May 2018

Machine Learning Applications in Graduation Prediction at the University of Nevada, Las Vegas

Elliott Collin Ploutz
philosopher.scholar@gmail.com

Repository Citation
Ploutz, Elliott Collin, "Machine Learning Applications in Graduation Prediction at the University of Nevada, Las Vegas" (2018). *UNLV Theses, Dissertations, Professional Papers, and Capstones*. 3309.
https://digitalscholarship.unlv.edu/thesesdissertations/3309

MACHINE LEARNING APPLICATIONS IN GRADUATION PREDICTION

AT THE UNIVERSITY OF NEVADA, LAS VEGAS

by

Elliott Collin Ploutz

Bachelor of Arts – Philosophy
University of Nevada, Las Vegas
2012

Bachelor of Science – Computer Science
University of Nevada, Las Vegas
2016

A thesis submitted in partial fulfillment of
the requirements for the

Master of Science in Computer Science

Department of Computer Science
Howard R. Hughes College of Engineering
The Graduate College

University of Nevada, Las Vegas
May 2018

UNLV GRADUATE COLLEGE

Thesis Approval

The Graduate College
The University of Nevada, Las Vegas

April 10, 2018

This thesis prepared by

Elliott Collin Ploutz

entitled

Machine Learning Applications in Graduation Prediction at the University of Nevada, Las Vegas

is approved in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science
Department of Computer Science

Fatma Nasoz, Ph.D.
Examination Committee Chair

Justin Zhan, Ph.D.
Examination Committee Member

Evangelo Yfantis, Ph.D.
Examination Committee Member

Matthew Bernacki, Ph.D.
Graduate College Faculty Representative

Kathryn Hausbeck Korgan, Ph.D.
Graduate College Interim Dean

Abstract

Graduation rates of four-year institutions are an increasingly important metric to incoming students and for ranking universities. To increase completion rates, universities must analyze available student data to understand trends and factors leading to graduation. Using predictive modeling, incoming students can be assessed as to their likelihood of completing a degree. If students are predicted to be most likely to drop out, interventions can be enacted to increase retention and completion rates.

At the University of Nevada, Las Vegas (UNLV), four-year graduation rates are 15% and six-year graduation rates are 39%. To improve these rates, we have gathered seven years worth of data on UNLV students who began in the fall 2010 semester or later up to the summer of 2017 which includes information from admissions applications, financial aid, and first year academic performance. The student group which is reported federally are first-time, full-time freshmen beginning in the summer or fall. Our data set includes all freshmen and transfer students within the time frame who meet our criteria. We applied data analysis and visualization techniques to understand and interpret this data set of 16,074 student profiles for actionable results by higher education staff and faculty. Predictive modeling such as logistic regression, decision trees, support vector machines, and neural networks are applied to predict whether a student will graduate. In this analysis, decision trees give the best performance.

Acknowledgements

"I am greatly indebted to many people for their help, more than I can list here. While I try to take hold of any opportunities that come my way, those opportunities are often curated and created by the kind and caring people I have met. For those who have most directly contributed to my thesis, I would like to especially thank Carrie Trentham, Rebecca Lorig, Daisy Duarte, and Kivanc Oner. For their years of commitment to my education as well as this thesis, I am greatly indebted to my committee members, Dr. Yfantis, Dr. Zhan, Dr. Bernacki, and my advisor Dr. Fatma Nasoz.

I would also like to thank my arms for being by my side, my legs for supporting me, and my fingers because I can always count on them."

Elliott Collin Ploutz

University of Nevada, Las Vegas
May 2018

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

One of the essential goals of universities is to aid student success. What student success means can change depending on the context. Student success could mean a high grade point average (GPA), self-assessed confidence in abilities, success in a specific course, graduation within a time-frame, or more specific milestones. With the available student data universities have at their disposal, the next logical step is to apply analytical techniques and statistical models to interpret the factors leading to, and the prediction of, student success.

A more objective definition for student success is degree completion, graduation. At the undergraduate level, bachelor degree graduation rates for American universities are particularly considered with six-year completion rates being reported to the U.S. Department of Education's National Center for Education Statistics (NCES). The NCES maintains the Integrated Postsecondary Education Data System (IPEDS) which compiles surveys from all institutions that receive federal aid. The working definition for this thesis derives from the IPEDS definition of completion within 150% of the expected time for graduation, i.e., within six years for four-year institutions [IPE17]. With analysis, universities can intervene on students who are likely to dropout and provide a fertile environment for those factors found to be conducive to graduation.

The ideal model would accurately classify incoming freshmen and transfer students with the likelihood of their success. At this stage, administrators can have the most potent effects in intervention and assistance. However, there are limitations on the data universities have access to. Gaevi et al. [GDRG16] identified three main types of data used in predictive models of academic success and retention, stored data, trace data, and a combination of the two.

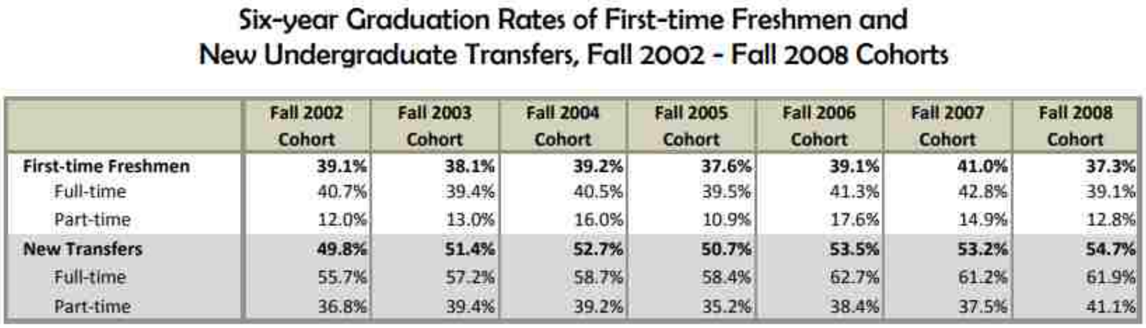
Stored data is all information provided to the school by the student, e.g., high school GPA, American College Testing (ACT)/Scholastic Aptitude Test (SAT) scores, biological sex, etc. Trace data or log data is data logged by Learning Management Systems (LMS). Online tools like class websites keep track of links clicked, time spent logged in, quizzes, forum posting, and other domain specific information. This data is later analyzed with educational data mining techniques for patterns and higher level information. Finally, combinations of the two types of data are used for a more complete picture. LMS data is typically used to predict success within a specific course rather than long term prediction of success within a major or graduation.

This problem is one area of the learning analytics and educational data mining fields. Learning analytics supports and optimizes the learning environment with methods applied to educational data sets [CLT+14]. The closely related field of educational data mining applies data mining, statistics, pattern recognition, and machine learning to automatically extract useful information from data generated by learners and learning environments. These research fields are robust and growing while only being recognized as an interdisciplinary field in the last few years. Papamitsiou & Economides [PE14] performed a systematic review of learning analytics research from 2008-2013 and found 209 mature articles, however only 40 met their inclusion criteria for key studies. One leader in the field, George Siemens, argues learning analytics has developed enough to be regarded as an emerging research field [Sie13]. While learning analytics is an important yet fledgling field, the use of trace data is beyond the scope of this thesis and left for future work.

1.1 Problem Description and Motivation

Given a set of student attributes, X, in the form of stored data such as application data, first year academic performance, and financial aid data, predict whether the student will graduate from the university (outcome = 1) or not (outcome = 0).

The specific student group that is tracked and reported nationally are full-time, first-time freshmen seeking a bachelor's degree who begin in the fall. These student groups are called cohorts. Students who begin in the summer are generally subsumed under the fall cohorts, however students starting in the spring go untracked. Transfer students are generally untracked at the national level. According to the NCES, six-year graduations rates for full-time, first-time freshmen seeking a bachelor's degree beginning in the fall of 2009 are at 59% nation wide for public institutions [IPE17]. As shown in the table below, the University of Nevada, Las Vegas' (UNLV's) graduation rates for first-time, full-time freshmen are well below the national average at about 39%.



Six-year Graduation Rates of First-time Freshmen and New Undergraduate Transfers, Fall 2002 - Fall 2008 Cohorts

	Fall 2002 Cohort	Fall 2003 Cohort	Fall 2004 Cohort	Fall 2005 Cohort	Fall 2006 Cohort	Fall 2007 Cohort	Fall 2008 Cohort
First-time Freshmen	**39.1%**	**38.1%**	**39.2%**	**37.6%**	**39.1%**	**41.0%**	**37.3%**
Full-time	40.7%	39.4%	40.5%	39.5%	41.3%	42.8%	39.1%
Part-time	12.0%	13.0%	16.0%	10.9%	17.6%	14.9%	12.8%
New Transfers	**49.8%**	**51.4%**	**52.7%**	**50.7%**	**53.5%**	**53.2%**	**54.7%**
Full-time	55.7%	57.2%	58.7%	58.4%	62.7%	61.2%	61.9%
Part-time	36.8%	39.4%	39.2%	35.2%	38.4%	37.5%	41.1%

Figure 1.1: Graduation rates as reported by [UNL14]

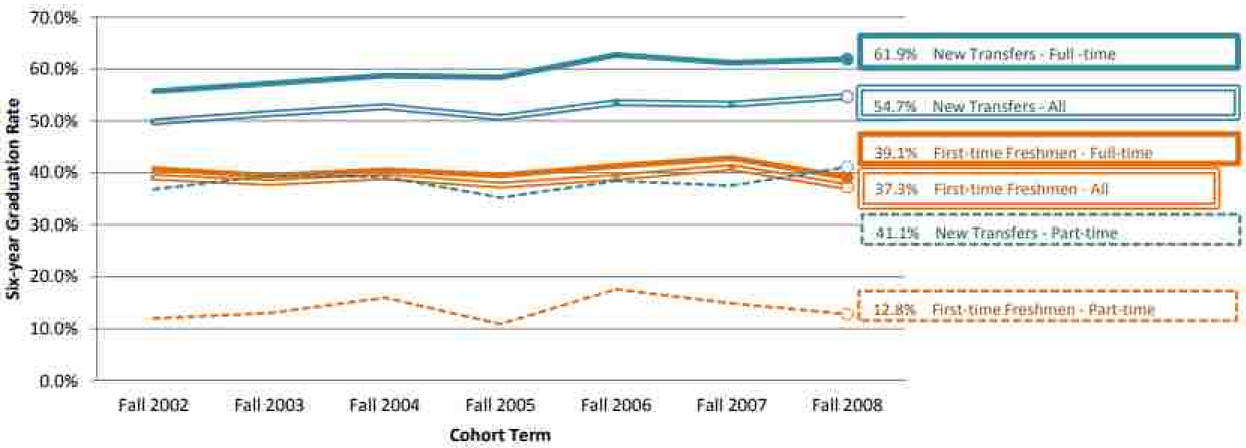


Figure 1.2: Graduation rates visualized by [UNL14]

Graduation rates are not only important to potential students, but also play a role in university rankings and funding opportunities. While full-time, first-year freshmen cohorts could be focused on to the exclusion of other groups, UNLV is committed to serving all of its student population. To that end, freshmen who began in the spring and transfer students at the undergraduate level are included in this analysis.

What makes this thesis particularly interesting is UNLV's ranking as the number one most ethnically diverse campus in the nation according to the 2017 listing by the U.S. News & World Report [div17]. Projections based on 2014 census data show a growing trend toward diversity such that "...by 2044, more than half of all Americans are projected to belong to a minority group (any group other than non-Hispanic White alone); and by 2060, nearly one in five of the nation's total population is projected to be foreign born," [CO15]. Forward looking institutions can refer to this work in expectation of changing student demographics.

Chapter 2

Literature Review

2.1 Literature Review

Many universities and institutional researchers have applied a wealth of techniques to student success. There are two main aspects, interpretation and prediction. Interpretation allows humans to understand the important factors leading to graduation, for example access to tutoring centers at key courses could lead to higher graduation rates. Prediction uses statistical models to estimate the number, rate, or range of students who will graduate within a specified amount of time. Typically, the more complex a model is the greater the degree of accuracy it can have. However, the more complexity a model has the less interpretable it is for humans.

2.1.1 Critique of Generalized Models Across Universities

The idea of a generalized model, one that can be applied to different universities, departments, and levels, is highly appealing. Much work has focused on this generalization [JML$^+$14] [OO16]. However, Gaevi et al. [GDRG16] have rebutted the idea of this ”one size fits all” model.

Gaevi argues generalized models and models which rely heavily on trace data are largely atheoretic with regards to learning theories, so the interpretability of results and interventions to be taken cannot draw from the long history of learning theory research. Variability in the predictive results of the learning analytics field could suggest contextual differences within each discipline resist a generalized model. Conijn et al. [CSKM17] analyzed 17 blended courses with N = 4,989 students at single institution using a Moodle LMS. Even though all the courses were from a single institution, they found high variance in prediction accuracy, suggesting the portability of models across courses is low.

2.1.2 Interpretation

Decision Trees

Broadly speaking, decision trees are tree-like, directed graphs. Beginning from the root, the tree splits up instances of the input where each level of the tree from the root specifies a narrower swath of the data. Decision trees can be used for classification or regression. At each branching node, a subset of the data is defined by a rule. This rule is of the form if and $else - if$ where the previous parent node rule is joined to the child rule by an *and* statement. At the bottom-most leaf node, the joined rule classifies a given instance or returns a predicted value.

In a comparison of decision trees, artificial neural networks, and multiple logistic regression by Serge Herzog [Her06], graduation prediction was based on four classes, less than three years, four years, five years, and six or more years, which Herzog claims ”generates a more balanced outcome in the dependent variable and ensures convergence in the regression model.” The decision trees were based on three-rule induction for the C&RT, CHAID-based, and C5.0 models. For graduation prediction, 15,457 undergraduate student profiles from 1995 to summer 2005 were used with 79 attributes. Variables used are listed in the appendix of the work. Missing values were imputed by a general linear model (earned-to-attempted credits) or by multiple regression (total campus-based credits). Mean value substitution was used for ACT scores.

Models were trained with and without transfer student data which lead to interesting changes in accuracy, a significant improvement in all models. Regarding the prediction of six or more years for graduation, the C5.0 tree performed the best of all tested models giving an accuracy of 93% which is 11 percentage points higher than the baseline logistic regression model. All models performed well in predicting the smaller portion of students who would graduate within three years.

Decision Sets

Lakkaraju et al. [LBL16] developed decision sets to bridge the gap between interpetability and high accuracy. Like decision trees and lists, decision sets provide a set of human readable rules to classify a given instance. Unlike trees and lists, the rules overlap between classes as little as possible and are non-hierarchical. Each rule within a set is independent of the other rules. Using association rule mining measures, the rules are assessed for accuracy with recall and precision. Interpretability is measured by conciseness, overlap, and coverage. The authors also proved solutions will be near optimal, within a range of at least $2/5^{ths}$ of the global, optimal solution.

The researchers applied their technique to the prediction of high school student graduation. With data gathered from grades 6-8 on nearly 21,000 students set to graduate high school in 2012 and 2013, the decision set technique showed accuracy levels comparable to other state of the art methods such as Bayesian decision lists and classification based on associations. Results were also compared to standard models such as logistic regression, random forests, gradient boosting, and decision trees. At this time, there appears to be no published work using decision sets for collegiate, graduation prediction.

2.1.3 Prediction

Machine learning techniques discussed here can be divided into two types, supervised and unsupervised. Supervised models are trained on labeled data. For our problem, this would mean labeling student profiles as successful (degree completed within a specified time-frame) or unsuccessful. Unsupervised models are label agnostic. Instead they try to autonomously learn new representations of the data that reveal hidden patterns.

Logistic Regression

Logistic regression is the most common technique [GDRG16] [BS12] [LBD^{+}12] [Pal13] due to how easily academic success corresponds to classifications like letter grades and graduation (on-track, at-risk, failing). It serves as a useful baseline to compare more advanced models.

Zhang et al. [ZAOT04] successfully applied multiple-logistic regression models to nine institutions. The large scale project analyzed student data from 1987 to 2002 from engineering disciplines with over 87,167 student records to evaluate pre-existing factors that most contribute to graduation. High school GPA and quantitative SAT scores were impactful for all models and all institutions. Gender, ethnicity, verbal SAT scores, and citizenship had a significant impact on graduation, but the impact for each attribute varied among institutions for the engineering students.

Regression analysis by Engle & O'Brien [EO07] across 20 institutions showed differing student factors and academic support change the most important attributes for models predicting graduation. A model trained on one institutional data set could have greatly reduced accuracy at another four-year institution.

Support Vector Machines

Support vector machines (SVMs) are supervised learning models with the goal of classification or regression. For classification, the SVM takes in data points and attempts to best separate them to their proper class in the attribute space, an N-dimensional space where N is the number of attributes. To do this, a hyperplane based on the data points is constructed, if it exists, such that the margin, the distance between a data point of any class, is maximized. Typically a maximized margin corresponds to a lower error in generalization [CM04].

Barker et al. [BTR04] used SVMs in graduation prediction for four-year universities. With 59 attributes forming a student profile, their results show a best-case prediction rate of 66.1% on the training set and 63.4% on the test set.

Artificial Neural Networks

Artificial neural networks are biologically inspired cognitive models. They consist of connected *neurons*, nodes, which form an input layer, hidden layer, and output layer. The input layer would in our case take in a student profile, each attribute corresponding to an input node. From that node, the information is sent to the next layer multiplied by a weight specific to the receiving node. The receiving node then processes the input from the previous nodes with an activation function. If the input surpasses a certain threshold, the neuron *fires* and passes weighted information to the next layer. The process is continued for the output layer, where the output of those nodes is the computation of the network to be interpreted as the result.

Another way of representing artificial neural networks would be with a directed, weighted graph. The learning aspect comes from fine tuning the weights to approximate the proper function for the given problem. It has long been shown that feedforward neural networks are a class of universal approximators [HSW89], meaning they can approximate any continuous function. This is a general and useful approach for classification tasks, particularly tasks that require a non-linear function.

Karamouzis & Vrettos [KV08] developed a three-layered perceptron, training it with backpropagation and tanh activation functions. The output of the model is two nodes, one for successful predicted graduation and one for failure. Their study was based around a two-year college. The data consisted of twelve attributes for 1,407 community college student profiles. The training set consisted of 1,100 profiles while the test set was 307. Their working definition of successful graduation from a two-year program is completion within three years. Their accuracy rates were 72%

for the training set and 68% for the test set, validating their model within 6,000 epochs.

An earlier attempt by Barker et al. [BTR04] dealt with four-year universities. Along with SVMs, their neural networks had a best-case prediction of 67.5% on the training set and 63.4% on the test set with 59 student attributes.

In Herzog's [Her06] comparison of decision trees and artificial neural networks, three types of backpropagation neural network topologies were used, simple topology, multitopology, and three-hidden-layer pruned. When new and transfer students were grouped together, the three-hidden-layers pruned model had 50% improvement in accuracy over the baseline regression model. Interestingly, sensitivity analysis on neural networks revealed influential attributes for the model where a similar sensitivity analysis on logistic regression showed little difference between variables.

