

A Comparative Analysis of Machine Learning Techniques for
Predicting Student First-Year Dropout

A Thesis

Submitted to the Graduate Faculty of the
National University, School of Engineering and Computing
in partial fulfillment of the requirements for the degree of
Masters of Science in Data Analytics

Prepared By:

Michael Kirkpatrick

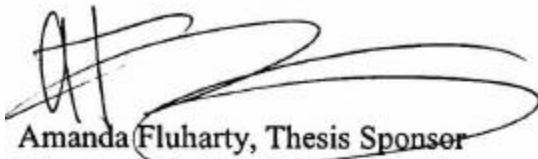
August 2017

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

MASTER'S THESIS APPROVAL FORM

We certify that we have read the project of Michael Kirkpatrick entitled A COMPARATIVE ANALYSIS OF MACHINE LEARNING TECHNIQUES FOR PREDICTING STUDENT FIRST-YEAR DROPOUT and that, in our opinion, it is satisfactory in scope and quality as the thesis for the degree of Master of Science in Data Analytics at National University.

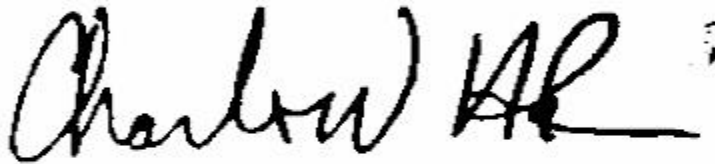
Approved:



Amanda Fluharty, Thesis Sponsor
Director of Institutional Research
National University

8/9/17

Date



Chuck Hahm, Thesis Advisor
Adjunct Professor, School of Engineering and Computing
National University

8/14/2017

Date



Dr. Jodi Reeves, Thesis Course Instructor
Academic Program Director – MS Data Analytics
Associate Professor, School of Engineering and Computing
National University

8/14/2017

Date

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

ABSTRACT

Student dropout is of the utmost concern in higher education and machine learning techniques have become a powerful tool for proactively identifying students at-risk of dropping out. Data from more than 17,000 National University students were used to train nine machine learning algorithms to predict first-year dropout under four conditions, thus resulting in thirty-six models. The algorithms included were Logistic Regression, Naïve Bayes, Neural Networks, k -Nearest Neighbor, Support Vector Machine with linear and polynomial kernels, Decision Tree, Random Forest, and XGBoost. Modeling conditions varied with regard to class balancing and feature reduction. Models were evaluated based on ROC area and accuracy. Ensemble tree-methods XGBoost and Random Forest were superior across all modeling conditions. Overall, class balancing and feature reduction did not improve model performance. Feature importance was examined and many novel features proved to be useful for dropout prediction. Recommendations for future research and specifically for National University are discussed.

Keywords: retention, attrition, machine learning, data mining, variable importance

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

TABLE OF CONTENTS

Master's Thesis Approval Form	ii
Abstract	iii
List of Tables	ix
List of Illustration	x
Chapter 1: Introduction	1
1.1 Background	1
1.2 Problem Statement	1
1.3 Objectives.....	3
1.4 Research Hypotheses.....	5
1.5 Limitations of the Study	5
1.6 Definition of Terms	7
1.6.1 Class Variable.....	7
1.6.2 Feature Variable	7
1.6.3 Instance.....	8
1.6.4 Dropout.....	8
1.7 Summary	9
Chapter 2: Literature Review	10
2.1 Introduction	10
2.2 Theoretical Foundation	10

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

2.3 Modeling Dropout	11
2.3.1 Algorithms	11
2.3.2 Class Balancing	15
2.3.3 Feature Reduction	18
2.3.4 Feature Importance	20
2.4 Summary	23
Chapter 3: Methodology	24
3.1 Introduction	24
3.2 Study Population	24
3.3 Data Sources	25
3.3.1 Student Class Activity Report	25
3.3.2 Alumni	26
3.3.3 Bio Demo	26
3.3.4 Academic Plans	26
3.3.5 Academic Advisement Report	26
3.3.6 End of Course Evaluations	27
3.3.7 Service Indicators	28
3.3.8 Transfer Course Detail	28
3.3.9 National Student Clearinghouse	28
3.3.10 NU External Education	29

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

3.3.11 FAFSA	30
3.3.12 Statistics of Income by ZIP Code	30
3.4 Features	30
3.5 Machine Learning Algorithms	33
3.5.1 Logistic Regression	35
3.5.2 Naïve Bayes	35
3.5.3 <i>k</i> -Nearest Neighbor	36
3.5.4 Neural Network	37
3.5.5 Support Vector Machine	38
3.5.6 Decision Tree	39
3.5.7 Random Forest	39
3.5.8 Extreme Gradient Boost	40
3.5.9 Comparison of Strengths and Weaknesses	41
3.6 Algorithm Training and Evaluation	42
3.6.1 Software	42
3.6.2 Training	42
3.6.3 Evaluation	46
3.6.4 Feature Importance	47
3.7 Summary	48
Chapter 4: Results	49

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

4.1 Introduction	49
4.2 Data Set Results	49
4.3 Modeling Results.....	51
4.3.1 Model Performance	51
4.3.2 Algorithm Performance	57
4.3.3 Data Set Performance	58
4.4 Feature Importance.....	60
4.4.1 All Models	61
4.4.2 Random Forest and XGBoost Models.....	62
4.4.3 Feature Interpretations	64
4.5 Summary	78
Chapter 5: Conclusions and Future Work.....	80
5.1 Algorithms.....	80
5.2 Class Balancing	81
5.3 Feature Reduction	81
5.4 Feature Importance.....	82
5.5 Summary	84
Appendix A: End of Course Evaluation Survey Items	85
Appendix B: SQL Code to Create Complete Unbalanced Data Set	86
Appendix C1: R Code – Data Preparation.....	103

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

Appendix C2: R Code – Model Training.....	106
Appendix C3: R Code – Model Testing	110
Appendix C4: R Code – Feature Importance.....	116
Appendix C5: R Code – Plots and Tables	125
Appendix D: Features Included	136
References.....	138

LIST OF TABLES

Table 1 Dropout Publication Modeling Techniques	12
Table 2 Dropout Publication Class Balancing Techniques.....	16
Table 3 Dropout Publication Feature Reduction Techniques	19
Table 4 Dropout Publication Feature Importance Techniques	21
Table 5 Features Included in the Current Study	31
Table 6 Algorithm Strengths and Weaknesses	41
Table 7 Model Naming Conventions	45
Table 8 Data Set Descriptive Statistics	49
Table 9 Modeling Results	52
Table 10 Algorithm Results	57
Table 11 Data Set Results	59
Table 12 Average Importance of 25 Most Important Features from All Models	61
Table 13 Average Importance of 25 Most Important Features from XGB and RF Models	63
Table 14 Chi-Square Tests of Independence for Categorical Features.....	65
Table 15 Student's t-tests for Continuous Features	73

LIST OF ILLUSTRATION

Figure 1. Tinto's (1975) conceptual model of student/institutional commitment.	11
Figure 2. Frequency of modeling techniques used for dropout prediction.	13
Figure 3. Illustration of the four data sets.	44
Figure 4. Example contingency table for binary classification.....	46
Figure 5. Recursive feature elimination results for Reduced Unbalanced data set.	50
Figure 6. Recursive feature elimination results for Reduced Balanced data set.....	51
Figure 7. ROC area by model.	53
Figure 8. Accuracy by model.....	54
Figure 9. Sensitivity by model.	55
Figure 10. Specificity by model.....	56
Figure 11. Evaluation metrics by algorithm.	58
Figure 12. Evaluation metrics by data set.	60
Figure 13. Average importance and 95% confidence interval for 25 most important features across all models.	62
Figure 14. Average importance and 95% confidence interval for 25 most important features across XGBoost and Random Forest models.	64
Figure 15. DFUWI by Dropout bar charts.	66
Figure 16. Degree Award Type by Dropout bar charts.	67
Figure 17. Relative Performance by Dropout bar charts.	68
Figure 18. Registrar Admission Flag (P_RFA) by Dropout bar charts.	69
Figure 19. Two-digit CIP Code by Dropout bar charts.	70
Figure 20. Perceptions of Learning by Dropout bar charts.....	71

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

Figure 21. Perceptions of Teaching by Dropout bar charts.	72
Figure 22. Previous Degree Level by Dropout bar charts.	73
Figure 23. Days to Fiscal Year Close by Dropout boxplot.....	74
Figure 24. Fiscal quarter start by Dropout bar charts.	75
Figure 25. Transfer units by Dropout boxplot.	76
Figure 26. Previous GPA by Dropout boxplot.....	77
Figure 27. Per-Capita Adjusted Gross Income by Dropout boxplot.....	78

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

Tens of thousands of students enroll at higher education institutions every year in the United States with the intent to complete their degree program. Unfortunately, many of them do not reach their goal of graduating and, in fact, many of them do not even complete their first year of schooling. One-year retention is considered the first stepping-stone to graduation, and one-year retention statistics are reported to the federal government for comparison and accountability purposes (Sousa, 2015). Therefore, the onus of student retention is not only on the student that is investing in their education; it is also on the institution that is investing in them.

Since the 1980s, higher education institutions have been developing and comparing predictive models to identify students at risk of dropping out (Bean, 1980). Institutions have utilized these predictions for student support, such as proactive student interventions. Such proactive interventions have been successful in improving student retention, thus improving the success of the students and the institution (Leitner, Khalil, & Ebner, 2017).

1.2 PROBLEM STATEMENT

National University is a private, nonprofit higher-education institution based in La Jolla, CA. Founded in 1971, NU strives to make education accessible to adults by offering over 100 accelerated graduate and undergraduate degree programs, with courses offered online in a one-month per class format (National University, 2017).

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

National University's one-year retention rate for first-time, full-time degree-seeking undergraduates in academic year 2015 was 64 percent, which is one percentage higher than the average retention rate for all four-year, private nonprofit, open admissions institutions in the United States (McFarland et al., 2017). Although National University's retention rate is slightly better than their immediate peer institutions, National University falls well below the average retention rate of all private nonprofit, four-year institutions in the U.S. These institutions, and coincidentally the average of all four-year institutions in the U.S., have an average one-year retention rate of 81 percent (McFarland et al., 2017).

National University's first-time, full-time, degree-seeking student population accounts for less than one percent of the student body. While statistics based on this population are useful for peer comparisons, they are not the best reflection of one-year retention within the university. In academic year 2015, the overall one-year retention for all degree-seeking students was 71 percent (NU Institutional Research, 2017). Although this rate is improved, it still means that 29 percent of students dropped out within their first year. A total of 9,744 students we included in this population, which means that roughly 2,826 dropped out.

Efforts to improve student retention at National University have historically been somewhat of a blanket approach. That is, interventions are directed at the entire student body, or at large groups of the student body (e.g., all undergraduate students). The challenges with this approach is that these interventions are too broad; they include a large proportion of the student body, some of which do not need intervention services.

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

Furthermore, they are not precise. They are generic, one-size-fits-all interventions that are not tailored to individual student needs.

Within the last year, efforts to improve student retention at National University have become more sophisticated. A targeted intervention program was initiated to predict student first-quarter retention and align interventions (Fluharty, Gallet, & Hower, 2016). The program has had promising results and has proven that predictive analytics can be implemented at National University's. However, the study modeled a narrow time-frame (i.e., one-quarter) with limited predictors (i.e., 25). National University would benefit from additional predictive analytic programs that support student retention beyond the first-quarter.

1.3 OBJECTIVES

The primary objective of the current study was to find the optimal model for predicting student first-year dropout at National University. To accomplish this, nine machine learning algorithms were modeled on 17,083 degree-seeking National University students from academic years 2015 and 2016. The machine learning algorithms were Logistic Regression, Naïve Bayes, Neural Networks, *k*-Nearest Neighbor, Support Vector Machine with a linear kernel, Support Vector Machine with a polynomial kernel, Decision Tree, Random Forest, and XGBoost. Models were evaluated based on ROC area and accuracy.

The secondary objective of the current study was to evaluate the effects of feature reduction and class balancing when predicting student dropout. To accomplish this, four data sets were created. The first data set was the Complete Unbalanced (CU) data set and

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

it contained all features and all 17,083 instances. This data set was considered Complete because it contained all features and it was considered Unbalanced because the class variable (i.e., Dropout) contained roughly 70% Persist instances and 30% Dropout instances. Feature reduction was applied to CU to remove non-predictive features and thus create the Reduced Unbalanced (RU) data set. This data set was considered Reduced because it contained a reduced feature set and was considered Unbalanced because the class contained roughly 70% Persist instances and 30% Dropout instances. Under-sampling was applied to the Persist class of the CU data set to create the Complete Balanced (CB) data set. This data set was considered Complete because it contained all features and it was considered balanced because the class variable contained 50% Persist instances and 50% Dropout instances. Finally, feature reduction was applied to CB to remove non-predictive features and thus create the Reduced Balanced (RU) data set. This data set was considered Reduced because it contained a reduced feature set and was considered Balanced because the class contained an equal proportion of Persist and Dropout instances. All nine machine learning algorithms were modeled on each data set, thus resulting in 36 models. The effects of feature reduction and class balancing were evaluated by comparing the group performance of the models.

The third objective of the current study was to examine which features were consistently most important for predicting dropout across modeling techniques. To accomplish this, feature importance was calculated for every feature within each of the 36 models. Since feature importance is calculated differently across algorithms, feature importance values were normalized. The normalized feature importance values were averaged across all models to obtain the average feature importance for each feature. The

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

average feature importance values were used to draw conclusions about which features are consistently important for dropout prediction.

1.4 RESEARCH HYPOTHESES

The strengths and weaknesses of each machine learning algorithm are discussed in detail in section 3.5.9 regarding the following six criteria: class imbalance, sparsity, outliers, high dimensionality, correlated and nonlinear features. It is hypothesized that Support Vector Machine with a polynomial kernel, Random Forest, and XGBoost (XGB) will perform best since these algorithms are robust to all six criteria.

There has been little research regarding the effects of class balancing on dropout prediction. Although sparse, the evidence suggests that class balancing improves dropout prediction (Delen, 2010; Lin, 2012; Thammasiri, Delen, Meesad & Kasap, 2014). Therefore, it is hypothesized that models created from balanced data sets will perform better than models created from unbalanced data sets.

There has been little research regarding the effects of feature reduction on dropout prediction. Although very limited, there is some evidence that feature reduction improves dropout prediction (Alkhasawneh & Hargraves, 2014). Therefore, it is hypothesized that models created from data sets with a reduced feature set will perform better than models created from data sets with the complete feature set.

1.5 LIMITATIONS OF THE STUDY

A considerable limitation is the availability of data due to national University's open admission policy. National University does not require students to report SAT, GRE or any sort of standardized test scores upon admission because National University seeks

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

to serve all students – not just those with the highest test scores. While this is beneficial for students, it makes dropout modeling more difficult. Such measures are typically very predictive of dropout (Cochran, Campbell, Baker, & Leeds, 2014).

There were many challenges with obtaining reliable external education data, specifically external degrees earned, time since last attendance and external GPA, all of which are considered significant predictors of dropout (Cochran et al., 2014). Three data sources were utilized for external education data but each had significant shortcomings. The first source was National Student Clearinghouse (NSC). NSC is nonprofit education organization that facilitates the verification of prior degree completion and/or previous enrollment of students (National Student Clearinghouse, 2017). Unfortunately, NSC verifies prior student enrollment activity by matching the students' first and last name, and then matching either the students' social security number or birth date. Individuals often change their name after marriage and it is common for marriage to occur after college. Therefore, in these circumstances, it is impossible to find prior educational activity. The second data source came from the student transcripts that National University collected during the application process. The benefit of this data source is that student name changes are not an issue. However, the significant drawback is that not all degree awards earned by the student are recorded. For example, if a student applies to a Master's program at National University and they previously completed a Master's and Bachelor's degree elsewhere, only the Bachelor's degree information is recorded. Furthermore, prior enrollment that does not result in a completed degree are omitted as well. Finally, the third data source was course-level transfer credits that students applied to their National University degree. The benefit of this data sources is that student name

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

changes are not a problem. However, the significant drawback is that data are available only if the student sought to apply their previous coursework.

Another limitation was how much historical data were available for model training. National University changed their end of course evaluation survey at the beginning of academic year 2015. Therefore, data prior to academic year 2015 could not be included.

1.6 DEFINITION OF TERMS

1.6.1 CLASS VARIABLE

In machine learning, a class variable is the categorical outcome variable in a classification algorithm that is comprised of two or more nominal categories (James, Witten, Hastie, & Tibshirani, 2013). The class variable for the current study is Dropout, which has class values of Dropout and Persist. Class variables are also called dependent or target variables, however, these terms are less specific since they refer to both continuous and categorical outcomes.

1.6.2 FEATURE VARIABLE

In machine learning, a feature is a variable that is used to describe an instance (James et al., 2013). Machine learning algorithms use features to learn patterns within data. An example of a feature from the current study is student Age. Features can be continuous or categorical and are often called independent or input variables.

1.6.3 INSTANCE

In machine learning, an instance is a data observation, which is described by features (James et al., 2013). In the current study, each instance represents a student. Instances are also called cases, records and examples.

1.6.4 DROPOUT

Dropout is used synonymously with attrition in the current study. Dropout and stop-out are sometimes defined as mutually exclusive sub-categories of attrition. For example, stop-out refers to students that discontinue due to non-enrollment whereas dropout refers to students that voluntarily withdrew or discontinue due to poor academic performance. In the current study, these distinctions are not made and therefore are not modeled. Dropouts, stop-outs, withdrawals and all other forms of attrition are labeled as dropout. What follows is a detailed definition of the Dropout variable used in the current study.

The class variable in the current study is Dropout and this variable has two levels. A value of “Dropout” signified that the student dropped out before their second academic year and a value of “Persist” indicated that the student persisted to their second academic year. Dropout is the class of interest in the current study, but in order to define dropout it is easiest to define persistence first. National University’s academic year spans from July 1st to June 30th. At National University, students are considered to persist when they are enrolled in a class in their second academic year or if they graduate from their program by their second academic year. For example, if a student takes their first class in July 2015 and they take a class anytime between July 2016 and June 2017, they have persisted. Or, if a student takes their first class in July 2015 and graduates during that

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

same academic year (July 2015 to June 2016) or anytime in the following academic year (July 2016 to June 2017), they have persisted. Dropout occurs when a student does not persist. That is, a student is considered to dropout if they do not graduate and do not take a class in their second academic year. Following the previous example, if a student takes their first class July 2015 and they do not graduate and are not enrolled in a class by the end of academic year 2016 (June 2017), they have dropped out.

It's important to note that if a student starts their first class later in the academic year, they have a shorter distance to their persistence goal. For example, if a student enrolls in their first class at the end of the academic year, let's say June 2016, they are considered to persist if they take a class the following month (July 2016) because that is when the next academic year starts.

1.7 SUMMARY

Student dropout is of the utmost concern for higher education institutions. Over the last couple decades, institutions have utilized predictive models to identify students at-risk of dropping out. National University is a private, nonprofit higher-education institution based in La Jolla, CA that does not have the ability to identify students at-risk of dropping out. The objective of the current study is to train and evaluate several machine learning algorithms in order to find the optimal model for predicting dropout at National University. Additionally, the current study will investigate the effects of feature reduction and class balancing on model performance as well as assess which features are most important for dropout prediction.

CHAPTER 2: LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter, a theoretical basis for explaining student dropout is discussed. Then an evaluation of all higher education, first-year dropout prediction publications is conducted. A total of 31 publications are systematically evaluated based on the modeling, class balancing, feature reduction, and feature importance techniques that were applied. Substantial gaps in the literature are noted, of which the current study seeks to fill.

2.2 THEORETICAL FOUNDATION

In 1975, Vincent Tinto proposed paradigm for explaining student dropout that has become widely accepted (Tinto, 1975). He argued that student dropout in higher education is a bidirectional relationship between the student and the university, and that this relationship is based on commitment (see Figure 1). In this relationship, the student enters the institution with their own history, as does the institution. Both parties, the student and the university, have their level of commitment to each other and it's this commitment that dictates student dropout throughout the student's academic career. The student expresses their commitment through their academic integration. For example, the student's academic performance, intellectual development and assimilation with the university are all signs of their commitment to completing their degree program. The institution expresses its commitment through their facilitation of social integration. Faculty interactions, classroom engagement, and extracurricular organizations are all examples of how the institution can express their commitment to the student. The more that both parties are committed to, and engage with each other, the less likely the student will dropout.

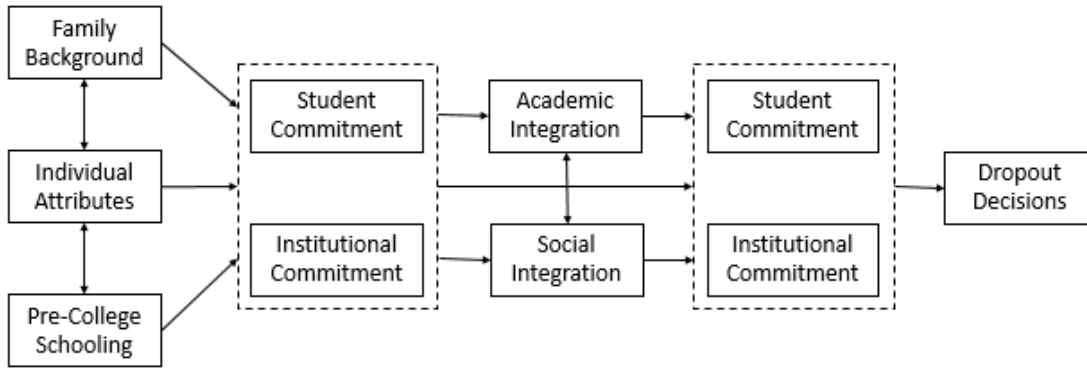


Figure 1. Tinto's (1975) conceptual model of student/institutional commitment.

2.3 MODELING DROPOUT

2.3.1 ALGORITHMS

Table 1 provides, to be the best of the author's ability, a list of all higher education, first-year dropout prediction publications and the predictive modeling techniques they utilized. All 31 of these publications specifically modeled first-year dropout. Publications that modeled other forms of dropout, such as course attrition, were not included. Figure 2 illustrates the frequency of modeling techniques used to predict first-year dropout. Across the 31 publications, 19 different modeling techniques were applied. However, Logistic Regression, Decision Trees, and Neural Networks accounted for the vast majority of applications. Logistic Regression was utilized 19 times, Decision Trees 12 times, Neural Networks 11 times, and the rest were utilized three or fewer times.

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

Table 1

<i>Dropout Publication Modeling Techniques</i>	
<u>Publication</u>	<u>Technique*</u>
Bean (1980)	LiR, PA
Stage (1989)	LR, PA
Cabrera, Nora, & Castafne (1993)	SEM
Dey & Astin (1993)	LR, PR, LiR
Murtaugh, Burns, & Schuster (1999)	SA
Bresciani & Carson, (2002)	LR
Garton & Ball (2002)	DA
Glynn, Sauer, & Miller (2003)	LR
Salazar, Goxalbez, Bosch, Miralles, & Vergara (2004)	OR
Zhang, Anderson, Ohland, & Thorndyke (2004)	LR
Herzog (2005)	LR
Herzog (2006)	DT, LR, NN
Sujitparapitaya (2006)	DT, LR, NN
Yu, DiGangi, Jannasch, Lo, & Kaprolet (2007)	DT
Fike & Fike (2008)	LR
Sutton & Nora (2008)	LR
Lin, Imbrie & Reid (2009)	DA, LR, NN, SEM
Park & Choi (2009)	LR
Veenstra, Dey, & Herrin (2009)	LR
Delen (2010)	DT, DTb, IF, LR, NN, RF, SVM
Jadrić, Garača, and Čukušić (2010)	DT, LR, NN
Yu, DiGangi, Jannasch, & Kaprolet (2010)	DT, MARS, NN
Zhang, Oussena, Clark, & Hyensook (2010)	DT, NB, SVM
Bogard, Helbig, Huff, & James (2011)	CE, DT, LR, NN
Nandeshwar, Menzies, & Nelson (2011)	BN, DT, DTb, NB, NN, OR
Lin (2012)	DT, DTb, NB
Yadav, Bharadwaj, & Saurabh, (2012)	DT
Alkhasawneh & Hargraves (2014)	NN
Chochran et al. (2014)	LR
Thammasiri et al. (2014)	DT, LR, NN, SVM
Aulck, Velagapudi, Blumenstock, & West (2016)	kNN, LR, RF

*BN = Bayesian Network; CE = Custom Ensemble; DA = Discriminant Analysis; DT = Decision Tree; DTb = Boosted Decision Tree; IF = Information Fusion; kNN = k-Nearest Neighbors; LiR = Linear Regression; LR = Logistic Regression; MARS = Multivariate Adaptive Regression Splines; NB = Naive Bayes; NN = Neural Networks; OR = One-Rule; PA = Path Analysis; PR = Probit Regression; RF = Random Forest; SA = Survival Analysis; SEM = Structural Equation Modeling; SVM = Support Vector Machine

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

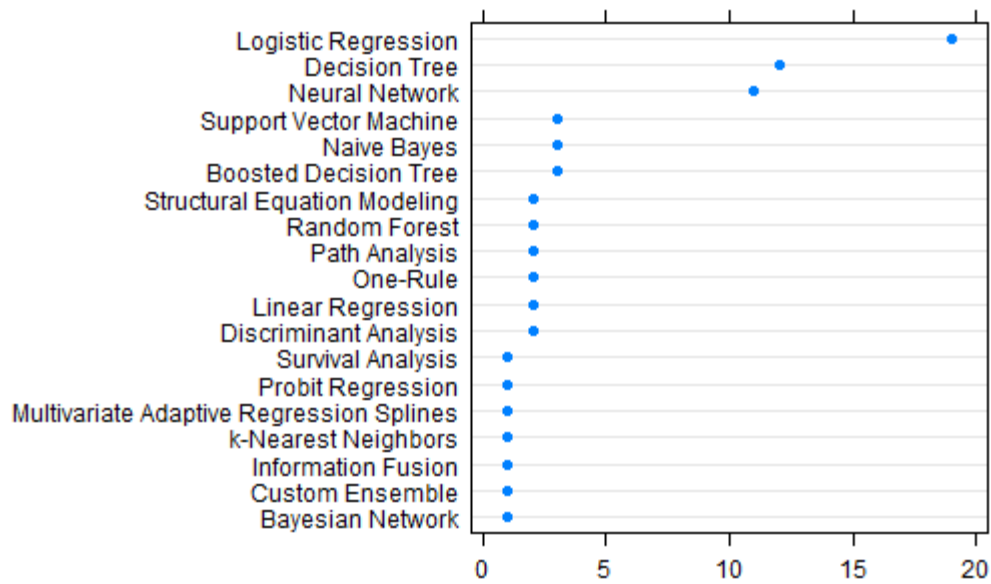


Figure 2. Frequency of modeling techniques used for dropout prediction.

There has been limited research on the comparison of modeling techniques for higher education, first-year dropout prediction. A total of 13 of the 31 publications compared the performance of modeling techniques and results have been inconsistent across studies. Dey and Astin (1993) were the first to conduct a comparative analysis of modeling techniques. They applied Logistic Regression, Probit Regression, and Linear Regression and found that these techniques produced similar results. Herzog (2006) compared C5 Decision Tree, Logistic Regression and Neural Network and found that the Decision Tree was most accurate. Sujitparapitaya (2006) and Jadrić et al. (2010) compared the exact same techniques and found that Neural Networks outperformed Logistic Regression and Decision Trees. Lin et al. (2009) compared Neural Networks, Logistic Regression, Discriminant Analysis, and Structural Equation Modeling and found that Neural Networks consistently performed the best. Delen (2010) compared individual

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

and ensemble models. The individual models were C5 Decision Trees, Logistic Regression, Neural Networks and Support Vector Machine. The ensemble methods were boosted Decision Trees, Information Fusion, and Random Forest. Delen found that the ensemble models outperformed the individual models. However, all of the final models had classification rates within one percent of each other. Yu et al. (2010) modeled Decision Trees, Multivariate Adaptive Regression Splines, and Neural Networks, but did not compare the classification rates of each model. Rather, they compared which variables were most influential in dropout prediction. Zhang et al. (2010) compared Naïve Bayes, Decision Trees, and Support Vector machine and found that Naïve Bayes had the highest accuracy rate. Based on misclassification rates, Bogard et al. (2011) found that Decision Trees outperformed Logistic Regression, Neural Networks and a custom ensemble model. Nandeshwar et al. (2011) compared One-R, C4.5 Decision Tree, boosted Decision Trees, Naïve Bayes, Bayesian Networks, and Neural Networks. They concluded that the more advanced modeling techniques did not significantly outperform the simple One-R algorithm. Lin (2012) compared Decision Trees, Naïve Bayes and boosted Decision Trees and found that the latter performed best based on Precision and Recall values. Thammasiri, Delen, Meesad, and Kasap (2014) compared Decision Trees, Logistic Regression, Neural Networks and Support Vector Machine with various class balancing techniques. They found that Support Vector Machines with SMOTE class balancing had the highest accuracy rate. Finally, Aulck et al. (2016) compared k -Nearest Neighbor, Logistic Regression and Random Forest and found that Logistic Regression had the highest ROC area and accuracy.

2.3.2 CLASS BALANCING

The class variable of a machine learning classification algorithms is the categorical outcome variable of model (James et al., 2013). For example, in the context of the current study, the class variable is dropout, which comprised of two levels: dropout and persist. The class variable is balanced if there is an equal proportion of dropout and persist instances. Conversely, the class variable is unbalanced if the proportion of dropout and persist instances are unequal. The performance of certain machine learning algorithms, such as Naïve Bayes, can be skewed if they are trained on unbalanced data (Witten, Frank, & Hall, 2011). For example, let's suppose the class variable is comprised of 99% persist instances and 1% dropout instances. The algorithm may learn to predict all instances as persist because this would be correct 99% of the time. Of course, this is not very useful.

Very few first-year dropout prediction publications have explored the effects of class balancing despite the regularity of class imbalance higher education. Table 2 lists the same publications from Table 1 and provides the sample size, the percent of the sample that dropped out, and any class balancing techniques that were applied. A total of 5 out of the 31 publications applied class balancing techniques. However, only 3 publications analyzed the effect of class balancing.

Table 2

<i>Dropout Publication Class Balancing Techniques</i>			
<u>Publication</u>	<u>Sample</u>	<u>% Dropout</u>	<u>Class Balancing</u>
Bean (1980)	906	12.2	-
Stage (1989)	323	9	-
Cabrera et al. (1993)	466	15.5	-
Dey & Astin (1993)	947	45.9	-
Murtaugh et al. (1999)	8,867	34.3	-
Bresciani & Carson, (2002)	3535	11.7	-
Garton & Ball (2002)	440	12	-
Glynn et al. (2003)	3244	50.9	Under-sampling
Salazar et al. (2004)	22,969	Unknown	-
Zhang et al. (2004)	14,084	49	-
Herzog (2005)	5261	23.7	-
Herzog (2006)	8,018	24.7	-
Sujitparapitaya (2006)	2445	20.5	-
Yu et al. (2007)	Unknown	Unknown	-
Fike & Fike (2008)	9,200	54.2	-
Sutton & Nora (2008)	576	Unknown	-
Lin et al. (2009)	1,508	20	-
Park & Choi (2009)	147	33.3	-
Veenstra et al. (2009)	Unknown	10	-
Delen (2010)	7,018	12	Unbalanced Under-sampling
Jadrić et al. (2010)	715	36.5	-
Yu et al. (2010)	6690	64.1	-
Zhang et al. (2010)	4223	Unknown	-
Bogard et al. (2011)	Unknown	Unknown	-
Nandeshwar et al. (2011)	33712	28.7	-
Lin (2012)	5943	15.7	Unbalanced Over-sampling
Yadav et al. (2012)	432	7.8	-
Alkhasawneh & Hargraves (2014)	1966	Unknown	-
Chochran et al. (2014)	2,314	16.8	-
Thammasiri et al. (2014)	21,654	21.3	Unbalanced Under-sampling Over-sampling SMOTE
Aulck et al. (2016)	32,500	23.5	Under-sampling only

Glynn et al. (2003) applied under-sampling, which is a class balancing technique that decreases the size of the majority class (i.e., persist) by randomly sampling the majority class until it is of equal size of the minority class (i.e., dropout). Glynn et al. (2003) modeled the balanced data set with Logistic Regression but did not model the

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

original unbalanced data set. Aulck et al. (2016) also applied under-sampling when training k -Nearest Neighbor, Logistic Regression and Random Forest models. However, Aulck et al. (2016) did not train the models on the original unbalanced data set. The effects of class balancing cannot be determined for both of these studies since the models were not trained on the unbalanced data sets.

Delen (2010) did a comparative analysis in which C5 Decision Trees, Logistic Regression, Neural Networks and Support Vector Machines were trained on the original unbalanced data set (12% dropout instances) and on a data set that was balanced via under-sampling (50% dropout instances). Delen (2010) found that class balancing decreased overall model accuracy from 87% to 79%, averaging across all four models. However, class balancing improved the accuracy of predicting dropout instances from 48% to 76%, averaging across all four models.

Lin (2012) trained Decision Trees, Naïve Bayes, and a series of boosted Decision Trees on the original unbalanced data set (15.7% dropout instances) and on a data set that was balanced via over-sampling. Over-sampling is a class balancing technique that increases the size of the minority class (i.e., dropout) by randomly sampling from the minority class (i.e., dropout) with replacement. That is, over-sampling simply duplicates dropout instances. Lin (2012) found that over-sampling increased the accuracy of predicting dropout instances from 9% to 55% when there were three-times as many dropout instances, averaging across all models.

Thammasiri et al. (2014) compared the effects of three class balancing techniques and the original unbalanced data set (21% dropout) on Decision Tree, Logistic

Regression, Neural Network and Support Vector Machine models. The class balancing techniques were under-sampling, over-sampling and synthetic minority over-sampling (SMOTE). SMOTE is a class balancing technique that creates synthetic minority class (i.e., dropout) instances (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). Thammasiri et al. (2014) found that all class balancing techniques improved the prediction of dropout instances. The average accuracy rate was 79% for models trained on the original data set, 83% for models trained on the over-sampled and under-sampled data sets, and 91% for models trained on the SMOTE data set.

2.3.3 FEATURE REDUCTION

Feature variables are the input variables that are used to predict the class variable in machine learning classification algorithms (James et al., 2013). Features are necessary for classification algorithms, however too many features, irrelevant features, or correlated features can impair model performance. Therefore, numerous feature reduction techniques have been created to reduce the number of features prior to model training, either by eliminating features entirely or by combining features through aggregation. Some feature reduction techniques require the researcher to analyze the feature reduction results and decide which features to remove. Conversely, some feature reduction techniques decide which features to eliminate all on their own.

There have been very few first-year, higher education, dropout prediction publications that have explored the effects of feature reduction. Table 3 lists the same publications from Table 1 and provides the number of features and any feature reduction techniques that were applied. A total of 4 out of the 31 publications applied feature

reduction techniques. However, only one of the publications analyzed the effect of feature reduction.

Table 3

<i>Dropout Publication Feature Reduction Techniques</i>		
<u>Publication</u>	<u>Features</u>	<u>Feature Reduction Technique</u>
Bean (1980)	58 → 28	Factor Analysis
Stage (1989)	39 → 17	Factor Analysis
Cabrera et al. (1993)	16	-
Dey & Astin (1993)	25	-
Murtaugh et al. (1999)	10	-
Bresciani & Carson, (2002)	17	-
Garton & Ball (2002)	5	-
Glynn et al. (2003)	250 → 62	Factor Analysis
Salazar et al. (2004)	16	-
Zhang et al. (2004)	6	-
Herzog (2005)	22	-
Herzog (2006)	40	-
Sujitparapitaya (2006)	14	-
Yu et al. (2007)	12	-
Fike & Fike (2008)	18	-
Sutton & Nora (2008)	24	-
Lin et al. (2009)	71	-
Park & Choi (2009)	8	-
Veenstra et al. (2009)	1	-
Delen (2010)	39	-
Jadrić et al. (2010)	24	-
Yu et al. (2010)	16	-
Zhang et al. (2010)	14	-
Bogard et al. (2011)	56	-
Nandeshwar et al. (2011)	103	-
Lin (2012)	21	-
Yadav et al. (2012)	9	-
Alkhasawneh & Hargraves (2014)	21 → 15	Genetic Algorithm
	20 → 16	
	20 → 11	
Chochran et al. (2014)	9	-
Thammasiri et al. (2014)	34	-
Aulck et al. (2016)	15	-

Bean (1980), Stage (1989), and Glynn et al. (2003) utilized factor analysis to reduce the number of features from 58 to 28, 39 to 17, and 250 to 62, respectively. In all three of these studies, the researchers used factor analysis to group similar survey items.

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

The researchers investigated the results of the factor analyses and determined if items were entered into the prediction model as latent factors, averaged to form composite features, or simply excluded altogether. In all cases, the researchers did not model the complete feature set and, therefore, conclusions cannot be made about the effects of feature reduction.

Alkhasawneh & Hargraves (2014) was the only study that compared the performance of models trained on a complete feature set and a reduced feature set. They utilized Neural Networks to predict student dropout for all students, ethnic majority students, and ethnic minority students. For all three samples, they applied the Genetic algorithm to automatically eliminate features. The number of features reduced for each data set was 21 features to 15 for all students, 20 to 16 for majority students, and 20 to 11 for minority students. They found that model accuracy improved from 74% to 75% for all students, 79% to 81% for majority students, and 60% to 63% for minority students.

2.3.4 FEATURE IMPORTANCE

The analysis of feature importance in higher education, first-year dropout prediction publications is relatively commonplace. Table 4 lists the same publications from Table 1 and provides the type of feature importance analysis that was conducted. A total of 24 out of the 31 publications investigated which features were most important to dropout prediction. However, most of these studies (19 out of 24) analyzed the feature importance for only one type of modeling technique. Only 5 out of the 31 studies assessed if features were important across multiple modeling techniques. However, only 3 of these studies made systematic comparisons.

