PREDICTING SUCCESS: AN APPLICATION OF DATA MINING TECHNIQUES TO STUDENT OUTCOMES

Noah Gilbert

Department of Computer Science, California State University San Marcos,
San Marcos, California, USA

ABSTRACT

This project examines the effectiveness of applying machine learning techniques to the realm of college student success, specifically with the intent of discovering and identifying those student characteristics and factors that show the strongest predictive capability with regards to successful graduation. The student data examined consists of first time freshmen and transfer students who matriculated at California State University San Marcos in the period of Fall 2000 through Fall 2010 and who either graduated successfully or discontinued their education. Operating on over 30,000 student observations, random forests are used to determine the relative importance of the student characteristics with genetic algorithms to perform feature selection and pruning. To improve the machine learning algorithm cross validated hyper-parameter tuning was also implemented. Overall predictive strength is relatively high as measured by the Matthews Correlation Coefficient, and both intuitive and novel features which provide support for the learning model are explored.

KEYWORDS

Machine Learning, Supervised Learning, Random Forests, Higher Education

1. INTRODUCTION

The problem of improving student outcomes at the level of secondary education has gained increasing importance over the last several decades. Tuition costs for both public and private institutions have consistently outpaced inflation by several percentage points for the last 30 years [1] and student loan debt has burgeoned, with students in 2014 having a debt burden 56% higher than comparable students in 2004 [2]. Yet in the same period that has seen double digit percentage increases in tuition and student loan costs graduation rates have remained relatively stagnant, with the 6-year graduation rate across all 4-year institutions standing at an unsatisfactory 57.7% for first time students who started in 2007, an increase from 51.7% in 1996 but falling far short of desired outcomes [3].

An exhaustive study conducted in 2014 examined over 2 million student records from cohorts starting in 2007 and 2008 and identified several segments of the student population whose completion rates actually decreased, particularly at for-profit institutions [4]. Given the incredibly high opportunity cost in terms of both time spent and financial outlay of an uncompleted secondary education, and when considered in light of studies showing the significant (and widening) earnings gap between college graduates and those without a 4-year degree [5] the necessity of addressing college dropout rates has taken on a more pressing and urgent tone. It is particularly concerning, as a lack of a college education and the attendant opportunities may further social inequity and disproportionately impact underserved and minority communities. Initiatives to improve degree completion rates at universities are therefore

DOI: 10.5121/ijdkp.2017.7201

widespread and one such effort, known as Graduation Initiative 2025, is currently underway at one of the largest university systems in the United States, the California State University (CSU) system [6]. The specific goals of this initiative are multi-fold, but primarily involve improving the 6-year and 4-year graduation outcomes for first time freshmen and transfer students.

The application of data mining techniques to practical problems of this nature has been going on for some time. With the contemporary and accelerated application of these techniques to everything from recommender systems [7] to forecasting stock market outcomes [8], there are quite a few models available to researchers for exploration. The random forest data mining algorithm, while a relatively established and straightforward technique, has nonetheless continued to be one of the more popular techniques for data mining, as it provides several highly desirable traits to researchers; simplicity of implementation, strong performance in both classification and regression problems, and a somewhat easier to understand degree of transparency (as compared to neural networks, for instance, in which the feature weights are difficult to extract).

The exploration conducted in this paper builds upon established research by applying the use of a much larger and diverse set of features than are usually considered in studies of this nature. Its main contributions are expanding upon existing research by incorporating multiple data mining techniques into a single pipeline, including feature imputation, feature selection using genetic algorithms, and random forests with hyper-parameter tuning.

In the following sections of this paper we will first provide information on related research as well as similarities and distinctions to the current work. A background section follows, in order to provide a basic understanding of the concepts and methodologies used in this work, as well as an explanation of why certain approaches were chosen over others. From here the implementation will be discussed; while specific technologies and tools will be noted, the focus will be on an explanation of the conceptual flow of the experiment as a whole. Following this the results of the experiment are analyzed and interpreted both from the perspective of quantitative analysis as well as through employing domain knowledge on higher education. Finally, the conclusion and ideas for future work and improvement of the research will be discussed.

2. RELATED WORK

2.1. Application of data mining to student outcomes

As a great deal of data mining and machine learning research occurs at institutions of higher learning it seems only natural that experiments often involve the readily available data on the local student populace. As such, data mining techniques have been applied in a variety of ways to student populations in prior research.

The University of Maryland conducted extensive research on over 250,000 students enrolled at the university of whom 30,000 were transfers from partner community colleges. Using logistic regression the researchers using predictive modelling to identify the factors leading to a variety of success outcomes, including GPA, retention, and graduation. Interestingly, the researchers identified the direction of change in GPA over time as a strong predictor of retention, an attribute which was also identified as significant in the current work [9].

Quadril and Kalyankar implemented decision trees and logistic regression in order to predict the likelihood that university students would drop out prior to completing their degrees, in order to provide advisors necessary information to perform direct or indirect intervention with the at-risk students [10].

Pandey examined a dataset of 600 students to determine the relative correlation between student performance factors including language medium, caste, and class through application of a linear Bayes classification system [11]. While the results of the research satisfied the parameters set by the experiment, the relatively small data size and the use of a simple linear system incapable of accounting for correlations between the input features could conceivably have been improved on.

2.2. Random Forests and Genetic Algorithms

Research on genetic algorithms and decision trees has also been explored in great detail. Researchers at Zhejiang Gongshang University classified mobile phone customers into different usage levels using a combination of C4.5 decision trees and genetic algorithms to evolve the bitwise representations of the feature set and attribute weights [12].

Similar to the work in this paper, Bala et. al applied genetic algorithms to bit-wise encoded feature space to generate feature sub-selections, which were then fed into an ID3 decision tree to evaluate fitness. The best performers were then recombined using crossover and mutation, with the resulting new feature set re-evaluated. This continued for 10 generations after which a final tree was evaluated against the holdout data. In the work of Bala et. al the focus was on general pattern classification, and not specific to student data [13]. Similarly, the work of Hansen et. al. focused on classification of Peptides using random forests and genetic algorithms to conduct feature selection [14].

3. BACKGROUND

3.1. Feature Processing

When dealing with imperfect data several techniques may be used to deal with situations involving missing or inadequate data, or data that is in a format incompatible with the machine learning estimator being used.