Oladokun et al. [OACO08] developed a model for predicting student performance at the university level for an engineering course using just ten attributes for a student profile. They provide an example of transforming the input data into a format acceptable for neural networks. The output class were of three types, good, average, and poor performance. The researchers used a multilayer perceptron with two hidden layers and five nodes per layer. Out of a total of 112 student records, 62 were used for training, 34 as the testing set, and 16 cross validation resulting in an overall accuracy of 74% where mean squared error is used to assess performance.

Hybrid Approach

Hybrid approaches use two or more models and aggregate the information into a whole. Oztekin et al. [OO16] used a hybrid approach involving decision trees, support vector machines, and artificial neural networks. They were also able to determine which features or attributes were important predictors based on sensitivity analysis, giving a level of interpretability.

Their hybrid approach consisted of training and testing the three analysis techniques using tenfold cross-validation. The model is then assessed. If it performs satisfactorily, the model is used in the next stage of information fusion-based sensitivity analyses, otherwise the model is discarded. The performance was measured by tenfold cross-validation, confusion matrices (accuracy, sensitivity, and specificity), and information fusion-based sensitivity analyses. Their derived equation for fused sensitivity analysis is defined as

$$\hat{y}_{fused} = \textstyle\sum_{i=1}^{m} \omega_i f_i = \omega_1 f_1 + \omega_2 f_2 + ... + \omega_m f_m$$

Where ω_i refers to the weight of the individual model and f_i is the model. The higher the performance of a model, the greater the weight. The weights are assumed to be normalized, so $\sum_{i=1}^{m} \omega_i = 1$. Finally, the ranking of the given attributes can be determined using the normalized sensitivity measure as

$$S_{n(fused)} = \sum_{i=1}^{m} \omega_i S_{in} = \omega_1 S_{1n} + \omega_2 S_{2n} + ... + \omega_m S_{mn}$$

Where S_{in} refers to the ith model and the nth attribute. The three models appear to be standard with no particular variants specific to the data.

It is important to note the student data that was used in the experiment. The researchers obtained the confidential and private data through official acquisition from a United States university. Any records missing data, e.g., no SAT scores, were dropped. The final data set used contained 30 input attributes and 1,204 records total.

Interestingly, using ten-fold cross-validation the SVM model performed the best, followed by the decision tree and artificial neural network. Logistic regression was not used due to its poor results, achieving an accuracy of only 50.18%. Accuracy ranged from 71.56% to 77.71%. Through sensitivity analysis, the most critical factors in predicting graduation rates were fall-term GPA, the high school the student attended, and living on-campus or off-campus. Students were more likely to graduate within six years if they lived on campus. The least critical factors were ethnicity, work-study, and if a student applied for financial aid.

2.2 Discussion

Given the high degree of diversity at UNLV, it is possible that models of community colleges may be more applicable to student success than university models. Community colleges tend to have higher diversity which leads to different factors influencing success [HC17]. Further research is needed on highly diverse, university institutions.

Care should be taken in the selection of training, test, and validation data sets. At many institutions, the student body is not neatly balanced, i.e., there tends to be more students academically successful or at risk. This can skew the predictive power of the model [LBD$^+$12]. Roughly equal amounts of students from the output classes of the model should be used to better assess the generality of the model. This is known as the *class imabalance problem* and is further detailed in Thai-Nghe et al. [TNBST09]. Typical ways of dealing with this issue are over-sampling and under-sampling to modify class distributions. The authors argue for three advanced methods: using Synthetic Minority Over-sampling Technique (SMOTE), cost-sensitive learning (CSL), and a combination of the two.

With the current aim of predicting graduation rates as a whole at UNLV, adding trace data would be unnecessarily cumbersome. After building a campus wide model, we can begin to compare this general model to discipline specific models. From there, we may be able to find the critical courses that predict success and graduation within the discipline. At that time, we can determine if trace data would be useful for evaluating course specific success, and a grounding in learning theory like [GDRG16] and [CSKM17] could prove useful to further extract variables and have meaningful results.

Towards the goal of early prediction for incoming students prior to their start at the university, we can look at past performance as whole in terms of cumulative GPA and credits taken, ignoring trace data at the outset. It is assumed that students will have no incoming trace data because they have not yet taken classes, so it is safely ignored.

In the cases where students have taken college level courses, we would likely not have access to the trace data. However, transfer students pose unique problems of their own. Some courses are not transferable or international students have an entirely different grading scale. One method as in [Her06] is to evaluate models with and without transfer students.

A pressing issue in the field is a standardization of a benchmark data set that researchers can use to compare models. This data set would need to be both anonymous, robust, and class balanced. As such, it would be more artificial than realistic. However, the current approach is that most researchers work only on their own data with some logistic regression model as their base or benchmark to compare to. This makes comparative experiments between researchers very difficult. However, as discussed [EO07], portability of models is unlikely to be accurate with different student demographics. One solution may be to have multiple benchmark data sets for different types of institutions. Of course, there are also hard to define differences between institutions like academic support and courses taught by adjunct faculty.

A separate motivation for not using public, benchmark data sets would be that they are far abstract from the institutional data which the researchers likely serve as current models and data vary wildly from one institution to the next. Attributes for modeling on stored data range from 10 to 80 with data samples ranging from 150 to 80,000.

Chapter 3

Data Description

Gathering the data proved to be one of the most difficult aspects of this thesis. UNLV has thousands of data tables stored in relational databases. Each table can have between two and a few hundred attributes. Simply pulling all known data on a student would quickly become impractical, so some level of selection is needed which causes a degree of bias in the analysis. Based on the reviewed literature and my interactions with domain experts, we gathered student data from their submitted admission applications, financial aid data, and their first year academic performance.

3.1 Data Gathering Process

The two main methods for extracting data are from UNLV Analytics and MyUNLV. UNLV Analytics is a licensed Oracle Business Intelligence product built specifically for UNLV student data. MyUNLV is an Oracle PeopleSoft licensed product with an interface to the databases. Care was taken to only collect data according to official census reporting dates to ensure accuracy. These systems were put in place for the fall 2010 semester and later, so previous records are excluded from this analysis due to their unreliability.

The criteria for a selected student is that they begin their work in the fall of 2010 or later and graduated during or before the summer 2017 semester. As stated previously, the range limit for a student to graduate is within six years of their starting term, so to be certain a student would not graduate and be considered non-graduated, the cut off mark for non-graduates are those students from the fall 2010 to summer 2011. The queries resulted in a total of 16,074 students with 12,677 graduating students and 3,397 non-graduating students, giving a class balance of 79% to 21%.

If we were to try to enlarge the search of non-graduates to look for students who had stopped enrolling in the next semester or took 3 years off, we cannot be certain this student did not take classes at another university only to transfer back to UNLV to graduate or that they returned to graduate after a break. With our strict range, we can be sure these students did not graduate at UNLV within the time frame.

From UNLV Analytics, a table of student data meeting the given criteria was taken from enrollment, admissions, and graduation relational databases. A query is automatically generated in SQL based on the constraints. The enrollment and admissions tables each needed to be downloaded for each semester and concatenated. Students meeting the criteria for non-graduates were retained in admissions along with graduating students found by left outer join on graduation and admissions using the NumPy and Pandas libraries in Python.

Four variables were generated, the starting term, second term, parents' highest education level, and 'startedSummer.' The starting term was given as the admission term for when a student had applied and enrolled in classes for the term. The second term was the spring or fall term following the starting term. For students who began in the summer, they were assigned to the next fall term and the startedSummer flag was set to one and zero if otherwise. The student table was then left outer joined with the enrollment table, giving the term information for the first and second terms. The financial aid tables carry granular information on the education level of each parent as reported by the student. These categories were consolidated into simpler groups, e.g., the categories master's degree, some graduate school, doctorate (professional), doctorate (academic), and post-doctorate became 'graduate school.' The highest education level achieved by either parent was taken as the attribute.

From MyUNLV, queries were generated to find the term GPA and cumulative GPA for each semester along with financial aid data, loans, scholarships, and grants. In similar methods as stated above, the GPA data for each term and financial aid information were left outer joined with the student data. There were many missing values because of these left out joins using student records. Students are not required to file for the Free Application for Federal Student Aid (FAFSA), so records will be empty. Similarly, term records are missing if a student did not enroll in that term.

3.2 Class Imbalance

Class imbalance occurs when one or more classes dominate the data. This becomes a serious issue for two main reasons, misleading evaluation metrics for models and an over-emphasis in models learning the majority class while placing less value on the minority class. In this case, we see the student data has a class imbalance of 79% to 21% or roughly four to one. If we had a simple model which predicted that every student would graduate, the accuracy of the model would be 79% and seemingly good. Of course this metric would be misguiding and useless. See section 4.1 for how this impacts the evaluation of models.

There are many complex techniques to combat class imbalance such as over-sampling of the minority class, under-sampling of the majority class, and combinations of the two. There are also class balancing weights which can be applied to the penalty of learning models where the classes are weighted proportionally to their size so the model ideally does not favor either class. Preliminary results with hyper-parameter searches in learning models showed that class balance weights rarely outperformed not having weights at all. When considering under or over-sampling, the small amount of data lends itself to over-sampling the minority class to equal the majority class which causes the class balance weights to become equal to not having class balances at all.

3.2.1 Over-Sampling Techniques

Over-sampling techniques attempt to balance the classes by either generating or using samples from the minority classes until they equal the size of the majority class. In this case, the training set contains 8,834 positive (graduating) samples and 2,417 negative (non-graduated) samples. Over-sampling is only used on the training set so that the test set still serves as an accurate measure of how well the model will generalize to new, incoming students. These techniques are applied after the data has been transformed, scaled, and imputed.

Random Over-Sampling

Random over-sampling is a simple technique that duplicates a sample with replacement at random from the minority class until the size matches the majority class.

SMOTE

The Synthetic Minority Over-Sampling Technique first described by Chawla et al. [BCHK11] creates new samples of the minority class by generating new attributes based on the surrounding samples in feature (attribute) space. Each student is a point in the multidimensional attribute space where each attribute forms a plane. To give a clear example of how a GPA attribute could be generated, we select a random student from the non-graduate minority class and its k nearest neighbors in this attribute space of the non-graduates. We find the difference between the selected student's GPA and one of its k nearest neighbors, say 2.9 - 3.1 which gives a difference of 0.2. This difference is then multiplied by a random value between zero and one, e.g., 0.5, which gives 0.1. This value is then added to the selected student's GPA which gives $2.9 + 0.1 = 3.0$, an attribute that lies between the two attributes in this case. The synthetic attribute is set to be the GPA of the synthetic student sample. This process continues for all attributes and the synthetic student is added to the minority class. The number of nearest neighbors, k, is dependent on the amount of over-sampling needed. In our case, the minority class needs to be over-sampled at 365%.

ADASYN

Adaptive synthetic (ADASYN) over-sampling is a technique developed by He et al. [HBGL08] which attempts to put more focus on hard-to-classify samples. However, by generating samples closer to the more difficult students for classification, ADASYN is sensitive to outliers. The algorithm generates minority samples similar to SMOTE, but it generates the number of samples according to the ratio of the class distribution. In our case, ADASYN generates 6,743 synthetic students for the non-graduated class for a total of 9,160 samples. This brings the class percentage to 50.9% for non-graduated and 49.1% for graduated students.

3.3 Data Cleaning

Data cleaning refers to detecting and correcting student records which have inaccurate or corrupt values resulting in ”dirty” data. Sometimes this is caused simply by values being incorrectly manually entered by a worker, e.g., a GPA of 40.1 instead of 4.01. Such records must be verified and corrected, or when the data cannot be verified, the record can be dropped from the analysis. A different issue is when two sets of data have the same information but separate representations. An example in this case are international exchange students who have a GPA model that is on the scale of 0-100. In this case, the GPA is scaled to within the range of 0.0-4.0 to retain the information.

The data is largely clean due to its census level of reporting at the federal level. Many of the attributes which needed cleaning were numerical in nature and the few dirty records could be manually verified by comparing a student's record in MyUNLV. Examples are an age of zero or a GPA of 101 which can be easily spotted in the exploratory phase via visualizations such as histogram plots. For handling missing values, see section 3.7.

3.4 Rationale for Attribute Choices

As noted in [GE03], "The objective of variable selection is three-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data." Certain features may play no role in determining the outcome and only serve as disruptive noise for models. Most models' computational complexity is directly tied to the number of attributes.

In addition to model complexity is the explanation and visualization of the datas' most salient attributes. Determining the most impactful attributes lead to actionable interventions in students. There are three main algorithms for attribute selection, filters (univariate statistical tests), wrappers (a search of the best feature combinations), and embedded (variable selection as a process of training a model).

3.4.1 The Chi-Squared Data Test

A good test to determine if an attribute is independent of the outcome class is the chi-squared test. The null hypothesis assumes that the attribute and outcome class are independent and calculates a p-value. In the results, 61 of the attributes had a p-value less than 0.05 which is generally accepted to be statistically significant, thus for those attributes the null hypothesis can be rejected. The table is given in appendix B in table B.1.

3.4.2 Recursive Feature Elimination

Recursive feature elimination is a wrapper method which takes a model, in this case logistic regression, and repeatedly trains the model on an increasing and varied subset of the attributes. By analyzing the coefficients learned by the logistic regression model, the least weighted attributes are pruned and replaced by new attributes until all attributes have been tried. Using a logistic regression model with an L1 penalty, the following were ranked as the top 30 attributes as shown in table B.3 of appendix B. The L1 penalty ensures that every attribute has a non-zero coefficient. F1 score is used to optimize training (see subsection 4.1.4). The entire data set was used in this case without over-sampling techniques.

We can also use recursive feature elimination with cross-validation (see section 4.2 for more on cross-validation). Cross-validation allows us to see how training on these subsets of attributes may generalize to new students.

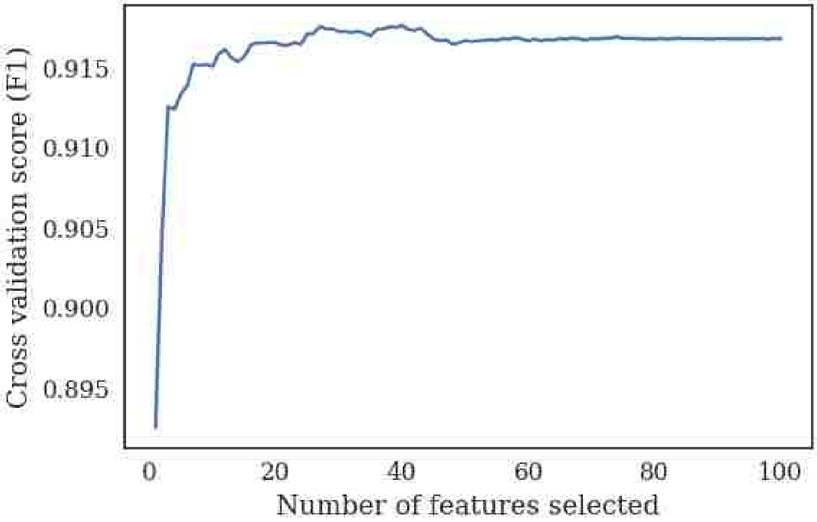


Figure 3.1: Recursive Feature Elimination with Cross Validation Using Logistic Regression - No Over Sampling

With no over-sampling, 40 is the number of optimal features returned.

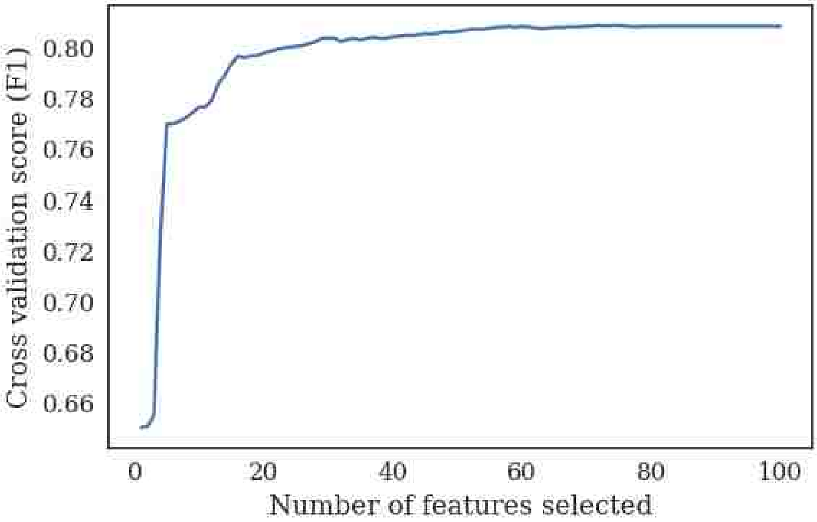


Figure 3.2: Recursive Feature Elimination with Cross Validation Using Logistic Regression - With Random Over Sampling

With random over-sampling, 72 is the number of optimal features returned. We see that the line is much smoother with more samples and class balancing, but the F1 score is about 10 percentage points less.

3.4.3 Decision Tree - Feature Importances

Decision trees split up the data by selecting an attribute and calculating a measure of information gain. This implementation uses Gini importance to find the optimal split of the data [PVG^{+}11]. By not giving a maximum depth and training on all data with all attributes, every attribute can be ranked by a score. The table is given in appendix B in table B.2.

3.5 Attributes Used For Prediction

Using a combination of the previous techniques and tests of mutual information, the attributes chosen are shown in table 3.1. Ideally, the features themselves would have a near zero Pearson correlation coefficient with each other, meaning they have unique information. Where attributes are highly correlated, greater than 85%, one is chosen and the other dropped, as is the case with the Millennium scholarship attribute for the first and second term. It is most likely that students will retain the scholarship into the second semester, so they are highly correlated. Attributes ending in x denote the first term and y denotes the second term.

3.5.1 Feature Scaling and Transformation

Categorical variables need to be transformed into an interpretable fashion for machine learning algorithms. One simple way is to create new columns corresponding to each possible category and use one-hot encoding. Take the attribute ’Academic Load x’ which corresponds to the amount of credits taken in the first semester with the categories ’Full-Time,’ ’Part-Time,’ and ’No Unit Load,’ then the table would be transformed as in table 3.2. The variable to be transformed would be removed from the data and the individual category variables would replace it, increasing the number of attributes by the number of categories.

Similarly to categorical features, binary features are transformed to one for positive and zero for negative. Numerical features are then scaled to a range of zero to one inclusive. At this time, no techniques are used to change numerical outliers to avoid a loss of information. Thus, all attributes are largely between zero and one for the machine learning algorithms.

Attribute
Admission Type
Gender
Millennium Scholar y
Taking Remedial x
Taking Remedial y
Western Undergraduate Exchange x
Non-Resident Alien
Nevada Resident
Honors College y
Term GPA x
Term GPA y
Cumulative GPA y
Cumulative Transfer GPA
Age
Cumulative Transfer GPA Credits
Core High School GPA
Unweighted High School GPA
loans
grants
Prmry EFC
Students Total Income
Parent Highest Ed Level Bachelor Level
Academic Load x Full-Time
Academic Load x No Unit Load
Academic Load y Full-Time
Academic Load y No Unit Load
SAP Not Meet
SAP Probation
SAP Meets SAP
IPEDS Race-Ethnicity Asian
IPEDS Race-Ethnicity Black or African American

Table 3.1: Selected attributes used for classification.