Table 4

<i>Dropout Publication Feature Importance Techniques</i>	
<u>Publication</u>	<u>Feature Importance Technique</u>
Bean (1980)	Single Modeling Technique Interpretation
Stage (1989)	Single Modeling Technique Interpretation
Cabrera et al. (1993)	Single Modeling Technique Interpretation
Dey & Astin (1993)	Multiple Modeling Technique Comparisons
Murtaugh et al. (1999)	Single Modeling Technique Interpretation
Bresciani & Carson, (2002)	Single Modeling Technique Interpretation
Garton & Ball (2002)	Single Modeling Technique Interpretation
Glynn et al. (2003)	Single Modeling Technique Interpretation
Salazar et al. (2004)	Not Conducted
Zhang et al. (2004)	Single Modeling Technique Comparisons
Herzog (2005)	Single Modeling Technique Interpretation
Herzog (2006)	Not Conducted
Sujitparapitaya (2006)	Single Modeling Technique Comparisons
Yu et al. (2007)	Single Modeling Technique Interpretation
Fike & Fike (2008)	Single Modeling Technique Interpretation
Sutton & Nora (2008)	Single Modeling Technique Interpretation
Lin et al. (2009)	Stepwise Multiple Modeling Techniques Comparison
Park & Choi (2009)	Single Modeling Technique Interpretation
Veenstra et al. (2009)	Not Conducted
Delen (2010)	Multiple Modeling Technique Comparisons and Cross-Model Aggregation
Jadrić et al. (2010)	Not Conducted
Yu et al. (2010)	Multiple Modeling Technique Comparisons
Zhang et al. (2010)	Single Modeling Technique Interpretation
Bogard et al. (2011)	Single Modeling Technique Interpretation
Nandeshwar et al. (2011)	Single Modeling Technique Interpretation
Lin (2012)	Not Conducted
Yadav et al. (2012)	Not Conducted
Alkhasawneh & Hargraves (2014)	Not Conducted
Chochran et al. (2014)	Single Modeling Technique Interpretation
Thammasiri et al. (2014)	Single Modeling Technique Interpretation
Aulck et al. (2016)	Single Modeling Technique Interpretation

Dey and Astin (1993) compared feature importance across Linear Regression, Logistic Regression, and Probit Regression models and found that the same five features were statistically significant across the modeling techniques. Additionally, the relative magnitude of importance for each feature was consistent across modeling techniques.

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

Sujitparapitaya (2006) modeled Decision Tree, Logistic Regression, and a Neural Network. Feature importance was conducted indirectly as a product of the author discussing the anatomy of each model. However, a formal comparison of all features across each model was not conducted and therefore substantial conclusions cannot be made.

Lin et al. (2009) trained a Neural Network, Logistic Regression, Discriminant Analysis, and a Structural Equation Model on six different groups of features that were subsets of themselves. Lin compared how the stepwise inclusion of each feature group affected model accuracy. Since Lin did not rank the importance of each feature group, this is at best an indirect examination of feature importance across modeling techniques.

Delen (2010) conducted an analysis of feature importance across Decision Tree, Logistic Regression, Neural Network, and Support Vector Machine models. The relative importance of features in each model was normalized, and then these normalized values were summed to measure the overall importance of each feature. Delen concluded that, regardless of modeling technique, the most important features for predicting dropout are those related to prior educational success, present educational success, and financial aid status.

Yu et al. (2010) utilized Decision Trees, Multivariate Adaptive Regression Splines and Neural Networks to model dropout. Additionally, he compared the most important features across the three models. Yu did not compare the importance of features systematically like Delen (2010) did. Rather, Yu simply noted that transferred

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

hours, residency, and ethnicity were consistently ranked as the most important features across each model.

2.4 SUMMARY

To be the best of the author's ability, all higher education, first-year dropout prediction publications were reviewed. A total of 31 publications were evaluated. Across all 31 publications, Logistic Regression was applied 19 times, Decision trees 12 times, Neural Networks 11, and all other modeling techniques were applied 3 or fewer times. There has been limited research on the comparison of modeling techniques for higher education, first-year dropout prediction. A total of 13 of the 31 publications compared the performance of modeling techniques and results have been inconsistent across studies. Very publications have explored the effects of class balancing despite the regularity of class imbalance higher education. A total of 5 out of the 31 publications applied class balancing techniques. However, only 3 publications analyzed the effect of class balancing. Very few publications have explored the effects of feature reduction. A total of 4 out of the 31 publications applied feature reduction techniques. However, only one of the publications analyzed the effect of feature reduction. The analysis of feature importance is relatively common across the 31 publications. However, only 3 out of the 31 studies systematically assessed if features were important across multiple modeling techniques.

CHAPTER 3: METHODOLOGY

3.1 INTRODUCTION

This chapter begins with a description of the study population. Then the data sources and associated variables are described. The machine learning algorithms are described in detail and their relative strengths and weaknesses are used to make hypotheses. The methods used to create the four data sets are explained. The criteria for evaluating models are defined as well as the methods used to evaluate feature importance.

3.2 STUDY POPULATION

Degree seeking, first-time National University students from academic year 2015 and 2016 were included in the study. That is, all students that attended National University for the first time (prior enrollments at other institutions were permitted) and started an Associate, Bachelor or Master degree program between July 1st 2014 and June 30th 2016 were included in the study. A student is considered to have started a program once they have been enrolled in a course for more than nine days.

National University follows a one class per month academic calendar. Except for project and independent study courses, all classes at National University are four weeks long and they start twelve times each year. The current study sought to predict student dropout at the earliest point possible in the students' academic career. Therefore, data used for dropout prediction were limited to those collected between the students' application date and the end the students' first class. Several data sources exist during this time period and they are discussed in the next section.

3.3 DATA SOURCES

National University utilizes PeopleSoft Campus Solutions to house their student data. At the time of this study, Institutional Research did not have a seamless connection to the data stored in PeopleSoft. Therefore, Institutional Research, with assistance from Information Technology, utilized PeopleSoft's query feature to create queries to access the data through a web-interface. PeopleSoft queries produce flat file extracts and Institutional Research used Microsoft SQL Server 2012 to create their own data warehouse of frequently used queries to expedite data retrieval. PeopleSoft query names were used as the table names within the Institutional Research data warehouse. Institutional Research warehoused data from several other sources in addition to PeopleSoft. The Institutional Research data warehouse was the exclusive source of data for the current study. A total of twelve tables were utilized to create the analysis data set and the following sections describe each of these tables.

3.3.1 STUDENT CLASS ACTIVITY REPORT

The Student Class Activity Report (SCAR) was the primary table that all other tables were joined to. This table provided student-level enrollment activity for all National University students since the University first began in 1971. Each row represents a student enrollment in a course and the columns provide information about the course, instructor and student.

SCAR was used for several purposes. This table was used to identify the study population and it provided essential information such as course attributes, instructor attributes, student academic plan attributes and student grades. SCAR was also used to calculate aggregate class statistics, such as total enrollment and median grades. Finally,

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

student dropout, the outcome measure of this study, was obtained from SCAR by examining the future enrollment activity of students.

3.3.2 ALUMNI

The Alumni table was used to identify students that graduated. It was used in conjunction with SCAR to identify student dropout. This table was used only for model development and it will not be used for dropout prediction of non-historical student data.

3.3.3 BIO DEMO

The Bio Demo table provided demographic information for each student, such as gender, ethnicity, address, date of birth, etc. These data are collected during the students' application but can be updated at any time. Bio Demo includes the most recent data available for each student.

3.3.4 ACADEMIC PLANS

The academic plans table provided additional academic plan descriptive data. This table was used exclusively for the Classification of Instructional Programs (CIP) codes that are associated with each academic plan. CIP codes were created by the U.S. Department of Education's National Center for Education Statistics to support the assessment of fields of study across all institutions (National Center for Education Statistics, 2010). CIP codes form a hierarchy and current study utilized the two-digit level (i.e., the highest level) in order to categorize student academic plans.

3.3.5 ACADEMIC ADVISEMENT REPORT

The Academic Advisement Report (AAR) defines the course requirements for each academic plan. This table was used to identify the major prep, major core, and

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

suggested major elective courses relative to each student. The purpose of this table is to identify if students enrolled in courses that directly supported their major (i.e., major prep, major core, and suggested electives courses) or if they enrolled in courses that did not directly support their major (i.e., general education courses).

3.3.6 END OF COURSE EVALUATIONS

During the last week of each course, students are given the opportunity to evaluate their course via an online standardized survey (see Appendix A). Students are not required to complete the survey, however, they are incentivized with a monetary raffle. The typical response rate for this survey is 40% each term. The end of course evaluation (ECE) survey consists of three sections and a total of twenty-one questions. The first two sections contain five-point Likert-scale questions with response options “Strongly Disagree”, “Disagree”, “Neutral”, “Agree”, “Strongly Agree”, and an additional “Not Applicable” option. The first section contains eight questions that address the students’ perception of learning for the given course. Averaging these eight items provided an overall student rating of the course content for each student. The second section contains twelve questions that address the students’ perception of teaching for the given course. Averaging these twelve items provides an overall student rating of the instructor for each student. The third section of the survey contains one free-response question in which each student can share whatever thoughts they have regarding the course. Although, this final question likely contains a wealth of valuable information, these unstructured data were not included in the current study.

3.3.7 SERVICE INDICATORS

The Service Indicator table provided an audit trail of various service events that students receive. Examples of services are financial holds, scholarships, military discounts, and so forth. Each row in the table represents the activation or removal of a service event associated with a students' account, and each of these events has a corresponding timestamp. The beauty of this table is that its structure makes it possible to examine what service events are active at any point in time.

3.3.8 TRANSFER COURSE DETAIL

The Transfer Course Detail table provided course-level transfer credits for each student. It identifies courses that were successfully applied to the students' NU academic plan and the table was used to calculate how many transfer credits were used to fulfill National University degree requirements. Additionally, this table was used to examine prior education enrollment.

3.3.9 NATIONAL STUDENT CLEARINGHOUSE

The National Student Clearinghouse (NSC) is nonprofit education organization that facilitates the exchange of student enrollment, performance and related information (National Student Clearinghouse, 2017). NSC enables institutions to verify prior degree completion and/or previous enrollment of students at external institutions. Institutional Research warehouses these data and the current study will examine the degree completion and enrollment behavior of students prior to their admission to National University. However, obtaining prior student enrollment activity through NSC is accomplished by matching the students' first and last name, and then also matching the students' social security number or birth date. Individuals often change their name after

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

marriage and it is common for marriage to occur after college. Therefore, in these circumstances, it is impossible to find prior educational activity.

3.3.10 NU EXTERNAL EDUCATION

Student transcripts collected during the application process populate the NU External Education table. The benefit of this table is that student name changes are not an issue. However, the drawback of this table is that not all degree awards earned are recorded. For example, if a student applies to a Master's program at National University and they previously completed a Master's and Bachelor's degree elsewhere, only the Bachelor's degree information is recorded. Furthermore, prior enrollment that does not result in a completed degree are omitted as well.

Transfer Course Detail, National Student Clearinghouse, and NU External Education tables each have their strengths and weaknesses. Transfer Course Detail provides prior enrollment activity only if the student wishes to apply their previous coursework to their National University academic plan. Transfer Course Detail also does not include prior degree completion data. National Student Clearinghouse provides detailed enrollment activity but such data cannot be found if students change their name. The NU External Education data are robust to student name changes but only prior degree completions are included (i.e., prior enrollments that did not result in a degree completion are not included). Therefore, Transfer Course Detail, National Student Clearinghouse and the NU External Education were used in conjunction to assess prior enrollment.

3.3.11 FAFSA

The FAFSA table provided student financial aid data from the Free Application for Federal Student Aid (FAFSA) form that students complete to determine their financial aid eligibility. This table provided valuable data such as parent's highest education, estimated family contribution, marital status, number of dependents, adjusted gross income, etc. However, these data are available only if a student completes a FAFSA application. Roughly half of the study population completed a FAFSA application.

3.3.12 STATISTICS OF INCOME BY ZIP CODE

Publicly available Statistics of Income (SOI) data by ZIP Code for tax year 2014 were obtained for the current study (United States Internal Revenue Service, 2014). These data were used to create a proxy measure of adjusted gross income for each student based on the per-capita adjusted gross income of the students' home ZIP Code. These data were included because actual adjusted gross income is only available for students that complete a FAFSA (i.e., half of the study population).

3.4 FEATURES

A total of 77 features were included in model building (one class features and 76 input features). The class feature, Dropout, was previously defined in section 1.6.4. The 76 features covered a wide domain of measures that were previously identified as predictive of dropout (Cochran et al., 2014; Leitner et al., 2017; Park & Choi, 2009; Sparkman, Maulding, & Roberts, 2012). Of the 77 features, 39 were observed measures and 38 were derived. Missing values for numeric features were substituted with the average feature value across all instances. Missing values for categorical features were assigned distinct value (e.g., "Unknown"). Since there are 76 features in the current

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

study, they will not be defined in detail. Rather, they are described in Table 5.

Additionally, Appendix B contains the SQL code used to gather/create all features and thus provides a detailed explanation of exactly how features were created.

Table 5

Features Included in the Current Study

<u>Variable</u>	<u>Data Type</u>	<u>Source(s)</u>	<u>Description</u>
Dropout*	Char (2)	SCAR & Alumni	Whether the student Dropped out or Persisted to their second academic year
AdjunctOnly*	Char (2)	SCAR	Whether the student had only Adjunct professors during their first NU term
DegreeAwardType	Char (3)	SCAR	Degree award type sought
ClassLength_avg*	Integer	SCAR	Length (in days) of students' first NU class. Average number of days if enrolled in more than one class during their first term
Classes*	Integer	SCAR	Number of classes taken during first NU term
Units*	Numeric	SCAR	Total units taken during first NU term
DFUWI*	Char (2)	SCAR	Student received a grade of D(+/-), F, Unsatisfactory, Withdrew, or Incomplete during their first term at NU
OnlineOnly*	Char (2)	SCAR	Student took only online classes during first term at NU
RelativePerf*	Char (2)	SCAR	The student's grade was compared to the median grade of the entire course and the student was categorized as being above or below the median. If the student took multiple classes, the calculation was weighted based on the unit load.
SIC_avg*	Integer	SCAR	Average class size for the student during first term at NU
DaysToFC*	Integer	SCAR	Number of days between the date the student registered for their first NU class and the date the class started
DaysToFYClose*	Integer	SCAR	Number of days between the end of the students first class at NU and the next academic year (i.e., how many days until they persist)
FCD_FYQ*	Char (4)	SCAR	The academic quarter that the student started. Q1 = July-Sept; Q2 = Oct-Dec; Q3 = Jan=Mar; Q4 = Apr-Jun
RemedialCrs*	Char (2)	SCAR	Whether the student took a remedial course
ClassCPE*	Char (2)	AAR & SCAR	Whether the student enrolled in at least one major core, major prep, or major suggested major elective course during their first term.
ActiveDuty*	Char (2)	BioDemo	Whether the student reported their military status as Active Duty

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

Gender	Char (3)	BioDemo	Self-reported gender
Ethnicity	Char (9)	BioDemo	Self-reported ethnicity
Age*	Integer	BioDemo & SCAR	Age at first class date
MilitaryYN*	Char (2)	BioDemo	Dichotomize Military Status to Y/N military
Probation	Char (2)	BioDemo	Whether the student entered NU on academic probation
CIP2D*	Char (21)	Acad Plans	Hierarchically grouped 6-digit CIP codes to 2-digits
LearningGrp*	Char (3)	ECE	Perception of Learning items from End of Course Evaluation survey were average and then categorized as Positive if > 3, Negative if <= 3, or No Response
TeachingGrp*	Char (3)	ECE	Perception of Teaching items from End of Course Evaluation survey were average and then categorized as Positive if > 3, Negative if <= 3, or No Response
N_BCW	Char (2)	SI	Partial Payments Due
N_BKA	Char (2)	SI	Service Indicator: Barred from Attendance
N_BLK	Char (2)	SI	Service Indicator: Business Office Lock
N_BPP	Char (2)	SI	Tuition Pre-Payment
N_CRD	Char (2)	SI	Credential Office Lock
N_ECD	Char (2)	SI	T2T Scholarship Denied
N_FPW	Char (2)	SI	Pell Lifetime Eligibility Warn
N_Hold*	Char (2)	SI	Any financial hold
N_IntlStu*	Char (2)	SI	International student
N_REC	Char (2)	SI	Registrar's Lock
N_WriteOff*	Char (2)	SI	At least one type of write off
P_B2B	Char (2)	SI	B2B Scholarship Tuition Disc
P_BCB	Char (2)	SI	Third Party Billing
P_BCR	Char (2)	SI	Business - Credit Balance
P_BOM	Char (2)	SI	Military Discount
P_BRR	Char (2)	SI	Refund Request
P_BSP	Char (2)	SI	Sponsored Program
P_BSS	Char (2)	SI	Staff Scholarship
P_BV3	Char (2)	SI	Military Chapter 33
P_BVA	Char (2)	SI	Veterans Affairs
P_BVO	Char (2)	SI	Vocational Rehabilitation
P_ECP	Char (2)	SI	CCC Preliminary Eligible
P_ECS	Char (2)	SI	CCC Scholarship Eligible
P_EPS	Char (2)	SI	Pathways to Success
P_FAC	Char (2)	SI	Campus FA Advisement
P_FAO	Char (2)	SI	Online FA Advisement
P_FAS	Char (2)	SI	Financial Aid Student
P_FBA	Char (2)	SI	Regent Military Discount
P_FRD	Char (2)	SI	Regent SOAR Disqualified
P_HIC	Char (2)	SI	Health Insurance - Intl Students
P_MiscSchlr*	Char (2)	SI	All other scholarships not already specified
P_NOS	Char (2)	SI	No Solicitation
P_OffsiteDisc*	Char (2)	SI	Any sort of offsite discount
P_REV	Char (2)	SI	Assigned for Formal Evaluation

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

P_RFA	Char (2)	SI	Registrar's Admissions Flag
P_ROR	Char (2)	SI	Release of Student Records
P_SOC*	Char (2)	SI	Any sort of military Service members Opportunity Colleges participant
P_VaTuit*	Char (2)	SI	Any Veteran's Affairs tuition assistance
TransUnits*	Numeric	Trans	Approved transfer units
DaysSinceLastAtt*	Integer	CHD, Ext Ed, Trans	Date difference between max external education date and NU first class date
PrevDeg*	Char (3)	CHD, Ext Ed, Trans	Previous degree is higher, lower or equal to degree award type sought
PrevAtt4Yr*	Char (3)	CHD, Ext Ed, Trans	Previously attended 4-year institution
PrevAtt2Yr*	Char (3)	CHD, Ext Ed, Trans	Previously attended 2-year institution
PreGPA*	Numeric	Ext Ed, Trans	External GPA - highest selected if multiple were available
FAFSAby*	Char (3)	FAFSA	When their FAFSA application was approved by. (first class start, first class end, or never)
ParentHiEd*	Char (4)	FAFSA	Highest education of any parent (Unknown; Middle School; High School; College)
AGI*	Integer	FAFSA	Actual adjusted gross annual income
GovtPrgmY*	Char (2)	FAFSA	Participated in food stamps, school lunch, SSI, TANF, or WIC
DependentsY	Char (2)	FAFSA	Has at least one dependent
ChildrenY	Char (2)	FAFSA	Has at least one child
MaritalStat	Char (5)	FAFSA	Marital Status. Single; Unknown; Divorced; Married; Separated
EFC	Integer	FAFSA	Expected family contribution
AGI_PerCapita*	Integer	SOI	Adjusted Gross Income for student's home ZIP Code

“*” = *derived variable*

3.5 MACHINE LEARNING ALGORITHMS

Nine classification machine learning algorithms were modeled to predict dropout. These algorithms were chosen because they offer a sample of regression, probabilistic, instance-based, kernel-based, tree-based, and ensemble machine learning methods. Each algorithm is described in the following sections and evaluated with the following

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

concerns: class imbalance, sparsity, outliers, high-dimensionality, correlated features, and nonlinearity.

Class imbalance, sparsity, outliers, high-dimensionality, correlated features, and nonlinearity were of issues of concern because they were known to exist in the study data. It was known that more students at NU persist instead of dropout, thus resulting in an unbalanced class. It was known that very few students have certain service indicators, thus resulting in sparsity. It was known that some students came from exceptionally high/low economic backgrounds, thus resulting in outliers. It was known that the interaction of the 76 features were going to be tested, thus resulting in potentially high-dimensionality. It was known that many of the features, particularly the financial ones, were very related, thus resulting in correlated features. Finally, although it was not known, it was expected that at least one of the 76 features had a nonlinear relationship with dropout, thus making nonlinearity a concern.

Common concerns such as training time and model interpretability were not considered because the goal of the study was to model dropout as accurately as possible. Training time and model interpretability are important concerns during the implementation phase of an advanced analytics project. Model interpretability is particularly important for garnering support from faculty, staff and administration. However, such concerns have zero effect on the model's ROC area or accuracy and are therefore not considered in the current study.

3.5.1 LOGISTIC REGRESSION

Logistic Regression (LR) is a commonly used statistical technique for modeling binary outcomes that was developed in the 1950s (Cox, 1958). LR is a linear model that has been transformed to handle binary outcomes via a logit function (Witten et al., 2011). It measures the relationship between the features and the class by estimating probabilities via the logit function. Each feature is assigned a weight and the weights are optimized to create the most accurate probability estimates. Since LR is probabilistic, it can be heavily influenced by class imbalance and sparsity. Infrequent instances are treated with equal weight as frequent instances so their unique contribution can be overshadowed. LR is a linear model so outlier values can cause problems by skewing the probabilities. LR is a statistical model that, due to constraining degrees of freedom, cannot handle high-dimensional spaces. LR can model nonlinear features but doing so requires manual manipulation of the data set. Furthermore, systematically testing the nonlinear relationship among all features has the potential to increase the dimensionality to the point of non-convergence. Correlated/redundant features can cause the feature weights to be unstable. The feature weights are unstable because they are shifting importance from one redundant feature to the other to maintain classification accuracy. From a statistical perspective, this is inconvenient because the interpretation of the feature weights may be misleading. However, from a machine learning perspective, classification is paramount and no harm is done.

3.5.2 NAÏVE BAYES

Naïve Bayes (NB) is a simple probabilistic classifier that has been widely used for decades (Witten et al., 2011). It is based on Bayes' theorem, which describes the

probability of an event based on prior knowledge of events that might be related to the event (i.e., the conditional probability). NB estimates the likelihood of an event (i.e., classification) by considering the conditional probabilities of each feature. NB assumes that each feature is independent (i.e., conditional independence) which results in the algorithm performing poorly when correlated features exist. Conditional independence also results in sparse features contributing little to the final classification. That is, infrequent occurrences are treated with equal weight as frequent occurrences so their unique contribution can be in effect overshadowed. However, this aspect causes NB to be very robust to outliers because, again, unusual occurrences have the same contribution as usual occurrences. NB can perform poorly when classes are heavily unbalanced because of the probabilistic nature of NB. If, for example, a data set contained very few instances of dropout, the model would learn that probability of dropout is low and would likely misclassify these instances. However, the probabilistic nature of NB results in the algorithm being well adapted to high-dimensional data, assuming the features are independent.

3.5.3 K-NEAREST NEIGHBOR

K-Nearest Neighbor (KNN) is an instance-based, non-parametric algorithm that is among the simplest of all machine learning algorithms (Witten et al., 2011). Instance-based methods measure the distance of instances to class labels during training. Typically, Euclidean distance is used as the distance measure. The training instances are literally saved in memory so that they can be used for classifying future instances. When future instances are passed through the model, they are compared to the k previously stored instances. Those k instances are then compared to k training instances and the

classification is determined by majority vote based on the k closest instances (i.e., the nearest neighbors). k is chosen during model training by trying a variety of different values. For binary classification, it is best to choose an odd value for k to avoid the possibility of a tie vote. KNN handles unbalanced classes by weighting the instances proportional to the class imbalance so that the voting is balanced. Since KNN is a nonparametric algorithm, it can handle nonlinearity and sparse features fine. However, since KNN is estimated by a distance measurement, it does not handle, outliers, high dimensionality or correlated/redundant features well. Outliers have large distance values and can skew the majority vote. High dimensionality and redundant features cause increased noise for KNN and thus negatively impact classification.

3.5.4 NEURAL NETWORK

Neural Networks (NN) are a category of algorithms that are inspired by the anatomy of the brain (Witten et al., 2011). A simple NN contains an input layer, a hidden layer, and the output classification. The input and hidden layers are comprised of perceptron algorithms. The perceptron is a simple algorithm, very similar to logistic regression, that takes a numeric input and outputs a binary classification. The input layer is fed the raw data and each feature corresponds to one perceptron input. The input layer outputs binary classifications that are then processed by the hidden layer. The hidden layer then outputs the class labels. NNs are fundamentally linear models and each perceptron has an associated weight. The perceptron weights are determined by processing the instances iteratively through a process known as backpropagation. The weights are adjusted after each input and gradient descent is employed to find the weights that have the least misclassification rate. NNs are probabilistic and suffer from some of

the same issues as Logistic Regression. Like logistic regression, they can have issues dealing with class imbalance, sparse features and outliers because the relative occurrence, or lack thereof, of the instances greatly impacts the model weights. However, unlike Logistic Regression, NNs work exceptionally well with nonlinear features and high-dimensionality. NNs work well under these circumstances because of the hidden layer, which can simultaneously model interaction effects and reduce the dimensionality of the feature space.

3.5.5 SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is a non-probabilistic linear model that was developed in the 1990s and designed for binary classification (James et al., 2013). SVM learns the maximum-margin hyperplane to find the greatest separation between classes within the data. The maximum-margin hyperplane is determined by support vectors, which are instances that lie on the maximum-margin hyperplane. Only these support vectors are used to determine class membership. Therefore, SVMs handle outliers exceptionally well because the results are not skewed by extreme values. Since SVM is non-probabilistic, class imbalance feature sparsity is not an issue. Arguably the greatest strengths of SVM is its ability to handle nonlinear and correlated features in high-dimensional space. SVM can handle high-dimensionality and correlated features because it only considers support vectors that lie on the maximum-marginal hyperplane. This makes SVM extremely efficient. Kernel functions can be applied to the vectors to map the data to a nonlinear space. Kernel functions (e.g., polynomial, radial, and sigmoid) are applied to the dot product of the features to create this nonlinear mapping. The current study modeled a polynomial SVM in addition to a linear SVM.

3.5.6 DECISION TREE

Decision tree classification algorithms were pioneered by Breiman, Friedman, Stone and Olshen (1984). They are a set of non-parametric, rule-based algorithms. In classification Decision Trees (DT), each node represents a feature, each branch represents a value, and the classification occurs at the terminal nodes. The DT is learned from the data and optimized to find the most direct classification path. Therefore, the most influential features occur at the top of the tree. Features that do not contribute to the decision process are not included in the model (i.e., they are “pruned” away). However, correlated/redundant features can misguide the decision process and incorrectly change the trajectory of the DT. Because DTs are non-parametric, they are very robust to class imbalance and sparse features. That is, because DTs do not make decisions based on the relative frequency of events, infrequent instances are not overshadowed by frequently occurring instances. DTs handle numeric features by discretizing them into categorical groups of numerical ranges. For example, the numeric feature Age might be discretized into three groups: less than 25 years old, 25 to 50 years old, and over 50 years old. The groups are optimized by the algorithm and this process has its pros and cons. It makes capable of dealing with nonlinear terms because the linearity of the feature is removed once discretized. However, DTs do not handle outliers very well because they influence the cut-point discretization.

3.5.7 RANDOM FOREST

Random Forest (RF) is an ensemble method that is an extension of Decision Trees. An ensemble model is an algorithm that generates many individual models and then combines their classifications to make a final decision. Ensemble models have been

shown to consistently outperform single models (James et al., 2013). Random Forest gets its name because it creates several *Decision Trees* (i.e., “Forest”) and because these trees are made with a random sample of instances (a.k.a., bagging) and a random sample features (Breiman, 2001). Since RFs are an extension of DTs, they are equally as good at handling class imbalance, sparsity, high-dimensionality and non-linearity. However, in contrast to simple DTs, RF are well equipped to handle correlated/redundant features because of the randomization of features. Furthermore, RF are well equipped to handle outliers since the random sampling of instances reduces the influence of unusual instances.

3.5.8 EXTREME GRADIENT BOOST

Extreme Gradient Boost (XGBoost) is another ensemble method that is an extension of Decision Trees (Chen & Guestrin, 2016). The algorithm produces multiple trees but the trees are not created independently. Rather, each tree is built in an iterative fashion and subsequent trees compensate for the mistakes of the previous tree by making the incorrectly classified instances more influential. This iterative process is called “boosting” because the relative importance of the hard to classify instances are “boosted” (i.e., increased). XGBoost also randomly samples features like Random Forest. Because of this, XGBoost is well equipped to handle correlated/redundant features. Since XGBoost is an extension of Decision Trees, it is equally good at handling class imbalance, sparsity, high-dimensionality and non-linearity. Additionally, XGBoost handles outliers well because the iterative “boosts” adapt to these extreme values.

3.5.9 COMPARISON OF STRENGTHS AND WEAKNESSES

The strengths and weaknesses of each algorithm from the preceding sections are summarized in Table 6. The last column of Table 6 provides a simple count of how many strengths each algorithm has. Based the number of strengths, it was hypothesized that Support Vector Machine with a polynomial kernel (SVMP), Random Forest (RF), and XGBoost (XGB) will perform best since these algorithms have six out of the six possible strengths.

Table 6

Algorithm Strengths and Weaknesses

Algorithm	<u>Class</u> <u>Imbalance</u>	<u>Sparsity</u>	<u>Outliers</u>	<u>Dimensionality</u>	<u>Correlated</u>	<u>Nonlinearity</u>	<u>#S</u>
LR	W	W	W	W	S	W	1
NB	W	W	S	S	W	W	2
NN	W	W	W	S	S	S	3
KNN	S	S	W	W	W	S	3
SVML	S	S	S	S	S	W	5
SVMP	S	S	S	S	S	S	6
DT	S	S	W	S	W	S	4
RF	S	S	S	S	S	S	6
XGB	S	S	S	S	S	S	6

S = Strength
W = Weakness

The following sections define how these algorithms were trained and evaluated. Although, efforts were taken to reduce the effects of class imbalance and high-dimensionality, it is still expected that SVMP, RF and XGB will still perform best under these conditions. That is, if the strengths/weaknesses for class imbalance and high-dimensionality are ignored from Table 6, SVMP, RF and XGB still have the most strengths and are therefore expected to perform best.

3.6 ALGORITHM TRAINING AND EVALUATION

3.6.1 SOFTWARE

R version 3.4.1 (2017-06-30) was used to analyze all data for the current study (R Core Team, 2017). All R code are provided in Appendices C1-C5. Version 6.0-76 of the caret package was used to train all algorithms (Kuhn, 2017). Caret stands for “classification and regression training” and the package is a set of functions that streamlines the process of creating predictive models (Kuhn, 2017). Caret streamlines the machine learning process by connecting various different R packages and gives common tasks uniform names. The package was primarily used for data splitting, pre-processing, feature selection, model tuning, and feature importance estimation.

3.6.2 TRAINING

Four different versions of the population data set were created to examine the effects of feature reduction and class imbalance. Figure 3 offers an illustrative example of the four data sets. The first data set (CU) was the population data set. This data set contained all features (i.e., complete) and all instances. The class for this data set is unbalanced since there are proportionally fewer students that dropped out than persisted.

The second data set (RU) contained a reduced feature set and all instances (i.e., unbalanced class). The features were reduced via recursive feature elimination (RFE). RFE is a wrapper method that evaluates multiple models by adding or removing features to find the optimal combination of features that maximizes model performance (John, Kogavi, & Pfleger, 1994). Specifically, RFE with a Random Forest function was applied and 76 models were tested to find the best combination. That is, the first model included

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

all features. Then the second model started out with all 76 features but was forced to finish with only 75 features, thus eliminating the weakest feature. Then the third model started out with all 76 features but was forced to end with only 74 features, thus eliminating the two weakest features. This process continued iteratively until the final model began with all 76 features but ended with one single, most predictive feature. The algorithm then compares the ROC area of all 76 models. The model with the highest ROC value determines what features should be retained. For example, if the model with the highest ROC area contained 40 features, these 40 features would then be included in the second data set.

The third data set (CB) contained all features (i.e., complete), however, several instances were removed from the persisted class to have an equal proportion of persist and dropout instances. That is, the class was balanced so that the proportion of dropout and persist instances was 50% each. Under-sampling (US) was applied to the persist class and zero instances from the dropout were removed. Once balanced, the order of the instances was randomized to avoid any bias that might arise during model development.

Finally, the fourth data set (RB) contained a reduced feature set and balanced class. This data set was created by applying RFE to the third data set (CB). Again, RFE with Random Forest as the function were applied.

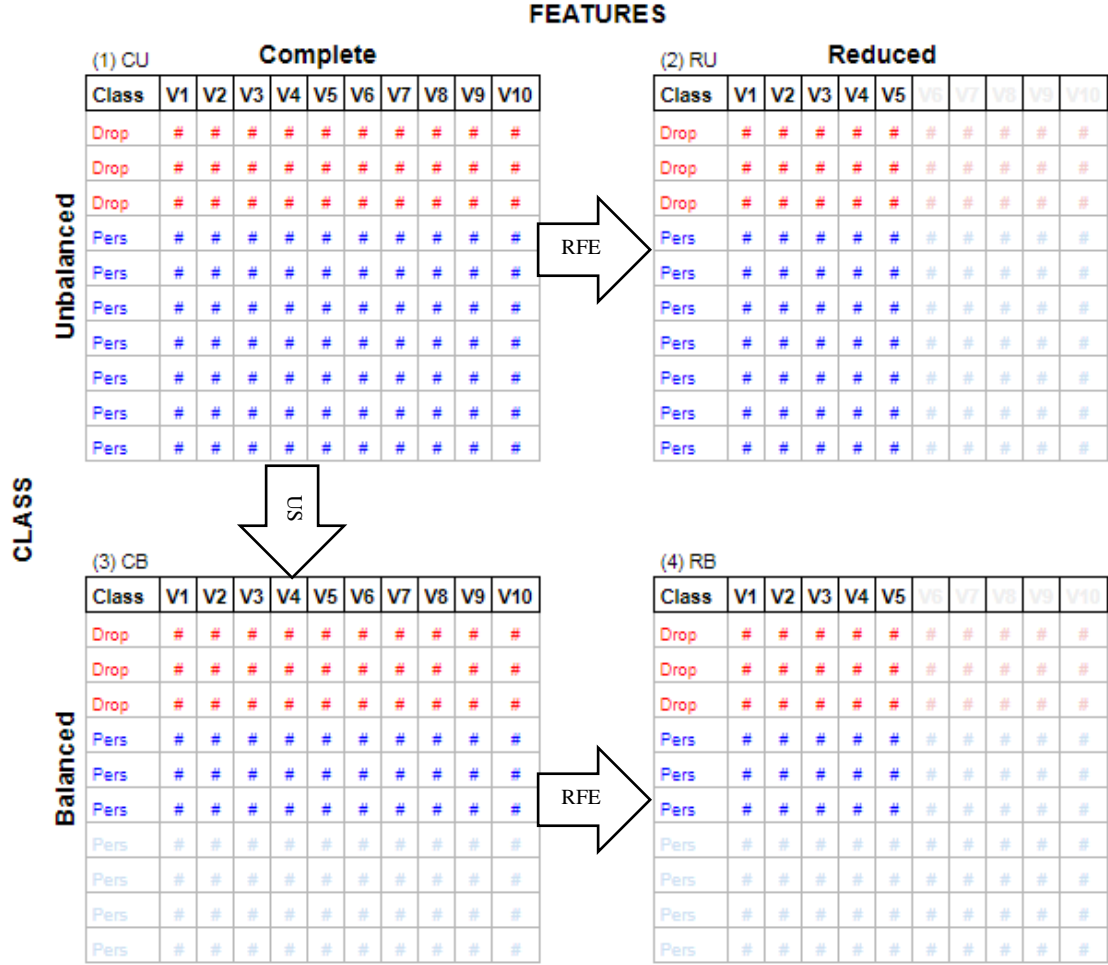


Figure 3. Illustration of the four data sets.

Each of the four data sets were split into training and testing data sets, thus producing eight data sets. Seventy percent of the data were allotted to training and thirty percent were allotted to testing. The algorithms were trained on the training data sets with tenfold cross-validation. The nine algorithms were applied to each of the four training data sets, thus resulting in 36 different models. Table 7 defines the model naming conventions that will be used throughout this paper. Once trained, the 36 models were applied to their corresponding testing data sets. Model predictions were compared to the

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

observed classifications. Three evaluation metrics were used to evaluate the predicted and observed classifications, as discussed in the following section.