3.1.1. Imputing Missing Values

Often when working with datasets of any size researchers may need to address the issue of missing values amongst the features or targets. The severity of this issue may vary from a high number of missing values (sparse data) to just a handful of missing values across several features. Different machine learning algorithms and specific implementations have varying sensitivities to missing data – some, like Naïve Bayes, deal with missing values seamlessly as it is linear and the features are treated independently. Others, particularly non-linear methods such as random forests of decision trees, may not allow for missing values.

For these situations the researcher is presented with various methods for dealing with missing data [15]. One option, removing any observations with one or more missing features, suffers from at least two shortcomings: removing observations reduces the effectiveness of a supervised algorithm's ability to train successfully; and observations with missing data may not be uniformly distributed across all target classes, leading to skew in the model's predictions. A second option is to instead interpolate the feature values based on methods as simple as using the mean of populated data in the same feature or as complex as using other machine learning techniques such as logarithmic regression to determine the values.

However, imputing too many values may also lead to model weakness. Imputing values when the number of missing values in a column is high relative to the number of total observations, or where the number of missing values for a particular observation (row) is high relative to the total number of features may distort the training of the model as imputation effectively 'creates' fake observations based on interpolation.

As the number of features in the data set with missing values was relatively small (only 3 out of over 100 features had missing values) and as the density of those features with missing values was greater than 75% (fewer than one missing value in any given feature for 4 observations) we focus instead on option 2, filling in missing values with an imputed value. For simplicity we imputed missing values using the mean of other data in the same feature, in spite of this having the potential of inducing bias [17]. Future work might involve devoting time to more computationally complex but potentially better alternatives such as using machine learning techniques to impute missing values based on other values in the observation [18].

$$f(x_i) = \begin{cases} x_i, & x_i \text{ not null} \\ mean(x_j), & x_j \text{ not null} \end{cases}$$

	SAT_Verbal
Observation 1	500
Observation 2	
Observation 3	600

	SAT_Verbal
Observation 1	500
Observation 2	550
Observation 3	600

Figure 1 - Imputing feature using mean

3.1.2. One-hot Encoding

In machine learning there are two primary classifications of features, quantitative and qualitative. Quantitative features are numeric values and can be broken down into either discrete values that may only be from a finite set (e.g. student level freshmen sophomore, etc. encoded as a numeric one through 4) or continuous numeric values within a bounded or unbounded range (e.g. age at entry or number of units completed in the first term).

Qualitative (or categorical) features, on the other hand, are usually encoded as strings and may possess a natural ordering (small, medium, large) in which case they are referred to as ordinal; they may only have two values (yes, no) in which case they are referred to as binary; or they may have no natural ordering (green, blue, red) in which case they are referred to as nominal. All three types of qualitative features are present in this research.

While some machine learning algorithms and implementations have the faculty to deal with categorical values, others do not and require the data to be preprocessed into a numeric format. The method of dealing with each type differs – for binary values we might use label encoding to change the two levels to 0 and 1. For ordinal values we use a similar technique, encoding each unique string into a numeric value matching the ordering of the feature values (e.g. small:0 , medium:1, large:2). However, for non-ordinal (values without a natural ordering) nominal values it may be dangerous to use this technique as the machine learning algorithm may interpret the values as having a natural ordering. Therefore we use a technique called one-hot encoding [19]. In this method for each unique feature value (or level) a new binary feature is created with either a 1 or 0, as seen below.

	Color
Observation 1	Red
Observation 2	Green
Observation 3	Blue

	Color_Red	Color_Green	Color_Blue
Observation 1	1	0	0
Observation 2	0	1	0
Observation 3	0	0	1

Figure 2 - One-hot encoding

27 features were encoded in this fashion, including GENDER, ETHNICITY, and MAJOR.

3.2. Feature Selection

Selecting appropriate features, also known as attributes or variables, occupies a place of key importance in the development of a successful data mining process. Whereas data mining algorithms are well-known and easily reproduced without subject matter expertise, manual feature selection requires a deep understanding of the data and the data domain. Omission of features may easily lead to outcomes with low predictive capabilities, as the model is unaware of key information in the dataset could reveal a significant pattern. On the other hand, inclusion of inconsequential features may also lead to a substandard model as it can lead to overfitting and excessive noise in the model, as well as generally reducing the speed with which the estimator is able to train and predict in a supervised learning environment [20].

3.3. Genetic Algorithms

The use of genetic algorithms has burgeoned, as the technique has proved applicable to many processes in data mining pipelines. Falling into the class of evolutionary algorithms and mimicking nature by embracing the paradigm of natural selection, genetic algorithms work on the concept of a *population*, a set of genetic representations in the solution domain hereafter referred to as *chromosomes*. Each individual genetic chromosome is encoded as an array of bits with each bit representing one aspect of the possible solution. The chromosomes are evaluated based on a fitness function, with the highest scoring chromosomes going on to 'reproduce' in a weighted but randomized fashion.

While this technique is applicable to multiple stages in the data processing pipeline, genetic algorithms are often applied (as they are in this case) to feature selection. In this paper, a binary feature mask is created which enables or disables features to which it is applied, with a 5% chance of any individual feature being enabled for any single chromosome. The first generation of the mask is generated randomly, with the chance of any individual feature enabled at a predetermined value. A snippet of a chromosome with the binary mask applied is show in Figure 3.

	Feature List																								
	ACADEMIC_LEVEL	ACAD_CAREER	ACAD_CAREER	ACAD_CAREER_LDESC	ACAD_CAREER_NBR	ACAD_PROG	ACAD_PROG	ACAD_PROG_EFFDT	ACAD_PROG_LDESC	ACTIVETERMS	ACTIVE_TERMS	CAL_GRANTS_AWARDED	CAMP	CAMP	CITIZENSHIP_CNTRY	COLLEGE_CHANGES	COMP_GE_ENGL_2	COMP_GE_MATH_2	COMP_GOLDEN4_2	COMP_LDGE_2	CSUSM_GPA_FINAL	CSUSM_GPA_FINAL	CSUSM_GPA_FIRST_TERM	CSUSM_GPA_FIRST_TERM	CSUSM_GPA_FIRST_YEAR	CSUSM_GPA_FIRST_YEAR
Feature Mask	1	0	0	0	1	1	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	1	0	1	0	1

Figure 3 - Sample feature mask

3.3.1. Crossover

The process of evolving children from the best scoring feature sets is done through crossover and mutation. Crossover is the key process in most genetic algorithms, entailing the recombination of sections of the encoded parents' chromosome into a newly defined child chromosome. Several specific implementations of crossover exist, with one of the most commonly seen in research single point crossover [21]. In single point crossover the chromosomes of two parents are combined by choosing a point randomly somewhere within the length of the parent, and then combining the gene of one parent to the left of this point with the remainder to the right of this point into the resulting child.