Academic Load x Full Time	Academic Load x Part Time	Academic Load x No Unit Load
1	0	0
0	1	0
0	0	1

Table 3.2: Example of a transformed categorical variable.

3.6 Visualizations

Visualizations are useful for getting a quick sense of the information contained in the data and the distribution. The trends between the two classes are often what would be expected. As shown in

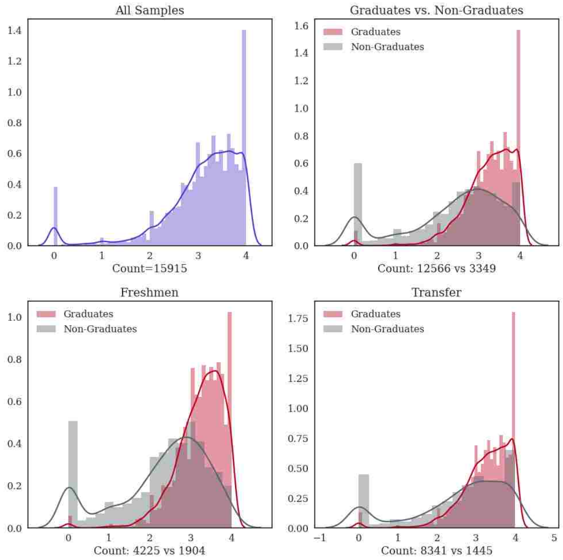


Figure 3.3: A histogram with kernel density estimation of first term GPA at UNLV.

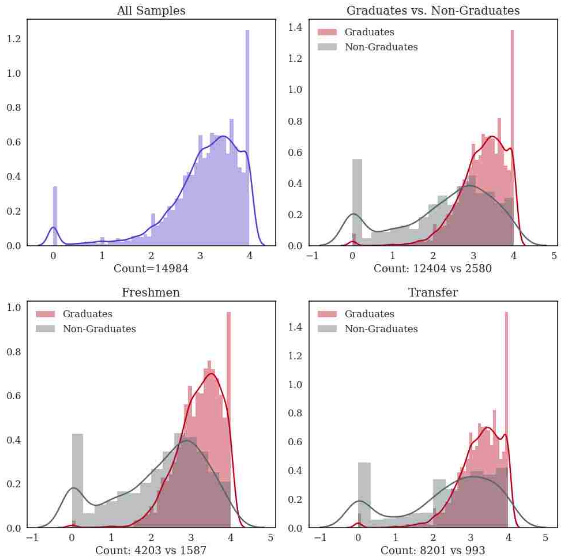


Figure 3.4: A histogram with kernel density estimation of second term GPA at UNLV.

figures 3.3 and 3.4, that graduate class has a higher mean and more narrow distribution than the non-graduate class. Interestingly, figures 3.5 and 3.6 show out-of-state students are more likely to be in the graduate class.

The satisfactory academic progress (SAP) attribute is likely to have high information due to its trying to measure academic progress directly. The requirements for this policy per semester are for undergraduate students to maintain above a 2.0 GPA, satisfactorily complete at least 70% of their attempted credits, and for students to complete their degrees within a credit limit of 186 credits. Failure to meet SAP results in financial aid being withheld by the university. Students who fail to

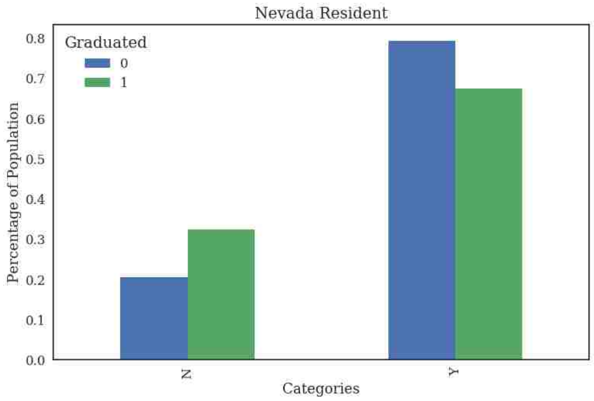


Figure 3.5: A normalized bar plot of Nevada residency.

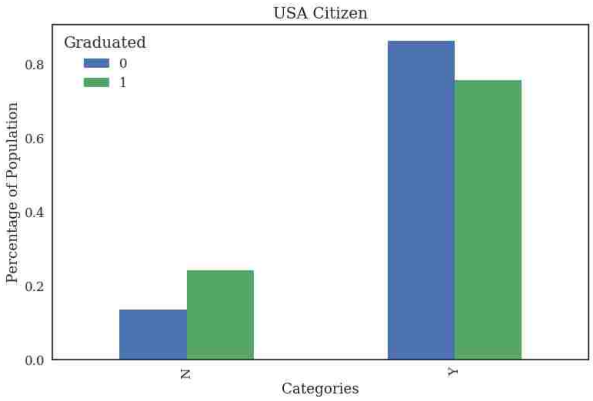


Figure 3.6: A normalized bar plot of US citizenship.

meet SAP must take bureaucratic steps to be in good standing with the university and continue. This likely contributes to struggling students not returning to take more classes. We can see the distribution in figure 3.7.

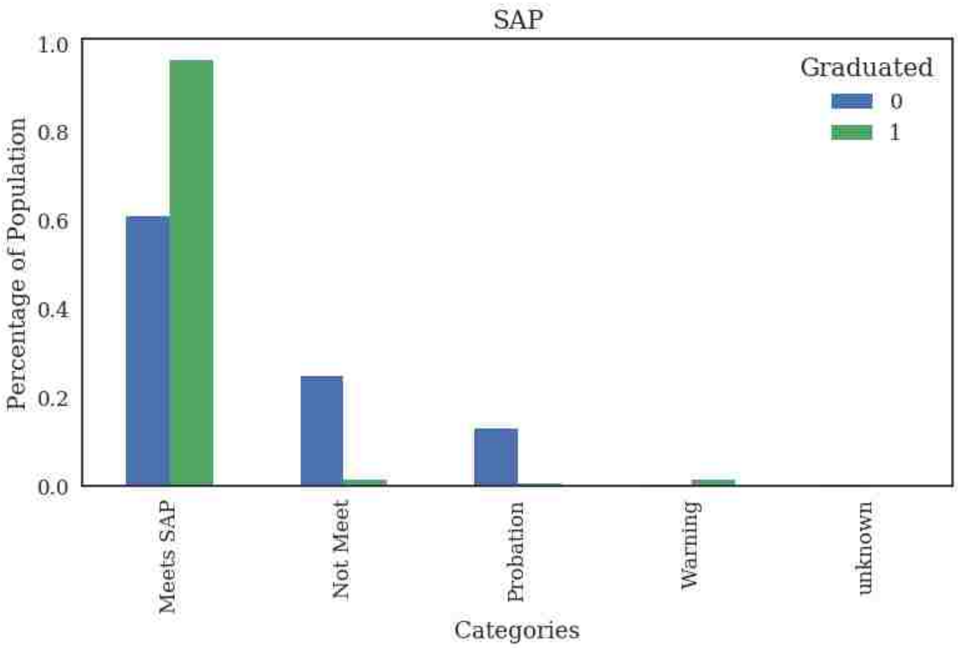


Figure 3.7: A normalized bar plot of the student academic progress attribute.

The Pearson correlation coefficient is shown in a heatmap in figure 3.8 for all the selected attributes. Ideally, a strong positive or negative correlation would indicate information related to the outcome variable.

3.6.1 Principal Component Analysis

Principal component analysis (PCA) is a technique to reduce the number of attributes without a significant loss in the variance of the attributes. PCA is in the class of dimensionality reduction techniques which finds a transformation of the data into a lower dimensional space. This lower dimension version of the data can be useful to train models more efficiently where a large number of attributes can slow the training or increase noise or bias. However, an important contribution is visualizing high dimensional data which is still an active area of research. By transforming the data into two or three-dimensional space, it can be visualized in traditional plots. This can help to detect clusters, outliers, and the general distribution of the data. This technique is applied after the data has been transformed, scaled, and imputed.

We can see from the three-dimensional and two-dimensional plots of figures 3.9 and 3.10 that

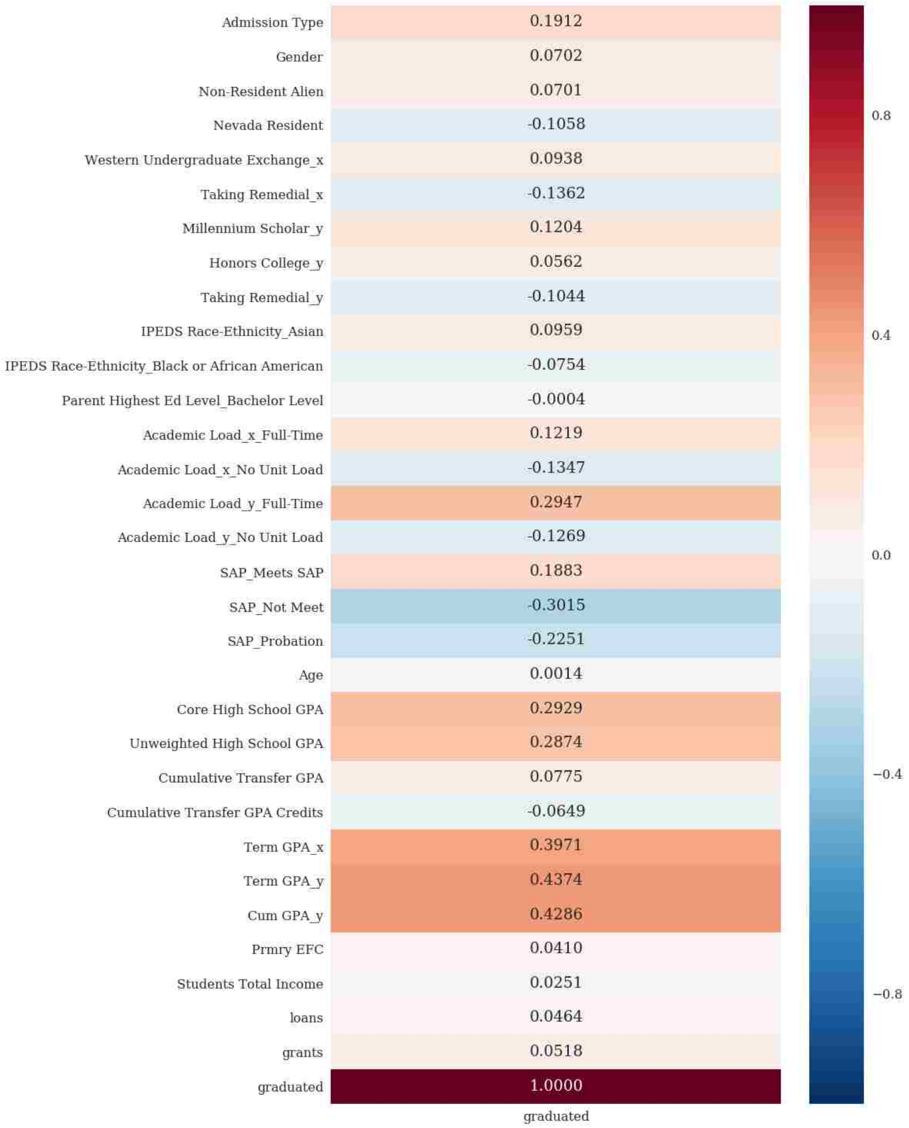


Figure 3.8: Correlation plot of selected features.

the majority class dominates the space. It is apparent that there are two major clusters in this space where all attributes of the data are used with PCA. Analyzing plots with just the selected attributes as in figures 3.11 and 3.12 shows there are many subgroups within the data and significant overlaps between the two classes. The two classes are visualized separately in figures 3.13 and 3.14

3.7 Imputation Method

Due to the left outer joins on the data, there are many missing values. Many machine learning models do not expect missing values in the data and will not properly predict for a sample. To run the models, some value must be given.

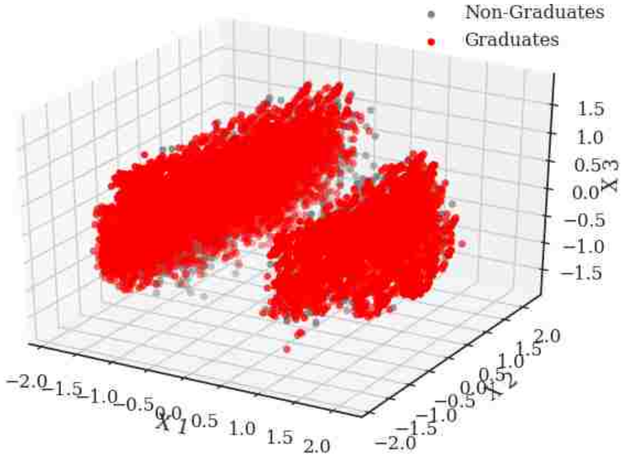


Figure 3.9: A transformation of the data to 3-dimensional space using PCA.

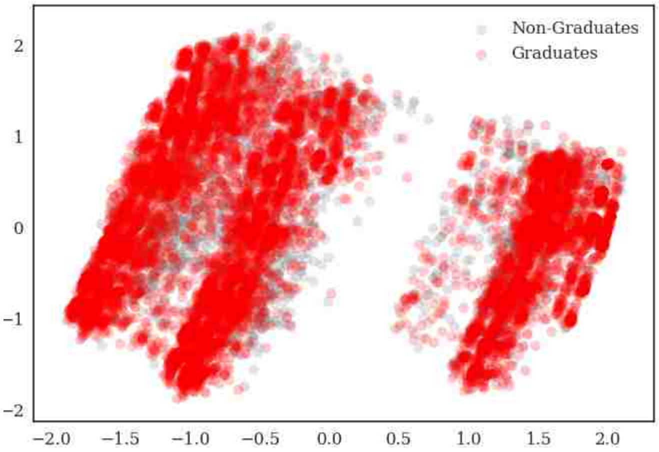


Figure 3.10: A transformation of the data to 2-dimensional space using PCA.

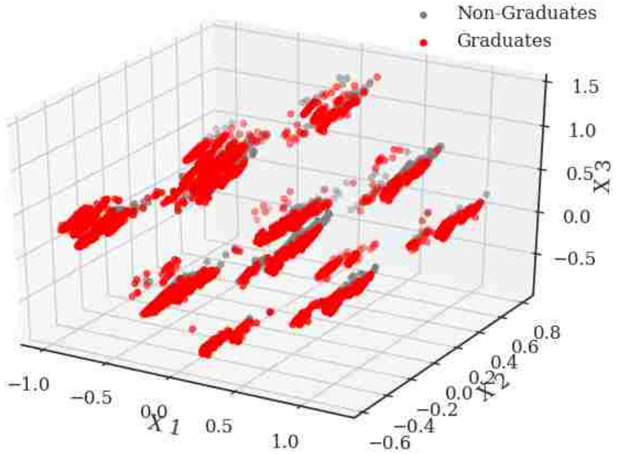


Figure 3.11: A transformation of the data to 3-dimensional space using PCA for the selected attributes.

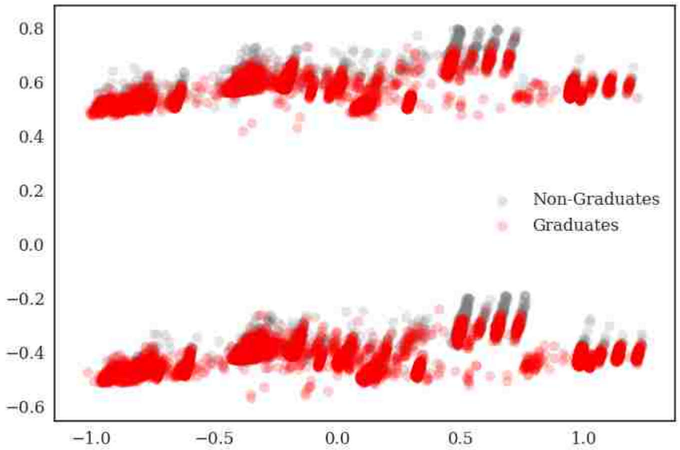


Figure 3.12: A transformation of the data to 2-dimensional space using PCA for the selected attributes.

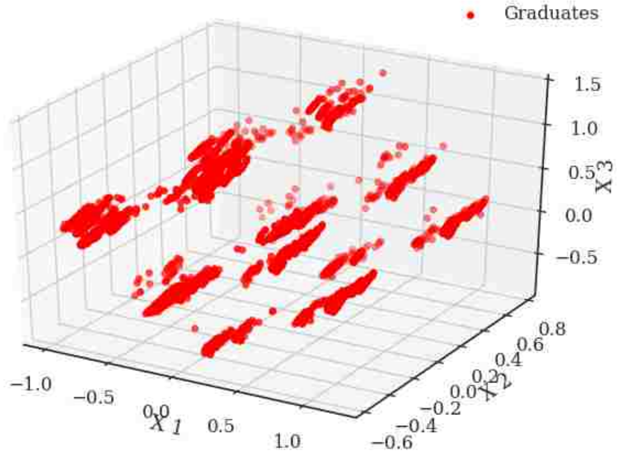


Figure 3.13: Selected attributes visualized in 3-dimensional space only for graduate students.

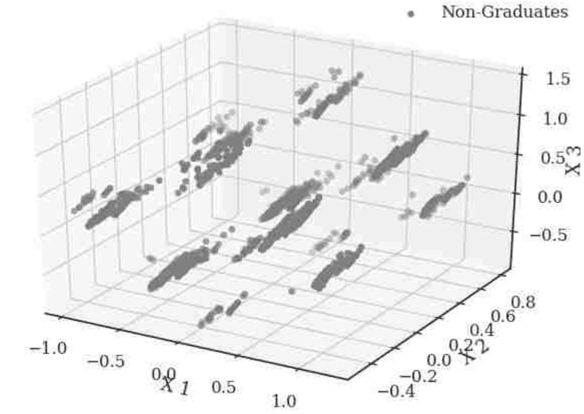


Figure 3.14: Selected attributes visualized in 3-dimensional space only for non-graduate students.

There are three well known types of missing data, missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). The most important type for this analysis is missing not at random where the data is missing due to a reason of that variable itself. A student who had a low ACT or SAT score may neglect to submit their scores, since it is not required for UNLV. Similarly, transfer students are not required to submit high school transcripts or test scores if they have sufficient transfer credits from another college. There are also many reasons why students do not complete FAFSA which contributes to many of the categorical variables in the data. It could be that students from wealthy families may not believe they will get any funding from FAFSA and choose not to complete it. This introduces a major bias into the samples and analysis since the data would be skewed to represent the middle class. International and non-resident students may not complete FAFSA because they do not believe they will be eligible for federal funds. Students may also be entirely ignorant of what FAFSA is.