Table 7

<i>Model Naming Conventions</i>				
<u>No.</u>	<u>Name</u>	<u>Algorithm</u>	<u>Features</u>	<u>Class</u>
1	LR_cu	Logistic regression	Complete	Unbalanced
2	NB_cu	Naïve Bayes	Complete	Unbalanced
3	NN_cu	Neural Network	Complete	Unbalanced
4	KNN_cu	k-Nearest Neighbor	Complete	Unbalanced
5	SVML_cu	Support Vector Machine - Linear	Complete	Unbalanced
6	SVMP_cu	Support Vector Machine - Polynomial	Complete	Unbalanced
7	DT_cu	Decision Tree	Complete	Unbalanced
8	RF_cu	Random Forest	Complete	Unbalanced
9	XGB_cu	XGBoost	Complete	Unbalanced
10	LR_ru	Logistic regression	Reduced	Unbalanced
11	NB_ru	Naïve Bayes	Reduced	Unbalanced
12	NN_ru	Neural Network	Reduced	Unbalanced
13	KNN_ru	k-Nearest Neighbor	Reduced	Unbalanced
14	SVML_ru	Support Vector Machine - Linear	Reduced	Unbalanced
15	SVMP_ru	Support Vector Machine - Polynomial	Reduced	Unbalanced
16	DT_ru	Decision Tree	Reduced	Unbalanced
17	RF_ru	Random Forest	Reduced	Unbalanced
18	XGB_ru	XGBoost	Reduced	Unbalanced
19	LR_cb	Logistic regression	Complete	Balanced
20	NB_cb	Naïve Bayes	Complete	Balanced
21	NN_cb	Neural Network	Complete	Balanced
22	KNN_cb	k-Nearest Neighbor	Complete	Balanced
23	SVML_cb	Support Vector Machine - Linear	Complete	Balanced
24	SVMP_cb	Support Vector Machine - Polynomial	Complete	Balanced
25	DT_cb	Decision Tree	Complete	Balanced
26	RF_cb	Random Forest	Complete	Balanced
27	XGB_cb	XGBoost	Complete	Balanced
28	LR_rb	Logistic regression	Reduced	Balanced
29	NB_rb	Naïve Bayes	Reduced	Balanced
30	NN_rb	Neural Network	Reduced	Balanced
31	KNN_rb	k-Nearest Neighbor	Reduced	Balanced
32	SVML_rb	Support Vector Machine - Linear	Reduced	Balanced
33	SVMP_rb	Support Vector Machine - Polynomial	Reduced	Balanced
34	DT_rb	Decision Tree	Reduced	Balanced
35	RF_rb	Random Forest	Reduced	Balanced
36	XGB_rb	XGBoost	Reduced	Balanced

3.6.3 EVALUATION

The 36 models were evaluated with four common metrics that are calculated from confusion matrixes: AUC, accuracy, sensitivity and specificity (Witten et al., 2011). A confusion matrix is a table that compares the predicted versus observed classifications (see Figure 4). The cells of the contingency table contain frequencies that are used to calculate various evaluation metrics. Sensitivity, also known as the true positive rate (TPR), is a measure of how well the model classifies the positive class and is equal to $TP / (TP + FN)$. In the current study, specificity measures how well the models predict dropout, and higher values of sensitivity indicate better predictions. The opposite of sensitivity is specificity, also known as the false positive rate (FPR). Specificity is equal to $TN / (TN + FP)$ and provides a measure of how well the models predict the negative class, persist in the current study. A Receiver Operator Curves (ROC) is a plot of sensitivity and specificity. The area underneath the ROC curve, often referred to as AUC (“Area Under the Curve”), provides a holistic measure of model performance because it considers the correctly classified and incorrectly classified instances. Finally, accuracy measures how well instances are correctly classified and is calculated by $(TP + TN) / (TP + TN + FP + FN)$. Accuracy does not consider the misclassified instances and may be misleading when the positive class is small. For example, if a model blindly classifies all instances as persisting and ninety percent of instances persist the model will be ninety percent accurate.

	Actual: Dropout	Actual: Persist
Predicted: Dropout	True Positive (TP)	False Positive (FP)
Predicted: Persist	False Negative (FN)	True Negative (TN)

Figure 4. Example contingency table for binary classification.

AUC was used as the primary evaluation metric for determining the best algorithm because this metric considers correctly and incorrectly classified instances. Accuracy was the secondary evaluation metric because it considers how often correct classifications are made. Sensitivity and specificity were analyzed for additional context since they are the measures used to calculate AUC.

3.6.4 FEATURE IMPORTANCE

Feature importance was evaluated for each of the 36 models. Within the caret package, feature importance was calculated two ways: by using model specific information or not using model specific information (Kuhn, 2017). Model specific feature importance is calculated by extrapolating model metrics. For example, feature importance for Logistic Regression utilizes the absolute value of the t-statistics for each feature, Naïve Bayes utilizes partial least squares for each feature, and tree-based algorithms assess the accuracy of the out-of-bag portion of the data for each feature. Models that do not have model specific metrics, such as neural networks, support vector machine and k-nearest neighbor, ROC curve analysis is conducted for each feature. Specifically, the area under the ROC curve is calculated for each feature and higher values indicate more importance. Whether or not feature importance is calculated based model specific information, the feature importance value for each feature is normalized and the maximum possible value is 100. The normalized feature importance values for all features averaged across all models examine which features were consistently most important for predicting dropout.

3.7 SUMMARY

Degree seeking, first-time National University students from academic year 2015 and 2016 were included in the study. Twelve data sources were utilized to create 79 features, roughly half of which were derived. Nine machine learning algorithms were included in the current study: Logistic Regression (LR), Naïve Bayes (NB), Neural Networks (NN), k-Nearest Neighbor (KNN), Support Vector Machine with a linear kernel (SVML), Support Vector Machine with a polynomial kernel (SVMP), Decision Tree (DT), Random Forest (RF), and XGBoost (XGB). It was hypothesized that SVMP, RF and XGB will outperform all models. R version 3.4.1 (2017-06-30) was used to analyze all data. Four different versions of the population data set were created to examine the effects of feature reduction and class imbalance: complete unbalanced (CU), reduced unbalanced (RU), completed balanced (CB) and reduced balanced (RB). All nine algorithms were trained on each data set, thus producing 36 models. Models were evaluated based on AUC and accuracy. Sensitivity and Specificity were investigated for additional context. Feature importance was estimated for each variable in all 36 models. Feature importance values were normalized so that feature importance could be examined across all models.

CHAPTER 4: RESULTS

4.1 INTRODUCTION

Study results are presented in this chapter. First, the four data sets created by applying recursive feature elimination and under-sampling are described. Then models built from these data sets are evaluated with regard to ROC area and accuracy followed by a similar evaluation of overall algorithm and data set performance. Finally, feature importance is assessed across all models and across only XGBoost and Random Forest models.

4.2 DATA SET RESULTS

The initial data set contained a complete feature set (76 features), all 17,083 instances. Dropout was the minority class, thus making this data set “complete” and “unbalanced” (CU). Under-sampling and recursive feature elimination were utilized to create three additional data sets. All four data sets are summarized in Table 8 and Appendix D lists which features were included in each data set.

Table 8

<i>Data Set Descriptive Statistics</i>			
<u>Data Set</u>	<u>Features</u>	<u>Dropout (%)</u>	<u>Instances</u>
(1) Complete Unbalanced (CU)	76	29%	17,083
(2) Reduced Unbalanced (RU)	53	29%	17,083
(3) Complete Balanced (CB)	76	50%	9,348
(4) Reduced Balanced (RB)	68	50%	9,348

Recursive feature elimination (RFE) was applied to the CU data set to eliminate non-predictive features, which once removed, produced the reduced-unbalanced (RU) data set. Figure 5 illustrates the RFE results. The highest ROC area was obtained when

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

53 features were included. These 53 features were retained in the RU data set, thus removing 23 non-predictive features. The RU data set contained all 17,083 instances.

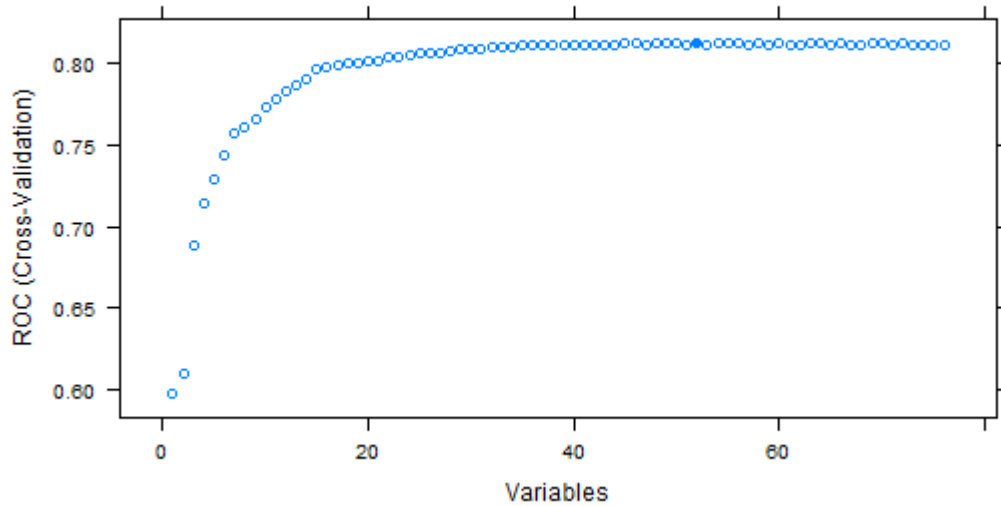


Figure 5. Recursive feature elimination results for Reduced Unbalanced data set.

Under-sampling was applied to the persist class of the CU data set to create the complete-balanced (CB) data set. A total of 7,249 persist instances were removed via under-sampling to balance the proportion of persist and dropout class instances – zero dropout instances were removed. Thus, the CB data set contained all 76 features and 9,348 instances; half of which were dropout and half were persist.

Finally, recursive feature elimination (RFE) was applied to the CB data set to eliminate non-predictive features in the balanced data set. Figure 6 illustrates the RFE results. The highest ROC area was obtained when 68 features were included. These 68 features were retained in the RB data set, thus removing only 8 non-predictive features. The RB data set contained 9,348 instances; half of which were dropout and half were persist.

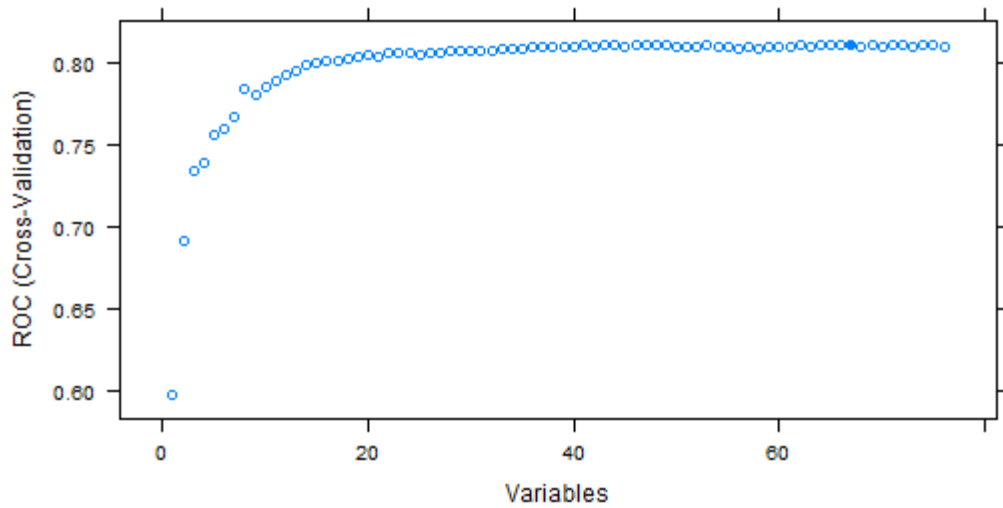


Figure 6. Recursive feature elimination results for Reduced Balanced data set.

4.3 MODELING RESULTS

4.3.1 MODEL PERFORMANCE

All models were evaluated based on ROC curve area (AUC) and Accuracy.

Models were also evaluated on Sensitivity and Specificity for added context. Table 9 provides a complete examination of all 36 models on all four of these measures as well as true-positive (TP), false-positive (FP), true-negative (TN) and false-negative (FN) proportions. The TP, FP, TN and FN proportions are equivalent to each models confusion matrix. The confusion matrix counts were simply converted to percentages and then pivoted to fit on one row.

Table 9

Modeling Results

<u>Model</u>	<u>AUC</u>	<u>Accuracy</u>	<u>Sensitivity</u>	<u>Specificity</u>	<u>TP</u>	<u>FP</u>	<u>TN</u>	<u>FN</u>
DT_cu	0.7090	0.7791	0.4156	0.9260	0.1196	0.0527	0.6594	0.1682
DT_ru	0.7090	0.7791	0.4156	0.9260	0.1196	0.0527	0.6594	0.1682
DT_cb	0.6711	0.6281	0.7356	0.5207	0.3678	0.2397	0.2603	0.1322
DT_rb	0.6711	0.6281	0.7356	0.5207	0.3678	0.2397	0.2603	0.1322
KNN_cu	0.7172	0.7406	0.2922	0.9219	0.0841	0.0556	0.6565	0.2037
KNN_ru	0.7263	0.7490	0.3322	0.9175	0.0956	0.0587	0.6534	0.1922
KNN_cb	0.7017	0.6369	0.6115	0.6624	0.3058	0.1688	0.3312	0.1942
KNN_rb	0.7060	0.6481	0.6346	0.6617	0.3173	0.1692	0.3308	0.1827
LR_cu	0.7893	0.7801	0.4251	0.9235	0.1224	0.0544	0.6577	0.1655
LR_ru	0.7883	0.7773	0.4142	0.9241	0.1192	0.0541	0.6581	0.1686
LR_cb	0.7787	0.7034	0.6671	0.7397	0.3336	0.1302	0.3698	0.1664
LR_rb	0.7792	0.7031	0.6664	0.7397	0.3332	0.1302	0.3698	0.1668
NB_cu	0.7390	0.7117	0.0000	0.9995	0.0000	0.0004	0.7117	0.2879
NB_ru	0.7397	0.7116	0.0000	0.9992	0.0000	0.0006	0.7116	0.2879
NB_cb	0.7341	0.6281	0.3214	0.9349	0.1607	0.0325	0.4675	0.3393
NB_rb	0.7344	0.5963	0.2292	0.9634	0.1146	0.0183	0.4817	0.3854
NN_cu	0.7651	0.7629	0.4617	0.8846	0.1329	0.0822	0.6300	0.1550
NN_ru	0.7719	0.7721	0.4915	0.8854	0.1415	0.0816	0.6306	0.1464
NN_cb	0.7638	0.7024	0.6814	0.7234	0.3407	0.1383	0.3617	0.1593
NN_rb	0.7695	0.7037	0.6881	0.7193	0.3441	0.1403	0.3597	0.1559
RF_cu	0.8033	0.7881	0.4298	0.9329	0.1237	0.0478	0.6643	0.1641
RF_ru	0.8040	0.7890	0.4386	0.9307	0.1263	0.0494	0.6628	0.1616
RF_cb	0.8022	0.7200	0.6990	0.7410	0.3495	0.1295	0.3705	0.1505
RF_rb	0.8024	0.7247	0.7092	0.7403	0.3546	0.1298	0.3702	0.1454
SVML_cu	0.7553	0.7572	0.2386	0.9668	0.0687	0.0236	0.6885	0.2192
SVML_ru	0.7642	0.7572	0.2386	0.9668	0.0687	0.0236	0.6885	0.2192
SVML_cb	0.7768	0.7058	0.6610	0.7505	0.3305	0.1247	0.3753	0.1695
SVML_rb	0.7766	0.7068	0.6603	0.7532	0.3302	0.1234	0.3766	0.1698
SVMP_cu	0.7728	0.7572	0.2271	0.9715	0.0654	0.0203	0.6918	0.2225
SVMP_ru	0.7722	0.7572	0.2271	0.9715	0.0654	0.0203	0.6918	0.2225
SVMP_cb	0.7872	0.7085	0.6712	0.7458	0.3356	0.1271	0.3729	0.1644
SVMP_rb	0.7881	0.7085	0.6664	0.7505	0.3332	0.1247	0.3753	0.1668
XGB_cu	0.8084	0.7923	0.4502	0.9307	0.1296	0.0494	0.6628	0.1583
XGB_ru	0.8110	0.7945	0.4237	0.9444	0.1220	0.0396	0.6725	0.1659
XGB_cb	0.7999	0.7231	0.6732	0.7729	0.3366	0.1136	0.3864	0.1634
XGB_rb	0.7968	0.7153	0.6766	0.7539	0.3383	0.1231	0.3769	0.1617

Figure 7 provides a dot plot of all models ordered by AUC. AUC is a measure of how well each model classifies all possible outcomes (i.e., true-positives, true-negatives, false-positives, and false-negatives). Model XGB_ru had an AUC of 81% and was

superior to all other models. XGBoost and four Random Forest models consistently had the highest AUC.

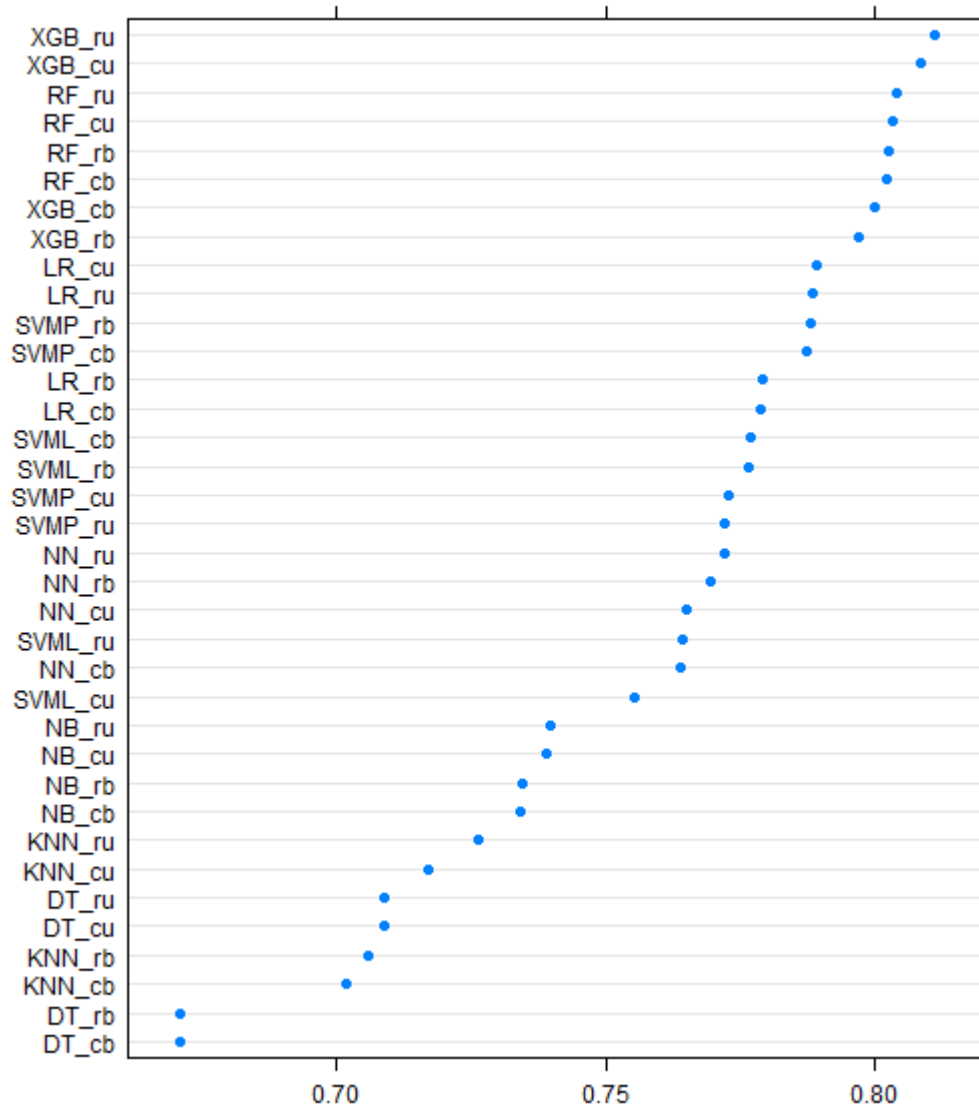


Figure 7. ROC area by model.

Figure 8 provides a dot plot of all models ordered by Accuracy. Accuracy is a measure of how well each model classifies correct outcomes (i.e., true-positives and true-negatives). Model XGB_ru had an accuracy of nearly 80% and was, again, the best

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

performing model. XGBoost and Random Forest models had the top-four highest accuracy rates.

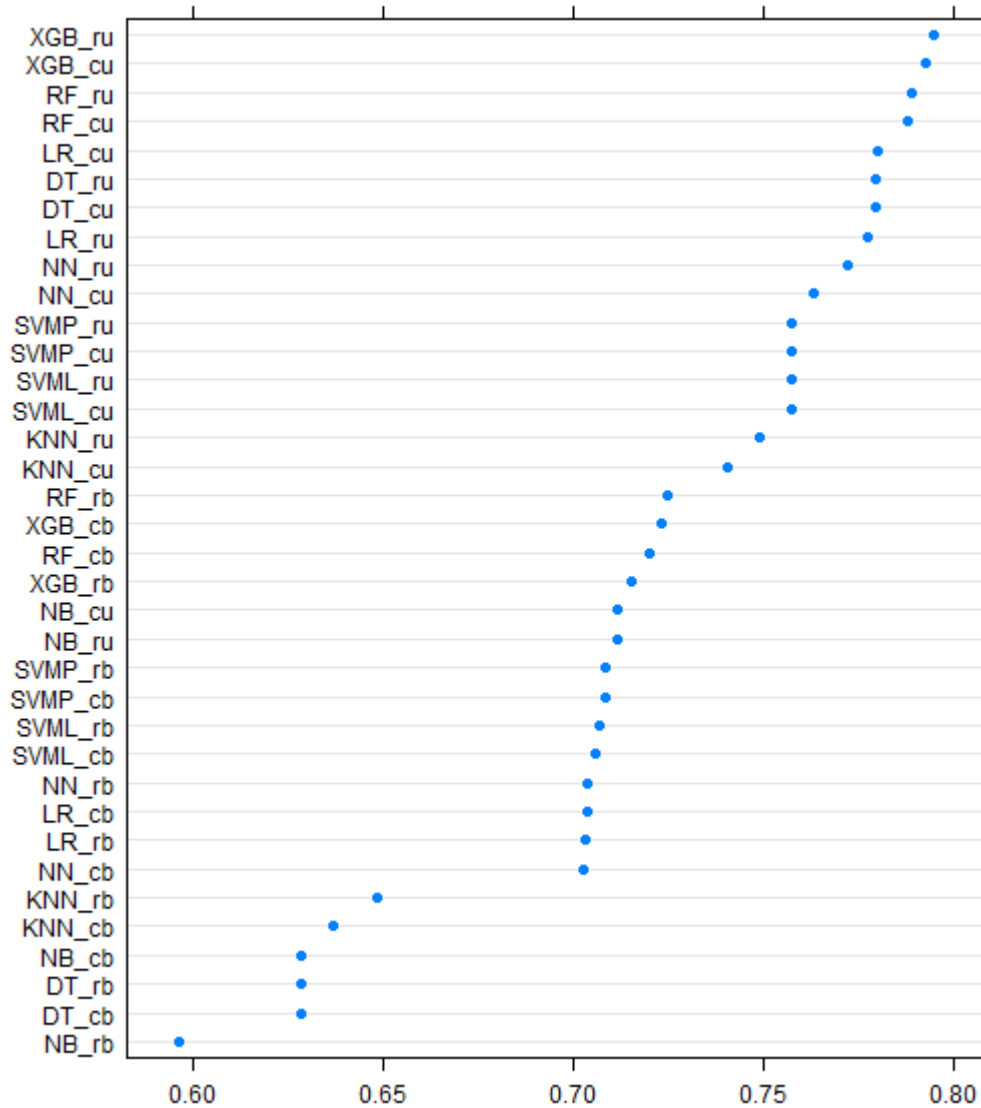


Figure 8. Accuracy by model.

Figure 9 provides a dot plot of all models ordered by Sensitivity. Sensitivity is a measure of how well each model classifies positive outcomes (i.e., true-positives and false-positives). Dropout was the positive outcome for the current study, therefore

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

Sensitivity is a measure of how well each model predicted Dropout. Models DT_rb and DT_cb had the highest Sensitivity (0.7356). However, several other models, predominately the balanced ones, had very similar levels of sensitivity.

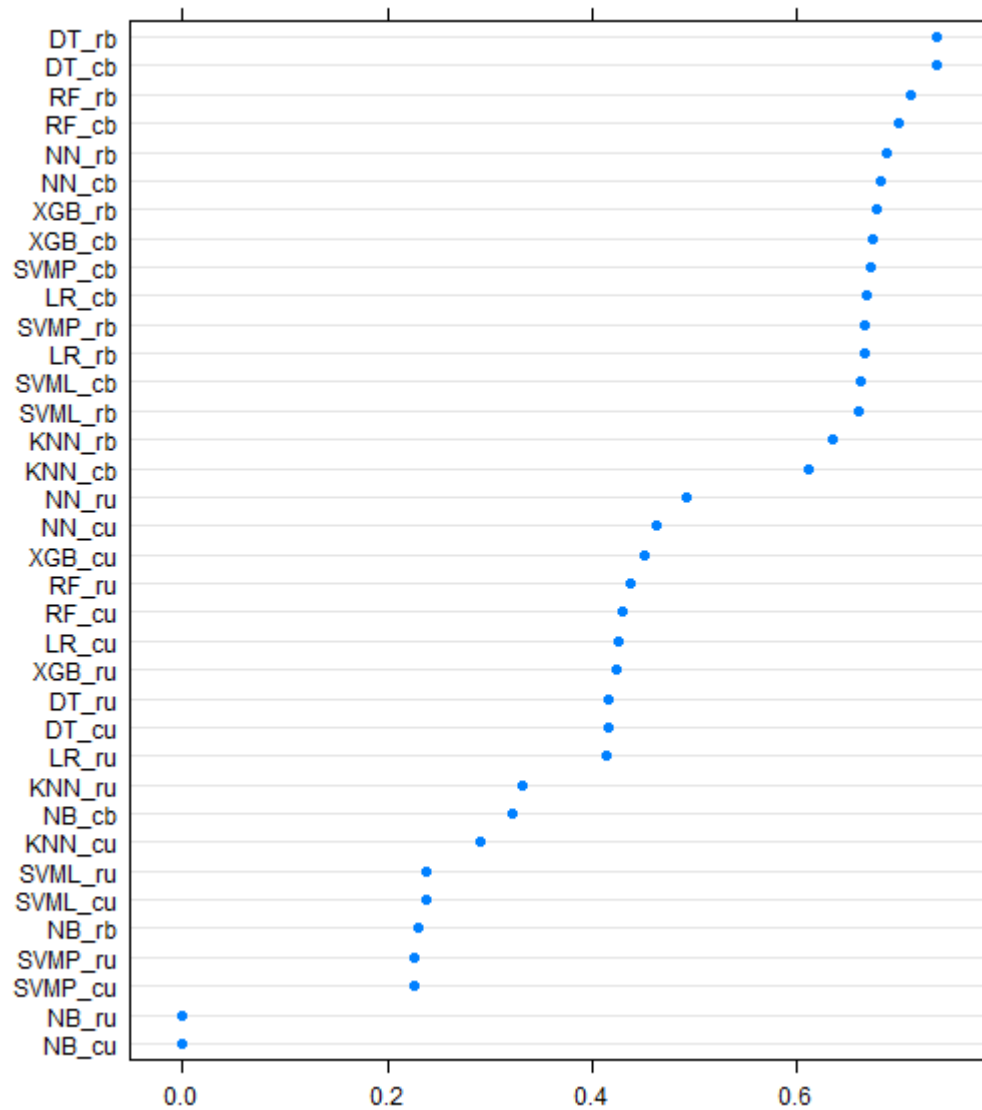


Figure 9. Sensitivity by model.

Figure 10 provides a dot plot of all models ordered by Specificity. Specificity is a measure of how well each model classifies negative outcomes (i.e., true-negatives and

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

false-negatives). Persist was the negative outcome for the current study, therefore Specificity is a measure of how well each model predicted Persist. The unbalanced Naïve Bayes and Support Vector Machine models (i.e., NB_cu, NB_ru, SVMP_ru, SVMP_cu, SVML_ru, and SVML_cu) had Specificity of nearly 100% which indicates that these models essentially predicted every student to persist. These predictions, of course, are not very useful. Unbalanced models consistently had higher levels of Specificity.

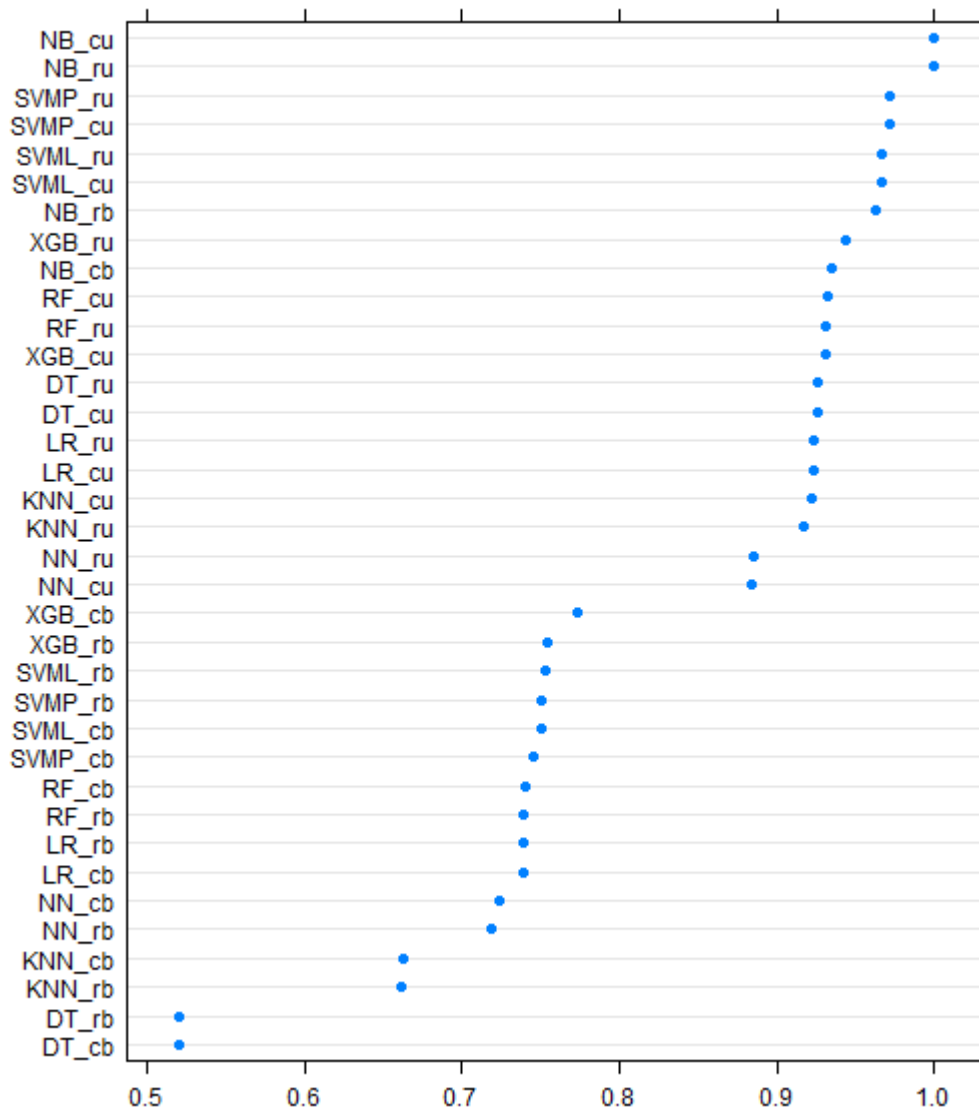


Figure 10. Specificity by model.

4.3.2 ALGORITHM PERFORMANCE

Model results from Table 9 were averaged to examine algorithm performance across the four data sets. Table 10 summarizes these algorithm results and Figure 11 provides additional visualization. The ensemble tree models performed best with regard to AUC. XGBoost and Random Forest averaged 0.8 AUC. Random Forest was very consistent across data sets, as indicated by the small range of AUC values. Ensemble tree models were also very accurate across data sets. XGBoost and Random Forest had an average accuracy of 75% across data sets. Overall, Neural Networks had the highest average Sensitivity. However, tree-based models (i.e., XGBoost, Random Forest, and Decision Tree) had very similar average Sensitivity. Naïve Bayes had an average Sensitivity of 0.14 and an average Specificity of 0.97 which was, respectively, considerably lower and higher than all other algorithms. As discussed previously, this is a result of Naïve Bayes essentially predicting all students to persist.

Table 10

<i>Algorithm Results</i>				
<u>Algorithm</u>	<u>AUC</u>	<u>Accuracy</u>	<u>Sensitivity</u>	<u>Specificity</u>
DT	0.6901	0.7036	0.5756	0.7234
KNN	0.7128	0.6935	0.4675	0.7907
LR	0.7839	0.7410	0.5432	0.8318
NB	0.7368	0.6619	0.1377	0.9743
NN	0.7676	0.7353	0.5807	0.8032
RF	0.8030	0.7554	0.5693	0.8359
SVML	0.7682	0.7318	0.4496	0.8593
SVMP	0.7801	0.7329	0.4480	0.8598
XGB	0.8040	0.7563	0.5559	0.8505

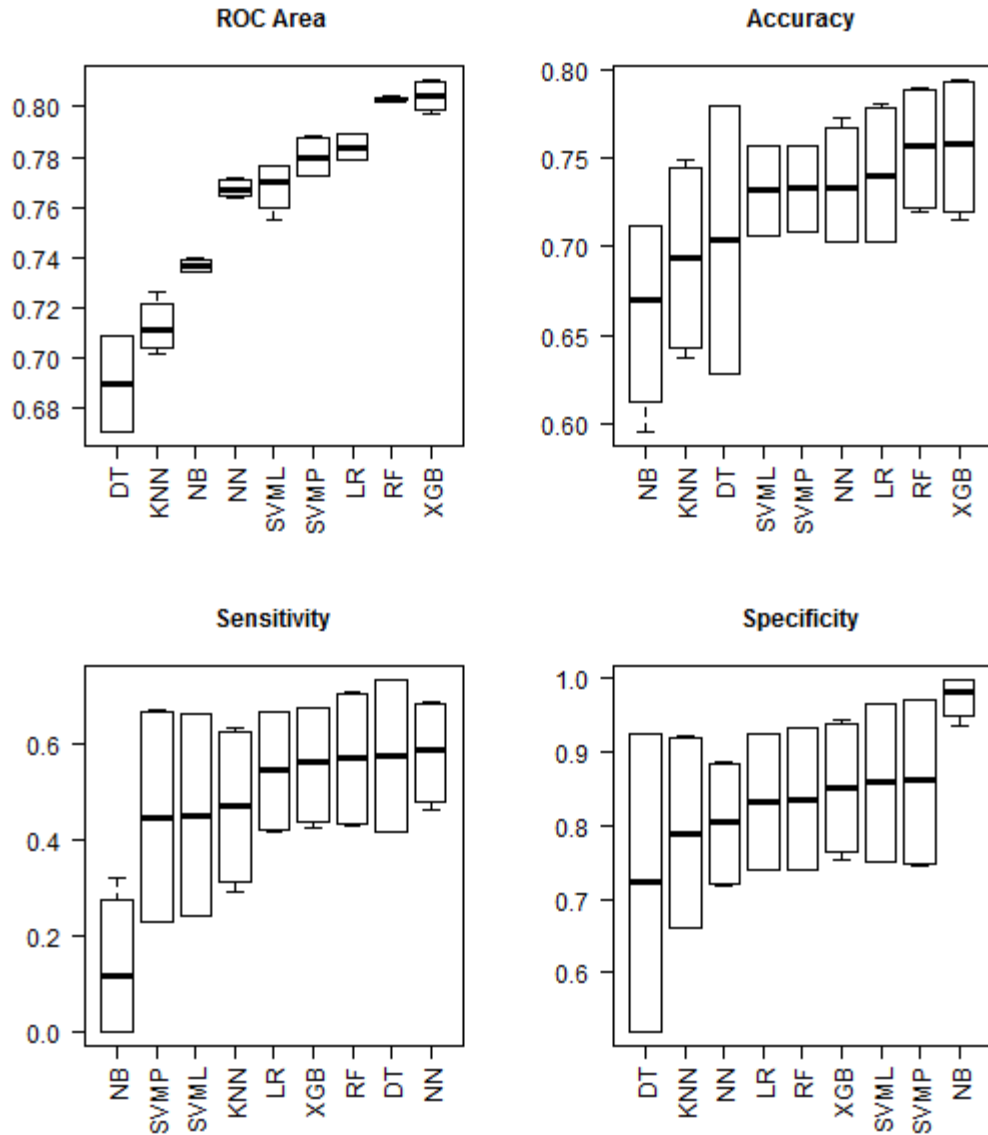


Figure 11. Evaluation metrics by algorithm.

4.3.3 DATA SET PERFORMANCE

Model results from Table 9 were averaged across the nine algorithms to examine the effect of balancing and feature reduction. These data set results are presented in Table 11 and Figure 12 provides additional visualization. All four data sets had equally high average AUC. Their average AUC was within 1% of each other. However, the

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

unbalanced data sets (i.e., cu and ru) were consistently more accurate than the balanced data sets (i.e., rb and cb). The unbalanced data sets were, on average, 8% more accurate than the balanced data sets. Balanced data sets consistently had higher average Sensitivity compared to unbalanced data sets, which is expected since balancing facilitates the prediction of the minority class. Relatedly, unbalanced data sets consistently had higher average Specificity compared to balanced data sets, which is expected since the majority class of unbalanced data are easier to predict.