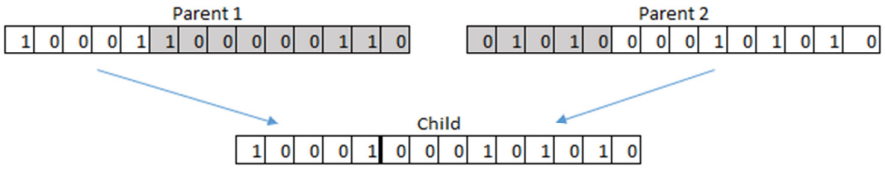

Figure 4 - Crossover

3.3.2. Mutation

Once a child has been generated through crossover it undergoes mutation. In this stage each bit of the child chromosome has a possibility of toggling from 0 to 1 or vice-versa. After some experimentation we set this value at 2%, which seemed high enough to provide enough variability in the children to incorporate features that might not have been selected in the initial random mask, but not so high that it caused good solutions to be lost

.

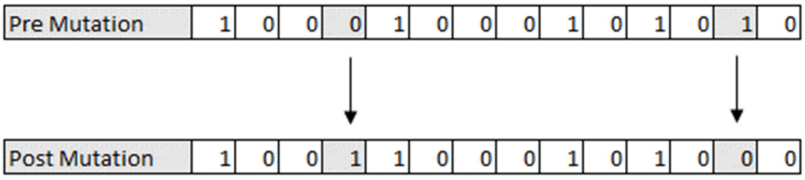

Figure 5 - Mutation

3.4. Classification Algorithms

3.4.1. Decision Trees

In machine learning, decision trees are a supervised learning method used for classification and regression which fall into the class of induction methods [22]. Decision trees are a particularly popular machine learning method due to their ability to handle both categorical and numeric features, as well as their relative ease of interpretation. Each internal node in a decision tree is composed of a feature identifier and a decision rule, or threshold, which directs observations to either the left or right child, until ultimately ending in a leaf node which identifies the classification.

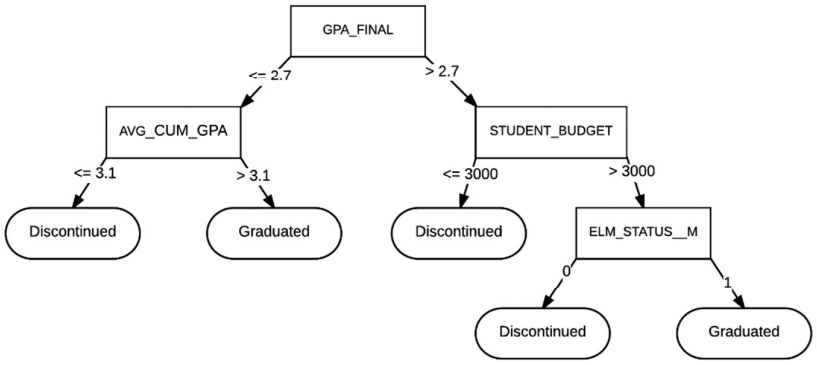

Figure 6 - Decision Tree

The construction of the tree is effected by a series of splits, wherein at each node starting at the root a specified number of features from the dataset are randomly sampled and the best split is determined. This best split is commonly the *gini impurity value*, defined as the summed square of all classification probabilities at a given node for the given feature and threshold $Gini(S) = 1 - \sum_{i=1}^{k} p_i^2$ [23]. Thus, those splits which come closest to evenly distributing the classifications along the branches are avoided in favor of those which more decisively segment the classifications, increasing the purity of the subsets.

3.4.2. Random Forests

In machine learning, random forests fall into the classification of learning algorithms known as ensemble methods, specifically combining by consensus [24]. Ensemble methods used in classification are collections of lower-level classifiers that train and predict independently – for each observation to be predicted the ensemble then returns a result based in some fashion on the classification results of the underlying estimators. Referred to as the wisdom of the crowd, this collective intelligence utilizes the majority result to provide superior results to individual underlying classifiers – the expectation is that the combination of the results will generally yield better performance as even when some portion of the underlying classifiers fail to make the correct prediction, enough of the other classifiers will pick the correct classification to override the erroneous trees.

Random forests are non-linear and as such may capture interrelationships between features that would otherwise escape detection in a purely linear classifier like Naïve Bayes. However, this comes at a cost. Unlike linear methods, most of which allow for a simple to interpret scalar value representing the correlation of a specific feature and the target variable, this simple interpretation is not available for non-linear ensemble methods. Instead, we are provided with feature importance, defined by the degree to which each feature minimizes the impurity of a node split, averaged across all trees in the forest. While not as concise as the correlation coefficient, feature importance allows us to see which features the random forest utilized most effectively in order to create predictive trees.

3.5. Hyper-parameter Optimization

While feature selection and dealing with missing or incorrect data prior to feeding to an estimator are of prime importance, other factors can also affect the ultimate performance of the classifier. Hyper-parameter optimization is the process of tuning the parameters that define the functioning of the estimator, as opposed to those values learned by the estimator; for instance, a hyper-parameter for random forest classifiers is the number of decision trees the random forest will generate, another is the number of features each decision tree in the forest will consider when generating a new node and split. Unlike values that are learned by the estimator during training, hyper-parameters are generally user defined and passed to the estimator upon initialization. While some hyper-parameters potentially impact the estimator's scoring performance, others are provided more for the speed with which the classifier may be trained.

Automated processes for hyper-parameter tuning function by running multiple iterations of the estimator with different combinations of the parameter sets and a scoring function and then relying on cross-validation to determine the highest scoring hyper-parameter set sampled. Some implementations are exhaustive, testing every possible combination of parameters against the model; however, this approach, while likely to find an optimal or near-optimal solution, nonetheless suffer from being incredibly taxing in terms of the performance with which the classifier can be trained, particularly for estimators for which a large number of hyper-parameters exist. An alternative, randomized grid search, works instead by randomly sampling from the provided parameter set a predetermined number of times and returning the best scoring parameter set found after cross-validation, as above.