From a databases perspective, it makes sense why some values are missing. Take the student athletes table. It is needless to keep a table with every student when you can simply insert athletes into the table as needed. In these and similar cases such as honors students or specific scholarships a value of ’no’ or zero is placed for all missing rows. For all other cases, a simple mean imputation for numerical values or the most frequent category in categorical columns was used as a filling value. As an alternate mean imputation, students were split by admission type (freshmen versus transfer) and missing values were filled with the corresponding mean. In a comparison of the feature ranking methods, the results were nearly identical. The mean imputation regardless of admission type was used so as to not introduce researcher bias on how students should be grouped. More advanced imputation methods are left for future work which can be compared to the results of mean imputation.

Chapter 4

Experimental Results

4.1 Evaluation Metrics

Evaluation metrics are the measures by which we can rate and understand the performance of a machine learning model. A quick way of visualizing the performance of a model is by a confusion matrix. For our purposes, the ”positive” class is considered the students who have graduated, and the ”negative” class are the non-graduated students.

Figure 4.1: General Confusion Matrix

The evaluation metrics discussed are quick interpretations of specific aspects shown by the confusion matrix, but all are based on TP, FP, FN, and TN.

4.1.1 Accuracy

Accuracy is the most intuitive evaluation metric; it is simply the percentage of correctly classified samples. However, in an imbalanced case in particular, it is a misleading metric. For this data set, simply predicting graduated for all student samples should give an accuracy of 79%, which is far above random guessing. This model would actually be useless for the problem despite having a good score. With imbalanced data sets, models can quickly develop a bias towards the majority class while ignoring type 1 and type 2 errors. Accuracy is not used to evaluate the models for these reasons.

4.1.2 Recall

Recall, also known as specificity and true positive rate, is the proportion of correctly predicted graduated students over the number of graduated students.

$$\frac{TP}{TP + FN} \tag{4.1}$$

An informal understanding of recall in this case would be how many of the students who are going to graduate that the model actually predicts correctly.

4.1.3 Precision

Precision is the proportion of correctly predicted graduated students over the number of students classified as graduated.

$$\frac{TP}{TP + FP} \tag{4.2}$$

An informal understanding of precision in this case would be how certain we can be when a model predicts a student will graduate that they will actually graduate.

4.1.4 F1 Score

It is often the case that recall and precision are equally important metrics. To quickly evaluate models by both measures, F1 score represents an equal contribution of the metrics by a harmonic mean given as:

$$\frac{2 * precision * recall}{precision + recall} \tag{4.3}$$

This gives a score between zero and one where one would represent perfect recall and precision. Recall and precision give scores which are both valuable and, being a proportion of classification, is largely independent of the class distribution. Due to F1 score's resistance to class imbalance and its equal weighting of recall and precision, it is the main score by which the models will be evaluated.

4.1.5 Area Under Curve

Area under curve (AUC) relates to receiver operating characteristics (ROC) analysis. ROC graphs provide a similar intuitive measure as do confusion matrices. It is a graph of the true positive rate versus the false positive rate. ROC analysis has a long history in medical diagnosis which then found great utility in the classification work of machine learning as described in the seminal work of Tom Fawcett [Faw06].

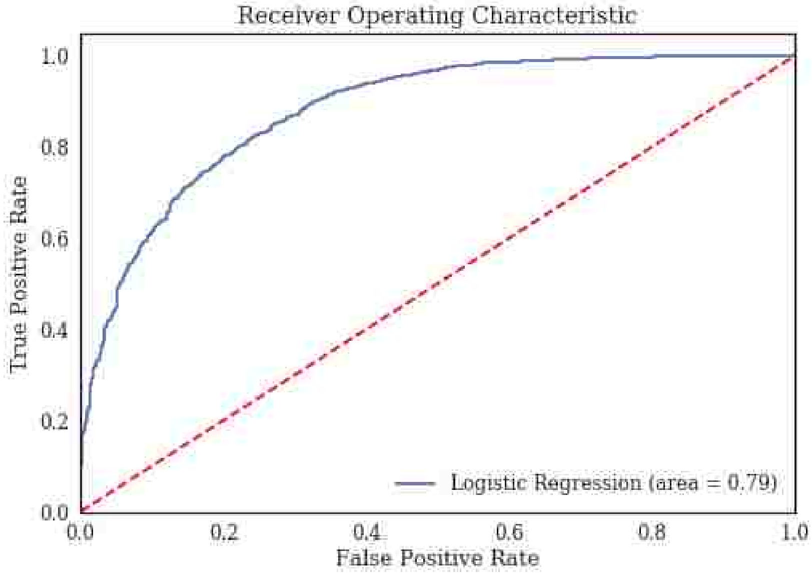


Figure 4.2: ROC Graph Example with AUC

The red dotted line of $x = y$ represents 50% of the area which corresponds to randomly guessing graduated or non-graduated. Ideally, the blue curve which represents the model would be up and to the left having 100% of the AUC.

4.2 10-Fold Cross-Validation

K-fold cross-validation, or in our case 10-fold cross-validation, is a technique to split the data into K subsets of the data. The motivation for this technique is to have the maximum student size possible to test (with cross-validation sets) and train with. Using ten split subsets, the model trains on nine and the held out subset, the tenth, is used to test. This process is continued by holding out the ninth subset and training a new model with the same parameters on the subsets one through eight and the tenth, et cetera. The results are found by taking the average of all ten models' performances on training and cross-validation sets, including the standard deviations. This gives a more detailed picture of how the model is learning over time which can be analyzed in a learning curve graph.

The particular implementation used here is scikit learn's stratified K-fold function. The data was shuffled. The random state of the function was set to an integer so all models would have the same splits for reproducibility between experiments. The splits of data are randomly selected from the samples and the balance of graduated to non-graduated classes are maintained for the training set. This means if over-sampling is applied, each split will have 50% of both graduated and non-graduated samples. If there is no over-sampling, the splits represent the training set, most likely four to one graduates to non-graduates.

4.2.1 Training, Testing, and Cross-Validation Splits

For the held out testing data to compare models, 30% of the students were randomly separated and scaled according to the training set. The 70% of remaining students were then used for 10-fold cross-validation.

4.3 Hyperparameter Search

A parameter for a model is something that is adjusted or tuned while the model is learning, like a coefficient for an attribute in logistic regression. A hyperparameter is what is set for a model prior to training, like setting the max depth of a decision tree. Hyperparameters play a key role in how long a model will need for training and contributes significantly into how effectively the model learns. It is rarely obvious what a good set of hyperparameters would be for a given problem and model. Instead, these hyperparameters are found experimentally. Multiple models are trained with varying combinations of hyperparameters.

Here is where the cross-validation set becomes particularly useful. While the parameters are tuned according to the training, the hyperparameters become biased towards testing well on the cross-validation set. This creates a type of ”fit” for the hyperparameters on the cross-validation set, since the hyperparameters that are selected are the set which performs well. This is why a test set of students are held out that the models have never seen before which are used to evaluate and compare the models, giving a more realistic assessment of performance on new cohorts of students. This hyperparameter search is different for each model since they all have different hyperparameters. The search is implemented via scikit learn's grid search function with cross-validation.

4.4 Logistic Regression

In many analyses, logistic regression serves as a baseline comparison for machine learning models. This is particularly true when there is no shared data set for a given problem that all researchers have access to. For student data which is protected by federal privacy laws, there are no reliable public data sets available. Logistic regression is thus the first model to choose as in [Her06].

4.4.1 Hyperparameters

The two main hyperparameters to search over for logistic regression are C and the type of solver. The solver is typically less important and is the optimization method used by the model to learn the parameters. The three solvers used here are liblinear, LBFGS, and newton-cg [PVG+11]. More importantly is the regularization term, C. This term helps to reign in overfitting. In the case of logistic regression, it learns a coefficient, a multiplier, for each attribute according to its influence on the outcome. If the student training set happens to have a split of the data where particular attributes, like whether they are full time in their second term, very strongly correlate with the outcome, but this is not true of the general student population, then the model can learn a very large coefficient for that attribute. However, the performance will then be poor in general because this attribute is incorrectly emphasized. C is the inverse of the regularization strength, so a large C corresponds to low penalization for strong coefficient changes in learning while a low C gives a higher penalization.

A Quick Note On Logspace

The values of C searched over are taken from NumPy's logarithmic space function, `'C': np.logspace(-4, 4, 8)`. This means from the range of $[10^{-4}, 10^{4}]$, eight points are chosen, giving .0001, .00138949549, .0193069773, .268269580, 3.72759372, 51.7947468, 719.685673, 1000.

4.4.2 Results

No Over-sampling

The best parameters are $C = 51.7947$ with the newton-cg solver.

```
F1 score of Logistic Regression - No Over-Sampling classifier on test set: 0.9044
```

Class	precision	recall	f1-score	support
0	0.70	0.41	0.52	1009
1	0.86	0.95	0.90	3814
avg/total	0.83	0.84	0.82	4823

Table 4.1: Evaluation metrics for logistic regression - no over-sampling.

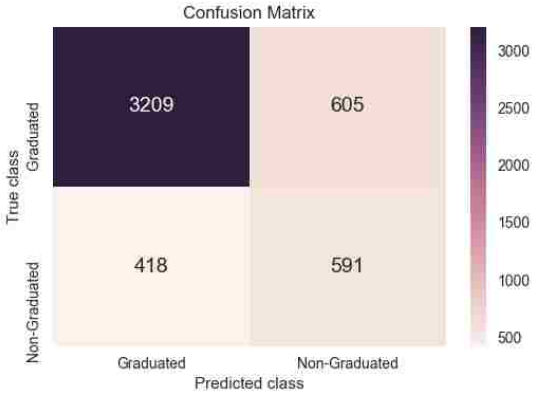


Figure 4.3: Logistic regression - no over-sampling

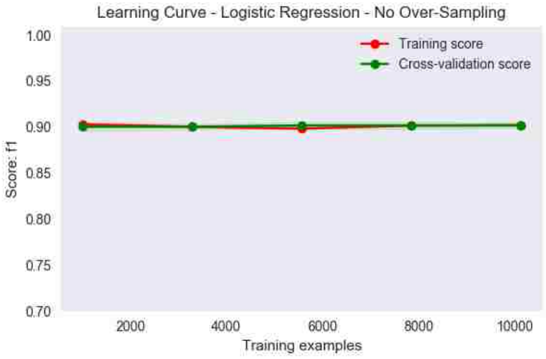


Figure 4.4: Logistic regression - no over-sampling

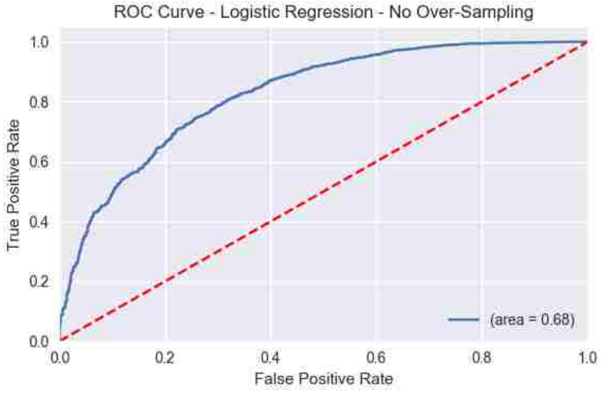


Figure 4.5: Logistic regression - no over-sampling

Class	precision	recall	f1-score	support
0	0.46	0.70	0.56	1009
1	0.91	0.78	0.84	3814
avg/total	0.82	0.77	0.78	4823

Table 4.2: Evaluation metrics for logistic regression - random over-sampling.

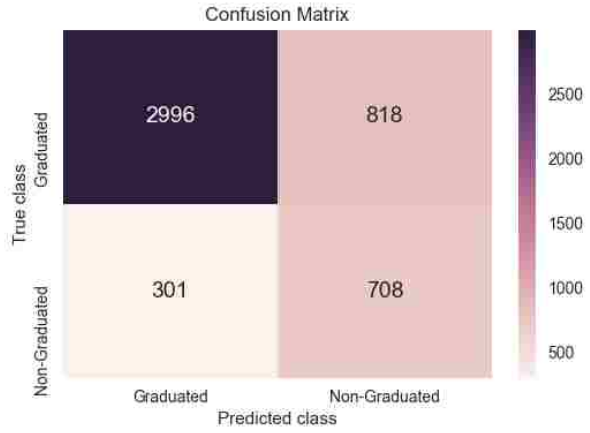


Figure 4.6: Logistic regression - random over-sampling

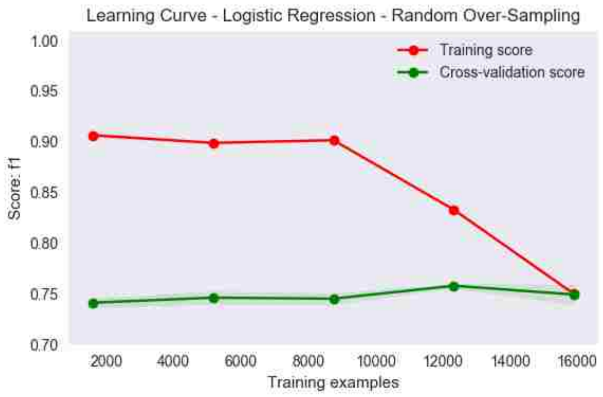


Figure 4.7: Logistic regression - random over-sampling

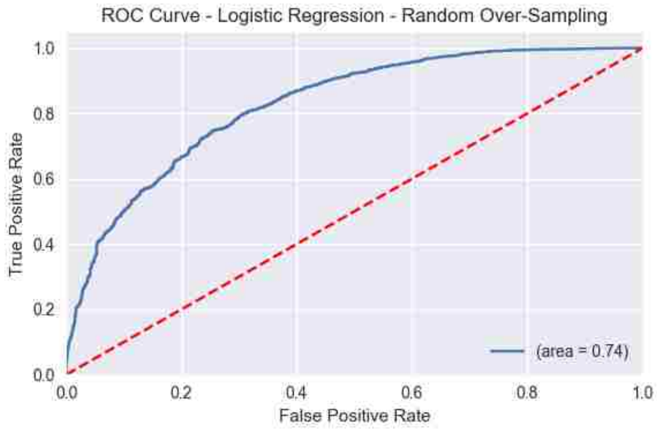


Figure 4.8: Logistic regression - random over-sampling

Random Over-sampling

The best parameters are $C = 3.7276$ with the newton-cg solver.

```
F1 score of Logistic Regression - Random Over-Sampling classifier on test set: 0.8426
```

Class	precision	recall	f1-score	support
0	0.46	0.70	0.56	1009
1	0.91	0.78	0.84	3814
avg/total	0.81	0.77	0.78	4823

Table 4.3: Evaluation metrics for logistic regression - SMOTE over-sampling.

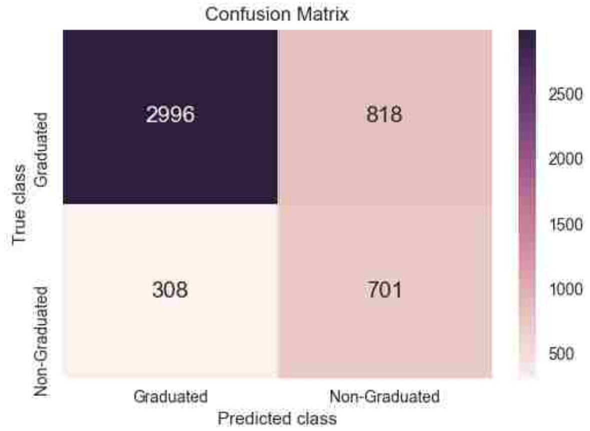


Figure 4.9: Logistic regression - SMOTE over-sampling

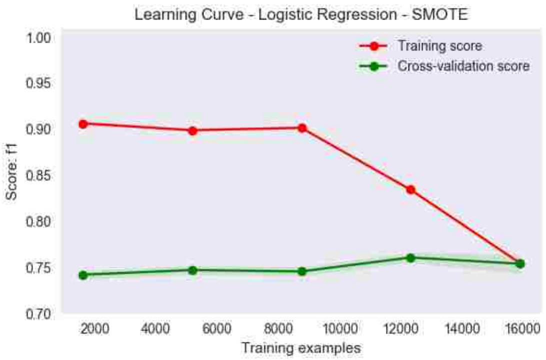


Figure 4.10: Logistic regression - SMOTE over-sampling

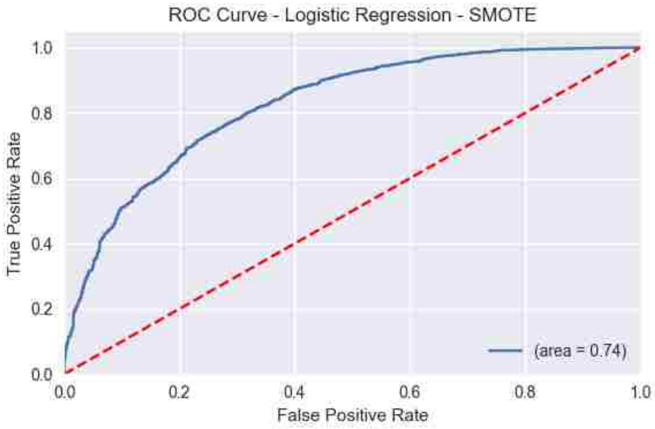


Figure 4.11: Logistic regression - SMOTE over-sampling

SMOTE Over-sampling

The best parameters are $C = 3.7276$ with a LBFGS solver.

```
F1 score of Logistic Regression - SMOTE classifier on test set: 0.8418
```

Class	precision	recall	f1-score	support
0	0.47	0.56	0.51	1009
1	0.88	0.83	0.86	3814
avg/total	0.79	0.78	0.78	4823

Table 4.4: Evaluation metrics for logistic regression - ADASYN over-sampling.