Table 11

<i>Data Set Results</i>				
<u>Data Set</u>	<u>AUC</u>	<u>Accuracy</u>	<u>Sensitivity</u>	<u>Specificity</u>
(1) Complete Unbalanced (CU)	0.7621	0.7631	0.3266	0.9396
(2) Reduced Unbalanced (RU)	0.7651	0.7652	0.3312	0.9406
(3) Complete Balanced (CB)	0.7572	0.6840	0.6357	0.7322
(4) Reduced Balanced (RB)	0.7582	0.6816	0.6296	0.7335

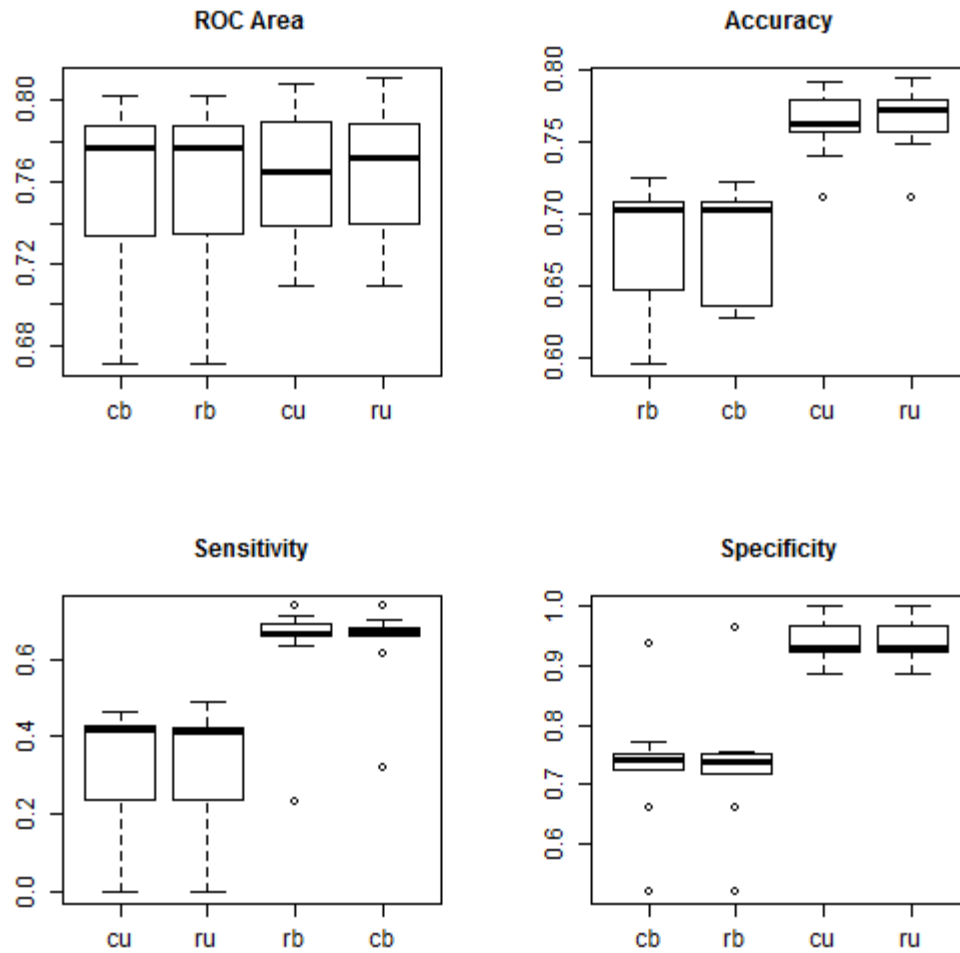


Figure 12. Evaluation metrics by data set.

4.4 FEATURE IMPORTANCE

Feature importance was evaluated for all 36 models and then for Random Forest and XGBoost models. In both scenarios, the relative importance of each feature within each model was scaled to 0 through 100 and these scaled values were averaged for each feature across each model.

4.4.1 ALL MODELS

Feature importance was evaluated for all 36 models. The 25 most important features across all models are listed in Table 12 and visualized in Figure 13.

Table 12

Average Importance of 25 Most Important Features from All Models

<u>Rank</u>	<u>Feature</u>	<u>N</u>	<u>Mean (95% C.I.)</u>	<u>Min</u>	<u>Max</u>
1	DFUWI	36	86.59 (82.27-90.91)	35.09	100.00
2	DegreeAwardType	36	76.31 (68.51-84.1)	26.57	100.00
3	DaysToFYClose	36	73.26 (65.48-81.05)	26.60	100.00
4	RelativePerf	36	67.51 (56.99-78.04)	7.85	100.00
5	P_RFA	36	50.35 (42.06-58.65)	9.85	77.68
6	FCD_FYQ	36	41.09 (29.78-52.4)	0.00	80.31
7	CIP2D	36	40.55 (33.58-47.52)	0.00	100.00
8	Probation	36	40.46 (35.33-45.59)	9.14	81.20
9	TransUnits	36	37.29 (25.63-48.95)	0.00	100.00
10	LearningGrp	36	36.84 (27.37-46.32)	0.00	68.89
11	TeachingGrp	36	34.37 (24.9-43.84)	0.00	66.33
12	ActiveDuty	36	28.63 (24.6-32.66)	10.02	41.41
13	PrevGPA	36	25.89 (17.33-34.45)	0.00	100.00
14	DaysSinceLastAtt	36	25.50 (17.99-33.00)	0.00	85.92
15	Age	36	24.38 (18.82-29.93)	0.00	58.51
16	PrevAtt4Yr	27	21.91 (16.07-27.75)	0.00	41.04
17	PrevDegLvl	36	21.81 (15.07-28.54)	0.00	76.20
18	PrevAtt2Yr	36	21.78 (15.83-27.73)	0.00	43.51
19	P_SOC	36	20.14 (15.29-24.98)	0.00	36.91
20	P_BOM	36	19.87 (13.95-25.78)	0.00	63.54
21	Ethnicity	36	18.61 (14.53-22.68)	0.00	42.57
22	MilitaryYN	36	17.68 (12.5-22.87)	0.00	35.00
23	Gender	36	16.43 (13.08-19.78)	0.00	36.62
24	AGI_PerCapita	36	16.19 (8.39-23.99)	0.00	85.05
25	ClassCPE	36	16.17 (11.28-21.06)	0.00	32.69

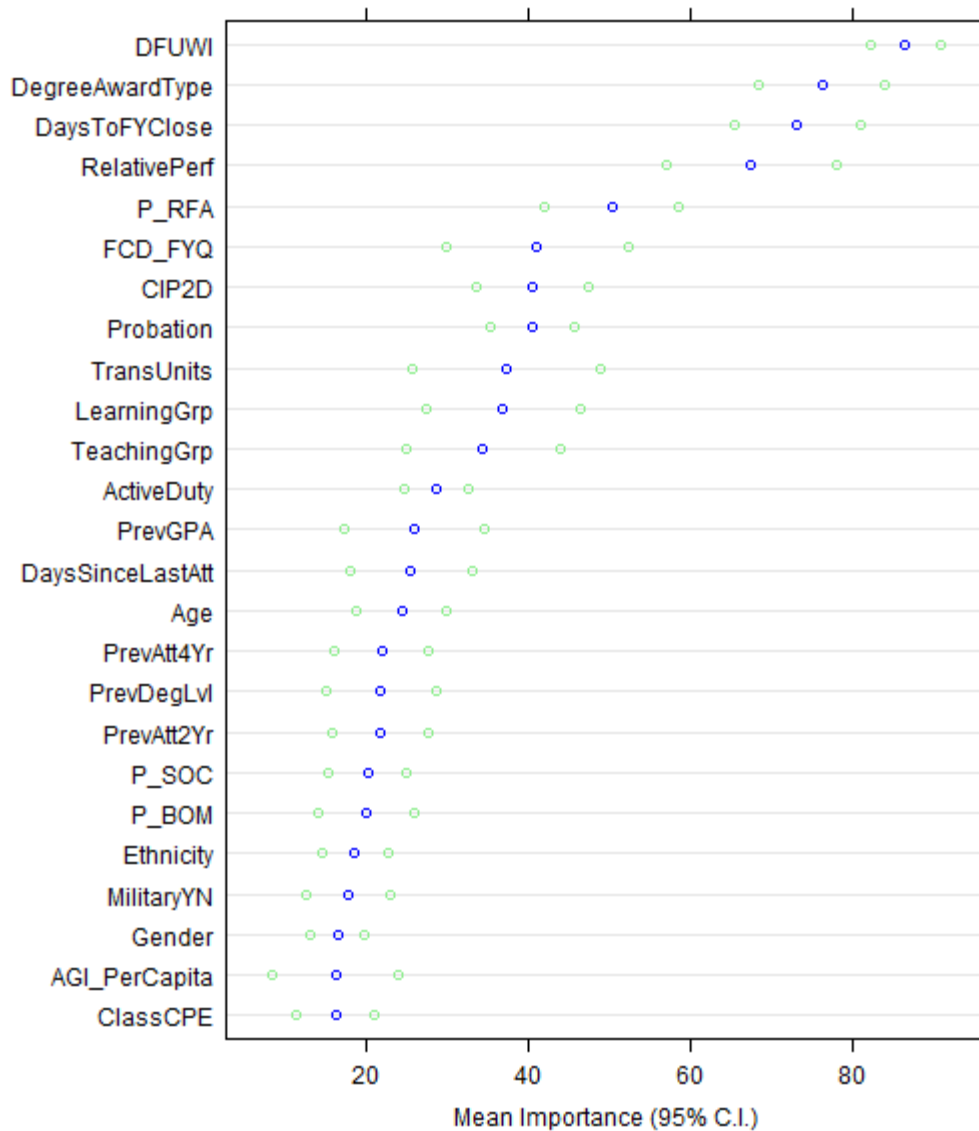


Figure 13. Average importance and 95% confidence interval for 25 most important features across all models.

4.4.2 RANDOM FOREST AND XGBOOST MODELS

As discussed in section 4.3.2, XGBoost and Random Forest models had the highest AUC and Accuracy. To compare the results of these models with all the models, the feature importance of only XGBoost and Random Forest models were examined and

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

the 25 most important features across all XGBoost and Random Forest models are listed in Table 13 and visualized in Figure 14.

Table 13

<i>Average Importance of 25 Most Important Features from XGB and RF Models</i>					
<u>Rank</u>	<u>Feature</u>	<u>N</u>	<u>Mean (95% C.I.)</u>	<u>Min</u>	<u>Max</u>
1	DFUWI	8	93.47 (87.37-99.57)	80.97	100.00
2	DaysToFYClose	8	81.76 (71.24-92.27)	60.81	100.00
3	TransUnits	8	72.01 (48.28-95.74)	17.94	100.00
4	PrevGPA	8	61.39 (37.01-85.76)	16.90	100.00
5	DegreeAwardType	8	53.19 (47.96-58.42)	41.80	63.23
6	DaysSinceLastAtt	8	44.44 (16.82-72.05)	0.46	85.92
7	AGI_PerCapita	8	43.75 (15.89-71.6)	2.40	85.05
8	CIP2D	8	41.72 (27.66-55.78)	20.32	62.80
9	DaysToFC	8	37.36 (13.9-60.81)	1.72	72.30
10	Age	8	30.8 (11.28-50.31)	2.14	58.51
11	RelativePerf	8	30.44 (26.63-34.26)	23.47	38.65
12	SIC_avg	8	30.35 (10.43-50.27)	1.21	60.22
13	Ethnicity	8	22.48 (7.8-37.16)	0.00	42.57
14	AGI	8	21.06 (8.67-33.45)	1.01	40.51
15	P_RFA	8	18.23 (14.62-21.85)	9.85	27.89
16	Probation	8	17.82 (9.52-26.12)	9.14	44.10
17	EFC	8	14.73 (6.63-22.84)	1.85	26.94
18	ActiveDuty	8	13.06 (11.84-14.27)	10.76	16.67
19	LearningGrp	8	12.31 (10.36-14.26)	6.92	15.02
20	FCD_FYQ	8	9.91 (5.37-14.46)	1.29	16.64
21	PrevDegLvl	8	9.77 (8.12-11.41)	7.75	14.90
22	ClassLength_avg	8	9.32 (3.77-14.86)	0.34	17.07
23	MaritalStat	8	8.06 (3.65-12.47)	0.92	14.16
24	Gender	8	7.96 (4.77-11.16)	2.16	13.21
25	OnlineOnly	8	7.61 (4.88-10.34)	3.00	12.51

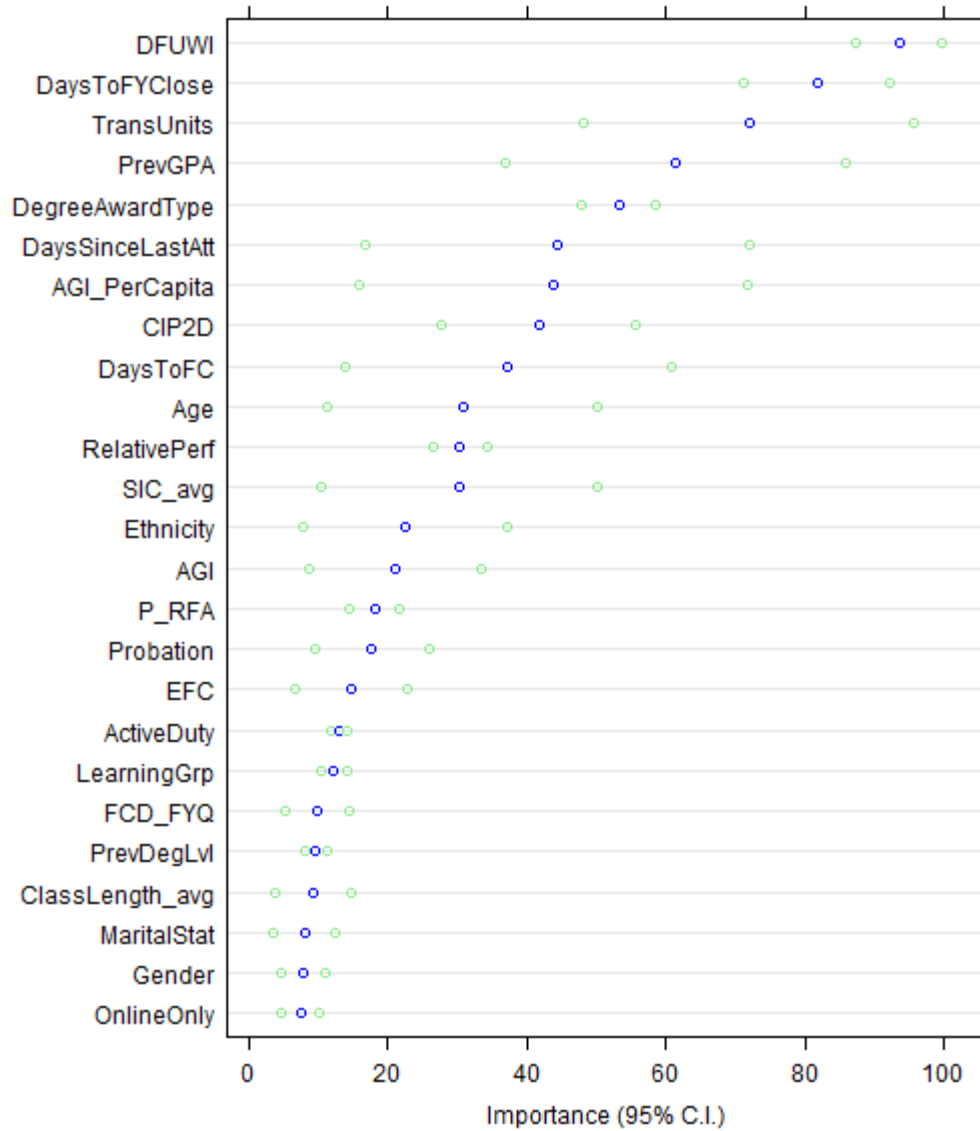


Figure 14. Average importance and 95% confidence interval for 25 most important features across XGBoost and Random Forest models.

4.4.3 FEATURE INTERPRETATIONS

Features that were consistently important across all models and across Random Forest and XGBoost models are examined in this section. Table 14 provides descriptive statistics and Pearson Chi-Square tests of significance for some of the most important categorical features crossed by dropout. What follows is a detailed examination of each of these categorical features.

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

Table 14

Chi-Square Tests of Independence for Categorical Features							
Variable	Population		Dropout		Persist		*p-value
	N (%)		N (%)		N (%)		
	N=17,083		N=4,917		N=12,166		
DFUWI							< .0001
DFUWI	1458	(8.5)	1103	(22.4)	355	(2.9)	
Non-DFUWI	15625	(91.5)	3814	(77.6)	11811	(97.1)	
Degree Award Type							< .0001
Associate Degree	970	(5.7)	496	(10.1)	474	(3.9)	
Bachelor Degree	8248	(48.3)	2918	(59.3)	5330	(43.8)	
Master's Degree	7865	(46)	1503	(30.6)	6362	(52.3)	
Relative Performance							< .0001
Above Median	11594	(67.9)	2479	(50.4)	9115	(74.9)	
Below Median	5489	(32.1)	2438	(49.6)	3051	(25.1)	
P_RFA							< .0001
Admission Approved	10558	(61.8)	2397	(48.7)	8161	(67.1)	
Admission Pending	6525	(38.2)	2520	(51.3)	4005	(32.9)	
CIP2D							< .0001
09 - Communication/Journalism	114	(0.7)	43	(0.9)	71	(0.6)	
11 - Computer and Info							
Sciences	676	(4)	150	(3.1)	526	(4.3)	
13 - Education	4924	(28.8)	939	(19.1)	3985	(32.8)	
14 - Engineering	213	(1.2)	72	(1.5)	141	(1.2)	
15 - Engineering Technologies	137	(0.8)	36	(0.7)	101	(0.8)	
16 - Foreign Languages	37	(0.2)	12	(0.2)	25	(0.2)	
22 - Legal Studies	63	(0.4)	17	(0.3)	46	(0.4)	
23 - English	267	(1.6)	81	(1.6)	186	(1.5)	
24 - Liberal Arts	665	(3.9)	355	(7.2)	310	(2.5)	
26 - Biological Sciences	307	(1.8)	127	(2.6)	180	(1.5)	
27 - Mathematics and Statistics	55	(0.3)	30	(0.6)	25	(0.2)	
30 - Interdisciplinary Studies	92	(0.5)	23	(0.5)	69	(0.6)	
31 - Fitness Studies	146	(0.9)	42	(0.9)	104	(0.9)	
42 - Psychology	1297	(7.6)	299	(6.1)	998	(8.2)	
43 - Homeland Security	794	(4.6)	217	(4.4)	577	(4.7)	
44 - Public Administration	352	(2.1)	51	(1)	301	(2.5)	
45 - Social Sciences	191	(1.1)	68	(1.4)	123	(1)	
50 - Arts	103	(0.6)	28	(0.6)	75	(0.6)	
51 - Health Professions	3933	(23)	1538	(31.3)	2395	(19.7)	
52 - Business	2614	(15.3)	755	(15.4)	1859	(15.3)	
54 - History	103	(0.6)	34	(0.7)	69	(0.6)	
Perception of Learning							< .0001
Negative	530	(3.1)	172	(3.5)	358	(2.9)	
No Response	7324	(42.9)	2671	(54.3)	4653	(38.2)	
Positive	9229	(54)	2074	(42.2)	7155	(58.8)	
Perception of Teaching							< .0001
Negative	464	(2.7)	128	(2.6)	336	(2.8)	
No Response	7357	(43.1)	2683	(54.6)	4674	(38.4)	
Positive	9262	(54.2)	2106	(42.8)	7156	(58.8)	

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

Previous Degree Level							< .0001
Equal or Higher	881	(5.2)	409	(8.3)	472	(3.9)	
Lower or Unknown	16202	(94.8)	4508	(91.7)	11694	(96.1)	

**p*-values based on Pearson Chi-Square test of independence

Overall, DFUWI was the most important feature across all models and XGBoost and Random Forest models. Students that received a grade of D, F, U, W or I during their first term at National University were more likely to dropout than students that did not receive a grade of D, F, U, W or I as shown in Figure 15.

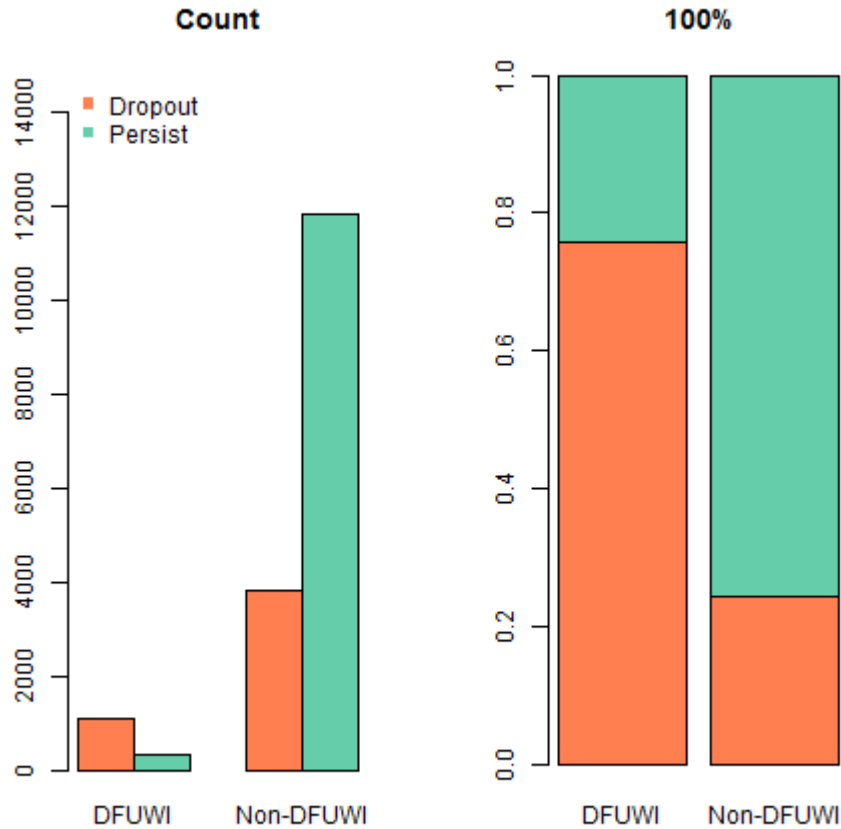


Figure 15. DFUWI by Dropout bar charts.

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

Degree award type (i.e., “DegreeAwardType”) was the second most important feature across all models and the fifth most important across XGBoost and Random Forest models. Students pursuing an Associate degree were more likely to dropout compared to students pursuing a Bachelor or Master degree. Students pursuing a Bachelor degree were more likely to dropout compared to students pursuing a Master degree as shown in Figure 16.

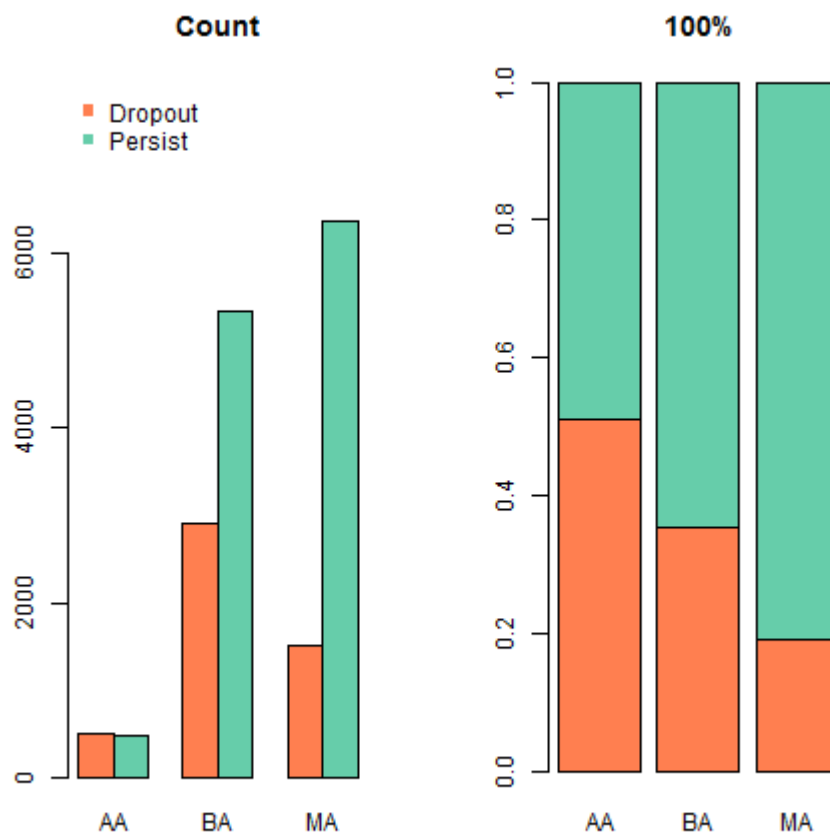


Figure 16. Degree Award Type by Dropout bar charts.

RelativePerf was the fourth most important feature across all models and the eleventh most important across XGBoost and Random Forest models. Students that

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

received a grade during their first class at National University that was below the median grade of that class were more likely to dropout compared to students that received grades above the median grade as shown in Figure 17.

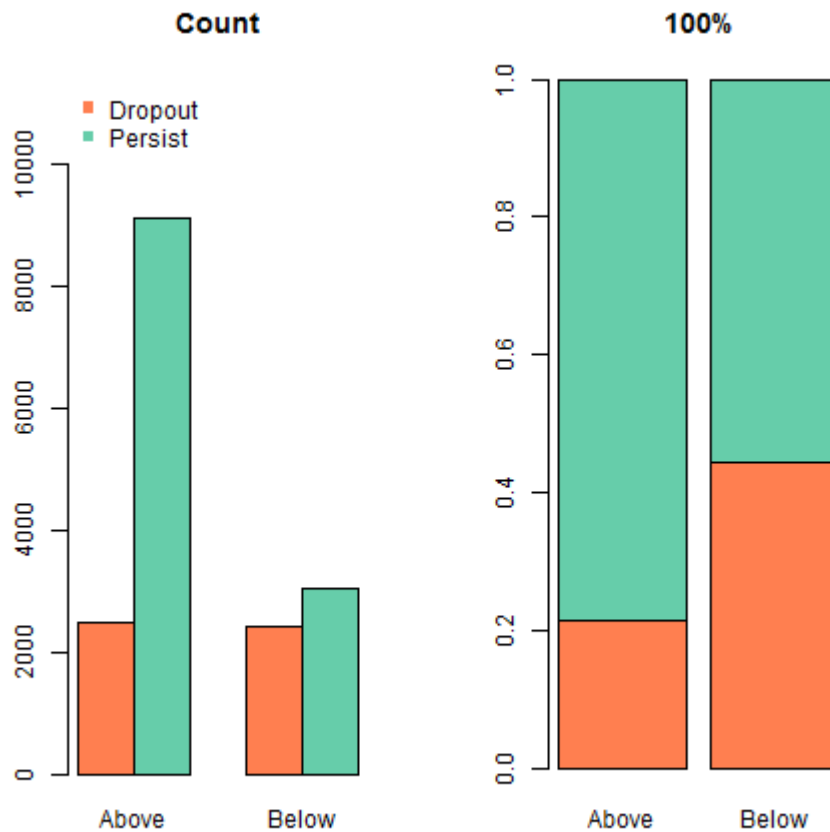


Figure 17. Relative Performance by Dropout bar charts.

P_RFA was the fifth most important feature across all models and the fifteenth most important across XGBoost and Random Forest models. This feature was derived from the service indicator table and indicates that the student's admission evaluation was still pending when the student began their first class at National University. Admission evaluations typically take a 30 to 60 days, so students with this service indicator began

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

their first class very soon after applying to National University. Students with this service indicator are more likely to dropout compared to those without it. That is, students whose admission evaluation is still pending are more likely to drop out than those that have been approved as shown in Figure 18.

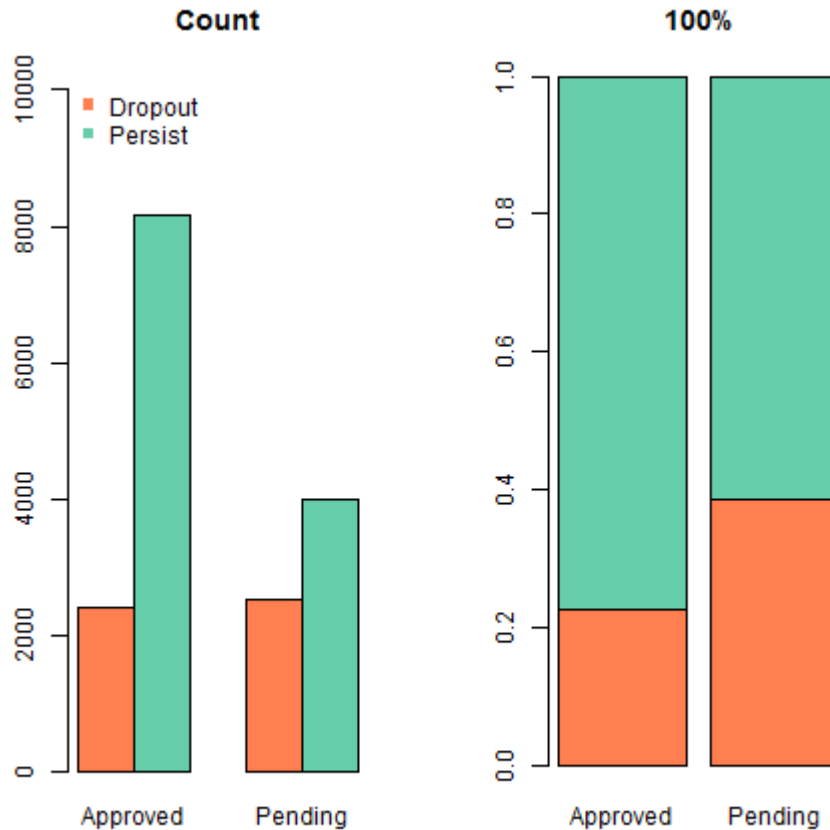


Figure 18. Registrar Admission Flag (P_RFA) by Dropout bar charts.

CIP2D was the seventh most important feature across all models and the eighth most important across XGBoost and Random Forest models. Student dropout rates were highest for Mathematics (55%), Liberal Studies (53%), Biology (41%), and Health Professions (39%) programs. Student dropout rates were lowest for Public

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

Administration (14%), Education (19%), Computer and Information Sciences (22%), and Psychology (23%) as shown in Figure 19.

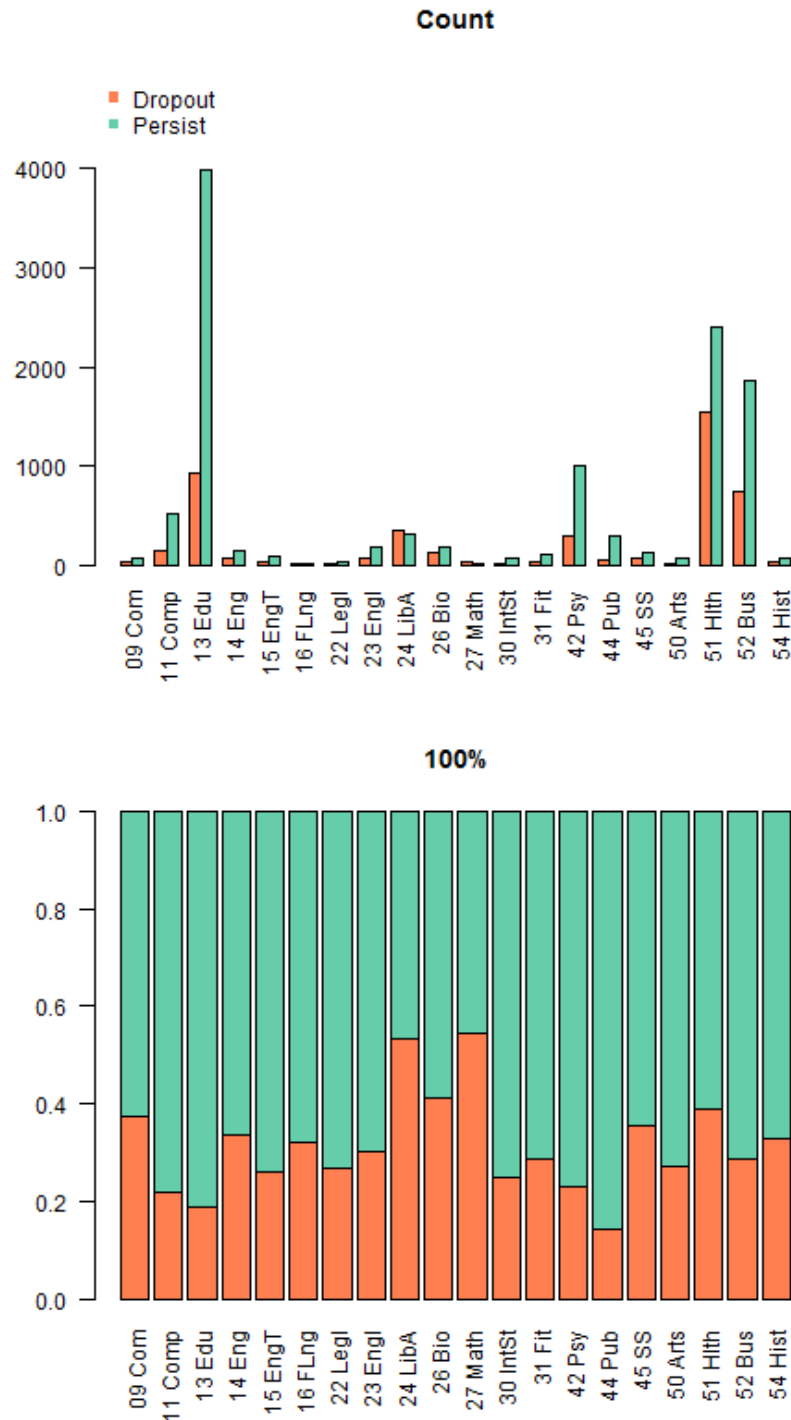


Figure 19. Two-digit CIP Code by Dropout bar charts.

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

LearningGrp and TeachingGrp were ranked tenth and eleventh across all models. This feature was derived from the students' end of course evaluation of their first class at National University. The students' responses were averaged and categorized as being positive, negative, or no response. Students that did not respond the end of course evaluation survey were the most likely to dropout compared to students that gave negative or positive ratings. This effect was consistent across items that assessed the students' perception of learning (see Figure 20) and items that assessed the students' perception of teaching (see Figure 21).

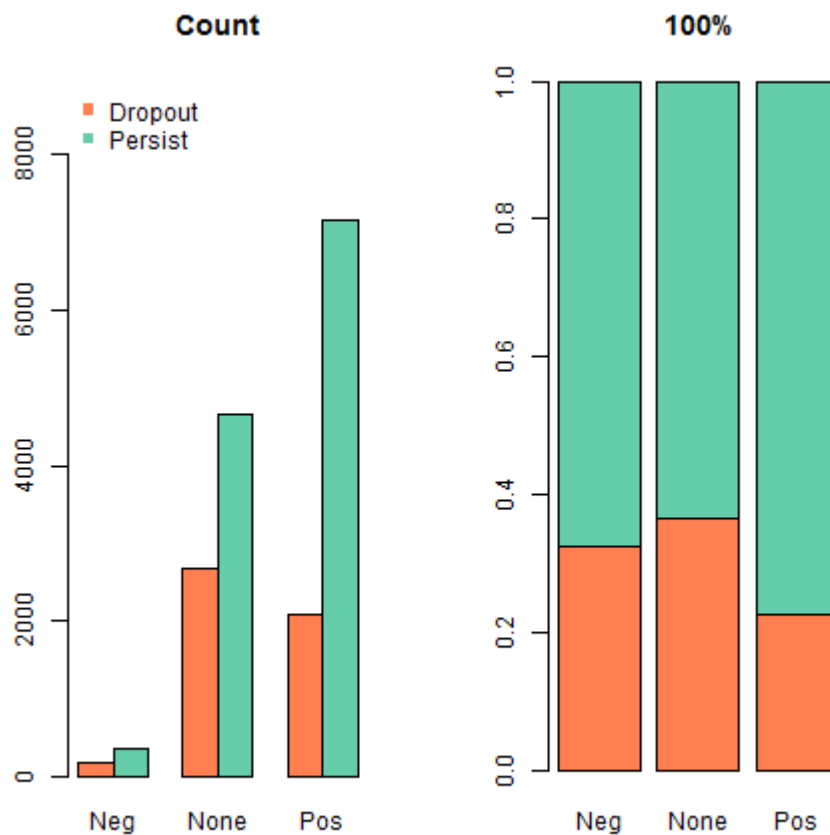


Figure 20. Perceptions of Learning by Dropout bar charts.

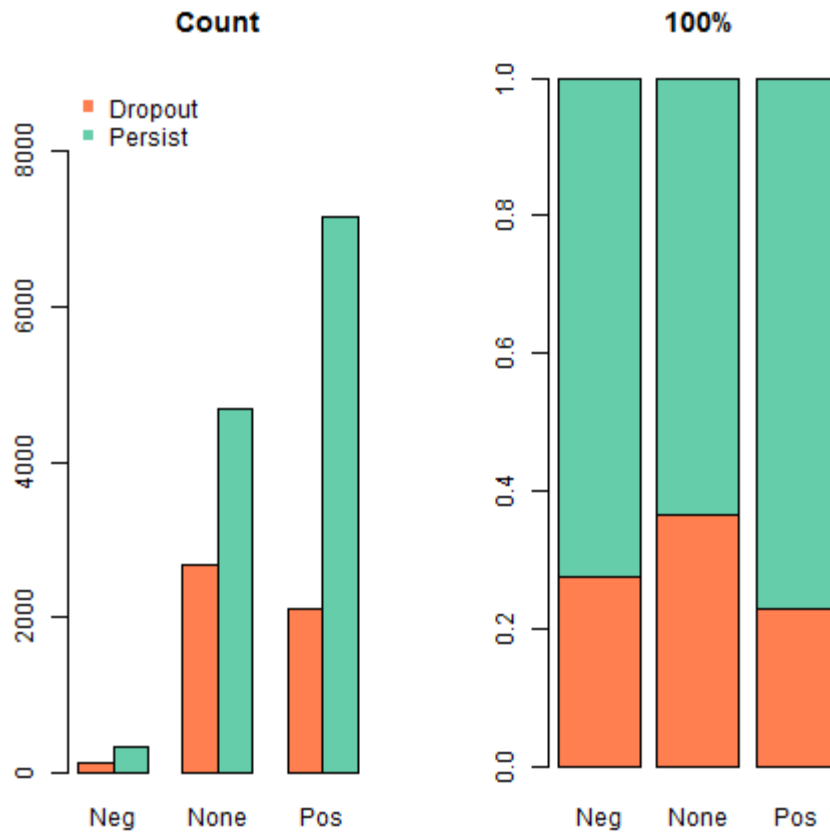


Figure 21. Perceptions of Teaching by Dropout bar charts.

