (n=1000)
CART	*cost complexity, min n, tree depth*	(0.1, 1E-10), (1, 15), (2, 40)	(1E-10, 15, 2)	(0.1, 15, 2)	54%, 65%, 86%	65%, 74%, 100%
Random Forest	*mtry, trees, min n*	(1, 14, 28), (1, 1000, 2000), (2, 21, 40)	(1, 1, 40)	(1, 1, 40)	70%, 50%, 62%	57%, 50%, 62%
XGBoost	*trees, min n, tree depth, learn rate, loss reduction, sample size*	(50, 100), (1, 5), (2, 8), (1.02, 1.58), (1, 1.99), (0.5, 1)	(50, 5, 8, 1.58, 1.99, .5)	(50, 5, 8, 1.02, 1.99, .5)	68%, 74%, 57%, 54%, 51%, 63%	77%, 68%, 52%, 57%, 71%, 60%
Elastic Net	*penalty, mixture*	(1E-10, 4.6e-5, 0.0059, 0.078, 1), (0, 0.33, 0.44, 0.55, 0.66, 0.77, 0.88, 1).	(0.077,0)	(0.077, 0)	67%, 17%	65%, 22%
Lasso	*penalty*	(1.00E-10, 4.6e-5, 5.9e-4, 0.077)	0.077	5.90E-04	74%	51%
Ridge	*penalty*	(1.E-10, 0.077, 1)	1	0.077	49%	40%

Note. Values in columns "Best Parameters" indicate the most frequently chosen hyperparameter values for both sample sizes, and columns "%" indicate the percentage of the time they were chosen (out of 35 outer folds total per method ´ sample size).

APPENDIX F
CODE

All R code, as well as a simulated dataset, can be found in a public repository at this address: https://github.com/rhess-io/Public---Attrition-Modeling-Script