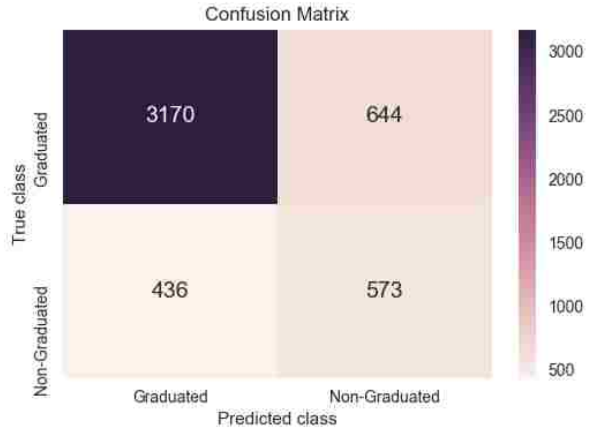


Figure 4.12: Logistic regression - ADASYN over-sampling

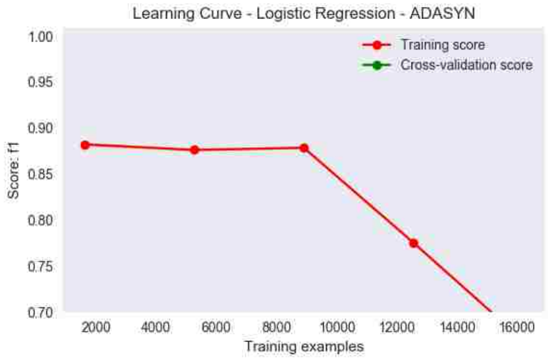


Figure 4.13: Logistic regression - ADASYN over-sampling

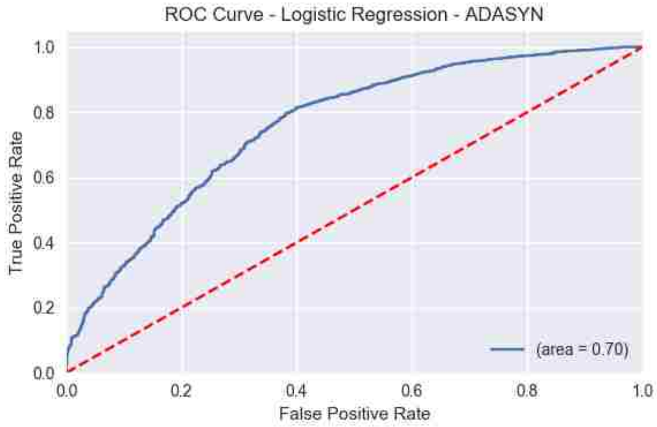


Figure 4.14: Logistic regression - ADASYN over-sampling

ADASYN Over-sampling

The best parameters are $C = 0.0001$ with the liblinear solver.

```
F1 score of Logistic Regression - ADASYN classifier on test set: 0.8544
```

Analysis of Results

The initial results of logistic regression look very promising with an F1 score of 90.45% even without over-sampling. However, the C value is very high, and the learning curve as shown in figure 4.4 further confirms that this model is overfitting the training. It is most likely learning the class distribution and class imbalance more than the usefulness of the attributes. This is further confirmed with the random over-sampling results showing an immediate drop of 15 percentage points on training and cross-validation. The ADASYN learning curves perform the worst with the F1 scores being below the viewing threshold of 70%. These learning curves show over-fitting even with smaller regularization terms. Having more student data with the same methods are unlikely to improve the models, it can only be alleviated with new attributes not included here.

4.5 Decision Tree

Decision trees have found great success in predicting and interpreting student retention and graduation rates as reported in section 2.1.2. They are particularly robust at handling categorical variables which form roughly two-thirds of the data. The implementation used here is scikit learn's version of an optimized CART algorithm. Rather than using a rule set to classify as some decision trees do, each node splits up the data according to a measure of information gain from the split.

4.5.1 Hyperparameters

The three main hyperparameters search over are the criterion, splitting type (splitter), and the max depth of the tree. The criterion is the measure of information gain of which there are two types, Gini and entropy. If the split type is best, then the optimal Gini or entropy feature is used. The random split chooses a random feature at each node and calculates the information gain. The depth of the tree can fight over-fitting, since the tree is less likely to form a split for every attribute of the training set. Instead the tree uses the most salient attributes. For max depth, the search is over five, ten, and fifteen levels deep.

4.5.2 Results

No Over-sampling

The best parameters are entropy, a 'best' splitter, and a max depth of five.

```
F1 score of Decision Tree - No Over-Sampling classifier on test set: 0.9177
```

Class	precision	recall	f1-score	support
0	0.78	0.48	0.60	1009
1	0.88	0.96	0.92	3814
avg/total	0.86	0.86	0.85	4823

Table 4.5: Evaluation metrics for decision tree - no over-sampling.

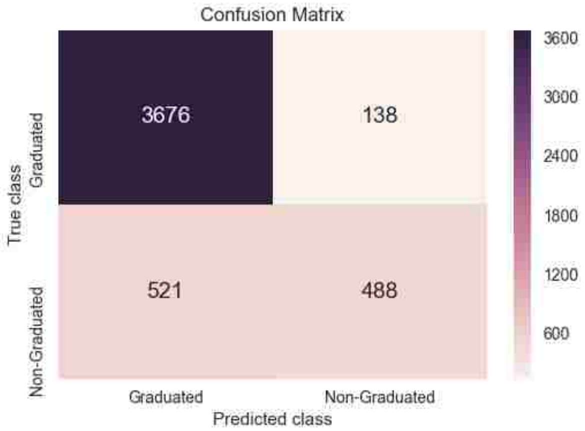


Figure 4.15: Decision tree - no over-sampling

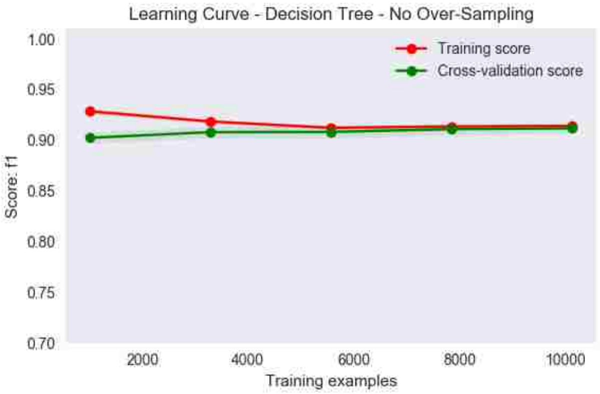


Figure 4.16: Decision tree - no over-sampling

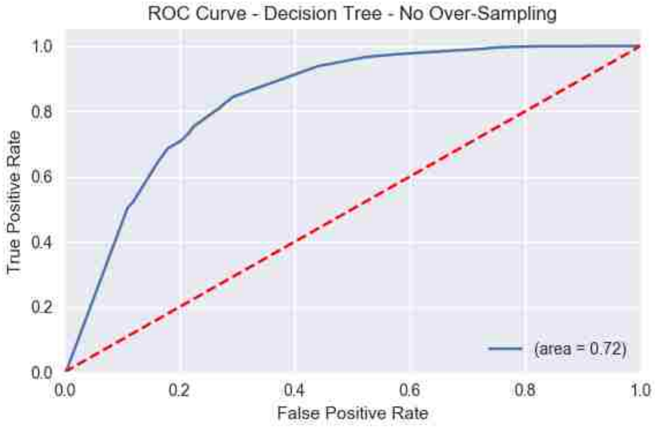


Figure 4.17: Decision tree - no over-sampling

Class	precision	recall	f1-score	support
0	0.50	0.60	0.55	1009
1	0.89	0.84	0.86	3814
avg/total	0.81	0.79	0.80	4823

Table 4.6: Evaluation metrics for decision tree - random over-sampling.

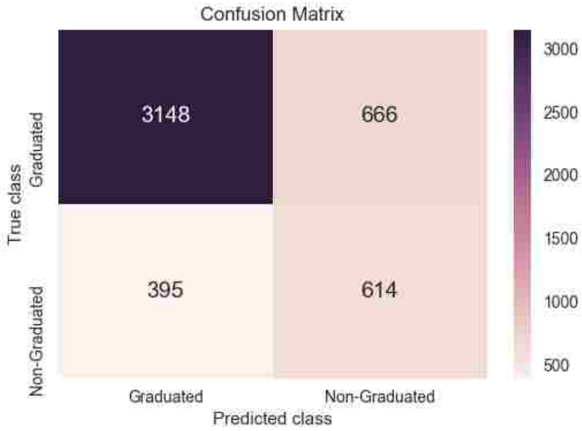


Figure 4.18: Decision tree - no over-sampling

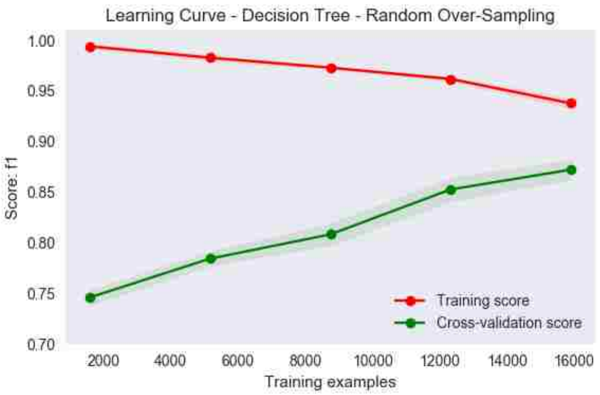


Figure 4.19: Decision tree - no over-sampling

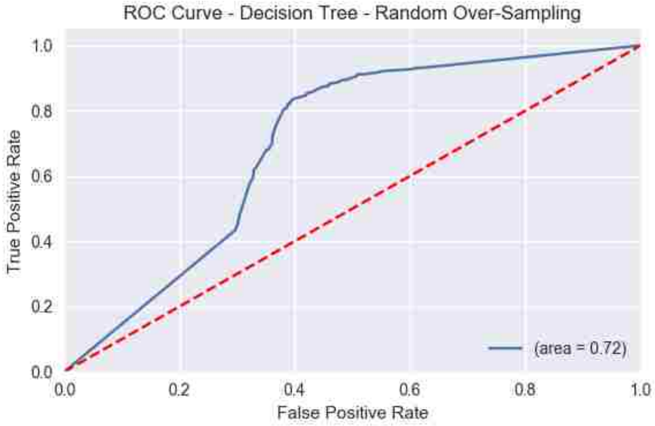


Figure 4.20: Decision tree - no over-sampling

Random Over-sampling

The best parameters are Gini, a splitter of best, and a max depth of 15.

```
F1 score of Decision Tree - Random Over-Sampling classifier on test set: 0.8719
```

Class	precision	recall	f1-score	support
0	0.49	0.63	0.55	1009
1	0.89	0.83	0.86	3814
avg/total	0.81	0.79	0.79	4823

Table 4.7: Evaluation metrics for decision tree - SMOTE over-sampling.

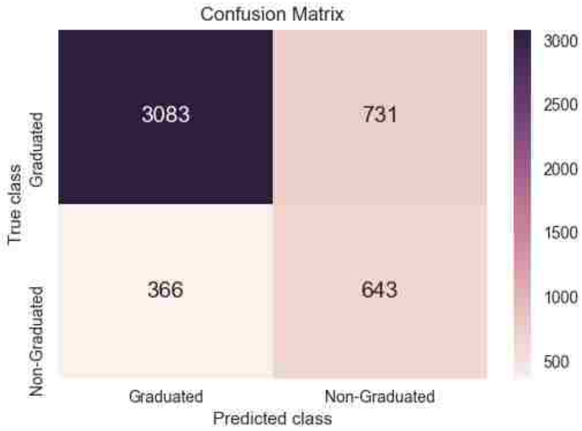


Figure 4.21: Decision tree - SMOTE over-sampling

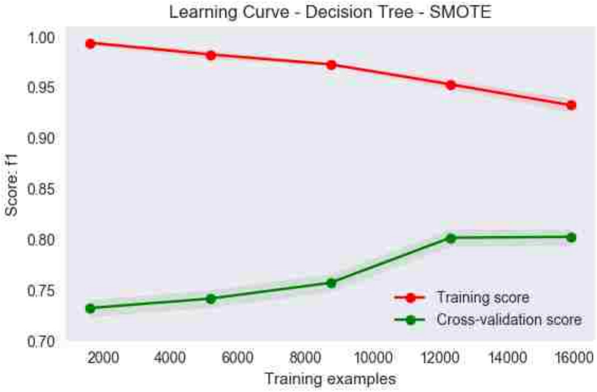


Figure 4.22: Decision tree - SMOTE over-sampling

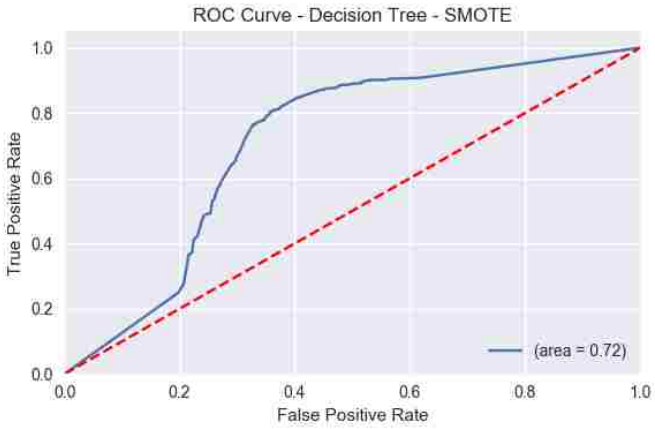


Figure 4.23: Decision tree - SMOTE over-sampling

SMOTE Over-sampling

The best parameters are Gini, a splitter of best, and a max depth of 15.

```
F1 score of Decision Tree - SMOTE classifier on test set: 0.8472
```

Class	precision	recall	f1-score	support
0	0.60	0.49	0.54	1009
1	0.87	0.92	0.89	3814
avg/total	0.82	0.83	0.82	4823

Table 4.8: Evaluation metrics for decision tree - ADASYN over-sampling.

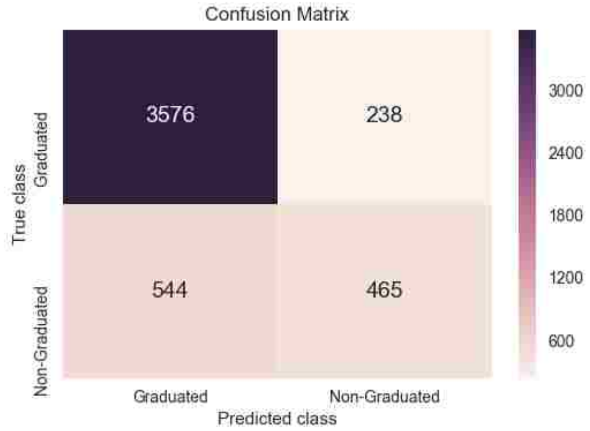


Figure 4.24: Decision tree - ADASYN over-sampling

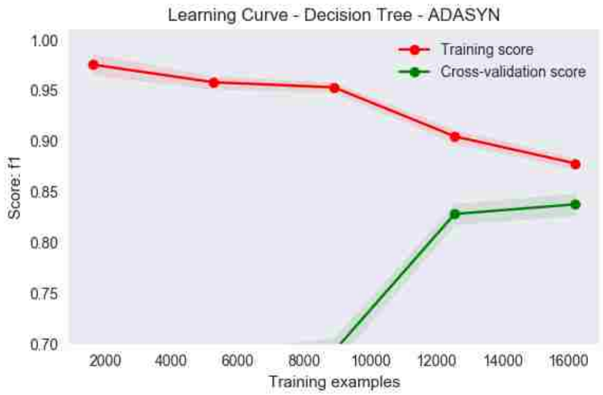


Figure 4.25: Decision tree - ADASYN over-sampling

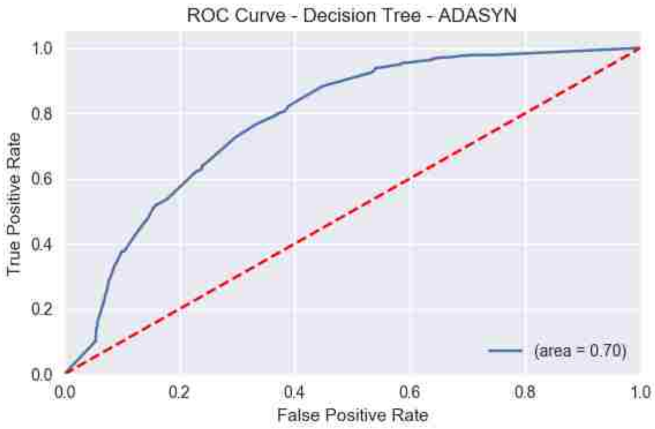


Figure 4.26: Decision tree - ADASYN over-sampling

ADASYN Over-sampling

The best parameters are Gini, a splitter of best, and a max depth of 10.

```
F1 score of Decision Tree - ADASYN classifier on test set: 0.8864
```

Analysis of Results

Decision trees tend to be sensitive to class imbalances due to their splitting criterions, since the algorithm tries to split the data optimally. This leads to the majority class generally giving the most information gain. We see that the decision tree performs very well with no sampling, achieving a slightly better result than logistic regression. However, like logistic regression it quickly overfit the data and left little room for growth in new student samples.

The algorithm's performance on the over-sampling student sets suggests a robustness and healthy growth in the presence of additional student data, so collecting more student samples should increase the predictive power of the model as evidenced by the learning curves. The algorithm has impressively high recall on the graduated student class at 96% with no over-sampling.

4.6 Support Vector Machines

While logistic regression tries to linearly separate the two classes, SVMs take a different approach by constructing a hyperplane and maximizing the marginal distance. SVMs have kernels, an essential feature of the algorithm that can be either linear or nonlinear. Not all data is linearly separable, so logistic regression can hit a limit where a nonlinear SVM can find a classification boundary with great complexity. This makes SVMs robust with high dimensional data and need less data as compared to other complex models. In this implementation, the radial basis function (RBF) kernel is used to accomplish nonlinearity.

4.6.1 Hyperparameters

The main hyperparameters searched for are the C and gamma terms. The C term functions similarly to regularization in logistic regression. A high C allows the SVM to select more student samples for support vectors, increasing its ability to classify all samples and overfit, while the gamma parameter controls the influence of each student sample [PVG+11]. Five points each are selected from the logarithmic space of $[10^{-2}, 10^{4}]$ for C and $[10^{-2}, 10^{3}]$ for gamma.

4.6.2 Results

No Over-sampling

The best parameters are $C = 316.2278$ and gamma $= 0.1778$.

```
F1 score of SVM - No Over-Sampling classifier on test set: 0.9090
```

Class	precision	recall	f1-score	support
0	0.78	0.35	0.48	1009
1	0.85	0.97	0.91	3814
avg/total	0.84	0.84	0.82	4823

Table 4.9: Evaluation metrics for SVM - no over-sampling.