3.6. Fitness Function

The choice of fitness function is heavily dependent on the classification problem in question. A common, albeit crude, fitness metric is accuracy, simply the number of correctly predicted observations in relation to the total number of samples. While this is appropriate in some circumstances, accuracy will often not adequately capture distortions in the data, particularly those involving unbalanced data sets in a binary classification algorithm, as it may yield high scores by simply predicting all samples in one direction (towards the over emphasized class in the samples). While this may be partially mitigated by using sampling techniques such as bagging which may somewhat even out the classes by in order to even out the sample classes.

The F1 score strikes a balance in this regard, as it provides a consolidated metric incorporating both recall and precision, thus ensuring that in cases of unbalanced classes consideration is given both to the ability of the estimator to correctly identify all instances of true positives as well as its ability to correctly exclude instances of false positives.

However, a shortcoming of the F1 score is that it focuses primarily on a single class, is focused on the majority class, and doesn't take into account true negatives [25]. This is problematic in the current paper as not only are the classes unbalanced for certain targets but additionally we are looking for strong predictive capabilities for the non-completion (true negative) events, which F1 completely ignores, as can be seen from
Figure 7.

$$F1 = 2\frac{pr}{p+r} \text{ where } p = \frac{tp}{tp+fp}, \; r = \frac{tp}{tp+fn}$$

Figure 7 - F1 Score Equation

Thus, after initially running all experiments using the F1 Score as the fitness function, I ultimately reran all tests using the Matthews Correlation Coefficient as the score to direct the genetic algorithm's choices of parents to evolve. Unlike the F1 Score, the Matthews Correlation Coefficient takes into account true negatives, and is regarded as a strong single-value measure of predictor performance in a two-class classification system [26].

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Figure 8 – Matthews Correlation Coefficient Equation

Interpretation of the MCC is relatively straightforward – MCC scores are between -1 and 1, with 1 representing perfect prediction, 0 representing results no better than random, and -1 representing a perfectly incorrect prediction.

4. ARCHITECTURAL MODEL

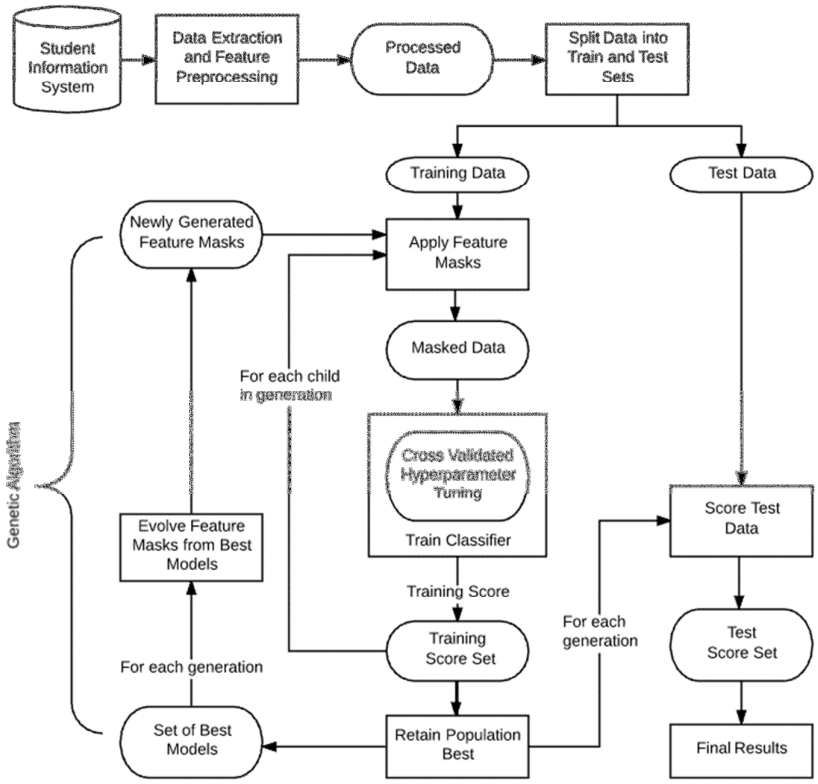

Figure 9 - Architectural model

5. METHODS

Source data was collected from the California State University San Marcos PeopleSoft Campus Solutions student information system, multifaceted software delivering functionality involved in all aspects of student administration and electronic student records. SQL views were used to extract and transform data from over 20 student data tables covering enrollments, grades, demographic information, application information, financial aid, and department of study. For the purpose of this research, only students who had either successfully completed their degree or discontinued their academic career were included.

As patterns of student success may change over time, included students were limited to first time freshmen and transfer students pursuing an Undergraduate degree who matriculated between Fall 2000 and Fall 2010. Students who started after Fall 2010 but were no longer enrolled by Fall 2016 would likely represent predominantly discontinuations (as they wouldn't have had the time to complete a 6-year graduation) and thus skew the results for the graduation rates. For the students in the dataset various outcome values (targets) were also collected, including whether they graduated and if so how many years were required to graduate.

Data collected in this way consisted of 31,048 completed academic careers (culminating as either a graduation or discontinuation) comprised of 19,548 transfer students and 11,502 first time freshmen. The transfer students were broken down into 13,978 graduation events and 5,576 discontinuations (71.5% graduation rate). The first time freshmen were broken down into 5,913

graduations and 5,589 discontinuations (51.4% graduation rate). 137 columns comprised of both quantitative and qualitative values were retrieved for each row.

Several simple data cleansing and refactoring transformations were implemented at the SQL view level. Boolean 'Yes/No' and 'True/False' columns were recoded as binary 1's and 0's; similarly, NULL values for Boolean fields were also recoded as 0.

The features then underwent processing to impute values for missing data based on the mean of non-missing values in the same feature. Features including GPA and SAT scores were imputed in this fashion. Depending on the type of experiment being run, GPA and unit load features were then dropped from the dataset. Subsequently, categorical features were one-hot encoded, increasing the number of features from 137 to between 311 and 333, depending on whether the experiment would include or exclude GPA and unit load information.

At this point the data was split into two sets randomly, with an 85%/15% split of training and test (holdout) data. As described above, a random boolean mask was then generated and applied to the feature set in order to generate the initial 20 chromosome population. With the chromosome applied to the training data to yield only those features that were enabled, the data was then passed to a randomized search hyper-parameter tuner. Internally, the randomized search generated 10 random forests, each of which was passed a random sampling of parameters from the candidate set. Each of the 10 random forests was then fit and scored against the training data using 3-fold cross validation, with the highest scoring random forest and associated hyper-parameter set returned. The best estimator was then rescored against the training data using the Matthews Correlation Coefficient and saved. This process was repeated for each chromosome in the population, with each chromosome's MCC score saved as above.