Previous degree level (PrevDegLvl) was the seventeenth most important feature across all models and the twenty-first most important across XGBoost and Random Forest models. Students that had previously earned a degree of the same level as the one they were pursuing at National University were less likely to dropout compared to those that had previously earned a lower degree or their previous degree was unknown as shown in Figure 22.

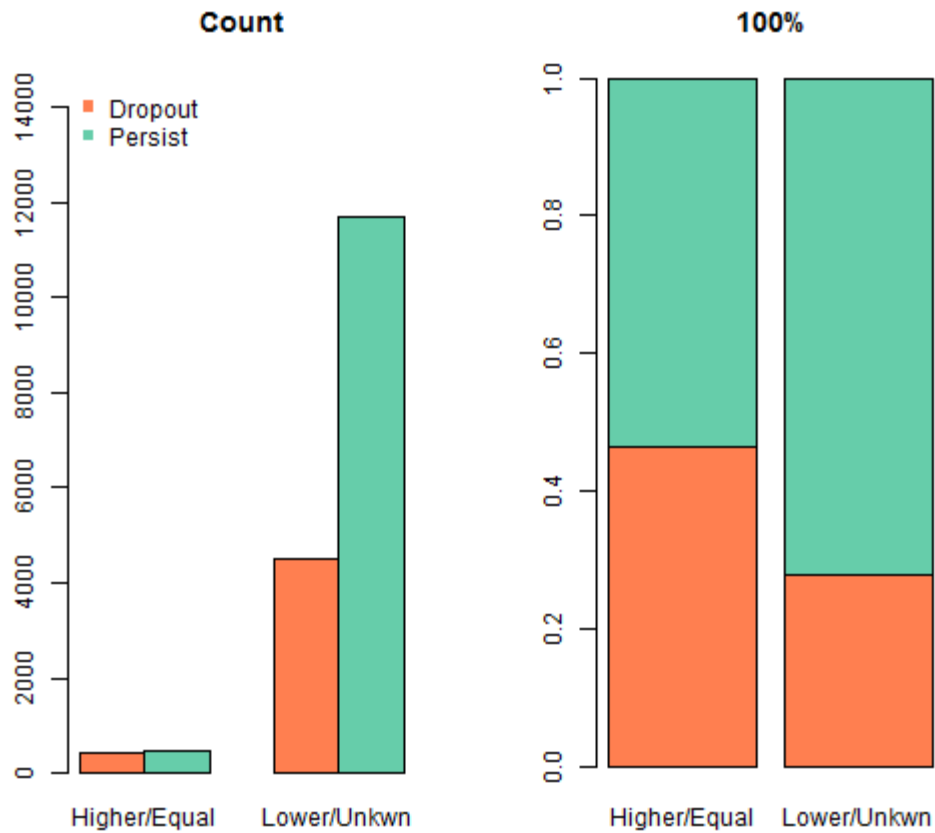


Figure 22. Previous Degree Level by Dropout bar charts.

Table 15 provides descriptive statistics for some of the most important continuous features crossed by dropout. What follows is a detailed examination of each of these continuous features.

Table 15

Feature	Dropout		Persist		95% C.I.	t	df
	M	SD	M	SD			
DaysToFYCclose	234.1	97.5	193.7	104.2	36.9, 43.8	23.35*	17081
TransUnits	42.2	66.3	42.6	61.8	-2.4, 1.7	-0.33	17081
PrevGPA	2.8	0.5	2.9	0.5	-0.5, -0.01	-2.77*	13585
AGI_PerCapita	31184.2	18892.9	31858.9	18248.0	-1285, -64.0	-2.16*	17081

* $p < .05$

DaysToFYClose was the third most important feature across all models and the second most importance across XGBoost and Random Forest models. On average, students that started their first class at National University further away from the end of the fiscal year (i.e., June 30th) were more likely to dropout compared to students that started their first class at national University closer to the end of the fiscal year as shown in Figure 23. The 100% stacked bar charts in Figure 24 illustrate that the percentage of students that dropout decreases linearly with each subsequent quarter.

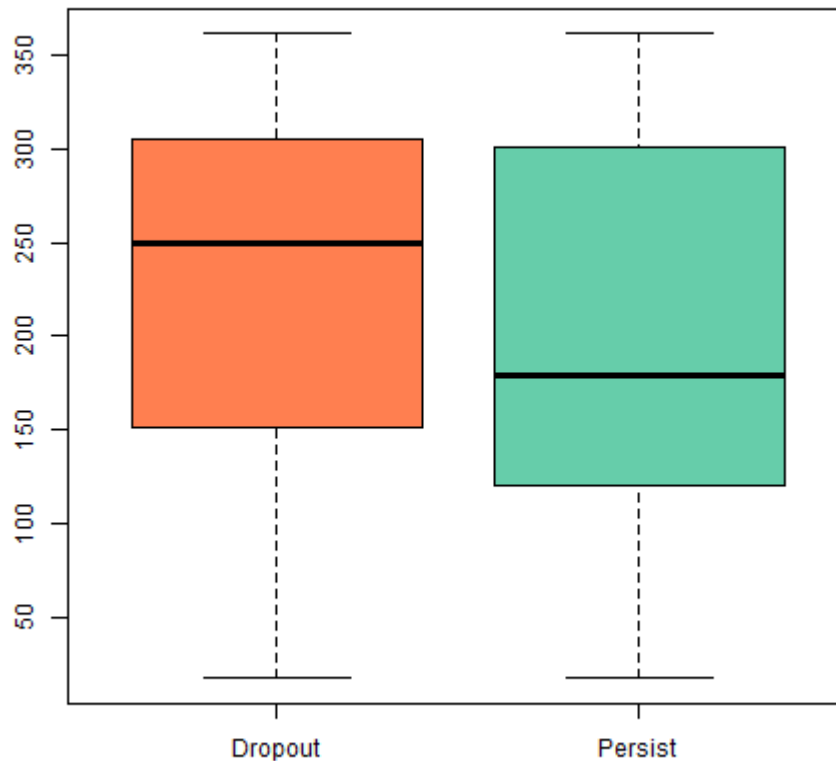


Figure 23. Days to Fiscal Year Close by Dropout boxplot.

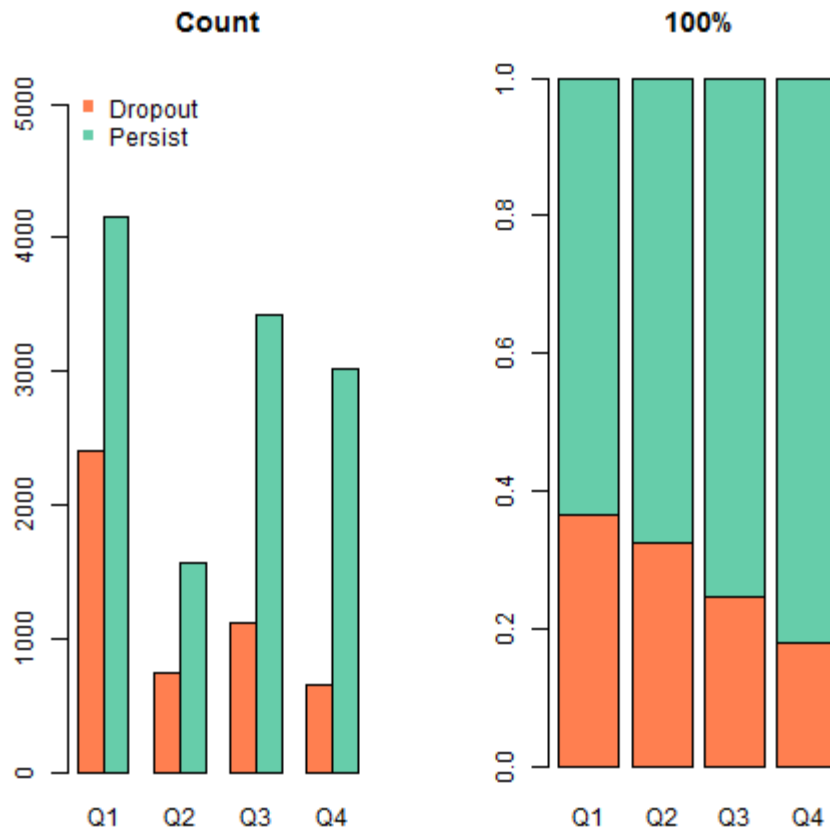


Figure 24. Fiscal quarter start by Dropout bar charts.

TransUnits was the third most important feature across XGBoost and Random Forest models and the ninth most importance across all models. On average, students with fewer transfer units were more likely to dropout. Although, this effect is rather weak, as shown in Figure 25. However, since this effect was not statistically significant and the boxplots illustrate the presence of many outliers, the models likely made their predictions based on extreme values.

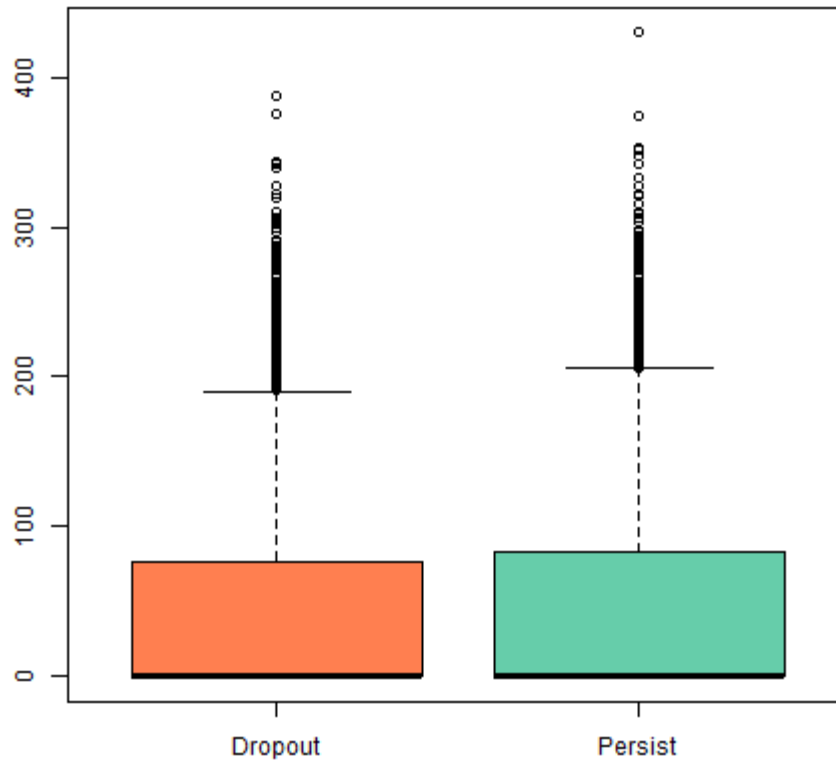


Figure 25. Transfer units by Dropout boxplot.

PrevGPA was the fourth most important feature across XGBoost and Random Forest models and the thirteenth most importance across all models. On average, students with lower grade point averages from previous institutions were more likely to dropout. Although, this effect is rather weak, as shown in Figure 26.

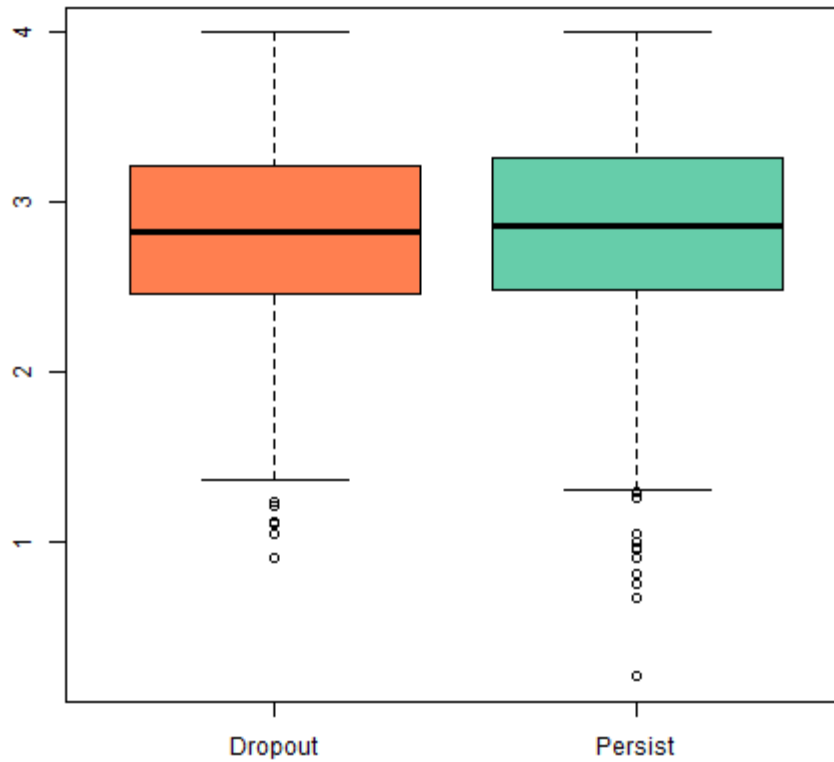


Figure 26. Previous GPA by Dropout boxplot.

AGI_PerCapita was the seventh most important feature across XGBoost and Random Forest models and the twenty-fourth most importance across all models. Per-capita adjusted gross income was derived from the students' home ZIP code and publicly available IRS statistics of income. On average, students with a home ZIP code that had a lower per-capita adjusted gross income were more likely to dropout as shown in Figure 27.

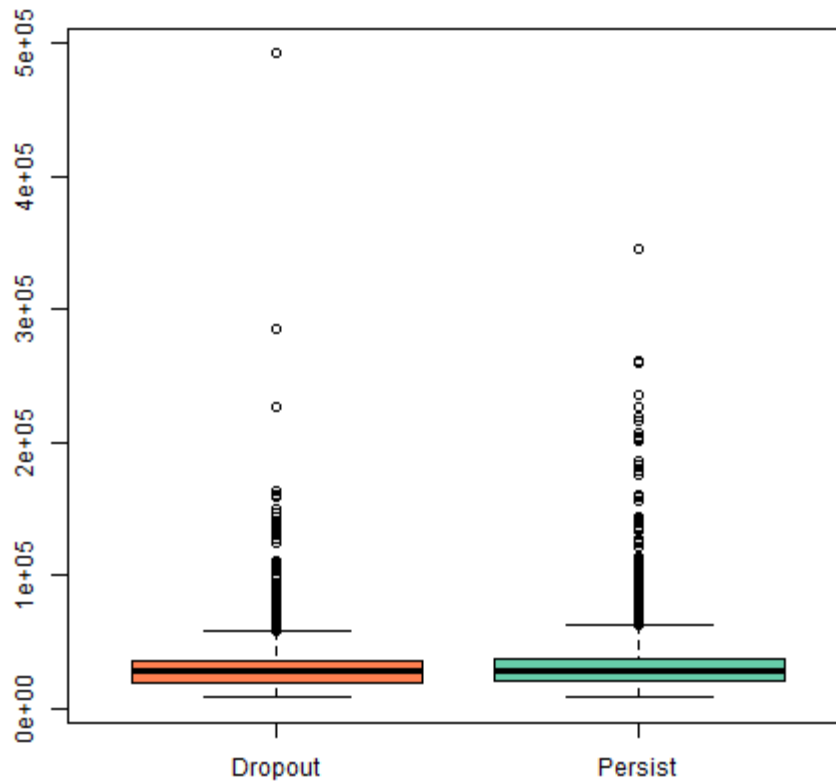


Figure 27. Per-Capita Adjusted Gross Income by Dropout boxplot.

4.5 SUMMARY

Four data sets were created by applying recursive feature elimination and under-sampling to the population data set. Nine machine learning algorithms were modeled on each data set, thus resulting in 36 models. All models were evaluated based on ROC area and accuracy and XGBoost and Random Forest models were consistently superior.

Feature reduction and class balancing did not improve model performance. In fact, class balancing impaired model accuracy. Feature importance was assessed across all models and across only XGBoost and Random Forest models. Features such as DFUWI, degree

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

ward type, and days to fiscal year close were consistently important features for predicting student dropout.

CHAPTER 5: CONCLUSIONS AND FUTURE WORK

Machine learning techniques were used to predict student dropout at National University. Nine machine learning algorithms were modeled on four different data sets: complete features with unbalanced class, reduced features with unbalanced class, complete features with balanced class, and reduced features with balanced class. Thus, a total of 36 models were trained and evaluated based on ROC Area and accuracy.

5.1 ALGORITHMS

It was hypothesized that Support Vector Machine with a polynomial kernel (SVMP), Random Forest (RF), and XGBoost (XGB) would perform best since they are robust to class imbalance, sparsity, outliers, high dimensionality, correlated features, and nonlinearity (Chen & Guestrin, 2016; James et al., 2013; Witten et al., 2011). This hypothesis was partially supported. XGBoost and Random Forest were indeed best at predicting dropout, however, SVMP did not perform as expected. SVMP had the fourth highest ROC area and fifth highest accuracy. The superiority of the two ensemble methods, XGB and RF, at predicting higher education student dropout were consistent with previous research (Delen, 2010). The ensemble methods were very robust across all data sets. They consistently had the highest ROC area and accuracy and therefore proved to be the most reliable techniques for predicting dropout at National University. The current study implemented the default tuning parameters associated with each algorithm. Future research could investigate if exhaustively manipulating the tuning parameters could improve model performance. RF has only one tuning parameter (i.e., the number of randomly selected predictors) whereas XGB has six (i.e., the number of boosting iterations, max tree depth, shrinkage, minimum loss reduction, subsample ratio of

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

columns, minimum sum of instance weight, and subsample percentage). XGB would be excellent candidate for exhaustive model tuning given the multitude of variations that can be trained with the six tuning parameters.

5.2 CLASS BALANCING

It was hypothesized that class balancing would improve model performance. This hypothesis was not supported. Class balancing did not improve model ROC area or accuracy. In fact, class balancing *decreased* model accuracy. This finding does not suggest that class balancing techniques are ineffective. Rather, it suggests that the data in the current study did not have class imbalance issues. Class balancing was advantageous for Delen (2010), Lin (2012) and Thammasiri et al. (2014) in which the dropout class represented 12%, 16% and 21% of instances, respectively. Class balancing was ineffective for the current study in which the dropout class represented 29% of instances. Future research could investigate the threshold at which class balancing becomes an effective technique for improving student dropout prediction.

5.3 FEATURE REDUCTION

It was hypothesized that feature reduction would improve model performance. This hypothesis was not supported. Feature reduction did not affect model ROC area or accuracy. ROC area and accuracy levels were essentially equivalent for models trained on complete or reduced features sets. This finding does not suggest that feature reduction techniques are ineffective. Rather, it suggest that the number of features included were not excessive. Alkhasawneh & Hargraves (2014) is the only other study that has examined the effects of feature reduction on student first-year dropout. Given the sparsity

of related research, future investigations of the effects of feature reduction on dropout prediction could be beneficial.

5.4 FEATURE IMPORTANCE

The relative importance of each feature was assessed across all models and across only the most predictive models (i.e., XGBoost and Random Forest). It was found that features that measured student perceptions of learning and teaching were very predictive of dropout. These features were derived from the student end of course evaluation survey and the aggregated Likert ratings were categorized as either positive, negative, or no response. Interestingly, students that did not respond compared to those with negative or positive ratings were the most likely to dropout. This finding suggests that end of course evaluations can serve as a measure of student engagement. Failure to respond to such surveys, and therefore be less engaged, were related to increased levels of dropout and agree with Tinto's theory of student commitment (Tinto, 1975). Future research could investigate if failure to respond to surveys is predictive of student dropout.

Features that measured the students in an absolute fashion, such as DFUWI, Probation, Previous GPA, Transfer Units and Active Duty military, were very predictive of student dropout and this is consistent with a large body of literature (Cochran et al., 2014; Leitner et al., 2017; Park & Choi, 2009). Interestingly, the current study found that features that measured student relative performance compared to their *immediate* peers (i.e., within a single course) were very predictive of dropout and this contributes to a growing body of literature (Herzog, 2005). The feature Relative Performance was derived for the current study and compared the letter-grade of each student to the median letter-grade of the course. Features like Relative Performance are superior to measures like

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

GPA because not all courses contribute to a student's GPA and because this measure is robust against courses with inflated grades. Future research should derive and investigate how other measures of relative performance impact student dropout prediction.

The feature Previous Degree Level (i.e., "PrevDegreeLvl") assessed if the student was getting a second degree (e.g., a second Bachelor's or a second Master's). This feature was an important predictor of dropout and it was found that students that pursue a second degree are less likely to dropout. As discussed in section 1.5, National University does not collect these data despite them being readily available during student matriculation. Given their predictive power and availability, National University should consider recording these data.

Finally, per-capita adjusted gross income (i.e., "AGI_PerCapita") was derived for the current study from publicly available Statistics of Income (SOI) data by ZIP Code for tax year 2014 (United States Internal Revenue Service, 2014). This feature served as a proxy measure of adjusted gross income for each student based on the per-capita adjusted gross income of the students' home ZIP Code. This proxy measure was used because actual adjusted gross income was available only for students that completed a FAFSA (i.e., half of the study population) and because previous research indicated that income is a significant predictor of dropout (Cochran et al., 2014; Leitner et al., 2017; Park & Choi, 2009). AGI_PerCapita proved to be an important feature in predicting student dropout. In fact, it outperformed actual AGI across all models. Future research could consider including this feature for dropout prediction, especially if actual AGI is unavailable. Additionally, future research should investigate if other socio-demographic features derived from the student's home ZIP code can facilitate dropout prediction.

5.5 SUMMARY

Overall, the current study demonstrated that machine learning techniques can accurately predict student one-year dropout with only one-month (i.e., one term) of data at National University. The models produced have the potential to support early intervention strategies and decrease student dropout. Pairing machine learning techniques with intervention strategies can be an effective way for National University to better support their students and improve the effectiveness of the entire university.

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

APPENDIX A: END OF COURSE EVALUATION SURVEY ITEMS

Response Options:

- 1) Strongly Disagree
- 2) Disagree
- 3) Neutral
- 4) Agree
- 5) Strongly Agree
- 6) N/A

Perception of Learning Items

- 1) My ability to write about this subject has improved.
- 2) The required speaking assignment(s) improved my oral communication skills. If there was no oral requirement, please mark NA.
- 3) I gained significant knowledge about this subject.
- 4) My ability to think critically about topics in this class has improved.
- 5) If research was required, my ability to do research has improved. If not, mark NA.
- 6) Discussions contributed to my learning.
- 7) I can apply what I learned in this course beyond the classroom.
- 8) I can apply what I learned in this course to my job or career goals.

Perception of Teaching Items

- 1) Instructor was well organized.
- 2) The instructor encouraged student interaction.
- 3) Instructor responded promptly to emails and other questions.
- 4) Method of assigning grades was clear.
- 5) The instructor gave clear explanations
- 6) Instructor was receptive to questions.
- 7) The instructor was an active participant in this class.
- 8) Instructor encouraged students to think independently.
- 9) Instructor was available for assistance.
- 10) Instructor provided timely feedback on my work.
- 11) I received useful comments on my work.
- 12) The instructor was an effective teacher.

APPENDIX B: SQL CODE TO CREATE COMPLETE UNBALANCED DATA SET

```

----- POPULATION CLASSES -----
IF OBJECT_ID('tempdb..#pop_classes') IS NOT NULL
    DROP TABLE #pop_classes
select *
into #pop_classes
from (
    select s.*
        ,coalesce([Initial Grade],[Final Grade]) Grade -- prioritizing
initial grade because these are historic data and I want the grade that was given
as soon as the course finished.
        ,case when coalesce([Initial Grade],[Final Grade]) = 'A' then 0
            when coalesce([Initial Grade],[Final Grade]) = 'A-' then 1
            when coalesce([Initial Grade],[Final Grade]) = 'B+' then 2
            when coalesce([Initial Grade],[Final Grade]) = 'B' then 3
            when coalesce([Initial Grade],[Final Grade]) = 'B-' then 4
            when coalesce([Initial Grade],[Final Grade]) = 'C+' then 5
            when coalesce([Initial Grade],[Final Grade]) = 'C' then 6
            when coalesce([Initial Grade],[Final Grade]) = 'C-' then 7
            when coalesce([Initial Grade],[Final Grade]) = 'D+' then 8
            when coalesce([Initial Grade],[Final Grade]) = 'D' then 9
            when coalesce([Initial Grade],[Final Grade]) = 'D-' then 10
            when coalesce([Initial Grade],[Final Grade]) = 'F' then 11
            when coalesce([Initial Grade],[Final Grade]) = 'AU' then 0
            when coalesce([Initial Grade],[Final Grade]) = 'H' then 0
            when coalesce([Initial Grade],[Final Grade]) = 'S' then 3
            when coalesce([Initial Grade],[Final Grade]) = 'U' then 11
            when coalesce([Initial Grade],[Final Grade]) = 'I' then 11
            when coalesce([Initial Grade],[Final Grade]) = 'W' then 11
            when coalesce([Initial Grade],[Final Grade]) = 'IP' then 11 --
none should exist in this data set but will exist in the future. considering a
poor outcome since the student didn't complete the course in time.
            end as GrOrd --this column is needed later
        from IR.dbo.vw_SCAR s
        join (select [Student ID],min([Completion Date]) as CmpDt_Min
            ,min([First Course Date]) as FCD_Min
            from IR.dbo.vw_SCAR
            where Units > 0 --this excludes ORI classes for all and BUS
500A for GRAD students
            and [Completed Course] = 1
            group by [Student ID]) m
        on s.[Student ID] = m.[Student ID]
        and s.[Completion Date] = m.CmpDt_Min --dont join on FCD - want
all the classes if they ended on the same date
        and s.[Completed Course] = 1
        where Units > 0
        and s.[Degree Award Type] in ('Associate Degree','Bachelor
Degree','Master's Degree')
        and s.Term between 1407 and 1606
    ) t

----- POPULATION DISTINCT -----
IF OBJECT_ID('tempdb..#pop') IS NOT NULL
    DROP TABLE #pop

```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```

select ROW_NUMBER ( ) over(order by StuID,FCD_Min,CmpDt_Min) as RowID
,*
into #pop
from (
    select [Student ID] as StuID,Career,[Degree Award Type],[Entity
Group],[Academic Plan]
        ,[Plan Description],[Acad Sub Plan],[Sub Plan Description]
        ,min([Class Enroll Date]) as Enrldt_Min
        ,min([First Course Date]) as FCD_Min
        ,min([Completion Date]) as CmpDt_Min
        ,min(Term) as Term_Min
        ,min([Fiscal Year]) as FY
    from #pop_classes
    group by [Student ID],Career,[Degree Award Type],[Entity Group],[Academic
Plan]
        ,[Plan Description],[Acad Sub Plan],[Sub Plan Description]
) d

```

```

----- OUTCOME MEASURE -----
IF OBJECT_ID('tempdb..#outcome') IS NOT NULL
    DROP TABLE #outcome
select *,case when Retained = 'N' and Graduated = 'N' then 'Y' else 'N' end as
Dropout
into #outcome
from (
    select p.RowID, p.StuID
        ,case when o.[Student ID] is not null then 'Y' else 'N' end as
Retained
        ,case when p.FY = a.GradFY then 'Y'
            when p.FY + 1 = a.GradFY then 'Y'
            else 'N' end as Graduated
    from #pop p
    left join (select distinct [Student ID],Career,[Fiscal Year]
        from IR.dbo.vw_SCAR s
        where Term >= 1507 and [Completed Course] = 1
            and s.[Degree Award Type] in ('Associate
Degree','Bachelor Degree','Master's Degree')) o
        on p.StuID = o.[Student ID]
        and p.Career = o.Career
        and p.FY + 1 = o.[Fiscal Year]
    left join (select distinct EMPLID, ACAD_CAREER, DEGREE, ACAD_PLAN,
ACAD_SUB_PLAN_1,cast([CONFER-COMP_DATE] as date) as GradDT
        ,case when month([CONFER-COMP_DATE]) >=7 then
year([CONFER-COMP_DATE]) + 1
            else year([CONFER-COMP_DATE])
            end as GradFY
        from IR.dbo.Alum12 a
        where ACAD_PLAN_TYPE = 'MAJ') a
        on p.StuID = a.EMPLID
        and p.Career = a.ACAD_CAREER
        and p.FY >= a.GradFY
) m

```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```

----- BIO Demo -----
IF OBJECT_ID('tempdb..#demo') IS NOT NULL
    DROP TABLE #demo
select *
into #demo
from (
    select p.RowID
        ,p.StuID
        ,case when b.EMPLID is not null then 1 else 0 end as BioDataExists
        ,b.[First Name], b.[Middle Name], b.[Last Name], b.[Pref Email],
b.[Pref Phone]
        ,isnull(b.Sex,'U') as Gender
        ,case when b.Ethnicity is null then 'Unknown'
            when b.Ethnicity = 'Elected not to respond' then 'Unknown'
            else b.Ethnicity end as Ethnicity
        ,DATEDIFF(year,b.DOB,p.FCD_min) as Age -- USE AGE GROUP INSTEAD?
        ,case when DATEDIFF(year,b.DOB,p.FCD_min) <= 24 then '<25'
            when DATEDIFF(year,b.DOB,p.FCD_min) <= 29 then '25-29'
            when DATEDIFF(year,b.DOB,p.FCD_min) <= 39 then '30-39'
            when DATEDIFF(year,b.DOB,p.FCD_min) >= 40 then '40+' end as
AgeCat
        ,case when b.[Military Status] in ('Active Duty','Dep of Active
Duty') then 'Y' else 'N' end as ActiveDuty
        ,case when b.[Military Status] in (select MStatus from
IRDev.xw.vw_MilitaryStatus where [Military Y/N] = 'Y') then 'Y' else 'N' end
MilitaryYN
        ,isnull(b.[Military Status],'Not indicated') as MilitaryStatus
        ,coalesce(AGI_PerCap_Zip,AGI_PerCap_ZipPre,AGI_PerCap_ST
            ,(select cast(sum(A00100)*1000 / (sum(mars1) + sum(mars2)*2 +
sum(mars4) + sum(NUMDEP)) as int) AGI_PerCap_US from IRDev.irs.SOI_2014))
AGI_PerCapita
        ,coalesce(AGI_zip,AGI_ZipPre,AGI_State,'4 US') AGI_Source
        ,j.CIP2D
        ,case when pr.EMPLID is not null then 'Y' else 'N' end as Probation
    from #pop p
    left join --get bio demo and reformat Zipcodes
        (select *
            ,case when isnumeric(left([Home Zip],5))=1 then
cast(left([Home Zip],5) as int) else null end as HomeZip
            ,case when isnumeric(left([Home Zip],5))=1 then
left(right('00000' + rtrim(cast(cast(left([Home Zip],5) as int) as char)),5),3)
else null end as HomeZipPref
            from IR.dbo.BioDemo) b
        on p.StuID = b.EMPLID
    left join --AGI for Zipcodes
        (select '1 Zip' as AGI_zip,zipcode,cast(sum(A00100)*1000 /
(sum(mars1) + sum(mars2)*2 + sum(mars4) + sum(NUMDEP)) as int) AGI_PerCap_Zip
        from IRDev.irs.SOI_2014 s where zipcode not in (0,99999) group by
STATEFIPS,[STATE],zipcode) z
        on b.HomeZip = z.zipcode
    left join --AGI for Zip Prefixes (3 digits)
        (select '2 ZipPrefix' as AGI_ZipPre,left(right('00000' +
rtrim(cast(zipcode as char)),5),3) as ZipPrefix
            ,cast(sum(A00100)*1000 / (sum(mars1) + sum(mars2)*2 +
sum(mars4) + sum(NUMDEP)) as int) AGI_PerCap_ZipPre

```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```

        from IRDev.irs.SOI_2014 s where zipcode not in (0,99999) group by
left(right('00000' + rtrim(cast(zipcode as char)),5),3)) zp
        on b.HomeZipPref = zp.ZipPrefix
    left join      --AGI for State
        (select '3 State' as AGI_State,[STATE],cast(sum(A00100)*1000 /
(sum(mars1) + sum(mars2)*2 + sum(mars4) + sum(NUMDEP)) as int) AGI_PerCap_ST
        from IRDev.irs.SOI_2014 s where zipcode not in (0,99999) group by
[STATE]) zs
        on b.[Home St] = zs.[STATE]
    left join (select distinct [Acad Plan],[CIP Code],c.[2D_CIPFamilyCode] + '
- ' + c.[2D_CIPTitle] as CIP2D
        from IR.soar.NU_IE_ACAD_PLAN_SETUP n
        left join IR.dbo.CIPCode2010_Hierarchical c
            on n.[CIP Code] = c.[6D_CIPCode]) j
        on p.[Academic Plan] = j.[Acad Plan]
    left join ir.dbo.PRO pr      --enter on probation
        on p.StuID = pr.EMPLID
        and p.Career = pr.ACAD_CAREER
        and p.Term_Min >= pr.EntryTerm
    ) b

----- AAR For Core Or Elective Class -----
IF OBJECT_ID('tempdb..#aar') IS NOT NULL
    DROP TABLE #aar
select *
into #aar
from (select distinct ACAD_PLAN1, CRSE_ID, 100 as CPE
        from IR..AAR
        where DESCRSHORT1 in ('CORE','PREP','ELEC')
        and CRSE_ID is not null
        and CRSE_ID <> ''
        and DESCR1 not like '%Minor%') a

----- Students in Class (SIC) -----
-- Get total enrollment for each class
IF OBJECT_ID('tempdb..#ClassSIC') IS NOT NULL
    DROP TABLE #ClassSIC
select *
into #ClassSIC
from (
select s.Term,s.[Course ID],s.[Class Nbr],s.[Subject],s.[Catalog],s.[Course Title]
    ,count(*) SIC
    from (
        select distinct Term,[Course ID],[Class Nbr]
        from #pop_classes) c
    left join ir.dbo.vw_SCAR s
        on c.Term = s.Term
        and c.[Course ID] = s.[Course ID]
        and c.[Class Nbr] = s.[Class Nbr]
    group by s.Term,s.[Course ID],s.[Class
Nbr],s.[Subject],s.[Catalog],s.[Course Title]
    ) c

```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```

----- Class Grades for All Students -----
--Then get every student that enrolled in those classes
IF OBJECT_ID('tempdb..#ClassGrades') IS NOT NULL
    DROP TABLE #ClassGrades
select *
into #ClassGrades
from (
select ROW_NUMBER () over (partition by Term,[Course ID],[Class Nbr] order by
GrOrd) rn
,*
from (
    select coalesce([Initial Grade],[Final Grade]) Grade
        ,case when coalesce([Initial Grade],[Final Grade]) = 'A' then 0
            when coalesce([Initial Grade],[Final Grade]) = 'A-' then 1
            when coalesce([Initial Grade],[Final Grade]) = 'B+' then 2
            when coalesce([Initial Grade],[Final Grade]) = 'B' then 3
            when coalesce([Initial Grade],[Final Grade]) = 'B-' then 4
            when coalesce([Initial Grade],[Final Grade]) = 'C+' then 5
            when coalesce([Initial Grade],[Final Grade]) = 'C' then 6
            when coalesce([Initial Grade],[Final Grade]) = 'C-' then 7
            when coalesce([Initial Grade],[Final Grade]) = 'D+' then 8
            when coalesce([Initial Grade],[Final Grade]) = 'D-' then 10
            when coalesce([Initial Grade],[Final Grade]) = 'F' then 11
            when coalesce([Initial Grade],[Final Grade]) = 'AU' then 0
            when coalesce([Initial Grade],[Final Grade]) = 'H' then 0
            when coalesce([Initial Grade],[Final Grade]) = 'S' then 3
            when coalesce([Initial Grade],[Final Grade]) = 'U' then 11
            when coalesce([Initial Grade],[Final Grade]) = 'I' then 11
            when coalesce([Initial Grade],[Final Grade]) = 'W' then 11
            when coalesce([Initial Grade],[Final Grade]) = 'IP' then 11 --
none should exist in this data set but will exist in the future. considering a
poor outcome since the student didn't complete the course in time.
            when coalesce([Initial Grade],[Final Grade]) is null then 0
        -- these are non-graded orientation or activity fee classes
        end as GrOrd
        ,c.SIC
        ,s.*
    from #ClassSIC c
    left join ir.dbo.vw_SCAR s
        on c.Term = s.Term
        and c.[Course ID] = s.[Course ID]
        and c.[Class Nbr] = s.[Class Nbr]
    ) c
) c

```

```

----- Class Median Grade -----
IF OBJECT_ID('tempdb..#ClassMedian') IS NOT NULL
    DROP TABLE #ClassMedian
select *
into #ClassMedian
from (
    select Median,Grade,GrOrd,SIC,Term,[Course ID],[Class
Nbr],[Subject],[Catalog],[Course Title]
    from (

```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```

        select case when SIC = 1 then 'Median'
                   when SIC/2 = rn then 'Median' end as Median
        ,*
    from #ClassGrades
    ) c
    where Median is not null
)c

```