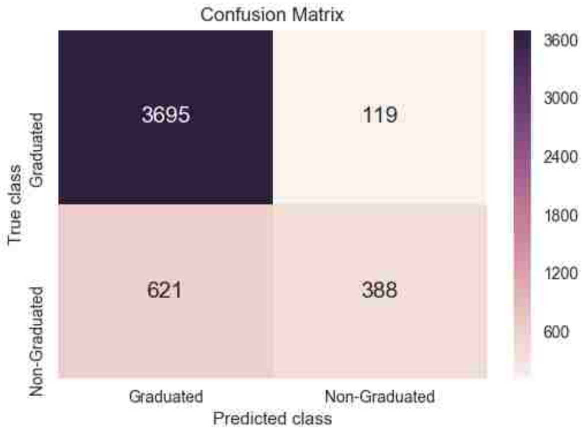


Figure 4.27: SVM - no over-sampling

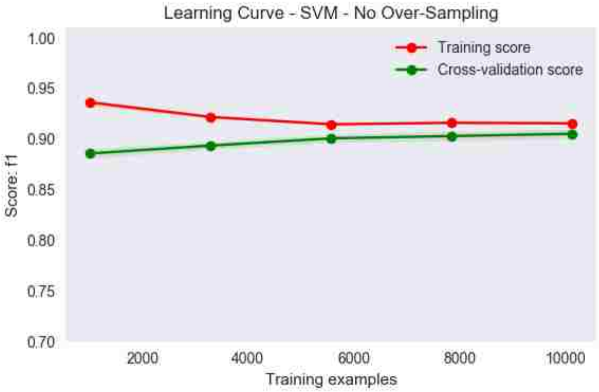


Figure 4.28: SVM - no over-sampling

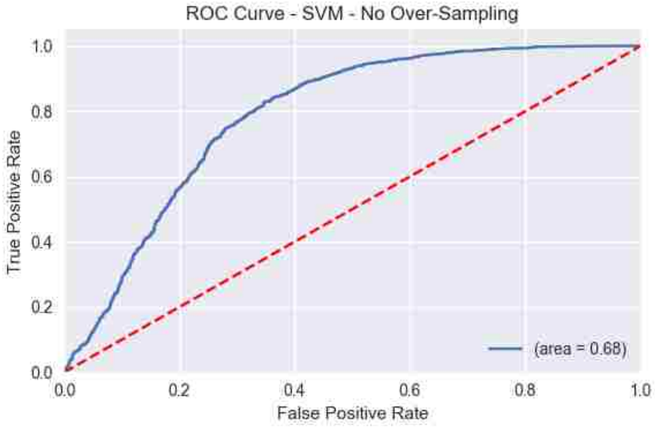


Figure 4.29: SVM - no over-sampling

Class	precision	recall	f1-score	support
0	0.43	0.01	0.02	1009
1	0.79	1.00	0.88	3814
avg/total	0.72	0.79	0.70	4823

Table 4.10: Evaluation metrics for SVM - random over-sampling.

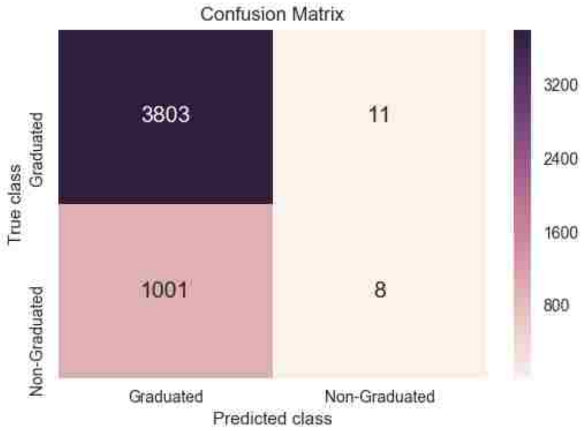


Figure 4.30: SVM - random over-sampling

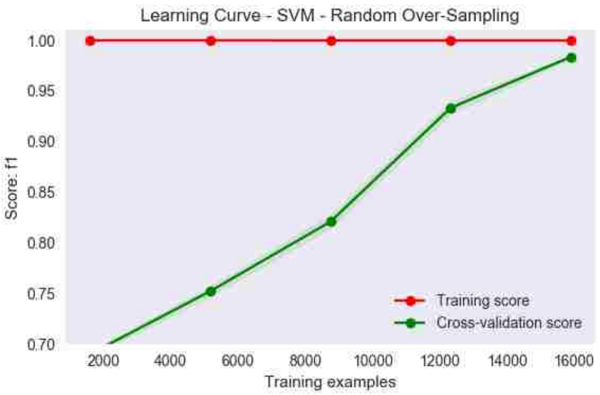


Figure 4.31: SVM - random over-sampling

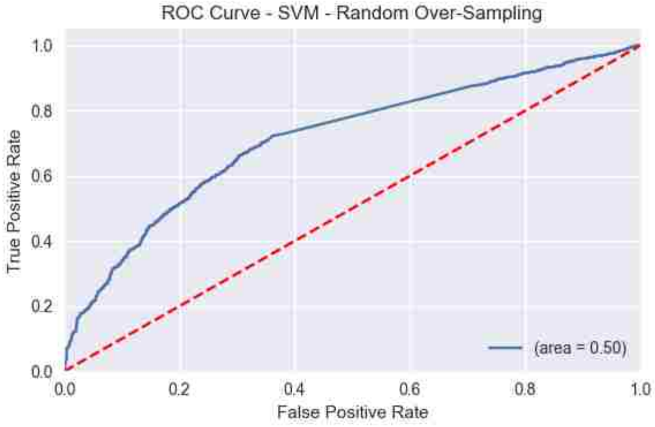


Figure 4.32: SVM - random over-sampling

Random Over-sampling

The best parameters are $C = 10$ and gamma = 1000.

```
F1 score of SVM - Random Over-Sampling classifier on test set: 0.8826
```

Class	precision	recall	f1-score	support
0	0.44	0.32	0.37	1009
1	0.83	0.89	0.86	3814
avg/total	0.75	0.77	0.76	4823

Table 4.11: Evaluation metrics for SVM - SMOTE over-sampling.

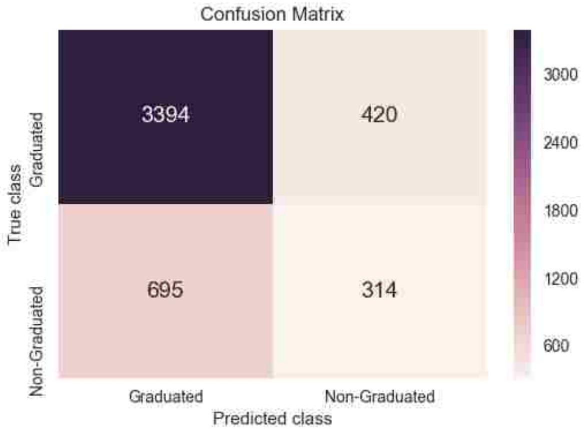


Figure 4.33: SVM - SMOTE over-sampling

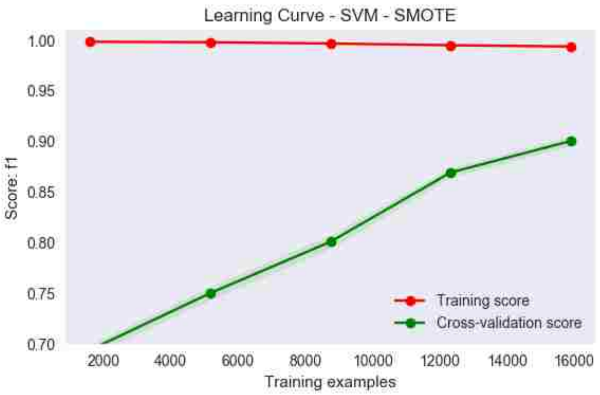


Figure 4.34: SVM - SMOTE over-sampling

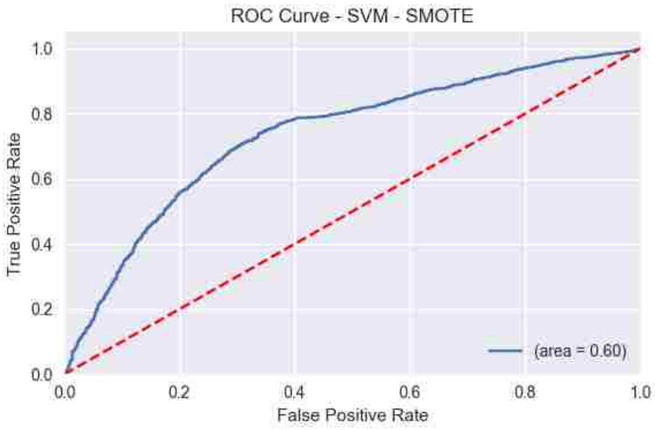


Figure 4.35: SVM - SMOTE over-sampling

SMOTE Over-sampling

The best parameters are $C = 10$ and gamma $= 56.2341$.

```
F1 score of SVM - SMOTE classifier on test set: 0.8589
```

Class	precision	recall	f1-score	support
0	0.68	0.45	0.55	1009
1	0.87	0.94	0.90	3814
avg/total	0.83	0.84	0.83	4823

Table 4.12: Evaluation metrics for SVM - ADASYN over-sampling.

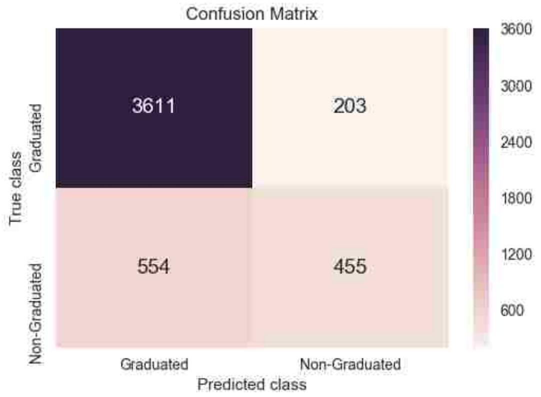


Figure 4.36: SVM - ADASYN over-sampling

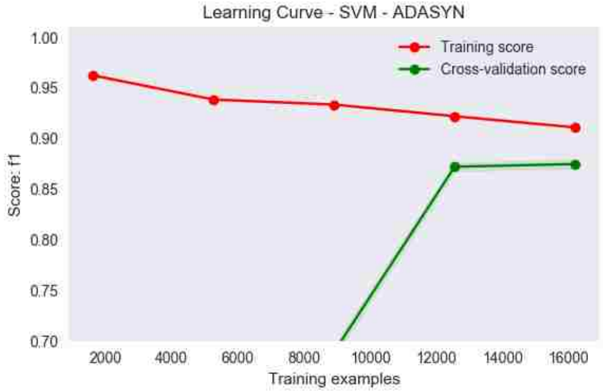


Figure 4.37: SVM - ADASYN over-sampling

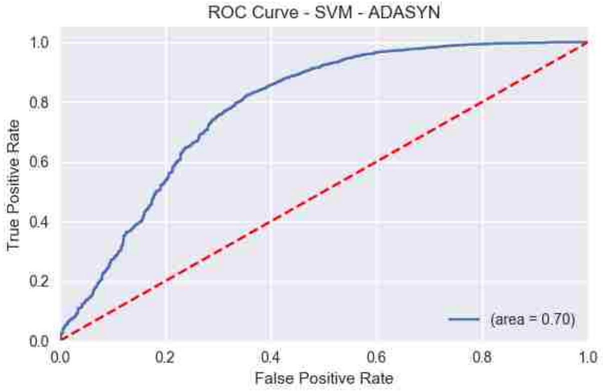


Figure 4.38: SVM - ADASYN over-sampling

ADASYN Over-sampling

The best parameters are $C = 10000$ and gamma $= 0.1778$.

```
F1 score of SVM - ADASYN classifier on test set: 0.9051
```

Analysis of Results

The two major hyperparameters can be graphed against each other as shown in figures 4.39 and 4.40 with a heatmap of the F1 score on the validation set. These graphs were created with five-fold cross-validation due to computational time. We see from the performance of these models that they prefer to create highly complex classification boundaries, thus overfitting the data. For no over-sampling, the SVM performs very similarly to logistic regression. In random over-sampling, the model predicts the majority class nearly always. The model is likely confused due to the over-sampling of the minority class with replacement. While there are more points for support vectors,

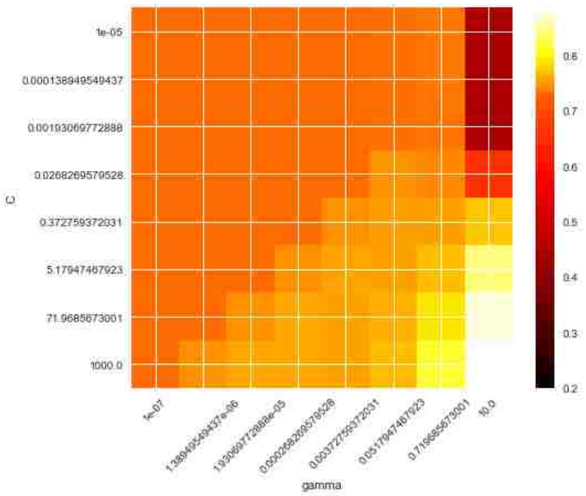


Figure 4.39: C versus gamma with respect to cross-validation scores.

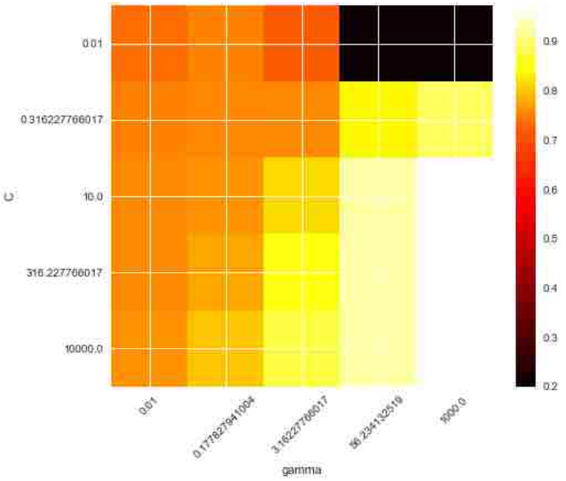


Figure 4.40: Extended ranges of the parameter search.

those additional points occupy the exact same space as the existing student samples, thus adding no new information. Interestingly, the synthetic samples in SMOTE and particularly ADASYN increase the generalization of the model. ADASYN over-sampling gives the best performance overall with a very high C of 10000. The over-sampling techniques show that an SVM would benefit from additional, diverse student samples to the data set.

4.7 Artificial Neural Network

Artificial neural networks (ANNs) are a fascinating class of algorithms which can approximate complex functions. This makes ANNs a useful choice when a proper function is unknown and too difficult to derive by analytical means. This implementation is specifically a multilayer perceptron with one hidden layer using a tanh activation function and the LBFGS solver [PVG^{+}11].

4.7.1 Hyperparameters

Two of the main hyperparameters which have the strongest affects on the model are the number of nodes in the hidden layer and the value of alpha, otherwise known as the learning rate. The number of hidden nodes corresponds to the complexity the model can learn. A high number of nodes can learn more complex classifications at the risk of overfitting. A smaller number of nodes can fight overfitting by only being able to hold information which is essential. The learning rate alpha sets the rate at which the model adjusts its parameters with each pass over the data or samples. Lower alpha rates slow training but grant finer grained learning. For these experiments, ten, thirty, fifty,

Class	precision	recall	f1-score	support
0	0.73	0.43	0.54	1009
1	0.86	0.96	0.91	3814
avg/total	0.84	0.85	0.83	4823

Table 4.13: Evaluation metrics for MLP - no over-sampling.

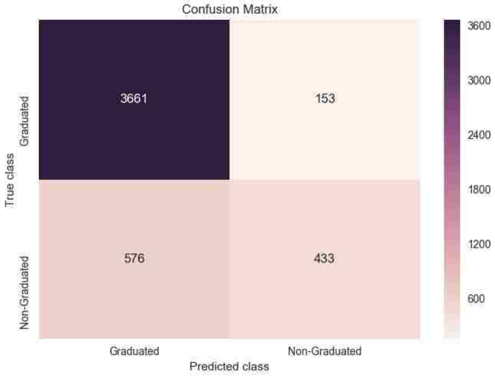


Figure 4.41: MLP - no over-sampling

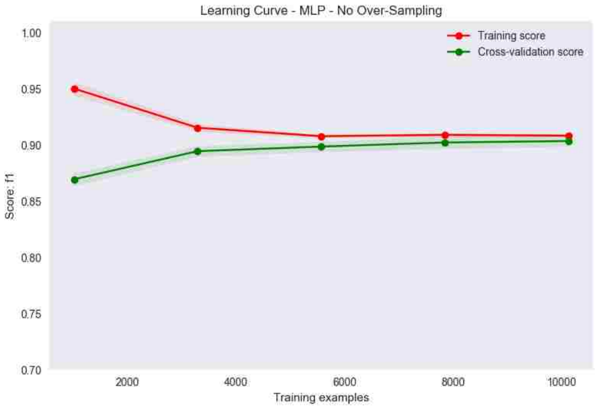


Figure 4.42: MLP - no over-sampling

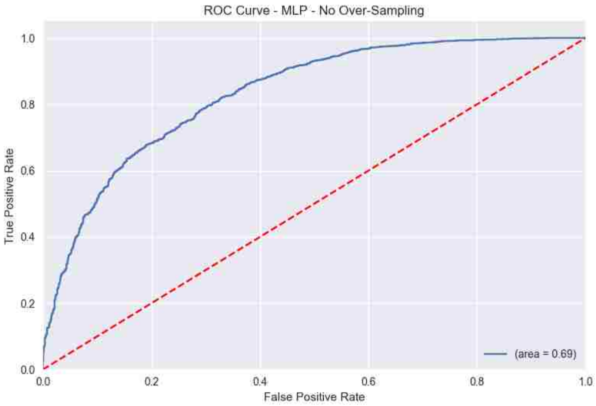


Figure 4.43: MLP - no over-sampling

and one hundred hidden nodes are considered with alpha values of either 0.1 or 0.01.

4.7.2 Results

No Over-sampling

The best parameters are an alpha of 0.01 and one hundred hidden nodes.

```
F1 score of MLP - No over-sampling classifier on test set: 0.9095
```

Class	precision	recall	f1-score	support
0	0.46	0.72	0.56	1009
1	0.91	0.78	0.84	3814
avg/total	0.82	0.76	0.78	4823

Table 4.14: Evaluation metrics for MLP - random over-sampling.

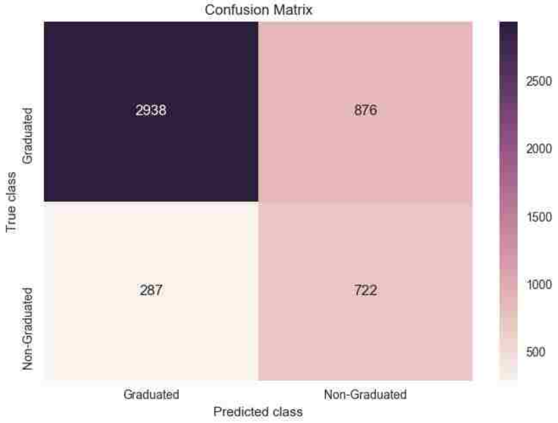


Figure 4.44: MLP - random over-sampling

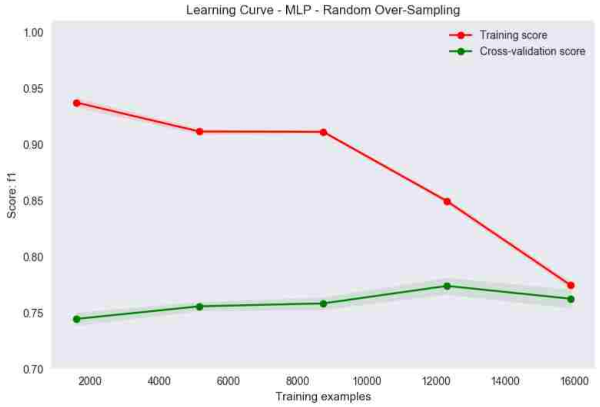


Figure 4.45: MLP - random over-sampling

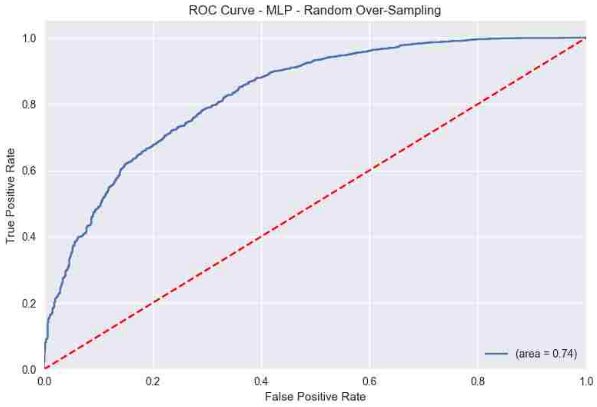


Figure 4.46: MLP - random over-sampling

Random Over-sampling

The best parameters are an alpha of 0.1 with thirty hidden nodes.

```
F1 score of MLP - Random over-sampling classifier on test set: 0.8348
```

Class	precision	recall	f1-score	support
0	0.46	0.70	0.55	1009
1	0.91	0.78	0.84	3814
avg/total	0.81	0.76	0.78	4823

Table 4.15: Evaluation metrics for MLP - SMOTE over-sampling.