Once the population was fully scored, the results were tabulated and the highest scoring chromosome (feature set), trained random forest classifier, and hyper-parameters were persisted to a global best variable. A new population of 20 chromosomes was then evolved from the highest scoring 50% of the population and all but the global best were then discarded. The global best trained classifier was then used to predict against the holdout test data. The resulting scores were then saved to a separate, persistent result set in order to capture the ultimate shift in population fitness from generation to generation. This process was repeated for 20 generations of populations, with each generation's best scores saved to the result set as above. The best score of each population after the first was compared to the existing global best, which was replaced if the new score was higher, otherwise it remained unchanged.

Once all 20 generations of 20 chromosomes were processed in this way, the final global best feature set and hyper-parameters were applied to a single decision tree which was then trained and scored against the training and test data. While this yielded a somewhat lower score than the global best random forest classifier due to its lack of the collective intelligence of the ensemble, a single decision tree nonetheless permits for interpretation of the resulting nodes and logical rules, which is very difficult to do when dealing with a random forest composed of numerous decision trees. The results of all generations and all scores are then plotted, and the decision tree is graphed.

Early in implementation it became apparent that GPA and unit load features had such high predictive capability other, more novel, features were generally being crowded out of the model. Thus for each student population (transfers and first time freshmen) I ran the experiment both with and without GPA and unit load information, by dropping these features from the dataset when appropriate; however, when removing the GPA and unit load features I only omitted the raw or averaged values while retaining those features showing rate of change of these measures.

Furthermore, I was interested in 5 common target outcomes for each student population as well as 2 outcomes specific to the student population, each of which was coded in the underlying data as a binary 0/1 column representing whether that particular target was attained by the student:

- Outcomes common to all students
  - Graduated (Graduated)
  - Graduation within 4 years of entry (Within 4 Years)
  - 1 year retention (Retention 1 Year)
  - 2 year retention (Retention 2 Years)
  - 3 year retention (Retention 3 Years)
- Transfer students
  - Graduated within 2 years of entry (Within 2 Years)
  - Graduated within 3 years of entry (Within 3 Years)
- First Time Freshmen
  - Graduated within 5 years of entry (Within 5 Years)
  - Graduated within 6 years of entry (Within 6 Years)

Note that different time to graduation targets are relevant to the two student populations. For transfer students we are interested in graduation between 2 and 4 years, while for first time freshmen our target is between 4 and 6 years. Thus the experiment was run 28 times in total (2 student populations, each with and without GPA and unit load, against 7 targets). While all common scoring metrics are included in the figures below, the primary focus of the evaluation is in consideration of the Matthews Correlation Coefficient (MCC) scores, for the reasons described in the background section of this paper. Other metrics are discussed in situations where the MCC does not provide sufficient information or where further investigation is warranted.

6. RESULTS

The Completion Rate shown in the figures below represents the percentage of observations in the test data for which the target in question was completed, e.g. for the Graduated target 2809 records of the 5676 in the holdout test data successfully graduated leading to a 49.49% completion in

Figure 10 - FTF with GPA and Units – Metric Scores. This provides the context necessary to understand issues of class imbalance.

6.1. First Time Freshmen with GPA and Unit Load information

First Time Freshmen w/GPA and Units

	Graduated	Within 4 Years	Within 5 Years	Within 6 Years	Retention 1 Year	Retention 2 Years	Retention 3 Years
Completion Rate	49.49%	15.58%	37.23%	46.59%	68.54%	57.83%	53.48%
Accuracy	0.8844	0.9009	0.8570	0.8762	0.9403	0.8945	0.8804
F1 Score	0.8917	0.5643	0.8228	0.8758	0.9583	0.9124	0.8936
Precision	0.8471	0.6974	0.8015	0.8270	0.9352	0.8845	0.8535
Recall	0.9413	0.4738	0.8452	0.9306	0.9824	0.9422	0.9376
Matthews Coefficient	0.7734	0.5228	0.7038	0.7584	0.8568	0.7825	0.7621
ROC Score	0.8837	0.7208	0.8549	0.8794	0.9127	0.8849	0.8760

Figure 10 - FTF with GPA and Units – Metric Scores

Scoring for first time freshmen was strong across most metrics and targets, with particular strength in MCC scores shown in overall graduation and retention targets. Retention 1 Year yielded very strong results; intuitively this can be explained by the fact that this is the first target

to be resolved chronologically in a student's academic career, i.e. the other targets cannot be reached if the student discontinues prior to reaching their first year, making it somewhat easier to predict. Furthermore, the Retention 1 Year target had the highest completion rate, with the samples skewed towards observations falling into the successfully completed classification.
Conversely, the Within 4 Years target had by far the highest classification skew, with the majority of students not graduating within 4 years as shown in the
Figure 10 Completion Rate for Within 4 Years value of 15.58%. This is likely the cause of the much weaker MCC, recall, and f1 scores. The relatively high accuracy and precision scores indicate that likely due to the class imbalance the model ended up misidentifying far too many observations as incomplete for this target.

	Graduated	Within 4 Years	Within 5 Years	Within 6 Years	Retention 1 Year	Retention 2 Years	Retention 3 Years
AVG_CSUSM_GPA	0.068			0.065			
AVG_CUM_GPA							0.068
AVG_DWF_PER_TERM				0.108			0.061
AVG_TERM_GPA	0.074		0.058				
AVG_TERM_GRADE_POINTS		0.084	0.143			0.079	0.086
AVG_TERM_UNITS_PASSED_GPA		0.126			0.087		
AVG_TERM_UNITS_TAKEN_GPA	0.065	0.086		0.084			
AVG_TERM_UNITS_TOTAL	0.083	0.218	0.162	0.107	0.071		
CSUSM_GPA_FINAL	0.105		0.147	0.164			
CSUSM_GPA_FIRST_YEAR	0.076				0.218	0.217	
HIP_RECEIVED_ADVISING						0.062	
MAX_TERM_GPA	0.090		0.063	0.081	0.080	0.093	0.094
PCT_CHANGE_LAST_OVERALL_GPA	0.061						
PLAN_CHANGES	0.056						
PREV_TERM_GPA	0.068			0.081			
TOT_UNITS_YEAR1					0.120		0.056
UNITS_LDGE_TOTAL_2Y						0.090	0.053

Figure 11 – FTF with GPA and Units - Feature Importances

Feature importances, as explained in the background section on random forests, showed some interesting patterns across the targets. The cell containing the highest feature importance value for each target is expressed through a heavy border in the figure above. We immediately note that 4 features in particular stand out in Figure 11, AVG_TERM_UNITS_TOTAL, CSUSM_GPA_FINAL, CSUSM_GPA_FIRST_YEAR, and MAX_TERM_GPA.