```

----- End Of Course Evaluations -----
IF OBJECT_ID('tempdb..#EndEval') IS NOT NULL
    DROP TABLE #EndEval
select *
into #EndEval
from (
    select *
    from (
        select [Stu ID],[Term],[Class Nbr],[First Class Date],[Last Class
Date],[Perception of]
        ,case when [Level] = 'U' then 'UGRD' when [Level] = 'G' then
'GRAD' when [Level] = 'E' then 'EXED' end Career
        ,AVG(cast(nullif(Response,6) as numeric)) A -- "NA" = 6 so
exclude these
        from IRDev.dbo.vw_EndOfCourseEval_t
        group by [Stu ID],[Term],[Class Nbr],[First Class Date],[Last Class
Date],[Perception of]
        ,case when [Level] = 'U' then 'UGRD' when [Level] = 'G' then
'GRAD' when [Level] = 'E' then 'EXED' end
        having count(Response) > (case when [Perception of] = 'Learning'
then 4 else 6 end) --needed to answer more than half of the questions ("NA"/"6" is
a valid response) (Learning has 8 and Teaching has 12)
    ) p
    PIVOT (avg(A) for [Perception of] in ([Learning],[Teaching])) as PivTable
    ) e

```

```

----- FIRST CLASS -----
-- Aggregated class data for analysis cohort
IF OBJECT_ID('tempdb..#FirstClass') IS NOT NULL
    DROP TABLE #FirstClass
select *
into #FirstClass
from (
    select RowID
    ,StuID
    ,Career
    ,avg(DATEDIFF(day,[First Course Date],[Completion Date])) as
ClassLength_avg
    ,count(*) as Classes
    ,sum(Units) as Units
    ,case when sum(DFUWI) > 0 then 'Y' else 'N' end DFUWI --flagging
any DFUWI

```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```

,case when sum(WW) = count(*) then 'Y' else 'N' end as OnlineOnly
,case when sum(Adjunct) = count(*) then 'Y' else 'N' end as
AdjunctOnly
,case when sum(RemedialCrS) > 0 then 'Y' else 'N' end as RemedialCrS
--flagging any RemedialCrS
,case when avg(Learning) > 3 then 'Positive' when avg(Learning) <= 3
then 'Negative' else 'No Response' end as LearningGrp
,case when avg(Teaching) > 3 then 'Positive' when avg(Teaching) <= 3
then 'Negative' else 'No Response' end as TeachingGrp
,case when sum(UnitWeight*RelativePerf) > 0.5 then 'Above Median'
else 'Below Median' end as RelativePerf
,avg(SIC) as SIC_avg
,case when avg(UnitWeight*CPE) >= 0.5 then 'Y' else 'N' end ClassCPE
--values are either 100% or 0% so collapsing to BIT
from (
select p.RowID, p.StuID
,s.TotalUnits
,c.Units / s.TotalUnits as UnitWeight
,case when coalesce([Initial Grade],[Final Grade]) in
('D+', 'D', 'D-', 'F', 'U', 'W', 'I', 'IP') then 100 else 0 end DFUWI--need to use
Initial Grade since this is what will be used in production. sometimes final grade
is populated without initial being populated
,case when c.[Instruction Mode] = 'WW' then 1 else 0 end as WW
,case when c.[Job Type] in ('Adjunct') then 1 else 0 end as
Adjunct
,case when c.subject + c.catalog in
('ENG13', 'MTH12A', 'MTH12B') then 1 else 0 end as RemedialCrS
,e.Learning
,e.Teaching
,case when c.GrOrd <= m.GrOrd then 100 --'At or Above
Median'
when c.GrOrd > m.GrOrd then 0 --'Below
Median'
end as RelativePerf
,m.SIC
,a.CPE
,c.*
from #pop p
left join #pop_classes c
on p.StuID = c.[Student ID]
and p.Career = c.Career
left join (select [Student ID],Career,sum(Units) as TotalUnits from
#pop_classes group by [Student ID],Career) s --get total units for tie-breakers
on p.StuID = s.[Student ID]
and p.Career = s.Career
left join #EndEval e
on c.[Student ID] = e.[Stu ID]
and c.Term = e.Term
and c.[Class Nbr] = e.[Class Nbr]
and c.Career = e.Career
and c.[First Course Date] = e.[First Class Date]
and c.[Completion Date] = e.[Last Class Date]
left join #ClassMedian m
on c.Term = m.Term
and c.[Course ID] = m.[Course ID]
and c.[Class Nbr] = m.[Class Nbr]
left join #aar a
on c.[Academic Plan] = a.ACAD_PLAN1

```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```

        and c.[Course ID] = a.CRSE_ID
    ) a
group by RowID,StuID,Career
) a

----- SCHEDULE -----
IF OBJECT_ID('tempdb..#Schedule') IS NOT NULL
    DROP TABLE #Schedule
select *
into #Schedule
from (
    select p.RowID
        ,p.StuID
        ,p.Career
        ,p.Term_Min
        ,p.FY
        ,p.Enrldt_Min
        ,p.FCD_Min
        ,p.CmpDt_Min
        ,max(s.[First Course Date]) as MaxSchldFCD
        ,max(s.[Class Enroll Date]) as MaxSchldEnrldt
        ,DATEDIFF(day,p.FCD_min,cast(cast(p.FY as char(4)) + '-07-01' as
date)) DaysToFYClose
        ,case when month(p.FCD_min) between 7 and 9 then 'Q1'
            when month(p.FCD_min) between 10 and 12 then 'Q2'
            when month(p.FCD_min) between 1 and 3 then 'Q3'
            when month(p.FCD_min) between 4 and 6 then 'Q4'
            end as FCD_FYQ
        ,DATEDIFF(day,p.Enrldt_Min,p.FCD_Min) DaysToFC
    from #pop p
    left join IR.dbo.vw_SCAR s
        on p.StuID = s.[Student ID]
        and p.Career = s.Career
    group by p.RowID
        ,p.StuID,p.Career,p.Term_Min,p.Enrldt_Min
        ,p.FCD_Min,p.CmpDt_Min,p.FY
    ) s

----- SERVICE INDICATORS -----
-- Create Working Service Indicator Table
IF OBJECT_ID('tempdb..#ServInd_work') IS NOT NULL
    DROP TABLE #ServInd_work
select *
into #ServInd_work
from (
    select RowID,StuID,Career

        ,SI_Stamp,SI_Code,SI_Desc,SI_Reason,SI_Reason_Desc,SI_Positive,SI_Positive_
Bit
        ,sum(Audit_Act_Value) AAV
    from #pop p
    left join IRDev.dbo.vw_NU_IR_SERVICE_INDICATOR_AUDITS a

```


MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```

        on p.StuID = a.ID
        and a.Audit_Stamp <= p.CmpDt_Min  --SI must be closed by last class
date
    group by RowID,StuID,Career

    ,ID,SI_Stamp,SI_Code,SI_Desc,SI_Reason,SI_Reason_Desc,SI_Positive,SI_Positive_Bit
    having sum(Audit_Act_Value) > 0  --leaves only open service indicators
) ServInd

-- CREATE WIDE TABLE TO JOIN TO ANALYSIS TABLE
IF OBJECT_ID('tempdb..#ServInd') IS NOT NULL
    DROP TABLE #ServInd
select *
into #ServInd
from (
    select *          --distinct SI_Group
    from (
        select RowID,StuID,Career
        ,case when SI_Positive = 'Y' then 'P' else SI_Positive end + '_' +
            case when SI_Code in
('BCB','BCR','BCW','BKA','BLK','BOM','BPP','BRR','BSP','BSS','BV3'
, 'BVA','BVO','CRD','ECD','ECP','ECS','EPS'
, 'FAC','FAO','FAS','FBA','FPW','FRD','HIC','NOS','REC','REV','RFA','ROR')
then SI_Code
            when SI_Code in ('BCH','BCS','BLH','RTR','SA1') then
'Hold'
            when SI_Code in ('BDT','BTP','BTA','EVA','EVN','VSC')
then 'VaTuit'
            when SI_Code in ('RSC','RSD','RSE') then 'SOC'
            when SI_Code in ('INT','IOS') then 'IntlStu'
            when SI_Code in ('ATN','FHB','FNG','ICM','S2S') then
'MiscSchlr'
            when SI_Code in ('BPW','BUW','BWO','BWR') then
'WriteOff'
            when SI_Desc like '%B2B%' then 'B2B'
            when SI_Desc like '%offsite cohort%' then 'OffsiteDisc'
            end as SI_Group
        , 'Y' as S
    ) p
    from #ServInd_work s
) p
PIVOT (min(S) for SI_Group in (N_BCW,      N_BKA, N_BLK, N_BPP, N_CRD, N_ECD,
N_FPW, N_Hold,      N_IntlStu,
N_REC, N_WriteOff,  P_B2B, P_BCB, P_BCR, P_BOM, P_BRR, P_BSP, P_BSS,
P_BV3, P_BVA, P_BVO,
P_ECP, P_ECS, P_EPS, P_FAC, P_FAO, P_FAS, P_FBA, P_FRD, P_HIC,
P_MiscSchlr, P_NOS,
P_OffsiteDisc,      P_REV, P_RFA, P_ROR, P_SOC, P_VaTuit)) as
PivTable
) si

```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
----- FAFSA -----
IF OBJECT_ID('tempdb..#FAFSA') IS NOT NULL
    DROP TABLE #FAFSA
select distinct
    p.RowID,p.StuID,p.Career
    ,case when p.FCD_Min >= F.[Eff Date] then 'BeginFirstClass'
          when p.CmpDt_Min >= F.[Eff Date] then 'EndFirstClass'
          else 'NoFAFSA' end as FAFSAby
    ,case when f.[PAR Fa Grd/Lvl] = 'College' or f.[PAR Mo Grd/Lvl] = 'College'
then 'College'
    when f.[PAR Fa Grd/Lvl] = 'High School' or f.[PAR Mo Grd/Lvl] =
'High School' then 'High School'
    when f.[PAR Fa Grd/Lvl] = 'Middle School' or f.[PAR Mo Grd/Lvl] =
'Middle School' then 'Middle School'
    else 'Unknown' end as ParentHiEd
    ,f.AGI
    ,case when f.[Food Stamps] = 'Yes' or f.[School Lunch] = 'Yes' or f.SSI =
'Yes' or f.TANF = 'Yes' or f.WIC = 'Yes' then 'Y'
          else 'N' end as GovtPrgmY
    ,case when f.Dependents = 'Yes' then 'Y' else 'N' end as DependentsY

    ,case when f.Children = 'Yes' then 'Y' else 'N' end as ChildrenY
    ,f.[In Family]
    ,isnull(f.[Mar Stat],'Unknown') as MaritalStat
    ,coalesce(f.[Prorated EFC],f.[Prmry EFC]) as EFC
into #FAFSA
from #pop p
left join IR.dbo.FAFSA f
    on p.StuID = f.ID
    and p.Career = f.Career
    and p.CmpDt_Min >= F.[Eff Date]
    and p.FY = f.[Aid Yr] -- need because students can apply for future
years so can't rely on effective date
```

```
----- Clearing House Extended Ed -----
IF OBJECT_ID('tempdb..#CHD') IS NOT NULL
    DROP TABLE #CHD
select *
into #CHD
from (
    select RowID,StuID,Career,[Degree Award Type],[Entity Group],FCD_Min
        ,max(Degree) as HiPrevDegree
        ,case when sum([4yr]) > 0 then 'Y' else 'N' end as PrevAtt4Yr
        ,case when sum([2yr]) > 0 then 'Y' else 'N' end as PrevAtt2Yr
        ,max(coalesce(PrevEnrllDate,GradDate)) as LastAttDate
        ,datediff(day,max(coalesce(PrevEnrllDate,GradDate)),FCD_Min) as
DaysSinceLastAtt
    from (
        select
            case when PrevEnrllDate <= CmpDt_Min then 1
                when GradDate <= CmpDt_Min then 1
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```

        end as ValidPriorEnrl
        ,case when GradDate <= CmpDt_Min then DegreeW else null end as
Degree
        ,*
    from (
        select p.*
        ,cast( left(cast(c.[Enrollment Begin]as int),4)
        + substring(cast(cast([Enrollment Begin]as int)
as char(8)),5,2)
        + right(cast(c.[Enrollment Begin]as int),2)
        as date) as PrevEnrllDate
        ,cast( left(cast(c.[Graduation Date]as int),4)
        + substring(cast(cast([Graduation Date]as int)
as char(8)),5,2)
        + right(cast(c.[Graduation Date]as int),2)
        as date) as GradDate
        ,case when [Degree Title] like '%cert%' then null else
(
        case when [Degree Title] like '%Assoc%' or
[Degree Title] like 'AA%' or [Degree Title] like '%A.A.%' or [Degree Title] like
'%A.S%' or [Degree Title] like 'AS %' or [Degree Title] like 'AS-%' or [Degree
Title] like '%ASS%' or [Degree Title] like '%ssociate%' or [Degree Title] in
('AS')
        then '1 AA'
        when [Degree Title] like '%bach%' or
[Degree Title] like 'ba%' or [Degree Title] like 'bs%' or [Degree Title] like 'b
%' or [Degree Title] like 'b.%' or [Degree Title] like '%achelor%' or [Degree
Title] in ('bba','bpa')
        then '2 BA'
        when [Degree Title] like '%mast%' or
[Degree Title] like 'ma%' or [Degree Title] like 'ms%' or [Degree Title] like 'm
%' or [Degree Title] like 'm.%' or [Degree Title] like 'mb%' or [Degree Title]
like 'med %' or [Degree Title] like 'mf%' or [Degree Title] like 'mh%' or [Degree
Title] like 'mp%' or [Degree Title] like '%aster%' or [Degree Title] in
('MED','MIS')
        then '3 MA'
        when [Degree Title] like '%doctor%' or
[Degree Title] like 'ED.%' or [Degree Title] like 'EDD%' or [Degree Title] like
'JD%' or [Degree Title] like '%PSY.D%' or [Degree Title] like '%PHD%' or [Degree
Title] in ('dba','dph','dpt','PSYD')
        then '4 Dr'
        end)
        end as DegreeW
        ,case when c.[2-year / 4-year] = 4 then 1 end as [4yr]
        ,case when c.[2-year / 4-year] = 2 then 1 end as [2yr]
        ,c.*
    from #pop p
    left join IR.dbo.CHD c
        on p.StuID = c.[Requester Return Field]
    where [College Name] <> 'NATIONAL UNIVERSITY'
    ) c
) d
where ValidPriorEnrl = 1
group by RowID,StuID,Career,[Degree Award Type],[Entity
Group],FCD_Min
) chd

```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```

----- NU Ext Ed -----
IF OBJECT_ID('tempdb..#NuExtEd') IS NOT NULL
    DROP TABLE #NuExtEd
select *
into #NuExtEd
from (
    select p.RowID,p.StuID,p.Career,p.[Degree Award Type],FCD_Min
        ,max(e.DegreeLvl) as HiDegr
        ,max(e.DegreeDate) as MaxDegrDt
        ,DATEDIFF(day,max(e.DegreeDate),p.FCD_Min) as DaysSinceDeg
        ,max(e.GPA) as GPA
        ,case when sum(e.[4Yr]) >= 1 then 'Y' else 'N' end as PrevAtt4yr
        ,case when sum(e.[2Yr]) >= 1 then 'Y' else 'N' end as PrevAtt2yr
    from #pop p
    left join (
        select coalesce(c.EMPLID,s.ID) StudentID
            ,case when c.EMPLID is not null and s.ID is not null then
                when c.EMPLID is not null then 'College Only'
                when s.ID is not null then 'Sum Only' end as Sources
            ,coalesce(c.EXT_ORG_ID,s.[Org ID]) as ORG_ID
            ,c.EXT_ORG_ID
            ,s.[Org ID]
            ,c.DEGREE
            ,c.DESCR
            ,case when c.DEGREE = 'GED' then '0 HS'
                when c.DESCR like '%assoc%' then '1 AA'
                when c.DESCR like '%bach%' or c.DESCR like 'b.%' then
                    when c.DESCR like '%mast%' then '3 MA'
                    when c.DESCR like '%Doct%' then '4 Dr'
                    when c.DESCR like '%High%' then '0 HS'
                end as DegreeLvl
            ,case when DEGREE_DT not like '%/190%' then cast(DEGREE_DT as
date) end as DegreeDate
            ,c.DEGREE_STATUS
            ,c.[Eff Date] as EffDateColl
            ,c.[Group Name]
            ,s.Career as careerColl
            ,s.[Sum Type]
            ,s.[Year]
            ,case when coalesce(s.[Ext GPA],s.[Conv GPA]) <= 4 then
nullif(coalesce(s.[Ext GPA],s.[Conv GPA]),0) end as GPA
            ,s.[Eff Date]
            ,case when c.DESCR like '%assoc%' then 1 else e.[2Yr] end as
[2Yr]
            ,case when c.DESCR like '%bach%' or c.DESCR like 'b.%' or
c.DESCR like '%mast%' or c.DESCR like '%Doct%' then 1 else e.[4Yr] end as [4Yr]
        from ir.dbo.ExtCollege c
        full outer join ir.dbo.ExtAcadSum s
            on c.EMPLID = s.ID
            and c.EXT_ORG_ID = s.[Org ID]
        left join (
            -- get 2/4yr data

```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```

select distinct e.*
,d.[2Yr]
,d.[4Yr]
from (
select distinct [Org Type],[Org ID],Descr
from IR.dbo.ExOrg) e
join (
select distinct INSTNM
,case when ICLEVEL_L = '2yr' then 1 else 0 end
,case when ICLEVEL_L = '4yr' then 1 else 0 end
from IR.ipedsDC.InstChars_Dir_Info) d
on e.Descr = d.INSTNM
) e
on coalesce(c.EXT_ORG_ID,s.[Org ID]) = e.[Org ID]
) e
on p.StuID = e.StudentID
and p.CmpDt_Min >= cast(e.DegreeDate as date)
group by p.RowID,p.StuID,p.Career,p.[Degree Award Type],FCD_Min
) e

----- Transfer Units -----
IF OBJECT_ID('tempdb..#trans_working') IS NOT NULL
DROP TABLE #trans_working
select *
into #trans_working
from (
select RowID,StuID,p.Career,[Degree Award Type],Term_Min
,isnull(sum(c.[Units Transferred]),0) as TransUnits
,cast(sum(c.[Units Transferred]*c.[Grd Pt Unt]) as numeric (8,3)) as
TransGrdPts
,cast(case when sum(c.[Units Transferred]) > 0 then
sum(c.[Units Transferred]*c.[Grd Pt Unt]) / sum(c.[Units
Transferred]) end as numeric (4,3)) as TransGPA
,sum(c.[Ext Units]) as ExtUnits
,cast(sum(c.[Ext Units]*g.[Grd Points]) as numeric (8,3)) as
ExtGrdPts
,cast(case when sum(c.[Ext Units]) > 0 then
sum(c.[Ext Units]*g.[Grd Points]) / sum(c.[Ext Units]) end as
numeric (4,3)) as ExtGPA
from #pop p
left join (select ID,Career,[Artic Term],EarnCredit,[Grade In],Grade
,case when EarnCredit = 'Y' then [Units Transferred]
end as [Units Transferred]
,case when EarnCredit = 'Y' then [Grd Pts Per Unt] end
as [Grd Pt Unt]
,[Ext Units]
from IR.soar.vw_NU_IR_TRNS_CRSE_DTL C
where ID in (select StuID from #pop)) C
on p.StuID = c.ID
and p.Career = c.Career

```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```

        and p.Term_Min >= c.[Artic Term]  -- some students get transfer
units approved just after their first term but I don't see how we would catch this
in production so excluding these
        left join (select distinct Grading,[grade in],[grd points]  --get grade
points for non transferred classes
                    ,case when [Grd Scheme] = 'NUG' then 'UGRD'
                        when [Grd Scheme] = 'NGR' then 'GRAD' end
as Car
                    from IRDev.soar.SR733B__GRADE_TABLE
                    where grading = 'TRN' and SetID = 'NATLU' and [Grd
Scheme] in ('NGR','NUG')) g
        on c.[Grade In] = g.[Grade In]
        and p.Career = g.Car
    group by RowID,StuID,p.Career,[Degree Award Type],Term_Min
) t

IF OBJECT_ID('tempdb..#trans') IS NOT NULL
    DROP TABLE #trans
select *
into #trans
from (
    select t.*, e.PrevAtt2yr, e.PrevAtt4yr
    from #trans_working t
    left join (
        select p.RowID,p.StuID,p.[Degree Award Type],p.Term_Min
        ,case when sum(e.[4Yr]) >= 1 then 'Y' else 'N' end as
PrevAtt4yr
        ,case when sum(e.[2Yr]) >= 1 then 'Y' else 'N' end as
PrevAtt2yr
        from #pop p
        left join IR.soar.vw_NU_IR_TRNS_CRSE_DTL t
        on p.StuID = t.ID
        and p.Career = t.Career
        and p.Term_Min >= t.[Artic Term]
        left join (
            select distinct e.*
            ,d.[2Yr]
            ,d.[4Yr]
            from (
                select distinct [Org Type],[Org ID],Descr
                from IR.dbo.ExOrg) e
            join (
                select distinct INSTNM
                ,case when ICLEVEL_L = '2yr' then 1 else 0 end
as [2Yr]
                ,case when ICLEVEL_L = '4yr' then 1 else 0 end
as [4Yr]
                from IR.ipedsDC.InstChars_Dir_Info) d
            on e.Descr = d.INSTNM
        ) e
        on t.[Source ID] = e.[Org ID]
    group by p.RowID,p.StuID,p.[Degree Award Type],p.Term_Min
    ) e
on t.RowID = e.RowID
) a

```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```

----- BRING ALL EXT ED TOGETHER -----
IF OBJECT_ID('tempdb..#ExtEd') IS NOT NULL
    DROP TABLE #ExtEd
select *
into #ExtEd
from (
    select e.*
        ,case when HiDegr >= NUdeg then 'Equal or Higher' -- combining since
groups are small
        else 'Lower' end as PrevDegLv1
    from (
        select RowID,StuID,Career,[Degree Award Type]
            ,case when [Degree Award Type] like 'A%' then '1 AA'
                when [Degree Award Type] like 'B%' then '2 BA'
                when [Degree Award Type] like 'M%' then '3 MA'
            end as NUdeg
            ,max(HiDegr) as HiDegr
            ,max(DaysSinceDeg) as DaysSinceLastAtt
            ,cast(max(GPA) as numeric(6,3)) as PrevGPA
            ,max(TransUnits) as TransUnits
            ,max(TransGPA) as TransGPA
            ,max(PrevAtt4Yr) as PrevAtt4Yr --max is Y since N < Y
            ,max(PrevAtt2Yr) as PrevAtt2Yr
        from (
            select RowID,StuID,Career,[Degree Award
Type],HiDegr,DaysSinceDeg,GPA,TransUnits,TransGPA
            ,case when Career = 'GRAD' then 'Y' else PrevAtt4yr end
as PrevAtt4yr
            ,PrevAtt2yr
        from (
            select RowID,StuID,Career,[Degree Award Type]
                ,HiDegr,DaysSinceDeg
                ,GPA
                ,null as TransUnits,null as TransGPA
                ,e.PrevAtt4yr
                ,e.PrevAtt2yr
            from #NuExtEd e
            union
            select RowID,StuID,Career,[Degree Award Type]
                ,HiPrevDegree,DaysSinceLastAtt
                ,null as GPA
                ,null as TransUnits,null as TransGPA
                ,c.PrevAtt4Yr,c.PrevAtt2Yr
            from #CHD c
            union
            select RowID,StuID,Career,[Degree Award Type]
                ,null as HiPrevDegree,null as DaysSinceLastAtt
                ,coalesce(t.ExtGPA,t.TransGPA) as GPA
                ,t.TransUnits, t.TransGPA
                ,t.PrevAtt4yr,t.PrevAtt2yr
            from #trans t
        ) u
    ) p
group by RowID,StuID,Career,[Degree Award Type]
        ,case when [Degree Award Type] like 'A%' then '1 AA'
            when [Degree Award Type] like 'B%' then '2 BA'

```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```

                                when [Degree Award Type] like 'M%' then '3 MA' end
                                ) e
        ) f

----- FINAL ANALYSIS TABLE -----
IF OBJECT_ID('tempdb..#analysis') IS NOT NULL
    DROP TABLE #analysis
select *
into #analysis
from (
select p.RowID
      ,p.StuID
      ,o.Dropout
      ,p.[Degree Award Type] as DegreeAwardType
      ,d.Gender
      ,d.Ethnicity
      ,d.Age
      ,d.ActiveDuty
      ,d.MilitaryYN
      ,d.AGI_PerCapita
      ,d.Probation
      ,d.CIP2D
      ,e.DaysSinceLastAtt
      ,e.PrevDegLvl
      ,e.PrevAtt4Yr
      ,e.PrevAtt2Yr
      ,e.PrevGPA
      ,e.TransUnits
      ,fa.FAFSAby
      ,fa.ParentHiEd
      ,fa.AGI
      ,fa.GovtPrgmY
      ,fa.DependentsY
      ,fa.ChildrenY
      ,fa.MaritalStat
      ,fa.EFC
      ,fc.ClassLength_avg
      ,fc.Classes
      ,fc.Units
      ,fc.DFUWI
      ,fc.OnlineOnly
      ,fc.AdjunctOnly
      ,fc.RemedialCrs
      ,fc.LearningGrp
      ,fc.TeachingGrp
      ,fc.RelativePerf
      ,fc.SIC_avg
      ,fc.ClassCPE
      ,sc.DaysToFC
      ,sc.DaysToFYClose
      ,sc.FCD_FYQ
      ,isnull(si.N_BCW, 'N') as N_BCW
      ,isnull(si.N_BKA, 'N') as N_BKA
      ,isnull(si.N_BLK, 'N') as N_BLK
      ,isnull(si.N_BPP, 'N') as N_BPP
      ,isnull(si.N_CRD, 'N') as N_CRD
      ,isnull(si.N_ECD, 'N') as N_ECD
```


MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
,isnull(si.N_FPW,'N') as N_FPW
,isnull(si.N_Hold,'N') as N_Hold
,isnull(si.N_IntlStu,'N') as N_IntlStu
,isnull(si.N_REC,'N') as N_REC
,isnull(si.N_WriteOff,'N') as N_WriteOff
,isnull(si.P_B2B,'N') as P_B2B
,isnull(si.P_BCB,'N') as P_BCB
,isnull(si.P_BCR,'N') as P_BCR
,isnull(si.P_BOM,'N') as P_BOM
,isnull(si.P_BRR,'N') as P_BRR
,isnull(si.P_BSP,'N') as P_BSP
,isnull(si.P_BSS,'N') as P_BSS
,isnull(si.P_BV3,'N') as P_BV3
,isnull(si.P_BVA,'N') as P_BVA
,isnull(si.P_BVO,'N') as P_BVO
,isnull(si.P_ECP,'N') as P_ECP
,isnull(si.P_ECS,'N') as P_ECS
,isnull(si.P_EPS,'N') as P_EPS
,isnull(si.P_FAC,'N') as P_FAC
,isnull(si.P_FAO,'N') as P_FAO
,isnull(si.P_FAS,'N') as P_FAS
,isnull(si.P_FBA,'N') as P_FBA
,isnull(si.P_FRD,'N') as P_FRD
,isnull(si.P_HIC,'N') as P_HIC
,isnull(si.P_MiscSchlr,'N') as P_MiscSchlr
,isnull(si.P_NOS,'N') as P_NOS
,isnull(si.P_OffsiteDisc,'N') as P_OffsiteDisc
,isnull(si.P_REV,'N') as P_REV
,isnull(si.P_RFA,'N') as P_RFA
,isnull(si.P_ROR,'N') as P_ROR
,isnull(si.P_SOC,'N') as P_SOC
,isnull(si.P_VaTuit,'N') as P_VaTuit
from #pop p
left join #outcome o on p.RowID = o.RowID
left join #demo d      on p.RowID = d.RowID
left join #ExtEd e      on p.RowID = e.RowID
left join #FAFSA fa     on p.RowID = fa.RowID
left join #FirstClass fc on p.RowID = fc.RowID
left join #Schedule sc   on p.RowID = sc.RowID
left join #ServInd si    on p.RowID = si.RowID
where d.BioDataExists = 1 --excluding these because of JFK data merge
) a
```

APPENDIX C1: R CODE – DATA PREPARATION

```
##### PREPARE DATA #####
raw <- read.csv("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Data.csv"
               ,na.strings = "NULL")
colnames(raw)[1] <- "RowID"

#replace missing values with mean
rawm <- raw
set.seed(951)
for(i in 1:ncol(rawm)){
  rawm[is.na(rawm[,i]), i] <- mean(rawm[,i], na.rm = TRUE)}

library(caret)
library(parallel)
library(doParallel)

##### 1) Complete Unbalanced #####
#Need to exclude all variables with only 1 level
cu <- rawm[,3:79]

set.seed(1)
inTraining <- createDataPartition(cu$Dropout, p = .70, list = FALSE)
cutr <- cu[ inTraining,]
cuts <- cu[ -inTraining,]

save(cu, file = "Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved
Objects/01 Data Prep/cu.rda")
save(cutr, file = "Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved
Objects/01 Data Prep/cutr.rda")
save(cuts, file = "Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved
Objects/01 Data Prep/cuts.rda")

##### 2) Reduced Unbalanced #####
load("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/01
Data Prep/cu.rda")

#setup parallel
cluster <- makeCluster(detectCores() - 1) # convention to leave 1 core for OS
registerDoParallel(cluster)

#random forest recursive feature elimination
rfFuncs$summary <- twoClassSummary #change the summary function to ROC
rfe_ctrl <- rfeControl(functions=rfFuncs, method="cv", number=10,allowParallel = TRUE)
set.seed(951)
rfe.ru <- rfe(cu[,2:77],cu$Dropout, sizes=c(1:76), rfeControl=rfe_ctrl
             ,metric = "ROC",preProc = c("center","scale"))
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
stopCluster(cluster)
registerDoSEQ()

#save/load the object
save(rfe.ru, file = "Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved
Objects/01 Data Prep/rfe.ru.rda")

#SUBSET THE FEATURES TO ONLY INCLUDE PREDICTIVE ONES
ru <- data.frame(cbind("Dropout"=cu[, "Dropout"], cu[,predictors(rfe.ru)]))

set.seed(1)
inTraining <- createDataPartition(ru$Dropout, p = .70, list = FALSE)
ruTr <- ru[ inTraining,]
ruTs <- ru[-inTraining,]

save(ru, file = "Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved
Objects/01 Data Prep/ru.rda")
save(ruTr, file = "Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved
Objects/01 Data Prep/ruTr.rda")
save(ruTs, file = "Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved
Objects/01 Data Prep/ruTs.rda")

##### 3) Complete Balanced #####
cb_n <- cu[cu$Dropout=="N",]
cb_y <- cu[cu$Dropout=="Y",]
set.seed(1)
cb_n <- cb_n[sample(1:nrow(cb_n),(table(cu$Dropout))[2], replace = F),]
cb <- rbind(cb_n,cb_y)
set.seed(1)
cb <- cb[sample(nrow(cb)),] #shuffle rows

set.seed(1)
inTraining <- createDataPartition(cb$Dropout, p = .70, list = FALSE)
cbTr <- cb[ inTraining,]
cbTs <- cb[-inTraining,]

save(cb, file = "Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved
Objects/01 Data Prep/cb.rda")
save(cbTr, file = "Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved
Objects/01 Data Prep/cbTr.rda")
save(cbTs, file = "Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved
Objects/01 Data Prep/cbTs.rda")

rm(inTraining,cb_n,cb_y)

##### 4) Reduced Balanced #####
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
#load("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/01
Data Prep/cb.rda")
#random forest recursive feature elimination
rfFuncs$summary <- twoClassSummary #change the summary function to ROC
rfe_ctrl <- rfeControl(functions=rfFuncs, method="cv", number=10,allowParallel = TRUE)
set.seed(951)
rfe.rb <- rfe(cb[,2:77],cb$Dropout, sizes=c(1:76), rfeControl=rfe_ctrl
            ,metric = "ROC",preProc = c("center","scale"))

stopCluster(cluster)
registerDoSEQ()

#save/load the object
save(rfe.rb, file = "Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved
Objects/01 Data Prep/rfe.rb.rda")
#load("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/01
Data Prep/rfe.rb.rda")

#SUBSET THE FEATURES TO ONLY INCLUDE PREDICTIVE ONES
rb <- data.frame(cbind("Dropout"=cb[, "Dropout"], cb[,predictors(rfe.rb)]))

set.seed(1)
inTraining <- createDataPartition(rb$Dropout, p = .70, list = FALSE)
rbtr <- rb[ inTraining,]
rbts <- rb[-inTraining,]

save(rb, file = "Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved
Objects/01 Data Prep/rb.rda")
save(rbtr, file = "Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved
Objects/01 Data Prep/rbtr.rda")
save(rbts, file = "Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved
Objects/01 Data Prep/rbts.rda")
```

APPENDIX C2: R CODE – MODEL TRAINING

```
library(caret)
library(parallel)
library(doParallel)

load("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/01
Data Prep/cutr.rda")
load("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/01
Data Prep/rutr.rda")
load("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/01
Data Prep/cbtr.rda")
load("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/01
Data Prep/rbtr.rda")

#set working directory for saving models
setwd("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/03
Model Building/")

# parallel setup
cluster <- makeCluster(detectCores() - 1) # convention to leave 1 core for OS
registerDoParallel(cluster)

##### Train Control #####
# use twoClassSummary so we evaluate the model wit ROC instead of Accuracy
tc <- trainControl(method="cv", number=10,savePredictions = "all", classProbs=T,
summaryFunction=twoClassSummary, allowParallel = T)

##### 1) COMPLETE UNBALANCED #####
dt.cu <- train(Dropout~., data=cutr, method="rpart", trControl=tc, preProc = c("center","scale"))
save(dt.cu,file = "dt.cu.rda")

knn.cu <- train(Dropout~., data=cutr, method="knn", trControl=tc, preProc = c("center","scale"))
save(knn.cu,file = "knn.cu.rda")

lr.cu <- train(Dropout~., data=cutr, method="glm", family="binomial", trControl=tc, preProc =
c("center","scale"))
save(lr.cu,file = "lr.cu.rda")

nb.cu <- train(Dropout~., data=cutr, method="nb", trControl=tc, preProc = c("center","scale"))
save(nb.cu,file = "nb.cu.rda")

nn.cu <- train(Dropout~., data=cutr, method="nnet", trControl=tc, preProc = c("center","scale"))
save(nn.cu,file = "nn.cu.rda")

rf.cu <- train(Dropout~., data=cutr, method="rf", trControl=tc, preProc = c("center","scale"))
save(rf.cu,file = "rf.cu.rda")
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
svml.cu <- train(Dropout~., data=cutr, method="svmLinear", trControl=tc, preProc =  
c("center","scale"))  
save(svml.cu,file = "svml.cu.rda")
```

```
svmp.cu <- train(Dropout~., data=cutr, method="svmPoly", trControl=tc, preProc =  
c("center","scale"))  
save(svmp.cu,file = "svmp.cu.rda")
```

```
xgb.cu <- train(Dropout~., data=cutr, method="xgbTree", trControl=tc, preProc =  
c("center","scale"))  
save(xgb.cu,file = "xgb.cu.rda")
```

```
##### 2) REDUCED UNBALANCED #####  
dt.ru <- train(Dropout~., data=rutr, method="rpart", trControl=tc, preProc = c("center","scale"))  
save(dt.ru,file = "dt.ru.rda")
```

```
knn.ru <- train(Dropout~., data=rutr, method="knn", trControl=tc, preProc = c("center","scale"))  
save(knn.ru,file = "knn.ru.rda")
```

```
lr.ru <- train(Dropout~., data=rutr, method="glm", family="binomial", trControl=tc, preProc =  
c("center","scale"))  
save(lr.ru,file = "lr.ru.rda")
```

```
nb.ru <- train(Dropout~., data=rutr, method="nb", trControl=tc, preProc = c("center","scale"))  
save(nb.ru,file = "nb.ru.rda")
```

```
nn.ru <- train(Dropout~., data=rutr, method="nnet", trControl=tc, preProc = c("center","scale"))  
save(nn.ru,file = "nn.ru.rda")
```

```
rf.ru <- train(Dropout~., data=rutr, method="rf", trControl=tc, preProc = c("center","scale"))  
save(rf.ru,file = "rf.ru.rda")
```

```
svml.ru <- train(Dropout~., data=rutr, method="svmLinear", trControl=tc, preProc =  
c("center","scale"))  
save(svml.ru,file = "svml.ru.rda")
```

```
svmp.ru <- train(Dropout~., data=rutr, method="svmPoly", trControl=tc, preProc =  
c("center","scale"))  
save(svmp.ru,file = "svmp.ru.rda")
```

```
xgb.ru <- train(Dropout~., data=rutr, method="xgbTree", trControl=tc, preProc =  
c("center","scale"))  
save(xgb.ru,file = "xgb.ru.rda")
```

```
##### 3) COMPLETE BALANCED #####  
dt.cb <- train(Dropout~., data=cbtr, method="rpart", trControl=tc, preProc = c("center","scale"))  
save(dt.cb,file = "dt.cb.rda")
```

```
knn.cb <- train(Dropout~., data=cbtr, method="knn", trControl=tc, preProc = c("center","scale"))
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
save(knn.cb,file = "knn.cb.rda")
```

```
lr.cb <- train(Dropout~., data=cbtr, method="glm", family="binomial", trControl=tc, preProc =  
c("center","scale"))  
save(lr.cb,file = "lr.cb.rda")
```

```
nb.cb <- train(Dropout~., data=cbtr, method="nb", trControl=tc, preProc = c("center","scale"))  
save(nb.cb,file = "nb.cb.rda")
```

```
nn.cb <- train(Dropout~., data=cbtr, method="nnet", trControl=tc, preProc = c("center","scale"))  
save(nn.cb,file = "nn.cb.rda")
```