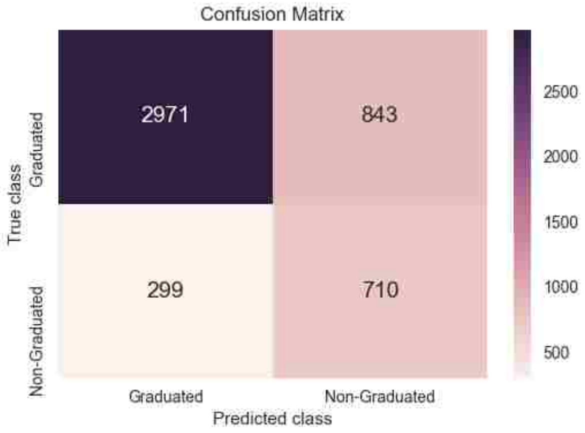
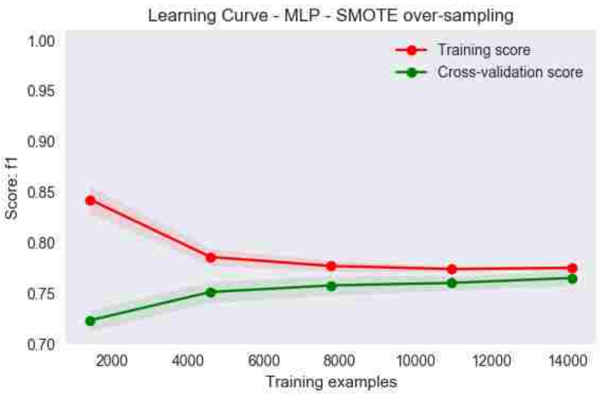


Figure 4.47: MLP - SMOTE over-sampling



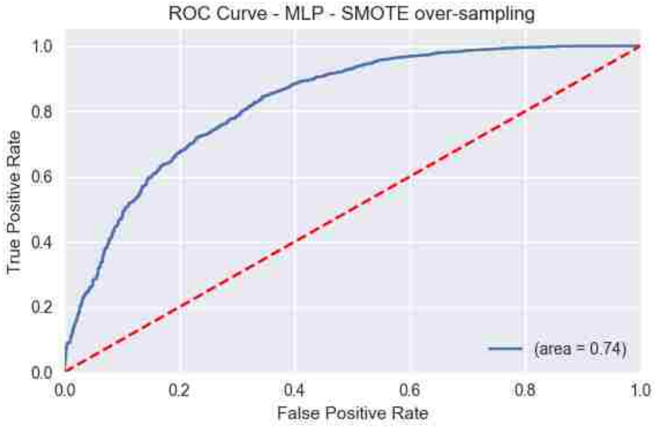


Figure 4.48: MLP - SMOTE over-sampling

Figure 4.49: MLP - SMOTE over-sampling

SMOTE Over-sampling

The best parameters are alpha = 0.01 and thirty hidden nodes.

```
F1 score of MLP - SMOTE over-sampling classifier on test set: 0.8439
```

Class	precision	recall	f1-score	support
0	0.63	0.50	0.56	1009
1	0.87	0.92	0.90	3814
avg/total	0.82	0.83	0.83	4823

Table 4.16: Evaluation metrics for MLP - ADASYN over-sampling.

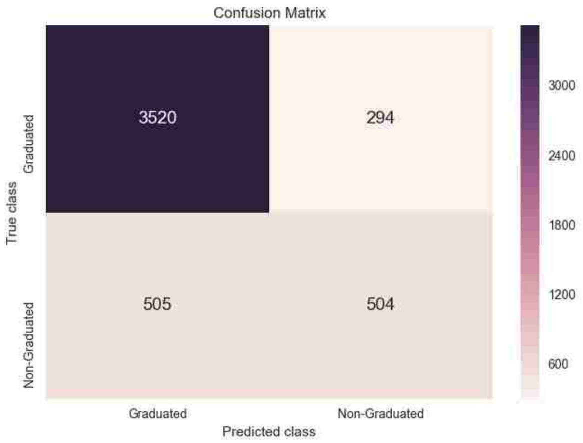


Figure 4.50: MLP - ADASYN over-sampling

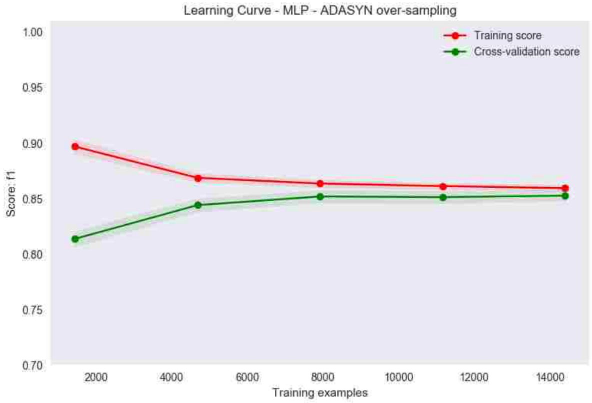


Figure 4.51: MLP - ADASYN over-sampling

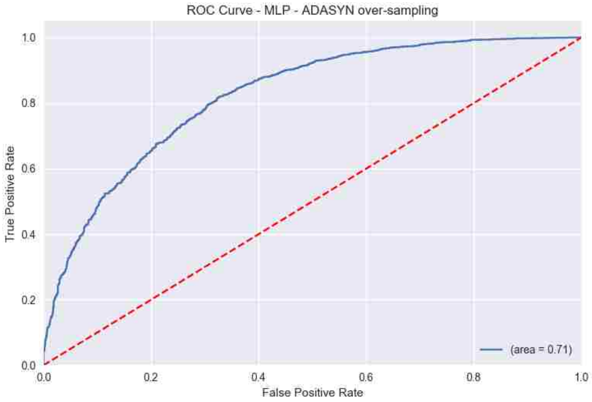


Figure 4.52: MLP - ADASYN over-sampling

ADASYN Over-sampling

The best parameters are alpha = 0.01 and one hundred hidden nodes.

```
F1 score of MLP - ADASYN over-sampling classifier on test set: 0.8961
```

Analysis of Results

We see that in nearly all cases, the multilayer perceptron classifier is robust in performance, coming very close to the evaluation metrics of the decision tree on the original data set with no over-sampling. It achieved this by having a smaller number hidden nodes at just ten. There is a sharp decline in score with random over-sampling as the model tends to over-classify the non-graduate students. SMOTE over-sampling gave a nearly identical performance as random over-sampling although with an alpha of 0.01 instead of 0.1. Interestingly, ADASYN over-sampling gave impressive performance overall with a hidden layer of fifty nodes and alpha of 0.01 the model was able to learn a complex and more accurate classification function.

Chapter 5

Conclusion

In this thesis, we described the problem of predicting student graduation rates. A review of the literature shows the difficulty in establishing standardized methods as universities have different demographics of students and there are no publicly available data sets for shared comparison of models. We detailed the process of collecting applicable UNLV student data for the purpose of analysis and prediction and pruning unnecessary and noisy attributes. Over-sampling techniques address the class imbalance problem of the data set. Finally, we applied a wide array of techniques currently used in the literature for this previously unanalyzed data set.

5.1 Summary of Results

We see as evidenced by the performance of all the models that class imbalance is highly influential. Worse yet is how biased many of the classifiers become with logistic regression being a clear example as shown in figure 4.4. Without over-sampling, all the models show a high bias in their learning curves which implies that simply gathering more student data with the current methods are unlikely

Class	precision	recall	f1-score
avg/0	0.5225	0.5925	0.5350
std/0	0.1184	0.1352	0.0238
avg/1	0.8900	0.8400	0.8575
std/1	0.0245	0.0757	0.0287
avg/avg	0.8125	0.7900	0.7900
std/avg	0.0171	0.0337	0.0200

Table 5.1: Logistic Regression - The average and standard deviation of scores across all experiments.

Class	precision	recall	f1-score
avg/0	0.5825	0.5575	0.5600
std/0	0.1391	0.0660	0.0294
avg/1	0.8850	0.8825	0.8825
std/1	0.0058	0.0634	0.0299
avg/avg	0.8200	0.8125	0.8150
std/avg	0.0271	0.0377	0.0289

Table 5.2: Decision Tree - The average and standard deviation of scores across all experiments.

Class	precision	recall	f1-score
avg/0	0.5775	0.2875	0.3600
std/0	0.1791	0.1936	0.2410
avg/1	0.8375	0.9525	0.8900
std/1	0.0359	0.0465	0.0245
avg/avg	0.7825	0.8125	0.7775
std/avg	0.0629	0.0386	0.0640

Table 5.3: Support Vector Machine - The average and standard deviation of scores across all experiments.

Class	precision	recall	f1-score
avg/0	0.5700	0.5850	0.5500
std/0	0.1364	0.1466	0.0082
avg/1	0.8875	0.8600	0.8700
std/1	0.0263	0.0942	0.0408
avg/avg	0.8225	0.8025	0.8025
std/avg	0.0126	0.0443	0.0263

Table 5.4: Multilayer Perceptron - The average and standard deviation of scores across all experiments.

Rank	precision	recall	f1-score
1	Multilayer Perceptron	Decision Tree/SVM	Decision Tree
2	Decision Tree	Decision Tree/SVM	Multilayer Perceptron
3	Logistic Regression	Multilayer Perceptron	Logistic Regression
4	SVM	Logistic Regression	SVM

Table 5.5: Ranking of models based on score.

to improve performance. One way to fight this high bias is to search for different student attributes than what is gathered here.

The more complex models of support vector machines and multilayer perceptrons still suffer from similar issues to logistic regression. The multilayer perceptron has a higher performance than logistic regression yet shows the same trend in the learning curves past 6,000 examples. The support vector machine models show promising results especially for more diverse data sets as evidenced from its reaction to synthetic student examples.

While nearly all models achieved high F1 scores with some method, the decision tree achieved the highest score at 0.9177 on the original student data set (table 4.5) as well as the highest, weighted F1 score average at 0.815. This is far from the only reason why this model is chosen as the best. The decision tree performed well across precision and recall. The support vector machine's recall for the minority suffers far worse than the decision tree but the majority score counter balances this to a much higher score. However, we can see that the support vector machine significantly fails at classifying the minority class. The decision tree consistently performed robustly across over-sampling techniques. More importantly, the learning curves of the over-sampling experiments show that the decision tree should be able to increase performance with more student samples. Unlike the other models, this is true even without generating synthetic samples. The additional benefit is the ability to visualize the model in a tree graph and interpret the meaning of the paths for a given student.

What is most difficult in this problem as we have seen from each of the confusion matrices of the experiments is the stubbornly misclassified group of non-graduated students. The lowest misclassification is seen in the multilayer perceptron's performance with random over-sampling at 282 student samples while many of the models have around 550. With a total of 1,009 non-graduated student samples in the test set, this is over 50% of those students consistently misclassified. This misclassification could be due to key attributes missing from this data set or to human factors. Many of these students likely fit the profile of graduating students but for some reason or other decide to leave UNLV or dropout altogether. However, in the multidimensional space these students would be difficult for any model to properly sift out without overfitting and decreasing performance in general.

5.2 Implications for Higher Education Practictioners

This work provides a foundation on which further data can be collected and a direction for additional modeling. The current models in conjunction with higher education staff can be an efficient and effective method to assist in raising graduation rates at UNLV. By training a decision tree model on all of the given data, future cohorts of students can be ranked according to the probability by which they will or will not graduate. Those students who have the highest confidence ratings can be addressed first.

Additionally, the feature ranking of this work suggests admissions officers should emphasize core high school GPA and unweighted high school GPA as a more accurate measure of predictive success at UNLV. For students who are not Nevada residents, eligibility for the Western Undergraduate Exchange program which reduces tuition fees is an important factor for incoming applicants. Furthermore, these results provide more empirical evidence for common practices by advising and retention coordinators. Students taking remedial math or English courses in the first or second semester are slightly less likely to graduate. Students who are less than full-time in either their first or second semester are less likely to graduate. The SAP attribute remains as one of the most predictive categorical variables.

The decision tree model provides the "path" which a student is on, allowing outreach efforts to focus on those attributes most likely to redirect a student to a path of success. Certain attributes are fixed, such as admission type, but the branching attributes provide a threshold for which coordinators can work with students to set clear goals. For larger college departments, a decision tree model can be trained on their subset of students, giving more nuanced thresholds for the specific major in addition to a university based model.

5.3 Future Work

With the current techniques employed and the analysis of the data set, there appears to be an upper limit to classification and evaluation scores. While methods such as a hybrid approach or ensemble modeling where different models are trained together and vote on each sample could be developed to handle some of the bias/variance problems, the confusion matrices and evaluation metrics show little variation in performance to improve information inference. Random forests could give a small performance boost which use many decision trees to interpret a given sample.

However, domain experts may have additional attributes that can be gathered such as survey data on motivation. Questions such as ”how likely are you to pursue graduate school” and ”how determined are you to get a bachelor’s degree on a scale of one to five” could help to provide essential information not yet tracked. The incorporation of the learning analytics field could prove to give a more dynamic and real-time analysis of student performance and attention. Using learning management systems, or in UNLV’s case, Blackboard and Moodle, metrics such as resources clicked, time active online, engagement in online discussion, and others combined with theories of learning could enhance the predictive power of student success, particularly on those students with little stored data.

Not to be overlooked is the issue of missing data and imputation. With the addition of any attributes based on survey data, we can apply this to newly incoming students but past students are unlikely to provide this data accurately or at all. The imputation method used here is a standard mean imputation, but it is likely that more advanced imputation techniques can provide entirely different results like fuzzy K nearest neighbors imputation.

To get a more accurate picture of hard-to-classify samples, students can be tracked through the National Student Clearinghouse to determine if they have dropped out of American universities completely or have simply transferred to another university to complete their degree. Students who complete a degree elsewhere likely had the ability to graduate from UNLV while students who drop out entirely likely form two separate groups. As evidenced by the PCA visualizations, there appear to be clusters of student groups which can be further explored.

Appendix A

Data Dictionary

The meaning and context of the attributes gathered.

Attribute	Description	Values
First Time Student		Binary (Y,N)
Admission Type	First year (freshmen) versus transfer students.	(FYR,TRN)
Gender		(M,F)
IPEDS Race/Ethnicity	IPEDS is the Integrated Postsecondary Education Data System. Their definition of race/ethnicity is federally reported.	Categorical
Non-Resident Alien		Binary (Y,N)
USA Citizen		Binary (Y,N)
Age		Discrete
Highest Education Level - Father		Categorical
Highest Education Level - Mother		Categorical
Parent Highest Ed Level	Takes the highest education level of either parent with consolidated categories	Categorical
Nevada Resident		Binary (Y,N)
ACT Composite Score		Discrete (0-36)
ACT Composite Score Range	A score range in addition to the composite score for comparison.	Ordinal, Range

ACT English Score		Discrete
ACT Math Score		Discrete
SAT Combined Score		Discrete (0-1600)
SAT Combined Score Range		Ordinal, Range
SAT Critical Reading Score		Discrete
SAT Math Score		Discrete
Core High School GPA		Discrete (0-4.8)
Unweighted High School GPA		Discrete (0-4.0)
Weighted High School GPA		Discrete (0-4.8)
Last High School - Postal Code		Zipcode
Last High School - Unweighted Percentile		Percent (0-100)
Last High School - Weighted Percentile		Percent (0-100)
Cumulative Transfer GPA		Discrete (0-4.0)
Cumulative Transfer GPA Credits		Discrete (0-4.0)
Date of Birth x		Date
Campus Resident x	Whether or not the student lived on campus (1st term).	Binary (Y,N)
Academic Load x	Academic load, full-time, part-time, etc. (1st term).	Categorical
Student Athlete x	(1st term)	Binary (Y,N)

Millennium Scholar x	If the student received the Millennium scholarship for this term (1st term).	Binary (Y,N)
Pell Recipient x	If the student received any Pell grants for this term (1st term).	Binary (Y,N)
Western Undergraduate Exchange x	If the student received WUE for this term (1st term).	Binary (Y,N)
Honors College x	If the student was in the honors college (1st term).	Binary (Y,N)
Taking Remedial x	If the student was taking at least one remedial class (1st term).	Binary
Campus Resident y	Whether or not the student lived on campus (2nd term).	Binary (Y,N)
Academic Load y	Academic load, full-time, part-time, etc. (2nd term).	Categorical
Student Athlete y	(2nd term)	Binary (Y,N)
Millennium Scholar y	If the student received the Millennium scholarship for this term (2nd term).	Binary (Y,N)
Pell Recipient y	If the student received any Pell grants for this term (2nd term).	Binary (Y,N)
Western Undergraduate Exchange y	If the student received WUE for this term (2nd term).	Binary (Y,N)
Honors College y	If the student was in the honors college (2nd term).	Binary (Y,N)
Taking Remedial y	If the student was taking at least one remedial class (2nd term).	Binary
Term GPA x	(1st term)	Discrete (0-4.0)
Cum GPA x	Cumulative GPA at the end of the 1st term.	Discrete (0-4.0)
Term GPA y	(2nd term)	Discrete (0-4.0)
Cum GPA y	Cumulative GPA at the end of the 2nd term.	Discrete (0-4.0)
PELL Elig	Eligible to receive the pell grant for this aid year.	Binary (Y,N)

Prmry EFC	Primary Estimated Family Contribution (EFC) used for financial aid determination.	Continuous
Total Income		Continuous
Student Income Contribution	To fees after considering costs of living.	Continuous
Students Total Income		Continuous
Calculated SC	Calculated student contribution to fees.	Continuous
Calculated PC	Calculated parent contribution to fees.	Continuous
Calculated EFC	Calculated Estimated Family Contribution (EFC).	Continuous
Parent Contribution	To fees.	Continuous
In Family	Number of family members.	Discrete
In College	Number of family members in college.	Discrete
Married		Categorical
Orphan	If the student was an orphan or ward of the state.	Categorical
AGI	Adjusted Gross Income.	Continuous
Care Dep	Number of dependents the student cares for.	Discrete
Dep Stat	If the student is a dependent or independent.	Categorical
SAP	Satisfactory Academic Progress - if the student meets the standards of staying on track for graduation.	Categorical
loans	Any loans recorded by the institution.	Continuous
grants	Any grants recorded by the institution.	Continuous
schol	Any scholarships recorded by the institution.	Continuous
aidYear	The financial aid-aid year.	[2011-2017]
firstTerm	The first spring or fall term the student attended (excludes summer).	The range is [2108-2175]
secondTerm	The second spring or fall term the student attended (excludes summer).	The range is [2108-2175]

startedSummer	If the student's admit term was a summer term.	Binary
graduated		Binary
class	Whether the student graduated and in what time span, 4 or 6 years	Categorical

Appendix B

Attribute Importances

The attribute importances of the used techniques.