Average term units total are particularly important for graduation within 4 or 5 years, and of less importance but still significant for graduation at all, graduation within 6 years, and 1 year retention targets. This appears to be intuitively correct, as the number of units taken per term has a direct impact on the number of years it takes to graduate. Notably, however, it is only weakly relevant to 1 year retention and does not appear to factor into 2 or 3 year retention, an observation warranting further exploration as it might not seem obvious that the number of units taken per term has little or no relation to whether or not the student is retained.

The CSUSM final GPA feature is of key importance to graduation and graduation within 6 years targets, and to a lesser extent graduation within 5 years. Reviewing the decision tree for these targets we note that a low final CSUSM GPA often indicates a high likelihood of failure to complete the target as shown in 12. Again this is intuitively true, as students whose overall GPA is near a C+ average may not have met all the requirements for graduation.

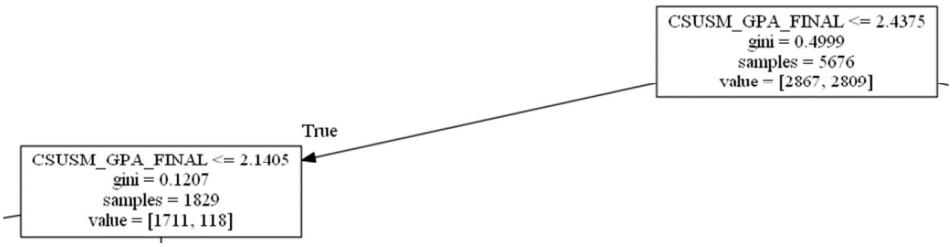

Figure 12 - FTF w/GPA and Units - Graduated Decision Tree Fragment

Finally, the CSUSM first year GPA is the most important feature of the 1 and 2 year retention targets; interestingly, it is also important to graduating at all, although to a lesser extent. This may indicate that students who do well in their first year are far more likely to return for their second and third years, although notably this effect appears to fade by the fourth year, as reflected by this feature having no importance to 3 year retention.

Also of interest, while the max term GPA (the highest single term GPA across the student's academic career) is not the most important feature for any target, it is nonetheless relevant to all targets with the exception of the graduation within 4 years outcome. This may be due, however, to the simple fact that the max term GPA may be strongly correlated with overall higher grades across all terms, as students who perform exceptionally well in a single term may generally be high performers throughout all terms.

While most targets had only 5 to 7 features that the random forest classifier used broadly enough in its ensemble of trees to be considered of some importance, the Graduated target had 10. This could possibly indicate that in order to build a sufficiently predictive model for this target a greater number of factors, each with a somewhat smaller individual importance, needed to be considered. Interestingly, the Within 4 Years target which delivered much lower metric scores than the other targets also had the fewest number of important features, with one feature discussed above, AVG_TERM_UNITS_TOTAL, yielding a very high significance.

Investigating, we find more clues from the decision tree generated for this target, the root node and left child of which are shown below:

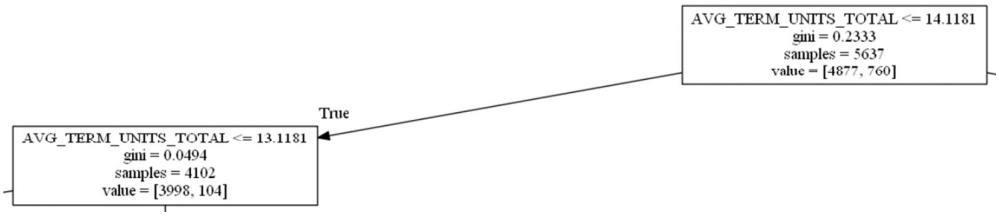

Figure 13 - FTF w/GPA and Units – Within 4 Years Decision Tree Fragment

On reflection it comes as no surprise that the average number of units taken per term is of such importance to the ability of a student to graduate within 4 years. The typical academic career in pursuit of an Undergraduate degree requires 120 units – in order to complete this degree within 4 years a student generally must take an average of 15 units per regular academic term. Viewing Figure 13 above we see that the decision tree root node splits on the threshold of 14.1181 units per term, with observations less than or equal to this value travelling down the left path. Note that value = [4877, 760] indicates that of the 5637 samples that entered the root node, 4877 had a classification of 0, indicating failure to complete within 4 years, and 760 had a classification of 1, indicating successful completion. Observations with an average unit load lower than 14.1181 then travel to the left node, pictured above, at which point only 2.5% (104/4102) represent successful completions. This is as expected given that students with fewer than 14 units per term

at CSUSM would need to make an exceptional effort to complete within 4 years (perhaps by taking classes over Summer at a community college).

As noted earlier, initial implementation efforts led to the realization that of the hundreds of feature candidates for the classifier, frequently only those involved with GPA and unit loads were being given consideration as they proved to have the highest feature importance.

6.2. First Time Freshmen without GPA and Unit Load information

	Graduated	Within 4 Years	Within 5 Years	Within 6 Years	Retention 1 Year	Retention 2 Years	Retention 3 Years
Completion Rate	49.49%	15.58%	37.23%	46.59%	68.54%	57.83%	53.48%
Accuracy	0.8643	0.8701	0.8267	0.8561	0.8684	0.8562	0.8626
F1 Score	0.8708	0.1265	0.7805	0.8517	0.9088	0.8812	0.8789
Precision	0.8217	0.7791	0.7574	0.8199	0.8739	0.8472	0.8398
Recall	0.9261	0.0689	0.8050	0.8861	0.9467	0.9180	0.9217
Matthews Coefficient	0.7348	0.2068	0.6384	0.7145	0.6810	0.7034	0.7252
ROC Score	0.8651	0.5329	0.8226	0.8580	0.8194	0.8442	0.8574

Figure 14 - FTF w/o GPA and Units - Metric Scores

Omitting the strongly predictive GPA and unit load features had the expected effect of lowering the scores somewhat across all metrics and targets. Additionally, we see a reversal of MCC strength across the retention targets; whereas in the FTF experiment including GPA and unit information we see the scores decreasing from 1 to 3 year retention, omitting GPA and unit information reverses this trend, with scores increasing from the 1 to 3 year retention targets. This may be due to the fact that removing the most highly predictive features, particularly the CSUSM_GPA_FIRST_YEAR which played such an important role in the feature matrix for the 1 and 2 year retention targets previously, disparately impacted the classifier's ability to form a random forest as capable of predicting the shorter term retention periods.