```
rf.cb <- train(Dropout~., data=cbtr, method="rf", trControl=tc, preProc = c("center","scale"))  
save(rf.cb,file = "rf.cb.rda")
```

```
svml.cb <- train(Dropout~., data=cbtr, method="svmLinear", trControl=tc, preProc =  
c("center","scale"))  
save(svml.cb,file = "svml.cb.rda")
```

```
svmp.cb <- train(Dropout~., data=cbtr, method="svmPoly", trControl=tc, preProc =  
c("center","scale"))  
save(svmp.cb,file = "svmp.cb.rda")
```

```
xgb.cb <- train(Dropout~., data=cbtr, method="xgbTree", trControl=tc, preProc =  
c("center","scale"))  
save(xgb.cb,file = "xgb.cb.rda")
```

```
##### 4) REDUCED BALANCED #####  
dt.rb <- train(Dropout~., data=rbtr, method="rpart", trControl=tc, preProc = c("center","scale"))  
save(dt.rb,file = "dt.rb.rda")
```

```
knn.rb <- train(Dropout~., data=rbtr, method="knn", trControl=tc, preProc = c("center","scale"))  
save(knn.rb,file = "knn.rb.rda")
```

```
lr.rb <- train(Dropout~., data=rbtr, method="glm", family="binomial", trControl=tc, preProc =  
c("center","scale"))  
save(lr.rb,file = "lr.rb.rda")
```

```
nb.rb <- train(Dropout~., data=rbtr, method="nb", trControl=tc, preProc = c("center","scale"))  
save(nb.rb,file = "nb.rb.rda")
```

```
nn.rb <- train(Dropout~., data=rbtr, method="nnet", trControl=tc, preProc = c("center","scale"))  
save(nn.rb,file = "nn.rb.rda")
```

```
rf.rb <- train(Dropout~., data=rbtr, method="rf", trControl=tc, preProc = c("center","scale"))  
save(rf.rb,file = "rf.rb.rda")
```

```
svml.rb <- train(Dropout~., data=rbtr, method="svmLinear", trControl=tc, preProc =  
c("center","scale"))  
save(svml.rb,file = "svml.rb.rda")
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
svmp.rb <- train(Dropout~., data=rbtr, method="svmPoly", trControl=tc, preProc =  
c("center","scale"))  
save(svmp.rb,file = "svmp.rb.rda")  
  
xgb.rb <- train(Dropout~., data=rbtr, method="xgbTree", trControl=tc, preProc =  
c("center","scale"))  
save(xgb.rb,file = "xgb.rb.rda")  
  
stopCluster(cluster)  
registerDoSEQ()
```


APPENDIX C3: R CODE – MODEL TESTING

```
##### Import All Models #####
#setwd("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/03
Model Building/")
setwd("C:/Users/MKirkpatrick/Desktop/R Files/03 Model Building/")
files = list.files(pattern="*.rda")
for (i in 1:length(files)) load(file=files[i])

##### Import All Test Data #####
setwd("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/01
Data Prep/")
setwd("C:/Users/MKirkpatrick/Desktop/R Files/01 Data Prep/")
files = list.files(pattern="*ts.rda")
for (i in 1:length(files)) load(file=files[i])

rm(files,i)

library(caret)
library(pROC)

##### 1) COMPLETE UNBALANCED #####
pred.dt.cu <- cbind(P=predict(dt.cu, cuts),predict(dt.cu, cuts, type="prob"))
cm.dt.cu <- confusionMatrix(pred.dt.cu$P,cuts$Dropout,positive = "Y")
roc.dt.cu <- roc(cuts$Dropout, pred.dt.cu$Y)

pred.knn.cu <- cbind(P=predict(knn.cu, cuts),predict(knn.cu, cuts, type="prob"))
cm.knn.cu <- confusionMatrix(pred.knn.cu$P,cuts$Dropout,positive = "Y")
roc.knn.cu <- roc(cuts$Dropout, pred.knn.cu$Y)

pred.lr.cu <- cbind(P=predict(lr.cu, cuts),predict(lr.cu, cuts, type="prob"))
cm.lr.cu <- confusionMatrix(pred.lr.cu$P,cuts$Dropout,positive = "Y")
roc.lr.cu <- roc(cuts$Dropout, pred.lr.cu$Y)

pred.nb.cu <- cbind(P=predict(nb.cu, cuts),predict(nb.cu, cuts, type="prob"))
cm.nb.cu <- confusionMatrix(pred.nb.cu$P,cuts$Dropout,positive = "Y")
roc.nb.cu <- roc(cuts$Dropout, pred.nb.cu$Y)

pred.nn.cu <- cbind(P=predict(nn.cu, cuts),predict(nn.cu, cuts, type="prob"))
cm.nn.cu <- confusionMatrix(pred.nn.cu$P,cuts$Dropout,positive = "Y")
roc.nn.cu <- roc(cuts$Dropout, pred.nn.cu$Y)

pred.rf.cu <- cbind(P=predict(rf.cu, cuts),predict(rf.cu, cuts, type="prob"))
cm.rf.cu <- confusionMatrix(pred.rf.cu$P,cuts$Dropout,positive = "Y")
roc.rf.cu <- roc(cuts$Dropout, pred.rf.cu$Y)

pred.svm.cu <- cbind(P=predict(svm.cu, cuts),predict(svm.cu, cuts, type="prob"))
cm.svm.cu <- confusionMatrix(pred.svm.cu$P,cuts$Dropout,positive = "Y")
roc.svm.cu <- roc(cuts$Dropout, pred.svm.cu$Y)
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
pred.svmc.cu <- cbind(P=predict(svmc.cu, cuts),predict(svmc.cu, cuts, type="prob"))
cm.svmc.cu <- confusionMatrix(pred.svmc.cu$P,cuts$Dropout,positive = "Y")
roc.svmc.cu <- roc(cuts$Dropout, pred.svmc.cu$Y)
```

```
pred.xgb.cu <- cbind(P=predict(xgb.cu, cuts),predict(xgb.cu, cuts, type="prob"))
cm.xgb.cu <- confusionMatrix(pred.xgb.cu$P,cuts$Dropout,positive = "Y")
roc.xgb.cu <- roc(cuts$Dropout, pred.xgb.cu$Y)
```

2) REDUCED UNBALANCED

```
pred.dt.ru <- cbind(P=predict(dt.ru, ruts),predict(dt.ru, ruts, type="prob"))
cm.dt.ru <- confusionMatrix(pred.dt.ru$P,ruts$Dropout,positive = "Y")
roc.dt.ru <- roc(ruts$Dropout, pred.dt.ru$Y)
```

```
pred.knn.ru <- cbind(P=predict(knn.ru, ruts),predict(knn.ru, ruts, type="prob"))
cm.knn.ru <- confusionMatrix(pred.knn.ru$P,ruts$Dropout,positive = "Y")
roc.knn.ru <- roc(ruts$Dropout, pred.knn.ru$Y)
```

```
pred.lr.ru <- cbind(P=predict(lr.ru, ruts),predict(lr.ru, ruts, type="prob"))
cm.lr.ru <- confusionMatrix(pred.lr.ru$P,ruts$Dropout,positive = "Y")
roc.lr.ru <- roc(ruts$Dropout, pred.lr.ru$Y)
```

```
pred.nb.ru <- cbind(P=predict(nb.ru, ruts),predict(nb.ru, ruts, type="prob"))
cm.nb.ru <- confusionMatrix(pred.nb.ru$P,ruts$Dropout,positive = "Y")
roc.nb.ru <- roc(ruts$Dropout, pred.nb.ru$Y)
```

```
pred.nn.ru <- cbind(P=predict(nn.ru, ruts),predict(nn.ru, ruts, type="prob"))
cm.nn.ru <- confusionMatrix(pred.nn.ru$P,ruts$Dropout,positive = "Y")
roc.nn.ru <- roc(ruts$Dropout, pred.nn.ru$Y)
```

```
pred.rf.ru <- cbind(P=predict(rf.ru, ruts),predict(rf.ru, ruts, type="prob"))
cm.rf.ru <- confusionMatrix(pred.rf.ru$P,ruts$Dropout,positive = "Y")
roc.rf.ru <- roc(ruts$Dropout, pred.rf.ru$Y)
```

```
pred.svml.ru <- cbind(P=predict(svml.ru, ruts),predict(svml.ru, ruts, type="prob"))
cm.svml.ru <- confusionMatrix(pred.svml.ru$P,ruts$Dropout,positive = "Y")
roc.svml.ru <- roc(ruts$Dropout, pred.svml.ru$Y)
```

```
pred.svmc.ru <- cbind(P=predict(svmc.ru, ruts),predict(svmc.ru, ruts, type="prob"))
cm.svmc.ru <- confusionMatrix(pred.svmc.ru$P,ruts$Dropout,positive = "Y")
roc.svmc.ru <- roc(ruts$Dropout, pred.svmc.ru$Y)
```

```
pred.xgb.ru <- cbind(P=predict(xgb.ru, ruts),predict(xgb.ru, ruts, type="prob"))
cm.xgb.ru <- confusionMatrix(pred.xgb.ru$P,ruts$Dropout,positive = "Y")
roc.xgb.ru <- roc(ruts$Dropout, pred.xgb.ru$Y)
```

3) COMPLETE BALANCED

```
pred.dt.cb <- cbind(P=predict(dt.cb, cbts),predict(dt.cb, cbts, type="prob"))
cm.dt.cb <- confusionMatrix(pred.dt.cb$P,cbts$Dropout,positive = "Y")
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
roc.dt.cb <- roc(cbts$Dropout, pred.dt.cb$Y)
```

```
pred.knn.cb <- cbind(P=predict(knn.cb, cbts),predict(knn.cb, cbts, type="prob"))  
cm.knn.cb <- confusionMatrix(pred.knn.cb$P,cbts$Dropout,positive = "Y")  
roc.knn.cb <- roc(cbts$Dropout, pred.knn.cb$Y)
```

```
pred.lr.cb <- cbind(P=predict(lr.cb, cbts),predict(lr.cb, cbts, type="prob"))  
cm.lr.cb <- confusionMatrix(pred.lr.cb$P,cbts$Dropout,positive = "Y")  
roc.lr.cb <- roc(cbts$Dropout, pred.lr.cb$Y)
```

```
pred.nb.cb <- cbind(P=predict(nb.cb, cbts),predict(nb.cb, cbts, type="prob"))  
cm.nb.cb <- confusionMatrix(pred.nb.cb$P,cbts$Dropout,positive = "Y")  
roc.nb.cb <- roc(cbts$Dropout, pred.nb.cb$Y)
```

```
pred.nn.cb <- cbind(P=predict(nn.cb, cbts),predict(nn.cb, cbts, type="prob"))  
cm.nn.cb <- confusionMatrix(pred.nn.cb$P,cbts$Dropout,positive = "Y")  
roc.nn.cb <- roc(cbts$Dropout, pred.nn.cb$Y)
```

```
pred.rf.cb <- cbind(P=predict(rf.cb, cbts),predict(rf.cb, cbts, type="prob"))  
cm.rf.cb <- confusionMatrix(pred.rf.cb$P,cbts$Dropout,positive = "Y")  
roc.rf.cb <- roc(cbts$Dropout, pred.rf.cb$Y)
```

```
pred.svm.cb <- cbind(P=predict(svm.cb, cbts),predict(svm.cb, cbts, type="prob"))  
cm.svm.cb <- confusionMatrix(pred.svm.cb$P,cbts$Dropout,positive = "Y")  
roc.svm.cb <- roc(cbts$Dropout, pred.svm.cb$Y)
```

```
pred.svm.cb <- cbind(P=predict(svm.cb, cbts),predict(svm.cb, cbts, type="prob"))  
cm.svm.cb <- confusionMatrix(pred.svm.cb$P,cbts$Dropout,positive = "Y")  
roc.svm.cb <- roc(cbts$Dropout, pred.svm.cb$Y)
```

```
pred.xgb.cb <- cbind(P=predict(xgb.cb, cbts),predict(xgb.cb, cbts, type="prob"))  
cm.xgb.cb <- confusionMatrix(pred.xgb.cb$P,cbts$Dropout,positive = "Y")  
roc.xgb.cb <- roc(cbts$Dropout, pred.xgb.cb$Y)
```

```
##### 4) REDUCED BALANCED #####
```

```
pred.dt.rb <- cbind(P=predict(dt.rb, rbts),predict(dt.rb, rbts, type="prob"))  
cm.dt.rb <- confusionMatrix(pred.dt.rb$P,rbts$Dropout,positive = "Y")  
roc.dt.rb <- roc(rbts$Dropout, pred.dt.rb$Y)
```

```
pred.knn.rb <- cbind(P=predict(knn.rb, rbts),predict(knn.rb, rbts, type="prob"))  
cm.knn.rb <- confusionMatrix(pred.knn.rb$P,rbts$Dropout,positive = "Y")  
roc.knn.rb <- roc(rbts$Dropout, pred.knn.rb$Y)
```

```
pred.lr.rb <- cbind(P=predict(lr.rb, rbts),predict(lr.rb, rbts, type="prob"))  
cm.lr.rb <- confusionMatrix(pred.lr.rb$P,rbts$Dropout,positive = "Y")  
roc.lr.rb <- roc(rbts$Dropout, pred.lr.rb$Y)
```

```
pred.nb.rb <- cbind(P=predict(nb.rb, rbts),predict(nb.rb, rbts, type="prob"))  
cm.nb.rb <- confusionMatrix(pred.nb.rb$P,rbts$Dropout,positive = "Y")  
roc.nb.rb <- roc(rbts$Dropout, pred.nb.rb$Y)
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
pred.nn.rb <- cbind(P=predict(nn.rb, rbts),predict(nn.rb, rbts, type="prob"))
cm.nn.rb <- confusionMatrix(pred.nn.rb$P,rbts$Dropout,positive = "Y")
roc.nn.rb <- roc(rbts$Dropout, pred.nn.rb$Y)
```

```
pred.rf.rb <- cbind(P=predict(rf.rb, rbts),predict(rf.rb, rbts, type="prob"))
cm.rf.rb <- confusionMatrix(pred.rf.rb$P,rbts$Dropout,positive = "Y")
roc.rf.rb <- roc(rbts$Dropout, pred.rf.rb$Y)
```

```
pred.svml.rb <- cbind(P=predict(svml.rb, rbts),predict(svml.rb, rbts, type="prob"))
cm.svml.rb <- confusionMatrix(pred.svml.rb$P,rbts$Dropout,positive = "Y")
roc.svml.rb <- roc(rbts$Dropout, pred.svml.rb$Y)
```

```
pred.svmp.rb <- cbind(P=predict(svmp.rb, rbts),predict(svmp.rb, rbts, type="prob"))
cm.svmp.rb <- confusionMatrix(pred.svmp.rb$P,rbts$Dropout,positive = "Y")
roc.svmp.rb <- roc(rbts$Dropout, pred.svmp.rb$Y)
```

```
pred.xgb.rb <- cbind(P=predict(xgb.rb, rbts),predict(xgb.rb, rbts, type="prob"))
cm.xgb.rb <- confusionMatrix(pred.xgb.rb$P,rbts$Dropout,positive = "Y")
roc.xgb.rb <- roc(rbts$Dropout, pred.xgb.rb$Y)
```

COMBINE INTO RESULTS TABLE

#ContingencyMatrixes

```
cm <- data.frame(rbind(dt.cu = as.vector(prop.table(cm.dt.cu$table)),
  knn.cu = as.vector(prop.table(cm.knn.cu$table)),
  lr.cu = as.vector(prop.table(cm.lr.cu$table)),
  nb.cu = as.vector(prop.table(cm.nb.cu$table)),
  nn.cu = as.vector(prop.table(cm.nn.cu$table)),
  rf.cu = as.vector(prop.table(cm.rf.cu$table)),
  svml.cu = as.vector(prop.table(cm.svml.cu$table)),
  svmp.cu = as.vector(prop.table(cm.svmp.cu$table)),
  xgb.cu = as.vector(prop.table(cm.xgb.cu$table)),

  dt.ru = as.vector(prop.table(cm.dt.ru$table)),
  knn.ru = as.vector(prop.table(cm.knn.ru$table)),
  lr.ru = as.vector(prop.table(cm.lr.ru$table)),
  nb.ru = as.vector(prop.table(cm.nb.ru$table)),
  nn.ru = as.vector(prop.table(cm.nn.ru$table)),
  rf.ru = as.vector(prop.table(cm.rf.ru$table)),
  svml.ru = as.vector(prop.table(cm.svml.ru$table)),
  svmp.ru = as.vector(prop.table(cm.svmp.ru$table)),
  xgb.ru = as.vector(prop.table(cm.xgb.ru$table)),

  dt.cb = as.vector(prop.table(cm.dt.cb$table)),
  knn.cb = as.vector(prop.table(cm.knn.cb$table)),
  lr.cb = as.vector(prop.table(cm.lr.cb$table)),
  nb.cb = as.vector(prop.table(cm.nb.cb$table)),
  nn.cb = as.vector(prop.table(cm.nn.cb$table)),
  rf.cb = as.vector(prop.table(cm.rf.cb$table)),
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
svml.cb = as.vector(prop.table(cm.svml.cb$table)),
svmp.cb = as.vector(prop.table(cm.svmp.cb$table)),
xgb.cb = as.vector(prop.table(cm.xgb.cb$table)),

dt.rb = as.vector(prop.table(cm.dt.rb$table)),
knn.rb = as.vector(prop.table(cm.knn.rb$table)),
lr.rb = as.vector(prop.table(cm.lr.rb$table)),
nb.rb = as.vector(prop.table(cm.nb.rb$table)),
nn.rb = as.vector(prop.table(cm.nn.rb$table)),
rf.rb = as.vector(prop.table(cm.rf.rb$table)),
svml.rb = as.vector(prop.table(cm.svml.rb$table)),
svmp.rb = as.vector(prop.table(cm.svmp.rb$table)),
xgb.rb = as.vector(prop.table(cm.xgb.rb$table))
)

#clean up contingency matrix
colnames(cm) <- c("TN", "FP", "FN", "TP")
cm$Sensitivity <- with(cm, TP/(TP+FN))
cm$Specificity <- with(cm, TN/(TN+FP))
cm$Accuracy <- with(cm, TP+TN)
cm <- cm[c(5,6,7,4,2,1,3)]

# ROC AREA
AUC <- data.frame(ROC.Area = rbind(dt.cu = auc(roc.dt.cu),
knn.cu = auc(roc.knn.cu),
lr.cu = auc(roc.lr.cu),
nb.cu = auc(roc.nb.cu),
nn.cu = auc(roc.nn.cu),
rf.cu = auc(roc.rf.cu),
svml.cu = auc(roc.svml.cu),
svmp.cu = auc(roc.svmp.cu),
xgb.cu = auc(roc.xgb.cu),

dt.ru = auc(roc.dt.ru),
knn.ru = auc(roc.knn.ru),
lr.ru = auc(roc.lr.ru),
nb.ru = auc(roc.nb.ru),
nn.ru = auc(roc.nn.ru),
rf.ru = auc(roc.rf.ru),
svml.ru = auc(roc.svml.ru),
svmp.ru = auc(roc.svmp.ru),
xgb.ru = auc(roc.xgb.ru),

dt.cb = auc(roc.dt.cb),
knn.cb = auc(roc.knn.cb),
lr.cb = auc(roc.lr.cb),
nb.cb = auc(roc.nb.cb),
nn.cb = auc(roc.nn.cb),
rf.cb = auc(roc.rf.cb),
svml.cb = auc(roc.svml.cb),
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
      svmp.cb = auc(roc.svmp.cb),
      xgb.cb = auc(roc.xgb.cb),

      dt.rb = auc(roc.dt.rb),
      knn.rb = auc(roc.knn.rb),
      lr.rb = auc(roc.lr.rb),
      nb.rb = auc(roc.nb.rb),
      nn.rb = auc(roc.nn.rb),
      rf.rb = auc(roc.rf.rb),
      svm1.rb = auc(roc.svm1.rb),
      svmp.rb = auc(roc.svmp.rb),
      xgb.rb = auc(roc.xgb.rb)
    )
  )

# CREATE FINAL TABLE
results <- cbind(AUC,cm)
results <- round(results,4)
results <- cbind(Model=row.names(results),results)
results$Model <- as.character(results$Model)
results <- cbind(Algorithm=apply(strsplit(results$Model,"[.]"), `[, 1)
                    ,Data=apply(strsplit(results$Model,"[.]"), `[, 2)
                    ,results)

results

setwd("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/04
Model Testing/")
setwd("C:/Users/MKirkpatrick/Desktop/R Files/04 Model Testing/")
write.csv(results,file = "Results.csv",row.names = F)
save(results, file = "results.rda")
```

APPENDIX C4: R CODE – FEATURE IMPORTANCE

```
##### Import All Models #####
setwd("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/03
Model Building/")
files = list.files(pattern="*.rda")
for (i in 1:length(files)) load(file=files[i])
rm(files,i)

library(caret)

##### 1) COMPLETE UNBALANCED #####
vi.dt.cu <- varImp(dt.cu)
vi.dt.cu <- vi.dt.cu$importance
vi.dt.cu$Model <- "dt.cu"
vi.dt.cu$rn <- gsub("^", "",rownames(vi.dt.cu))

vi.knn.cu <- varImp(knn.cu)
vi.knn.cu <- vi.knn.cu$importance
t <- data.frame(Overall = vi.knn.cu[, "Y"])
t$Model <- "knn.cu"
t$rn <- rownames(vi.knn.cu)
vi.knn.cu <- t

vi.lr.cu <- varImp(lr.cu)
vi.lr.cu <- vi.lr.cu$importance
vi.lr.cu$Model <- "lr.cu"
vi.lr.cu$rn <- gsub("^", "",rownames(vi.lr.cu))

vi.nb.cu <- varImp(nb.cu)
vi.nb.cu <- vi.nb.cu$importance
t <- data.frame(Overall = vi.nb.cu[, "Y"])
t$Model <- "nb.cu"
t$rn <- rownames(vi.nb.cu)
vi.nb.cu <- t

vi.nn.cu <- varImp(nn.cu)
vi.nn.cu <- vi.nn.cu$importance
vi.nn.cu$Model <- "nn.cu"
vi.nn.cu$rn <- gsub("^", "",rownames(vi.nn.cu))

vi.rf.cu <- varImp(rf.cu)
vi.rf.cu <- vi.rf.cu$importance
vi.rf.cu$Model <- "rf.cu"
vi.rf.cu$rn <- gsub("^", "",rownames(vi.rf.cu))

vi.svml.cu <- varImp(svml.cu)
vi.svml.cu <- vi.svml.cu$importance
t <- data.frame(Overall = vi.svml.cu[, "Y"])
t$Model <- "svml.cu"
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
t$rn <- rownames(vi.svml.cu)
vi.svml.cu <- t
```

```
vi.svmp.cu <- varImp(svmp.cu)
vi.svmp.cu <- vi.svmp.cu$importance
t <- data.frame(Overall = vi.svmp.cu[, "Y"])
t$Model <- "svmp.cu"
t$rn <- rownames(vi.svmp.cu)
vi.svmp.cu <- t
```

```
vi.xgb.cu <- varImp(xgb.cu)
vi.xgb.cu <- vi.xgb.cu$importance
vi.xgb.cu$Model <- "xgb.cu"
vi.xgb.cu$rn <- gsub("", "", rownames(vi.xgb.cu))
```

```
##### 2) REDUCED UNBALANCED #####
```

```
vi.dt.ru <- varImp(dt.ru)
vi.dt.ru <- vi.dt.ru$importance
vi.dt.ru$Model <- "dt.ru"
vi.dt.ru$rn <- gsub("", "", rownames(vi.dt.ru))
```

```
vi.knn.ru <- varImp(knn.ru)
vi.knn.ru <- vi.knn.ru$importance
t <- data.frame(Overall = vi.knn.ru[, "Y"])
t$Model <- "knn.ru"
t$rn <- rownames(vi.knn.ru)
vi.knn.ru <- t
```

```
vi.lr.ru <- varImp(lr.ru)
vi.lr.ru <- vi.lr.ru$importance
vi.lr.ru$Model <- "lr.ru"
vi.lr.ru$rn <- gsub("", "", rownames(vi.lr.ru))
```

```
vi.nb.ru <- varImp(nb.ru)
vi.nb.ru <- vi.nb.ru$importance
t <- data.frame(Overall = vi.nb.ru[, "Y"])
t$Model <- "nb.ru"
t$rn <- rownames(vi.nb.ru)
vi.nb.ru <- t
```

```
vi.nn.ru <- varImp(nn.ru)
vi.nn.ru <- vi.nn.ru$importance
vi.nn.ru$Model <- "nn.ru"
vi.nn.ru$rn <- gsub("", "", rownames(vi.nn.ru))
```

```
vi.rf.ru <- varImp(rf.ru)
vi.rf.ru <- vi.rf.ru$importance
vi.rf.ru$Model <- "rf.ru"
vi.rf.ru$rn <- gsub("", "", rownames(vi.rf.ru))
```


MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
vi.svml.ru <- varImp(svml.ru)
vi.svml.ru <- vi.svml.ru$importance
t <- data.frame(Overall = vi.svml.ru[, "Y"])
t$Model <- "svml.ru"
t$rn <- rownames(vi.svml.ru)
vi.svml.ru <- t
```

```
vi.svmp.ru <- varImp(svmp.ru)
vi.svmp.ru <- vi.svmp.ru$importance
t <- data.frame(Overall = vi.svmp.ru[, "Y"])
t$Model <- "svmp.ru"
t$rn <- rownames(vi.svmp.ru)
vi.svmp.ru <- t
```

```
vi.xgb.ru <- varImp(xgb.ru)
vi.xgb.ru <- vi.xgb.ru$importance
vi.xgb.ru$Model <- "xgb.ru"
vi.xgb.ru$rn <- gsub("", "", rownames(vi.xgb.ru))
```

3) COMPLETE BALANCED

```
vi.dt.cb <- varImp(dt.cb)
vi.dt.cb <- vi.dt.cb$importance
vi.dt.cb$Model <- "dt.cb"
vi.dt.cb$rn <- gsub("", "", rownames(vi.dt.cb))
```

```
vi.knn.cb <- varImp(knn.cb)
vi.knn.cb <- vi.knn.cb$importance
t <- data.frame(Overall = vi.knn.cb[, "Y"])
t$Model <- "knn.cb"
t$rn <- rownames(vi.knn.cb)
vi.knn.cb <- t
```

```
vi.lr.cb <- varImp(lr.cb)
vi.lr.cb <- vi.lr.cb$importance
vi.lr.cb$Model <- "lr.cb"
vi.lr.cb$rn <- gsub("", "", rownames(vi.lr.cb))
```

```
vi.nb.cb <- varImp(nb.cb)
vi.nb.cb <- vi.nb.cb$importance
t <- data.frame(Overall = vi.nb.cb[, "Y"])
t$Model <- "nb.cb"
t$rn <- rownames(vi.nb.cb)
vi.nb.cb <- t
```

```
vi.nn.cb <- varImp(nn.cb)
vi.nn.cb <- vi.nn.cb$importance
vi.nn.cb$Model <- "nn.cb"
vi.nn.cb$rn <- gsub("", "", rownames(vi.nn.cb))
```

```
vi.rf.cb <- varImp(rf.cb)
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
vi.rf.cb <- vi.rf.cb$importance
vi.rf.cb$Model <- "rf.cb"
vi.rf.cb$rn <- gsub("", "",rownames(vi.rf.cb))
```

```
vi.svml.cb <- varImp(svml.cb)
vi.svml.cb <- vi.svml.cb$importance
t <- data.frame(Overall = vi.svml.cb[, "Y"])
t$Model <- "svml.cb"
t$rn <- rownames(vi.svml.cb)
vi.svml.cb <- t
```

```
vi.svmp.cb <- varImp(svmp.cb)
vi.svmp.cb <- vi.svmp.cb$importance
t <- data.frame(Overall = vi.svmp.cb[, "Y"])
t$Model <- "svmp.cb"
t$rn <- rownames(vi.svmp.cb)
vi.svmp.cb <- t
```

```
vi.xgb.cb <- varImp(xgb.cb)
vi.xgb.cb <- vi.xgb.cb$importance
vi.xgb.cb$Model <- "xgb.cb"
vi.xgb.cb$rn <- gsub("", "",rownames(vi.xgb.cb))
```

```
##### 4) REDUCED BALANCED #####
```

```
vi.dt.rb <- varImp(dt.rb)
vi.dt.rb <- vi.dt.rb$importance
vi.dt.rb$Model <- "dt.rb"
vi.dt.rb$rn <- gsub("", "",rownames(vi.dt.rb))
```

```
vi.knn.rb <- varImp(knn.rb)
vi.knn.rb <- vi.knn.rb$importance
t <- data.frame(Overall = vi.knn.rb[, "Y"])
t$Model <- "knn.rb"
t$rn <- rownames(vi.knn.rb)
vi.knn.rb <- t
```

```
vi.lr.rb <- varImp(lr.rb)
vi.lr.rb <- vi.lr.rb$importance
vi.lr.rb$Model <- "lr.rb"
vi.lr.rb$rn <- gsub("", "",rownames(vi.lr.rb))
```

```
vi.nb.rb <- varImp(nb.rb)
vi.nb.rb <- vi.nb.rb$importance
t <- data.frame(Overall = vi.nb.rb[, "Y"])
t$Model <- "nb.rb"
t$rn <- rownames(vi.nb.rb)
vi.nb.rb <- t
```

```
vi.nn.rb <- varImp(nn.rb)
vi.nn.rb <- vi.nn.rb$importance
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
vi.nn.rb$Model <- "nn.rb"  
vi.nn.rb$rn <- gsub("", "",rownames(vi.nn.rb))
```

```
vi.rf.rb <- varImp(rf.rb)  
vi.rf.rb <- vi.rf.rb$importance  
vi.rf.rb$Model <- "rf.rb"  
vi.rf.rb$rn <- gsub("", "",rownames(vi.rf.rb))
```

```
vi.svml.rb <- varImp(svml.rb)  
vi.svml.rb <- vi.svml.rb$importance  
t <- data.frame(Overall = vi.svml.rb[, "Y"])  
t$Model <- "svml.rb"  
t$rn <- rownames(vi.svml.rb)  
vi.svml.rb <- t
```

```
vi.svmp.rb <- varImp(svmp.rb)  
vi.svmp.rb <- vi.svmp.rb$importance  
t <- data.frame(Overall = vi.svmp.rb[, "Y"])  
t$Model <- "svmp.rb"  
t$rn <- rownames(vi.svmp.rb)  
vi.svmp.rb <- t
```

```
vi.xgb.rb <- varImp(xgb.rb)  
vi.xgb.rb <- vi.xgb.rb$importance  
vi.xgb.rb$Model <- "xgb.rb"  
vi.xgb.rb$rn <- gsub("", "",rownames(vi.xgb.rb))
```

COMBINE ALL FEATURES

#special step for Logistic/NNet

```
vil <- rbind(vi.lr.cu, vi.nn.cu, vi.lr.ru, vi.nn.ru, vi.lr.cb, vi.nn.cb, vi.lr.rb, vi.nn.rb)
```

```
vil$Feature <- sapply(vil$rn, function(i)  
  if (grepl("^Units",i)) {"Units"}  
  else if (grepl("^TransUnits",i)) {"TransUnits"}  
  else if (grepl("^TeachingGrp",i)) {"TeachingGrp"}  
  else if (grepl("^SIC_avg",i)) {"SIC_avg"}  
  else if (grepl("^SchldUnits",i)) {"SchldUnits"}  
  else if (grepl("^SchldToNxtFY",i)) {"SchldToNxtFY"}  
  else if (grepl("^SchldMaxDays",i)) {"SchldMaxDays"}  
  else if (grepl("^RemedialCrns",i)) {"RemedialCrns"}  
  else if (grepl("^RelativePerf",i)) {"RelativePerf"}  
  else if (grepl("^Probation",i)) {"Probation"}  
  else if (grepl("^PrevGPA",i)) {"PrevGPA"}  
  else if (grepl("^PrevDegLvl",i)) {"PrevDegLvl"}  
  else if (grepl("^PrevAtt4Yr",i)) {"PrevAtt4Yr"}  
  else if (grepl("^PrevAtt2Yr",i)) {"PrevAtt2Yr"}  
  else if (grepl("^ParentHiEd",i)) {"ParentHiEd"}  
  else if (grepl("^P_VaTuit",i)) {"P_VaTuit"}  
  else if (grepl("^P_SOC",i)) {"P_SOC"}  
  else if (grepl("^P_ROR",i)) {"P_ROR"}
```

```

else if (grepl("^P_RFA",i)) {"P_RFA"}
else if (grepl("^P_REV",i)) {"P_REV"}
else if (grepl("^P_OffsiteDisc",i)) {"P_OffsiteDisc"}
else if (grepl("^P_NOS",i)) {"P_NOS"}
else if (grepl("^P_MiscSchlr",i)) {"P_MiscSchlr"}
else if (grepl("^P_HIC",i)) {"P_HIC"}
else if (grepl("^P_FRD",i)) {"P_FRD"}
else if (grepl("^P_FBA",i)) {"P_FBA"}
else if (grepl("^P_FAS",i)) {"P_FAS"}
else if (grepl("^P_FAO",i)) {"P_FAO"}
else if (grepl("^P_FAC",i)) {"P_FAC"}
else if (grepl("^P_EPS",i)) {"P_EPS"}
else if (grepl("^P_ECS",i)) {"P_ECS"}
else if (grepl("^P_ECP",i)) {"P_ECP"}
else if (grepl("^P_BVO",i)) {"P_BVO"}
else if (grepl("^P_BVA",i)) {"P_BVA"}
else if (grepl("^P_BV3",i)) {"P_BV3"}
else if (grepl("^P_BSS",i)) {"P_BSS"}
else if (grepl("^P_BSP",i)) {"P_BSP"}
else if (grepl("^P_BRR",i)) {"P_BRR"}
else if (grepl("^P_BOM",i)) {"P_BOM"}
else if (grepl("^P_BCR",i)) {"P_BCR"}
else if (grepl("^P_BCB",i)) {"P_BCB"}
else if (grepl("^P_B2B",i)) {"P_B2B"}
else if (grepl("^OnlineOnly",i)) {"OnlineOnly"}
else if (grepl("^N_WriteOff",i)) {"N_WriteOff"}
else if (grepl("^N_REC",i)) {"N_REC"}
else if (grepl("^N_IntlStu",i)) {"N_IntlStu"}
else if (grepl("^N_Hold",i)) {"N_Hold"}
else if (grepl("^N_FPW",i)) {"N_FPW"}
else if (grepl("^N_ECD",i)) {"N_ECD"}
else if (grepl("^N_CRD",i)) {"N_CRD"}
else if (grepl("^N_BPP",i)) {"N_BPP"}
else if (grepl("^N_BLK",i)) {"N_BLK"}
else if (grepl("^N_BKA",i)) {"N_BKA"}
else if (grepl("^N_BCW",i)) {"N_BCW"}
else if (grepl("^MilitaryYN",i)) {"MilitaryYN"}
else if (grepl("^MaritalStat",i)) {"MaritalStat"}
else if (grepl("^LearningGrp",i)) {"LearningGrp"}
else if (grepl("^GovtPrgmY",i)) {"GovtPrgmY"}
else if (grepl("^Gender",i)) {"Gender"}
else if (grepl("^FCD_FYQ",i)) {"FCD_FYQ"}
else if (grepl("^FAFSAbY",i)) {"FAFSAbY"}
else if (grepl("^Ethnicity",i)) {"Ethnicity"}
else if (grepl("^EFC",i)) {"EFC"}
else if (grepl("^DFUWI",i)) {"DFUWI"}
else if (grepl("^DependentsY",i)) {"DependentsY"}
else if (grepl("^DegreeAwardType",i)) {"DegreeAwardType"}
else if (grepl("^DaysToFYClose",i)) {"DaysToFYClose"}
else if (grepl("^DaysToFC",i)) {"DaysToFC"}
else if (grepl("^DaysSinceLastAtt",i)) {"DaysSinceLastAtt"}

```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
else if (grepl("^ClassLength_avg",i)) {"ClassLength_avg"}
else if (grepl("^Classes",i)) {"Classes"}
else if (grepl("^ClassCPE",i)) {"ClassCPE"}
else if (grepl("^CIP2D",i)) {"CIP2D"}
else if (grepl("^ChildrenY",i)) {"ChildrenY"}
else if (grepl("^AGI_PerCapita",i)) {"AGI_PerCapita"}
else if (grepl("^AGI",i)) {"AGI"}
else if (grepl("^Age",i)) {"Age"}
else if (grepl("^AdjunctOnly",i)) {"AdjunctOnly"}
else if (grepl("^ActiveDuty",i)) {"ActiveDuty"}
else {NA}
)

vil[is.na(vil)] <- 0
vil <- aggregate(vil[, "Overall"], by=list(Model=vil$Model, Feature=vil$Feature), FUN=max)