B.1 Chi Squared Test

Attribute	Score	P-value
Academic Load y No Unit Load	2322.24552094	0.0
SAP Not Meet	1400.98842422	1.2812752103e-306
SAP Probation	797.833657045	1.59611387245e-175
Academic Load y Full-Time	356.604418464	1.54515368674e-79
Academic Load x No Unit Load	347.022910073	1.88560587287e-77
Taking Remedial x	267.542983292	3.89416408306e-60
SAP Meets SAP	249.921951317	2.7005673127e-56
Admission Type	225.31033319	6.28240154001e-51
Millennium Scholar y	176.203937667	3.26815503007e-40
Taking Remedial y	163.82814175	1.64932107205e-37
Term GPA x	144.773718735	2.40675517221e-33
Western Undergraduate Exchange x	136.551581493	1.51136134481e-31
Western Undergraduate Exchange y	133.575639951	6.76553542713e-31
Term GPA y	128.035482183	1.10254168456e-29
IPEDS Race-Ethnicity Asian	120.942139953	3.93441050378e-28

IPEDS Race-Ethnicity Black or African American	85.3441526044	2.50702269399e-20
Non-Resident Alien	74.142239666	7.26862635707e-18
IPEDS Race-Ethnicity Nonresident Alien	74.142239666	7.26862635707e-18
Cum GPA y	70.8953775776	3.76679465248e-17
Academic Load x Full-Time	62.5875836228	2.54855134244e-15
Nevada Resident	53.8582567671	2.15487617185e-13
Honors College y	49.6302431647	1.85630021142e-12
Parent Highest Ed Level Not Indicated	48.6560551648	3.05025092709e-12
Cumulative Transfer GPA	44.8605123269	2.11581913178e-11
Honors College x	39.8075847839	2.80254649167e-10
USA Citizen	39.0791953294	4.06957636613e-10
Academic Load y Half-Time	38.3572256009	5.89095647621e-10
Academic Load x Part-Time	38.0232916976	6.99051102305e-10
SAT Combined Score Range 700-799	36.8760674846	1.25881709899e-09
Academic Load y Part-Time	36.4004519337	1.60665960651e-09
Gender	32.352204753	1.28611202781e-08
Academic Load x Half-Time	31.9097664944	1.61503082295e-08
ACT Composite Score Range 12-17	31.5257113808	1.96816854512e-08
Campus Resident y	30.3769316476	3.55733519085e-08
SAT Combined Score Range 800-899	20.9051108943	4.8260523269e-06
Last High School - Weighted Percentile	18.89192945	1.38336681412e-05
Parent Highest Ed Level Graduate Level	18.4889071243	1.70896077587e-05

IPEDS Race-Ethnicity Native Hawaiian or Other Pacific Islander	17.7527982778	2.51547224591e-05
SAT Combined Score Range 900-999	16.785016892	4.18625397316e-05
Pell Recipient x	15.6468111721	7.63411982817e-05
SAP Warning	15.3280870161	9.0362966847e-05
schol	14.5253801045	0.000138283933811
SAT Combined Score Range 1100-1199	13.9119055132	0.00019158110756
IPEDS Race-Ethnicity American Indian or Alaska Native	12.1516652986	0.00049044053864
Core High School GPA	10.3290403905	0.00130953273309
Student Athlete y	10.1867936004	0.00141450064697
IPEDS Race-Ethnicity White	9.31705768165	0.00227030272126
Parent Highest Ed Level HS Level	9.13947863079	0.00250152374163
IPEDS Race-Ethnicity Two or more races	9.00912237456	0.00268635382634
Student Athlete x	8.86357188288	0.0029091775804
SAT Combined Score Range 600-699	8.768842254	0.00306419886865
PELL Elig	8.15460511079	0.00429519516391
Academic Load x Three Quarter Time	7.27453246049	0.00699390951424
ACT Composite Score Range 24-29	6.72977432689	0.00948165951282
Parent Highest Ed Level Some College	6.66349081509	0.00984079572272
IPEDS Race-Ethnicity Unknown race and ethnicity	6.60408672543	0.0101744975064
Dep Stat IND	5.97041031605	0.0145478953395

SAT Combined Score Range 1200-1299	5.69282299745	0.0170344302359
ACT Composite Score Range 30-36	5.5459305859	0.0185237049235
SAT Combined Score Range 1400-1499	4.55127180003	0.0328943045165
In Family	4.18583561071	0.0407631128494
Pell Recipient y	3.77579622183	0.0519991468939
Cumulative Transfer GPA Credits	3.60583455684	0.0575771631932
Unweighted High School GPA	3.30193532677	0.0691983099747
SAT Combined Score Range 1300-1399	3.23489225009	0.0720850450949
IPEDS Race-Ethnicity Hispanic	2.78397172765	0.0952117880088
In College	2.71939400496	0.0991356462848
Campus Resident x	2.56333731192	0.109367490703
Weight HS GPA Diff	2.13506557498	0.143964305592
Last High School - Unweighted Percentile	2.08408525923	0.148841357843
SAT Combined Score Range 1000-1099	1.57449400707	0.20955557337
Prmry EFC	1.47710434947	0.224228123823
startedSummer	1.09214116563	0.295997282306
SAT Combined Score	1.07765937157	0.299221761217
ACT Math Score	1.06935728474	0.301090665076
SAT Math Score	0.988108610718	0.320205086325
loans	0.960359338545	0.32709635931
Dep Stat DEP	0.922655788035	0.336778613945
SAT Critical Reading Score	0.800747796198	0.370869884771
Academic Load y Three Quarter Time	0.782375794589	0.376415520828
SAP unknown	0.780949167049	0.376851007896

grants	0.779401340276	0.377324293749
Parent Contribution	0.768757056424	0.380601841711
Parent Highest Ed Level Less Than HS Level	0.745130820782	0.388022234119
ACT Composite Score	0.601637165996	0.437954053999
Student Income Contribution	0.596950193696	0.439744036956
Married unknown	0.590626503569	0.442176955891
Dep Stat unknown	0.590626503569	0.442176955891
Total Income	0.482148600161	0.487450697591
ACT English Score	0.478202089918	0.489237826586
SAT Combined Score Range 1500-1600	0.370567403138	0.542695147175
Married Yes	0.368182267094	0.543996759265
Millennium Scholar x	0.205755119208	0.650115040133
Married No	0.170741389213	0.679453750481
Calculated SC	0.121373773215	0.727549271715
Students Total Income	0.061849695915	0.803595950189
SAT Combined Score Range 500-599	0.0324061535241	0.857139149679
Age	0.00185330208634	0.965661680365
Parent Highest Ed Level Bachelor Level	0.00184248153897	0.96576200792
ACT Composite Score Range 18-23	2.74985115007e-06	0.998676894559

Table B.1: Score and p-value by the chi squared test.

B.2 Decision Tree Feature Importances

Attribute	Importance
Admission Type	0.0256872321898
Gender	0.0146572939734

Non-Resident Alien	0.00169027668642
USA Citizen	0.0098159158602
Nevada Resident	0.0108186507132
startedSummer	0.00339090870063
Campus Resident x	0.00505359752128
Student Athlete x	0.00204546679355
Millennium Scholar x	0.0074017071158
Pell Recipient x	0.00719171204443
Western Undergraduate Exchange x	0.00233113840596
Honors College x	0.000707535736932
Taking Remedial x	0.00965470285629
Campus Resident y	0.00478175221768
Student Athlete y	0.00219500030384
Millennium Scholar y	0.00955464655222
Pell Recipient y	0.00748218267976
Western Undergraduate Exchange y	0.00157272956668
Honors College y	0.000789915619138
Taking Remedial y	0.00837291709928
PELL Elig	0.00519146633973
Age	0.0265180521709
ACT Composite Score	0.00826495368349
ACT English Score	0.00891002643302
ACT Math Score	0.00874100972167
SAT Combined Score	0.00920299473248
SAT Critical Reading Score	0.00948506353776
SAT Math Score	0.00984548338182
Core High School GPA	0.0162573477348
Unweighted High School GPA	0.016355530548

Weight HS GPA Diff	0.0135503234487
Last High School - Unweighted Percentile	0.0106667692976
Last High School - Weighted Percentile	0.0116051855539
Cumulative Transfer GPA	0.0179309025197
Cumulative Transfer GPA Credits	0.0175161912785
Term GPA x	0.0546319516342
Term GPA y	0.0654040006214
Cum GPA y	0.0592091490623
Prmry EFC	0.0114509236112
Total Income	0.0127352844276
Student Income Contribution	0.00504807785698
Students Total Income	0.0112281316919
Calculated SC	0.0108894730244
Parent Contribution	0.00888148224469
In Family	0.0099084468345
In College	0.00808328283602
loans	0.0160127487828
grants	0.0159680641545
schol	0.00804013144111
IPEDS Race-Ethnicity American Indian or Alaska Native	0.00115846739447
IPEDS Race-Ethnicity Asian	0.00789996317396
IPEDS Race-Ethnicity Black or African American	0.00590474447088
IPEDS Race-Ethnicity Hispanic	0.00931259641922

IPEDS Race-Ethnicity Native Hawaiian or Other Pacific Islander	0.00271905011269
IPEDS Race-Ethnicity Nonresident Alien	0.00170674963836
IPEDS Race-Ethnicity Two or more races	0.00597280846502
IPEDS Race-Ethnicity Unknown race and ethnicity	0.00289558839742
IPEDS Race-Ethnicity White	0.0115157097937
Parent Highest Ed Level Bachelor Level	0.0117331270376
Parent Highest Ed Level Graduate Level	0.000290000483746
Parent Highest Ed Level HS Level	0.0104979730832
Parent Highest Ed Level Less Than HS Level	0.00369716932682
Parent Highest Ed Level Not Indicated	0.00649652902633
Parent Highest Ed Level Some College	0.0110260913134
ACT Composite Score Range 12-17	0.00181657853202
ACT Composite Score Range 18-23	0.00368150112902
ACT Composite Score Range 24-29	0.00263928796946
ACT Composite Score Range 30-36	0.000397145977431

SAT Combined Score Range 1000-1099	0.0060868623827
SAT Combined Score Range 1100-1199	0.00279190913229
SAT Combined Score Range 1200-1299	0.00204128230973
SAT Combined Score Range 1300-1399	0.000622846113003
SAT Combined Score Range 1400-1499	8.26998452649e-05
SAT Combined Score Range 1500-1600	9.9896993928e-05
SAT Combined Score Range 500-599	8.37631063366e-05
SAT Combined Score Range 600-699	0.000389075947517
SAT Combined Score Range 700-799	0.00128625285404
SAT Combined Score Range 800-899	0.00282541311967
SAT Combined Score Range 900-999	0.00384593311345
Academic Load x Full-Time	0.00886771348429
Academic Load x Half-Time	0.00545814130541
Academic Load x No Unit Load	0.00715284289733
Academic Load x Part-Time	0.00425694568151
Academic Load x Three Quarter Time	0.00610707640688
Academic Load y Full-Time	0.0380959181101

Academic Load y Half-Time	0.00562812266552
Academic Load y No Unit Load	0.0760781049071
Academic Load y Part-Time	0.0033948623837
Academic Load y Three Quarter Time	0.00845284246029
Married No	0.00453504345391
Married Yes	0.00222688557677
Married unknown	0.00290234784083
Dep Stat DEP	0.00488083895945
Dep Stat IND	0.00321584157013
Dep Stat unknown	0.00476018350288
SAP Meets SAP	0.0162269276036
SAP Not Meet	0.0286569999113
SAP Probation	0.0179083127718
SAP Warning	0.00354259658422
SAP unknown	0.00340670401822

Table B.2: Scores given by the decision tree.

B.3 Recursive Feature Elimination

Attribute
Admission Type
Gender
Millennium Scholar y
Taking Remedial x
Taking Remedial y
Western Undergraduate Exchange x
Non-Resident Alien
Nevada Resident
Honors College y
Term GPA x
Term GPA y
Cum GPA y
Cumulative Transfer GPA
Age
Cumulative Transfer GPA Credits
Core High School GPA
Unweighted High School GPA
loans
grants
Prmry EFC
Students Total Income
Parent Highest Ed Level Bachelor Level
Academic Load x Full-Time
Academic Load x No Unit Load
Academic Load y Full-Time
Academic Load y No Unit Load
SAP Not Meet
SAP Probation
SAP Meets SAP
IPEDS Race-Ethnicity Asian
IPEDS Race-Ethnicity Black or African American

Table B.3: Chosen attributes of the RFE algorithm by logistic regression.

Bibliography

[BCHK11] Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813, 2011.

[BS12] Rebecca Barber and Mike Sharkey. Course correction: using analytics to predict course success. In *Proceedings of the 2nd international conference on learning analytics and knowledge*, pages 259–262. ACM, 2012.

[BTR04] Kash Barker, Theodore Trafalis, and Teri Reed Rhoads. Learning from student data. In *Systems and Information Engineering Design Symposium, 2004. Proceedings of the 2004 IEEE*, pages 79–86. IEEE, 2004.

[CLT$^+$14] Mohamed Amine Chatti, Vlatko Lukarov, Hendrik Thüs, Arham Muslim, Ahmed Mohamed Fahmy Yousef, Usman Wahid, Christoph Greven, Arnab Chakrabarti, and Ulrik Schroeder. Learning analytics: Challenges and future research directions. *eleed*, 10(1), 2014.

[CM04] Vladimir Cherkassky and Yunqian Ma. Practical selection of svm parameters and noise estimation for svm regression. *Neural networks*, 17(1):113–126, 2004.

[CO15] S. Colby and J. Ortman. Projections of the size and composition of the u.s. population: 2014 to 2060, Mar 2015.

[CSKM17] R. Conijn, C. Snijders, A. Kleingeld, and U. Matzat. Predicting student performance from lms data: A comparison of 17 blended courses using moodle lms. *IEEE Transactions on Learning Technologies*, 10(1):17–29, Jan 2017.

[div17] Campus ethnic diversity national universities, 2017.

[EO07] Jennifer Engle and Colleen O'Brien. Demography is not destiny: Increasing the graduation rates of low-income college students at large public universities. *Pell Institute for the Study of Opportunity in Higher Education*, 2007.

[Faw06] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861 – 874, 2006. ROC Analysis in Pattern Recognition.

[GDRG16] Dragan Gašević, Shane Dawson, Tim Rogers, and Danijela Gasevic. Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting academic success. *The Internet and Higher Education*, 28:68–84, 2016.

[GE03] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003.

[HBGL08] Haibo He, Yang Bai, E. A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, June 2008.

[HC17] Monica L Heller and Jerrell C Cassady. Predicting community college and university student success: A test of the triadic reciprocal model for two populations. *Journal of College Student Retention: Research, Theory & Practice*, 18(4):431–456, 2017.

[Her06] Serge Herzog. Estimating student retention and degree-completion time: Decision trees and neural networks vis--vis regression. *New Directions for Institutional Research*, 2006(131):17–33, 2006.

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[IPE17] Undergraduate Retention and Graduation Rates. Technical report, U.S. Department of Education. Institute of Education Sciences, National Center for Education Statistics., Apr 2017.

[JML$^+$14] Sandeep M Jayaprakash, Erik W Moody, Eitel JM Lauría, James R Regan, and Joshua D Baron. Early alert of academically at-risk students: An open source analytics initiative. *Journal of Learning Analytics*, 1(1):6–47, 2014.

[KV08] Stamos T Karamouzis and Andreas Vrettos. An artificial neural network for predicting student graduation outcomes. In *Proceedings of the World Congress on Engineering and Computer Science*, pages 991–994, 2008.

[LBD$^+$12] Eitel JM Lauría, Joshua D Baron, Mallika Devireddy, Venniraiselvi Sundararaju, and Sandeep M Jayaprakash. Mining academic data to improve college student retention: An open source perspective. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, pages 139–142. ACM, 2012.

[LBL16] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1675–1684. ACM, 2016.

[OACO08] VO Oladokun, AT Adebanjo, and OE Charles-Owaba. Predicting students academic performance using artificial neural network: A case study of an engineering course. *The Pacific Journal of Science and Technology*, 9(1):72–79, 2008.

[OO16] Asil Oztekin and Asil Oztekin. A hybrid data analytic approach to predict college graduation status and its determinative factors. *Industrial Management & Data Systems*, 116(8):1678–1699, 2016.

[Pal13] Stuart Palmer. Modelling engineering student academic performance using academic analytics. *International journal of engineering education*, 29(1):132–138, 2013.

[PE14] Zacharoula Papamitsiou and Anastasios A Economides. Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence. *Journal of Educational Technology & Society*, 17(4):49, 2014.

[PVG^{+}11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[Sie13] George Siemens. Learning analytics: The emergence of a discipline. *American Behavioral Scientist*, 57(10):1380–1400, 2013.

[TNBST09] Nguyen Thai-Nghe, Andre Busche, and Lars Schmidt-Thieme. Improving academic performance prediction by dealing with class imbalance. In *Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on*, pages 878–883. IEEE, 2009.

[UNL14] Six-year graduation rates of first-time freshmen and new undergraduate transfers, fall 2002 - fall 2008 cohorts. Technical report, University of Nevada, Las Vegas, Dec 2014.

[ZAOT04] Guili Zhang, Timothy J Anderson, Matthew W Ohland, and Brian R Thorndyke. Identifying factors influencing engineering student graduation: A longitudinal and cross-institutional study. *Journal of Engineering education*, 93(4):313–320, 2004.

Curriculum Vitae

Graduate College
University of Nevada, Las Vegas

Elliott Collin Ploutz
Email: philosopher.scholar@gmail.com

Degrees:

Bachelor of Science in Computer Science 2016
University of Nevada, Las Vegas

Bachelor of Arts in Philosophy 2012
University of Nevada, Las Vegas

Thesis Title: Machine Learning Applications in Graduation Prediction at the University of Nevada, Las Vegas

Thesis Examination Committee:

Chairperson, Fatma Nasoz, Ph.D.
Committee Member, Dr. Justin Zhan, Ph.D.
Committee Member, Dr. Evangelo Yfantis, Ph.D.
Graduate Faculty Representative, Dr. Matthew Bernacki, Ph.D.