	Graduated	Within 4 Years	Within 5 Years	Within 6 Years	Retention 1 Year	Retention 2 Years	Retention 3 Years
AVG_DROPS_PER_TERM		0.0517			0.0635	0.0530	0.0526
AVG_DWF_PER_TERM	0.2017	0.2072	0.2475	0.2828	0.0863	0.0935	0.1108
DROPPED_COURSES					0.0537	0.0856	0.0555
DWF_GRADES		0.1232		0.0572			
ENROLLED_SERVICE_LEARNING							0.0557
GPA_CHANGE_FIRST_LAST_TERM	0.0819		0.0996				
HIP_RECEIVED_ADVISING		0.0548	0.1030			0.1071	0.0937
PCT_CHANGE_FIRST_LAST_GPA	0.1056			0.1016	0.0566		
PCT_CHANGE_LAST_OVERALL_GPA				0.1133	0.0978	0.0581	0.0711
PCT_CHANGE_LAST_PREV_TERM_GPA	0.0567		0.1011				
PLAN_CHANGES	0.1254			0.1030	0.1171	0.0892	0.0884
UNITS_LDGE_TOTAL_2Y			0.0557	0.0779	0.1544	0.0851	0.0849
HS_GPA		0.0711					

Figure 15 - FTF w/o GPA and Units - Feature Importances

When depriving the model of the highly predictive GPA and unit information we see other, more novel features emerge. AVG_DWF_PER_TERM, a feature valued at the average number of D's, F's or withdrawals a student earns per term, provides the primary strength for all graduation outcomes. While not explicitly a measure of GPA, this feature nonetheless is intuitively correlated as students receiving higher numbers of non-passing grades in courses will undoubtedly have generally lower GPAs than those students with fewer non-passing grades. In this way, the average DWF per term feature acts as a proxy for the highly predictive GPA features omitted from this run of the experiment.

Yet in addition to the DWF feature other features which might not be as intuitively connected to student success also demonstrate importance. AVG_DROPS_PER_TERM, a measure of the average number of courses dropped by a student prior to receiving a grade is relevant to all three retention outcomes, as is DROPPED_COURSES, the total number of courses dropped during the student's academic career. While it is likely that these features are strongly correlated with one another and thus somewhat redundant, it nevertheless provides the interesting observation that a student's retention may be predicted by merely unenrolling from courses. Also interesting is the importance of the PLAN_CHANGES feature, measuring the number of times a student has changed their major. Taken together these two features may involve student uncertainty which might lead to a higher chance of dropping out.

The three features involving the direction and magnitude of change in GPA over time (PCT_CHANGE_FIRST_LAST_GPA, PCT_CHANGE_LAST_OVERALL_GPA, and PCT_CHANGE_LAST_PREV_TERM_GPA) show varying levels of importance across the target outcomes. While related to GPA, these features are particularly interesting in that as they only measure change a student they can only show a relative improvement or decline of GPA, e.g. a student with a consistent 4.0 GPA who receives a B will see a negative percentage, while a student with a 2.0 GPA who receives a B will see a positive percentage, and the student with a 3.0 GPA who receives a B will see no change at all.

We can see this at work in
Figure 16 below, showing a fragment of the Within 5 Years decision tree. While the percentage of completions at the parent node is 25.96% (283/1090), a student whose last term GPA drops by 19.49% or more from the previous term (e.g. from a 3.0 term GPA to a 2.4 term GPA) moves along the left branch to a node of which only 9.09% (25/275) continue on to graduate within 5 years.

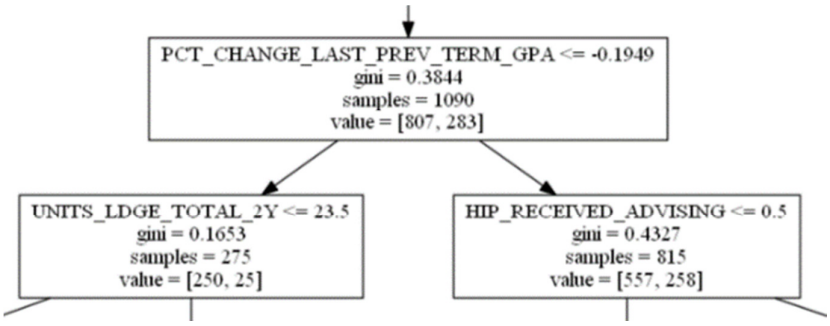

Figure 16 - FTF w/o GPA and Units – Within 5 Years - Decision Tree Fragment

6.3. Transfer Students with GPA and Unit Load Information

	Graduated	Within 2 Years	Within 3 Years	Within 4 Years	Retention 1 Year	Retention 2 Years	Retention 3 Years
Completion Rate	71.14%	25.24%	55.67%	64.86%	74.71%	66.30%	64.94%
Accuracy	0.8914	0.9229	0.8690	0.8763	0.9522	0.9116	0.8995
F1 Score	0.9266	0.8455	0.8858	0.9101	0.9684	0.9365	0.9244
Precision	0.8817	0.8436	0.8544	0.8746	0.9558	0.9033	0.8885
Recall	0.9763	0.8475	0.9195	0.9487	0.9813	0.9723	0.9634
Matthews Coefficient	0.7342	0.7942	0.7354	0.7186	0.8724	0.7978	0.7807
ROC Score	0.8340	0.8977	0.8630	0.8420	0.9245	0.8804	0.8750

Figure 17 - Transfer students with GPA and Units – Metric Scores

As with freshmen when including GPA and unit information, the strongest predictive value by far was against the Retention 1 Year target. Interestingly, as with freshmen the predictive strength of the model decreased as retention period increased (from 1 to 3 years), but unlike the first time freshmen the predictive strength of the model also decreased as the time to graduate increased (from within 2 years to within 4 years), whereas with freshmen this latter pattern was reversed, with predictive strength increasing across time to graduate (from within 4 years to within 6 years).