##### Tree Based Models #####
rm(vit)
vit <- rbind(vi.dt.cu, vi.rf.cu, vi.xgb.cu,
             vi.dt.ru, vi.rf.ru, vi.xgb.ru,
             vi.dt.cb, vi.rf.cb, vi.xgb.cb,
             vi.dt.rb, vi.rf.rb, vi.xgb.rb)

vit$Feature <- sapply(vit$rn, function(i)
  if (grepl("^Units",i)) {"Units"}
  else if (grepl("^TransUnits",i)) {"TransUnits"}
  else if (grepl("^TeachingGrp",i)) {"TeachingGrp"}
  else if (grepl("^SIC_avg",i)) {"SIC_avg"}
  else if (grepl("^SchldUnits",i)) {"SchldUnits"}
  else if (grepl("^SchldToNxtFY",i)) {"SchldToNxtFY"}
  else if (grepl("^SchldMaxDays",i)) {"SchldMaxDays"}
  else if (grepl("^RemedialCrs",i)) {"RemedialCrs"}
  else if (grepl("^RelativePerf",i)) {"RelativePerf"}
  else if (grepl("^Probation",i)) {"Probation"}
  else if (grepl("^PrevGPA",i)) {"PrevGPA"}
  else if (grepl("^PrevDegLvl",i)) {"PrevDegLvl"}
  else if (grepl("^PrevAtt4Yr",i)) {"PrevAtt4Yr"}
  else if (grepl("^PrevAtt2Yr",i)) {"PrevAtt2Yr"}
  else if (grepl("^ParentHiEd",i)) {"ParentHiEd"}
  else if (grepl("^P_VaTuit",i)) {"P_VaTuit"}
  else if (grepl("^P_SOC",i)) {"P_SOC"}
  else if (grepl("^P_ROR",i)) {"P_ROR"}
  else if (grepl("^P_RFA",i)) {"P_RFA"}
  else if (grepl("^P_REV",i)) {"P_REV"}
  else if (grepl("^P_OffsiteDisc",i)) {"P_OffsiteDisc"}
  else if (grepl("^P_NOS",i)) {"P_NOS"}
  else if (grepl("^P_MiscSchlr",i)) {"P_MiscSchlr"}
  else if (grepl("^P_HIC",i)) {"P_HIC"}
  else if (grepl("^P_FRD",i)) {"P_FRD"}
  else if (grepl("^P_FBA",i)) {"P_FBA"}
```

```

else if (grepl("^P_FAS",i)) {"P_FAS"}
else if (grepl("^P_FAO",i)) {"P_FAO"}
else if (grepl("^P_FAC",i)) {"P_FAC"}
else if (grepl("^P_EPS",i)) {"P_EPS"}
else if (grepl("^P_ECS",i)) {"P_ECS"}
else if (grepl("^P_ECP",i)) {"P_ECP"}
else if (grepl("^P_BVO",i)) {"P_BVO"}
else if (grepl("^P_BVA",i)) {"P_BVA"}
else if (grepl("^P_BV3",i)) {"P_BV3"}
else if (grepl("^P_BSS",i)) {"P_BSS"}
else if (grepl("^P_BSP",i)) {"P_BSP"}
else if (grepl("^P_BRR",i)) {"P_BRR"}
else if (grepl("^P_BOM",i)) {"P_BOM"}
else if (grepl("^P_BCR",i)) {"P_BCR"}
else if (grepl("^P_BCB",i)) {"P_BCB"}
else if (grepl("^P_B2B",i)) {"P_B2B"}
else if (grepl("^OnlineOnly",i)) {"OnlineOnly"}
else if (grepl("^N_WriteOff",i)) {"N_WriteOff"}
else if (grepl("^N_REC",i)) {"N_REC"}
else if (grepl("^N_IntlStu",i)) {"N_IntlStu"}
else if (grepl("^N_Hold",i)) {"N_Hold"}
else if (grepl("^N_FPW",i)) {"N_FPW"}
else if (grepl("^N_ECD",i)) {"N_ECD"}
else if (grepl("^N_CRD",i)) {"N_CRD"}
else if (grepl("^N_BPP",i)) {"N_BPP"}
else if (grepl("^N_BLK",i)) {"N_BLK"}
else if (grepl("^N_BKA",i)) {"N_BKA"}
else if (grepl("^N_BCW",i)) {"N_BCW"}
else if (grepl("^MilitaryYN",i)) {"MilitaryYN"}
else if (grepl("^MaritalStat",i)) {"MaritalStat"}
else if (grepl("^LearningGrp",i)) {"LearningGrp"}
else if (grepl("^GovtPrgmY",i)) {"GovtPrgmY"}
else if (grepl("^Gender",i)) {"Gender"}
else if (grepl("^FCD_FYQ",i)) {"FCD_FYQ"}
else if (grepl("^FAFSABY",i)) {"FAFSABY"}
else if (grepl("^Ethnicity",i)) {"Ethnicity"}
else if (grepl("^EFC",i)) {"EFC"}
else if (grepl("^DFUWI",i)) {"DFUWI"}
else if (grepl("^DependentsY",i)) {"DependentsY"}
else if (grepl("^DegreeAwardType",i)) {"DegreeAwardType"}
else if (grepl("^DaysToFYClose",i)) {"DaysToFYClose"}
else if (grepl("^DaysToFC",i)) {"DaysToFC"}
else if (grepl("^DaysSinceLastAtt",i)) {"DaysSinceLastAtt"}
else if (grepl("^ClassLength_avg",i)) {"ClassLength_avg"}
else if (grepl("^Classes",i)) {"Classes"}
else if (grepl("^ClassCPE",i)) {"ClassCPE"}
else if (grepl("^CIP2D",i)) {"CIP2D"}
else if (grepl("^ChildrenY",i)) {"ChildrenY"}
else if (grepl("^AGI_PerCapita",i)) {"AGI_PerCapita"}
else if (grepl("^AGI",i)) {"AGI"}
else if (grepl("^Age",i)) {"Age"}

```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
else if (grepl("^AdjunctOnly",i)) {"AdjunctOnly"}
else if (grepl("^ActiveDuty",i)) {"ActiveDuty"}
else {NA}
)

vit[is.na(vit)] <- 0
vit <- aggregate(vit[, "Overall"], by=list(Model=vit$Model, Feature=vit$Feature), FUN=sum)

##### All Other Models #####
vio <- rbind(vi.knn.cu, vi.knn.ru, vi.knn.cb, vi.knn.rb,
            vi.nb.cu, vi.nb.ru, vi.nb.cb, vi.nb.rb,
            vi.svm.cu, vi.svm.ru, vi.svm.cb, vi.svm.rb,
            vi.svm.cu, vi.svm.ru, vi.svm.cb, vi.svm.rb)

vio <- vio[c(2,3,1)]
colnames(vio)[2] <- "Feature"
colnames(vio)[3] <- "x"

##### Combine Models #####
v <- rbind(vil, vit, vio)

vi <- cbind(Algorithm=sapply(strsplit(v$Model, "."), `[`, 1)
            ,Data=sapply(strsplit(v$Model, "."), `[`, 2)
            ,v)

save(vi, file = "Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved
Objects/05 Variable Importance/vi.rda")
```

APPENDIX C5: R CODE – PLOTS AND TABLES

```
##### Recursive Feature Elimination #####
library(caret)
setwd("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/01
Data Prep/")
load("rfe.ru.rda")
load("rfe.rb.rda")

png("RFE ru plot.png", width = 528, height = 288)
plot(rfe.ru)
dev.off()

png("RFE rb plot.png", width = 528, height = 288)
plot(rfe.rb)
dev.off()

##### ANALYZE RESULTS #####
setwd("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/04
Model Testing/")
load("results.rda")

# Capitalize names
results$Algorithm <- toupper(results$Algorithm)
results$Model <- with(results,paste(Algorithm,"_",Data, sep = ""))

##### EXPORT RESULTS PLOTS #####
##### BY MODEL #####
png("Model ROC Area.png", width = 528, height = 600)
results$order <- with(results,reorder(Model,ROC.Area))
dotplot( order ~ ROC.Area, results, xlab = NULL #,main="ROC Area by Model"
        ,scales=list(x=list(cex=1.02),y=list(cex=1.02)))
dev.off()

png("Model Accuracy.png", width = 528, height = 600)
results$order <- with(results,reorder(Model,Accuracy))
dotplot( order ~ Accuracy, results,xlab = NULL#, main="Accuracy by Model"
        ,scales=list(x=list(cex=1.02),y=list(cex=1.02)))
dev.off()

png("Model Sensitivity.png", width = 528, height = 600)
results$order <- with(results,reorder(Model,Sensitivity))
dotplot( order ~ Sensitivity, results, xlab = NULL #,main="Sensitivity by Model"
        ,scales=list(x=list(cex=1.02),y=list(cex=1.02)))
dev.off()

png("Model Specificity.png", width = 528, height = 600)
```


MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
results$order <- with(results,reorder(Model,Specificity))
dotplot( order ~ Specificity, results, xlab = NULL#,main="Specificity by Model"
        ,scales=list(x=list(cex=1.02),y=list(cex=1.02)))
dev.off()
```

```
##### BY ALGORITHM #####
```

```
results.algo <- aggregate(results[,4:7], by=list(results$Algorithm), FUN=mean)
#setwd("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/04
Model Testing/")
#write.csv(results.algo, file = "Algorithm Results.csv", row.names = F)
```

```
png("Algorithm Metrics.png", width = 528, height = 600)
par(mfrow=c(2,2))
results$order <- with(results,reorder(Algorithm,ROC.Area,mean))
boxplot(ROC.Area ~ order,las=2,
        data = results,
        main = "ROC Area",par(cex.axis=1.2))
results$order <- with(results,reorder(Algorithm,Accuracy,mean))
boxplot(Accuracy ~ order,las=2,
        data = results,
        main = "Accuracy",par(cex.axis=1.2))
results$order <- with(results,reorder(Algorithm,Sensitivity,mean))
boxplot(Sensitivity ~ order,las=2,
        data = results,
        main = "Sensitivity",par(cex.axis=1.2))
results$order <- with(results,reorder(Algorithm,Specificity,mean))
boxplot(Specificity ~ order,las=2,
        data = results,
        main = "Specificity",par(cex.axis=1.2))
par(mfrow=c(1,1))
dev.off()
```

```
##### BY Data #####
```

```
results.data <- aggregate(results[,4:7], by=list(results$Data), FUN=mean)
```

```
png("Dataset Metrics.png", width = 528, height = 528)
par(mfrow=c(2,2))
```

```
results$order <- with(results,reorder(Data,ROC.Area,mean))
boxplot(ROC.Area ~ order,
        data = results,
        main = "ROC Area",par(cex.axis=1.2))
```

```
results$order <- with(results,reorder(Data,Accuracy,mean))
boxplot(Accuracy ~ order,
        data = results,
        main = "Accuracy",par(cex.axis=1.2))
results$order <- with(results,reorder(Data,Sensitivity,mean))
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
boxplot(Sensitivity ~ order,
        data = results,
        main = "Sensitivity",par(cex.axis=1.2))
results$order <- with(results,reorder(Data,Specificity,mean))
boxplot(Specificity ~ order,
        data = results,
        main = "Specificity",par(cex.axis=1.2))

par(mfrow=c(1,1))
dev.off()
```

```
##### VARIABLE IMPORTANCE OVERALL #####
setwd("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/05
Variable Importance/")
load(file = "means.feature.rda")
```

```
feature.plot <- means.feature[order(-means.feature$Importance),]
feature.plot <- head(feature.plot,n=25)
```

```
png("Feature Importance All Models.png", width = 528, height = 600)
dotplot( with(feature.plot,reorder(Feature,Importance)) ~ LB + Importance + UB, feature.plot,
        xlab = "Mean Importance (95% C.I.)",
        #main="Feature Importance Across All Models",
        col = c("lightgreen","blue","lightgreen"),
        scales=list(x=list(cex=1.02),y=list(cex=1.02)))
dev.off()
```

```
##### VARIABLE IMPORTANCE XGB and RF #####
setwd("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/05
Variable Importance/")
load(file = "fi.xf.means.rda")
feature.plot <- fi.xf.means[order(-fi.xf.means$Importance),]
feature.plot <- head(feature.plot,n=25)
```

```
png("Feature Importance XBG and RF.png", width = 528, height = 600)
dotplot( with(feature.plot,reorder(Feature,Importance)) ~ LB + Importance + UB, feature.plot,
        xlab = "Importance (95% C.I.)",
        #main="Feature Importance for XGB and RF Models",
        col = c("lightgreen","blue","lightgreen"),
        scales=list(x=list(cex=1.02),y=list(cex=1.02)))
dev.off()
```

```
##### INDIVIDUAL FEATURES #####
raw <- read.csv("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Data.csv"
               ,na.strings = "NULL")
```

```
raw$Outcome <- as.factor(sapply(raw$Dropout, function(i)
  if (i == "Y") {"Dropout"}
  else if (i == "N") {"Persist"}))
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
else {NA}))

raw$DFUWI <- as.factor(sapply(raw$DFUWI, function(i)
  if (i == "Y") {"DFUWI"}
  else if (i == "N") {"Non-DFUWI"}
  else {NA}))

raw$DegreeAwardType <- as.factor(sapply(raw$DegreeAwardType, function(i)
  if (i == "Associate Degree") {"AA"}
  else if (i == "Bachelor Degree") {"BA"}
  else if (i == "Master's Degree") {"MA"}
  else {NA}))

raw$RelativePerf <- as.factor(sapply(raw$RelativePerf, function(i)
  if (i == "Above Median") {"Above"}
  else if (i == "Below Median") {"Below"}
  else {NA}))

raw$RFA <- as.factor(sapply(raw$P_RFA, function(i)
  if (i == "Y") {"Pending"}
  else if (i == "N") {"Approved"}
  else {NA}))

raw$Learning <- as.factor(sapply(raw$LearningGrp, function(i)
  if (i == "Negative") {"Neg"}
  else if (i == "No Response") {"None"}
  else if (i == "Positive") {"Pos"}
  else {NA}))

raw$Teaching <- as.factor(sapply(raw$TeachingGrp, function(i)
  if (i == "Negative") {"Neg"}
  else if (i == "No Response") {"None"}
  else if (i == "Positive") {"Pos"}
  else {NA}))

raw$PrevDeg <- as.factor(sapply(raw$PrevDegLvl, function(i)
  if (i == "Lower") {"Lower/Unkwn"}
  else if (i == "Equal or Higher") {"Higher/Equal"}
  else {NA}))

raw$CIP <- as.factor(sapply(raw$CIP2D, function(i)
  if (i == "09 - COMMUNICATION, JOURNALISM, AND RELATED PROGRAMS.") {"09
Com"}
  else if (i == "11 - COMPUTER AND INFORMATION SCIENCES AND SUPPORT
SERVICES.") {"11 Comp"}
  else if (i == "13 - EDUCATION.") {"13 Edu"}
  else if (i == "14 - ENGINEERING.") {"14 Eng"}
  else if (i == "15 - ENGINEERING TECHNOLOGIES AND ENGINEERING-RELATED
FIELDS.") {"15 EngT"}
  else if (i == "16 - FOREIGN LANGUAGES, LITERATURES, AND LINGUISTICS.") {"16
FLng"}
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
else if (i == "22 - LEGAL PROFESSIONS AND STUDIES.") {"22 Legl"}
else if (i == "23 - ENGLISH LANGUAGE AND LITERATURE/LETTERS.") {"23 Engl"}
else if (i == "24 - LIBERAL ARTS AND SCIENCES, GENERAL STUDIES AND
HUMANITIES.") {"24 LibA"}
else if (i == "26 - BIOLOGICAL AND BIOMEDICAL SCIENCES.") {"26 Bio"}
else if (i == "27 - MATHEMATICS AND STATISTICS.") {"27 Math"}
else if (i == "30 - MULTI/INTERDISCIPLINARY STUDIES.") {"30 IntSt"}
else if (i == "31 - PARKS, RECREATION, LEISURE, AND FITNESS STUDIES.") {"31 Fit"}
else if (i == "42 - PSYCHOLOGY.") {"42 Psy"}
else if (i == "43 - HOMELAND SECURITY, LAW ENFORCEMENT, FIREFIGHTING AND
RELATED PROTECTIVE SERVICES") {"43 Home"}
else if (i == "44 - PUBLIC ADMINISTRATION AND SOCIAL SERVICE PROFESSIONS.")
{"44 Pub"}
else if (i == "45 - SOCIAL SCIENCES.") {"45 SS"}
else if (i == "50 - VISUAL AND PERFORMING ARTS.") {"50 Arts"}
else if (i == "51 - HEALTH PROFESSIONS AND RELATED PROGRAMS.") {"51 Hlth"}
else if (i == "52 - BUSINESS, MANAGEMENT, MARKETING, AND RELATED SUPPORT
SERVICES.") {"52 Bus"}
else if (i == "54 - HISTORY.") {"54 Hist"}
else {NA}}))
```

```
#defaults for plots
color <- c("coral","aquamarine3")
yrange <- function(y) c(0,max(y)*1.25) #function to make Y axis 25% bigger
setwd("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/05
Variable Importance/")
```

```
# DFUWI
png("DFUWI.png", width = 480, height = 480)
par(mfrow=c(1,2))
t <- with(raw, table(Outcome,DFUWI))
barplot(t, beside=T, col=color, ylim = yrange(t),
        main = "Count")
legend("topleft", levels(raw$Outcome),pch=15,
        col=color, bty="n")
p <- prop.table(t,2) #if want a propotion table
barplot(p, col=color,main = "100%")
par(mfrow=c(1,1))
dev.off()
```

```
# DegreeAwardType
png("DegreeAwardType.png", width = 480, height = 480)
par(mfrow=c(1,2))
t <- with(raw, table(Outcome,DegreeAwardType))
barplot(t, beside=T, col=color, ylim = yrange(t),
        main = "Count")
legend("topleft", levels(raw$Outcome),pch=15,
        col=color, bty="n")
p <- prop.table(t,2) #if want a propotion table
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
barplot(p, col=color, main = "100%")
par(mfrow=c(1,1))
dev.off()

# RelativePerf
png("RelativePerf.png", width = 480, height = 480)
par(mfrow=c(1,2))
t <- with(raw, table(Outcome, RelativePerf))
barplot(t, beside=T, col=color, ylim = yrange(t),
        main = "Count")
legend("topleft", levels(raw$Outcome), pch=15,
       col=color, bty="n")
p <- prop.table(t, 2) #if want a propotion table
barplot(p, col=color, main = "100%")
par(mfrow=c(1,1))
dev.off()

# RFA
png("RFA.png", width = 480, height = 480)
par(mfrow=c(1,2))
t <- with(raw, table(Outcome, RFA))
barplot(t, beside=T, col=color, ylim = yrange(t),
        main = "Count")
legend("topleft", levels(raw$Outcome), pch=15,
       col=color, bty="n")
p <- prop.table(t, 2) #if want a propotion table
barplot(p, col=color, main = "100%")
par(mfrow=c(1,1))
dev.off()

# CIP2D
png("CIP2D.png", width = 480, height = 800)
par(mfrow=c(2,1))
t <- with(raw, table(Outcome, CIP))
barplot(t, beside=T, col=color, ylim = yrange(t),
        main = "Count", las=2)
legend("topleft", levels(raw$Outcome), pch=15,
       col=color, bty="n")
p <- prop.table(t, 2) #if want a propotion table
barplot(p, col=color, main = "100%", las=2)
par(mfrow=c(1,1))
dev.off()

# Perception of Learning
png("PercOfLearning.png", width = 480, height = 480)
par(mfrow=c(1,2))
t <- with(raw, table(Outcome, Learning))
barplot(t, beside=T, col=color, ylim = yrange(t),
        main = "Count")
legend("topleft", levels(raw$Outcome), pch=15,
       col=color, bty="n")
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
p <- prop.table(t,2) #if want a propotion table
barplot(p, col=color,main = "100%")
par(mfrow=c(1,1))
dev.off()

# Perception of Teaching
png("PercOfTeaching.png", width = 480, height = 480)
par(mfrow=c(1,2))
t <- with(raw, table(Outcome,Teaching))
barplot(t, beside=T, col=color, ylim = yrange(t),
        main = "Count")
legend("topleft", levels(raw$Outcome),pch=15,
        col=color, bty="n")
p <- prop.table(t,2) #if want a propotion table
barplot(p, col=color,main = "100%")
par(mfrow=c(1,1))
dev.off()

# PrevDegLvl
png("PrevDegLvl.png", width = 528, height = 480)
par(mfrow=c(1,2))
t <- with(raw, table(Outcome,PrevDeg))
barplot(t, beside=T, col=color, ylim = yrange(t),
        main = "Count")
legend("topleft", levels(raw$Outcome),pch=15,
        col=color, bty="n")
p <- prop.table(t,2) #if want a propotion table
barplot(p, col=color,main = "100%")
par(mfrow=c(1,1))
dev.off()

# FCD_FYQ
png("FCD_FYQ.png", width = 480, height = 480)
par(mfrow=c(1,2))
t <- with(raw, table(Outcome,FCD_FYQ))
barplot(t, beside=T, col=color, ylim = yrange(t),
        main = "Count")
legend("topleft", levels(raw$Outcome),pch=15,
        col=color, bty="n")
p <- prop.table(t,2) #if want a propotion table
barplot(p, col=color,main = "100%")
par(mfrow=c(1,1))
dev.off()

png("DaysToFYClose.png", width = 480, height = 480)
with(raw, boxplot(DaysToFYClose ~ Outcome, col = color))
dev.off()

png("TransUnits.png", width = 480, height = 480)
with(raw, boxplot(TransUnits ~ Outcome, col = color))
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
dev.off()

png("PrevGPA.png", width = 480, height = 480)
with(raw, boxplot(PrevGPA ~ Outcome, col = color))
dev.off()

png("AGI_PerCapita.png", width = 480, height = 480)
with(raw, boxplot(AGI_PerCapita ~ Outcome, col = color))
dev.off()

##### VARIABLE IMPORTANCE CHI SQUARE TABLE #####
raw <- read.csv("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Data.csv"
               ,na.strings = "NULL")
summary(raw)

dropout <- with(raw,data.frame(Dropout))
features <- raw[,4:79]

# Get list of variable types
l <- lapply(features,is.factor)
# create categorical data set
cat <- features[,l==T]
# create continuous data set
con <- features[,l==F]

# Recode Values
dropout$Dropout <- gsub("Y","Dropout", dropout$Dropout)
dropout$Dropout <- gsub("N","Persist", dropout$Dropout)

cat$DFUWI <- gsub("Y","DFUWI", cat$DFUWI)
cat$DFUWI <- gsub("N","Non-DFUWI", cat$DFUWI)
cat$P_RFA <- gsub("Y","Admission Pending", cat$P_RFA)
cat$P_RFA <- gsub("N","Admission Approved", cat$P_RFA)

#DFUWI, DegreeAwardType, RelativePerf, P_RFA, CIP2D, LearningGrp, TeachingGrp,
PrevDegLvl
as.matrix(table(cat$DFUWI))
t.count <- rbind(
  as.matrix(table(cat$DFUWI)),
  as.matrix(table(cat$DegreeAwardType)),
  as.matrix(table(cat$RelativePerf)),
  as.matrix(table(cat$P_RFA)),
  as.matrix(table(cat$CIP2D)),
  as.matrix(table(cat$LearningGrp)),
  as.matrix(table(cat$TeachingGrp)),
  as.matrix(table(cat$PrevDegLvl)))

t.perc <- rbind(
```

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
as.matrix(prop.table(table(cat$DFUWI))),
as.matrix(prop.table(table(cat$DegreeAwardType))),
as.matrix(prop.table(table(cat$RelativePerf))),
as.matrix(prop.table(table(cat$P_RFA))),
as.matrix(prop.table(table(cat$CIP2D))),
as.matrix(prop.table(table(cat$LearningGrp))),
as.matrix(prop.table(table(cat$TeachingGrp))),
as.matrix(prop.table(table(cat$PrevDegLvl))))

t.perc <- round(t.perc*100,1)

d.count <- rbind(
  table(cat$DFUWI,dropout$Dropout),
  table(cat$DegreeAwardType,dropout$Dropout),
  table(cat$RelativePerf,dropout$Dropout),
  table(cat$P_RFA,dropout$Dropout),
  table(cat$CIP2D,dropout$Dropout),
  table(cat$LearningGrp,dropout$Dropout),
  table(cat$TeachingGrp,dropout$Dropout),
  table(cat$PrevDegLvl,dropout$Dropout))

d.perc <- rbind(
  prop.table(table(cat$DFUWI,dropout$Dropout),2),
  prop.table(table(cat$DegreeAwardType,dropout$Dropout),2),
  prop.table(table(cat$RelativePerf,dropout$Dropout),2),
  prop.table(table(cat$P_RFA,dropout$Dropout),2),
  prop.table(table(cat$CIP2D,dropout$Dropout),2),
  prop.table(table(cat$LearningGrp,dropout$Dropout),2),
  prop.table(table(cat$TeachingGrp,dropout$Dropout),2),
  prop.table(table(cat$PrevDegLvl,dropout$Dropout),2))

d.perc <- round(d.perc*100,1)

d <- cbind(t.count,t.perc,d.count,d.perc)

setwd("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/05
Variable Importance/")
write.csv(d, file = "Feature Frequencies.csv", row.names = T)

#DegreeAwardType, RelativePerf, P_RFA, CIP2D, LearningGrp, TeachingGrp
with(raw,chisq.test(Dropout,DFUWI, correct=F))
with(raw,chisq.test(Dropout,DegreeAwardType, correct=F))
with(raw,chisq.test(Dropout,RelativePerf, correct=F))
with(raw,chisq.test(Dropout,P_RFA, correct=F))
with(raw,chisq.test(Dropout,CIP2D, correct=F))
with(raw,chisq.test(Dropout,LearningGrp, correct=F))
with(raw,chisq.test(Dropout,TeachingGrp, correct=F))
with(raw,chisq.test(Dropout,PrevDegLvl, correct=F))
```


MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

```
table(raw$Dropout)

##### Continuous Variables #####
raw <- read.csv("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Data.csv"
               ,na.strings = "NULL")

raw$Dropout <- gsub("Y","Dropout", raw$Dropout)
raw$Dropout <- gsub("N","Persist", raw$Dropout)

# DaysToFYClose TransUnits PrevGPA AGI_PerCapita
library(psych)
t <- raw[,c("Dropout","DaysToFYClose","TransUnits","PrevGPA","AGI_PerCapita")]
describeBy(t[2:5],group = t$Dropout,mat = T, digits = 1)

with(raw, t.test(DaysToFYClose ~ Dropout,var.equal = T))
with(raw, t.test(TransUnits ~ Dropout,var.equal = T))
with(raw, t.test(PrevGPA ~ Dropout,var.equal = T))
with(raw, t.test(AGI_PerCapita ~ Dropout,var.equal = T))

##### Features used in Models #####

setwd("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/01
Data Prep/")
load("cu.rda")
load("ru.rda")
load("cb.rda")
load("rb.rda")

n.cu <- data.frame(colnames(cu))
n.ru <- data.frame(colnames(ru))
n.cb <- data.frame(colnames(cb))
n.rb <- data.frame(colnames(rb))

n.cu$j <- colnames(cu)
n.ru$j <- colnames(ru)
n.cb$j <- colnames(cb)
n.rb$j <- colnames(rb)

features <- merge(n.cu,n.ru, by="j", all=T)
features <- merge(features,n.cb, by="j", all=T)
features <- merge(features,n.rb, by="j", all=T)

setwd("Z:/Advanced Analytics/One Year Retention Risk Indicator/Analyses/Saved Objects/04
Model Testing/")
write.csv(features, file = "Features Included.csv", row.names = T, na = "")
```


MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

APPENDIX D: FEATURES INCLUDED

Feature	Complete Unbalanced	Reduced Unbalanced	Complete Balanced	Reduced Balanced
ActiveDuty	X	X	X	X
AdjunctOnly	X	X	X	X
Age	X	X	X	X
AGI	X	X	X	X
AGI_PerCapita	X	X	X	X
ChildrenY	X	X	X	X
CIP2D	X	X	X	X
ClassCPE	X	X	X	X
Classes	X	X	X	X
ClassLength_avg	X	X	X	X
DaysSinceLastAtt	X	X	X	X
DaysToFC	X	X	X	X
DaysToFYClose	X	X	X	X
DegreeAwardType	X	X	X	X
DependentsY	X		X	X
DFUWI	X	X	X	X
Dropout	X	X	X	X
EFC	X	X	X	X
Ethnicity	X	X	X	X
FAFSAbY	X	X	X	X
FCD_FYQ	X	X	X	X
Gender	X	X	X	X
GovtPrgmY	X		X	X
LearningGrp	X	X	X	X
MaritalStat	X	X	X	X
MilitaryYN	X	X	X	X
N_BCW	X	X	X	X
N_BKA	X		X	X
N_BLK	X	X	X	X
N_BPP	X		X	
N_CRD	X		X	
N_ECD	X	X	X	X
N_FPW	X		X	X
N_Hold	X		X	X
N_IntlStu	X	X	X	X
N_REC	X		X	X
N_WriteOff	X		X	X
OnlineOnly	X	X	X	X
P_B2B	X	X	X	X
P_BCB	X		X	X
P_BCR	X		X	
P_BOM	X	X	X	X
P_BRR	X		X	
P_BSP	X		X	
P_BSS	X		X	X
P_BV3	X	X	X	X
P_BVA	X		X	

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

P_BVO	X		X	X
P_ECP	X		X	X
P_ECS	X	X	X	X
P_EPS	X		X	X
P_FAC	X	X	X	X
P_FAO	X	X	X	X
P_FAS	X	X	X	X
P_FBA	X	X	X	X
P_FRD	X		X	X
P_HIC	X		X	X
P_MiscSchlr	X		X	
P_NOS	X	X	X	X
P_OffsiteDisc	X	X	X	X
P_REV	X	X	X	X
P_RFA	X	X	X	X
P_ROR	X		X	
P_SOC	X	X	X	X
P_VaTuit	X	X	X	X
ParentHiEd	X	X	X	X
PrevAtt2Yr	X	X	X	X
PrevAtt4Yr	X		X	X
PrevDegLvl	X	X	X	X
PrevGPA	X	X	X	X
Probation	X	X	X	X
RelativePerf	X	X	X	X
RemedialCrs	X		X	
SIC_avg	X	X	X	X
TeachingGrp	X	X	X	X
TransUnits	X	X	X	X
Units	X	X	X	X

REFERENCES

- Alkhasawneh, R., & Hargraves, R. H. (2014). Developing a hybrid model to predict student first year retention in STEM disciplines using machine learning techniques. *Journal of STEM Education: Innovations and Research*, 15(3), 35.
- Aulck, L., Velagapudi, N., Blumenstock, J., & West, J. (2016). Predicting Student Dropout in Higher Education. *arXiv preprint arXiv:1606.06364*.
- Bean, J. P. (1980). Dropouts and turnover: The synthesis and test of a causal model of student attrition. *Research in higher education*, 12(2), 155-187.
- Bogard, M., Helbig, T., Huff, G., & James, C. (2011). A comparison of empirical models for predicting student retention. *White paper. Office of Institutional Research, Western Kentucky University*.
- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.
- Bresciani, M. J., & Carson, L. (2002). A study of undergraduate persistence by unmet need and percentage of gift aid. *NASPA Journal*, 40(1), 104-123.
- Cabrera, A. F., Nora, J.-E. A., & Castafne, M. B. (1993). Structural equations modeling test of an integrated model of student retention. *The Journal of Higher Education*, 64(2), 123-139.

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE:

Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 341–378.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system.

In *Proceedings of the 22nd Annual International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). ACM.

Cochran, J. D., Campbell, S. M., Baker, H. M., & Leeds, E. M. (2014). The role of student characteristics in predicting retention in online courses. *Research in Higher Education*, 55(1), 27-48.

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 215-242.

Delen, D. (2010). A comparative analysis of machine learning techniques for student retention management. *Decision Support Systems*, 49(4), 498-506.

Dey, E. L., & Astin, A. W. (1993). Statistical alternatives for studying college student retention: A comparative analysis of logit, probit, and linear regression. *Research in higher education*, 34(5), 569-581.

Fike, D. S., & Fike, R. (2008). Predictors of first-year student retention in the community college. *Community College Review*, 36(2), 68-88.

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

- Fluharty, A., Gallet, K., & Hower, Z. (2016). *Predicting First Quarter Student Retention at National University with Data Mining Techniques* (Unpublished master's thesis). National University, San Diego, CA.
- Garton, B. L., Ball, A. L., & Dyer, J. E. (2002). The academic performance and retention of college of agriculture students. *Journal of Agricultural Education*, 43(1), 46-56.
- Glynn, J. G., Sauer, P. L., & Miller, T. E. (2003). Signaling student retention with prematriculation data. *NASPA Journal*, 41(1), 41-67.
- Herzog, S. (2005). Measuring determinants of student return vs. dropout/stopout vs. transfer: A first-to-second year analysis of new freshmen. *Research in higher education*, 46(8), 883-928.
- Herzog, S. (2006). Estimating student retention and degree-completion time: Decision trees and neural networks vis-à-vis regression. *New directions for institutional research*, 2006(131), 17-33.
- Jadrić, M., Garača, Ž., & Čukušić, M. (2010). Student dropout analysis with application of data mining methods. *Management: Journal of Contemporary Management Issues*, 15(1), 31-46.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: with applications in R* (Vol. 103). Springer Science & Business Media.

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

- John, G. H., Kohavi, R., & Pflieger, K. (1994). Irrelevant features and the subset selection problem. In *Machine learning: proceedings of the eleventh international conference* (pp. 121-129).
- Kuhn, M. (2017). caret: Classification and Regression Training. R package version 6.0-76. <https://CRAN.R-project.org/package=caret>
- Leitner, P., Khalil, M., & Ebner, M. (2017). Learning Analytics in Higher Education—A Literature Review. *Learning Analytics: Fundamentals, Applications, and Trends*, 1-23.
- Lin, J. J., Imbrie, P. K., & Reid, K. J. (2009). Student retention modelling: An evaluation of different methods and their impact on prediction results. *Research in Engineering Education Symposium*, 1-6.
- Lin, S. H. (2012). Data mining for student retention management. *Journal of Computing Sciences in Colleges*, 27(4), 92-99.
- McFarland, J., Hussar, B., de Brey, C., Snyder, T., Wang, X., WilkinsonFlicker, S., Gebrekristos, S., Zhang, J., Rathbun, A., Barmer, A., Bullock Mann, F., and Hinz, S. (2017). The condition of education 2017. *National Center for Education Statistics*, NCES 2017144.
- Murtaugh, P. A., Burns, L. D., & Schuster, J. (1999). Predicting the retention of university students. *Research in higher education*, 40(3), 355-371.
- Nandeshwar, A., Menzies, T., & Nelson, A. (2011). Learning patterns of university student retention. *Expert Systems with Applications*, 38(12), 14984-14996.

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

National Center for Education Statistics. (2010). *What is the CIP?* Retrieved from

<https://nces.ed.gov/ipeds/cipcode>

National Student Clearinghouse. (2017). *NCS Factsheet-0117*. Retrieved from

<http://studentclearinghouse.info/onestop/wp-content/uploads/NSCFactSheet.pdf>

National University. (2017). *For the greater*. Retrieved from

<http://www.nu.edu/forthegreater.html>

NU Institutional Research. (2017). *Student Success Metrics*. National University, Office of Institutional Research. La Jolla, CA: Internal Report. Retrieved June 15, 2016

Park, J. H., & Choi, H. J. (2009). Factors influencing adult learners' decision to drop out or persist in online learning. *Educational Technology & Society*, 12(4), 207-217.

R Core Team (2017). R: A language and environment for statistical computing. R

Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

Salazar, A., Gosalbez, J., Bosch, I., Miralles, R., & Vergara, L. (2004). A case study of knowledge discovery on academic achievement, student desertion and student retention. In *Information Technology: Research and Education, 2004. ITRE 2004. 2nd International Conference on* (pp. 150-154). IEEE.

Sousa, T. (2015). Student Retention is More Important Than Ever [Web log post]. *Higher Ed Live*. Retrieved from <http://higheredlive.com/3-reasons-student-retention-is-more-important-than-ever/>

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

- Sparkman, L., Maulding, W., & Roberts, J. (2012). Non-cognitive predictors of student success in college. *College Student Journal*, 46(3), 642-652.
- Stage, F. K. (1989). Motivation, academic and social integration, and the early dropout. *American Educational Research Journal*, 26(3), 385-402.
- Sujitparapitaya, S. (2006). Considering student mobility in retention outcomes. *New Directions for Institutional Research*, 2006(131), 35-51.
- Sutton, S. C., & Nora, A. (2008). An exploration of college persistence for students enrolled in web-enhanced courses: A multivariate analytic approach. *Journal of College Student Retention: Research, Theory & Practice*, 10(1), 21-37.
- Tinto, V. (1975). Dropout from higher education: A theoretical synthesis of recent research. *Review of educational research*, 45(1), 89-125.
- Thammasiri, D., Delen, D., Meesad, P., & Kasap, N. (2014). A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition. *Expert Systems with Applications*, 41(2), 321-330.
- United States Internal Revenue Service. (2014). *SOI Tax Stats – Individual Income Tax Statistics – 2014 ZIP Code Data (SOI)* [Data file]. Retrieved from <https://www.irs.gov/uac/soi-tax-stats-individual-income-tax-statistics-zip-code-data-soi>
- Veenstra, C. P., Dey, E. L., & Herrin, G. D. (2009). A Model for Freshman Engineering Retention. *Advances in Engineering Education*, 1(3), n3.

MACHINE LEARNING TECHNIQUES AND STUDENT DROPOUT

- Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.
- Yadav, S. K., Bharadwaj, B., & Pal, S. (2012). Mining education data to predict student's retention: A comparative study. *International Journal of Computer Science and Information Security*, 10(2), 113-117.
- Yu, C. H., DiGangi, S. A., Jannasch-Pennell, A., Lo, W., & Kaprolet, C. (2007). A Data-Mining Approach to Differentiate Predictors of Retention. *Online Submission*.
- Yu, C. H., DiGangi, S., Jannasch-Pennell, A., & Kaprolet, C. (2010). A data mining approach for identifying predictors of student retention from sophomore to junior year. *Journal of Data Science*, 8(2), 307-325.
- Zhang, G., Anderson, T. J., Ohland, M. W., & Thorndyke, B. R. (2004). Identifying Factors Influencing Engineering Student Graduation: A Longitudinal and Cross-Institutional Study. *Journal of Engineering education*, 93(4), 313-320.
- Zhang, Y., Oussena, S., Clark, T., and Hyensook, K. (2010) Using data mining to improve student retention in HE: a case study. In: ICEIS - 12th International Conference on Enterprise Information Systems, 2010., 8-12 June, Portugal.