	Graduated	Within 2 Years	Within 3 Years	Within 4 Years	Retention 1 Year	Retention 2 Years	Retention 3 Years
AVG_CSUSM_GPA	0.0803						
AVG_DWF_PER_TERM	0.0590						
AVG_TERM_GPA	0.0633						0.0624
AVG_TERM_GRADE_POINTS						0.0577	
AVG_TERM_UNITS_PASSED_GPA	0.1560	0.1131		0.1660		0.0633	0.0748
AVG_TERM_UNITS_TAKEN_GPA		0.1331	0.1515				
AVG_TERM_UNITS_TOTAL	0.0909	0.3104	0.1618	0.1184	0.0956	0.1005	0.0907
CSUSM_GPA_FIRST_YEAR	0.1123		0.0501	0.1041	0.4867	0.1787	0.2166
DWF_GRADES		0.0516					
ELM_STATUS__P						0.0709	
ELM_STATUS__T							0.0544
EPT_STATUS_LDESC__Exempt						0.0714	0.0566
GPA_CHANGE_FIRST_LAST_TERM				0.0572			
LAST_TERM_GPA			0.0725	0.1293			
MAX_CSUSM_GPA	0.0515						
MIN_TERM_GPA			0.1105	0.0721			
PREV_TERM_GPA	0.1589			0.1346		0.0959	0.0983
TOT_24_DEGR_UNITS_YEAR1			0.0631				
TOT_24_UNITS_YEAR1			0.0653	0.0664			
TOT_UNITS_YEAR1		0.0969	0.0816		0.1012	0.1009	

Figure 18 - Transfer students with GPA and Units - Feature Importances

Reviewing the feature importance grid for transfer students we note several interesting findings. As seen with freshmen, the average term units taken feature occupies a place of significant influence for all targets. As discussed previously, this is intuitively due to the fact that the time it takes to graduate is dependent on the number of units taken per term. Similarly, we also see the CSUSM_GPA_FIRST_YEAR providing high significance for all but one target. In fact, its contribution of predictive strength to the Retention 1 Year target is such that in spite of there only being 3 important features for this target, the Retention 1 Year MCC score is the highest of all targets. Note that in the decision tree fragment below showing the root node, a first year CSUSM GPA of less than or equal to 2.1045 leads from a node where 74.7% (6642/8890) of the student samples are successfully retained for 1 year to a node where only 4.1% (76/1832) will still be retained. This may represent transfer students who are ill-prepared for the academic rigors of a 4 year institution and quickly drop out after poor performance within their first year after transfer.

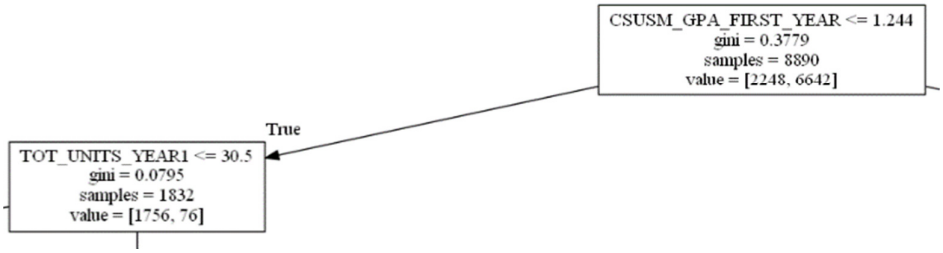

Figure 19 - Transfer students w/GPA and Units - Decision Tree Fragment

6.4. Transfer Students without GPA and Unit Load Information

	Graduated	Within 2 Years	Within 3 Years	Within 4 Years	Retention 1 Year	Retention 2 Years	Retention 3 Years
Completion Rate	71.14%	25.24%	55.67%	64.86%	74.71%	66.30%	64.94%
Accuracy	0.8450	0.7814	0.7957	0.7935	0.8407	0.8337	0.8370
F1 Score	0.8971	0.3513	0.8276	0.8588	0.9003	0.8852	0.8813
Precision	0.8407	0.6733	0.7814	0.7746	0.8404	0.8205	0.8251
Recall	0.9615	0.2376	0.8797	0.9636	0.9693	0.9610	0.9458
Matthews Coefficient	0.6112	0.3045	0.5852	0.5337	0.5478	0.6148	0.6401
ROC Score	0.7656	0.5997	0.7848	0.7194	0.7199	0.7695	0.7948

Figure 20 - Transfer students w/o GPA and Units – Metric Scores

Scoring for transfer students without GPA and unit load information, while not entirely unsatisfactory, was nevertheless much lower than any other experiment. Omission of GPA and units led to a similar outcome as that with freshmen, lower scores overall than the experiment when GPA and units were included and a sensitivity to the unbalanced class representing the shortest graduation period. As with freshmen we see an interesting shift in MCC scores for retention; once again omission of the GPA and units led to a reversal of predictive power for retention periods. Whereas the experiment with transfer students including GPA and units showed decreasing MCC scores moving from 1 year to 3 year retention, as with the freshmen excluding these features leads to increasing scores across this range.

	Graduated	Within 2 Years	Within 3 Years	Within 4 Years	Retention 1 Year	Retention 2 Years	Retention 3 Years
ADMIT_ACAD_LOAD__P		0.1249					
ADMIT_ACAD_LOAD_LDESC__Enrolled Part-Time		0.1075	0.1263				
AVG_DROPS_PER_TERM					0.0650		
AVG_DWF_PER_TERM	0.2283	0.1179	0.2132	0.2958	0.1501	0.1652	0.1548
DWF_GRADES		0.1466	0.1084	0.1422			
ELM_STATUS__P						0.0893	
ELM_STATUS__T						0.0558	
EPT_STATUS__T						0.1001	
GPA_CHANGE_FIRST_LAST_TERM	0.0872	0.0519	0.1018	0.0982	0.1123	0.0510	0.0774
HIP_RECEIVED_ADVISING							0.0968
NUM_PROBATION_TERMS			0.0748				
PCT_CHANGE_FIRST_LAST_GPA	0.1059			0.1129	0.1218	0.0809	0.0726
PCT_CHANGE_LAST_OVERALL_GPA	0.1582		0.1379		0.0664	0.0774	0.0822
PCT_CHANGE_LAST_PREV_TERM_GPA	0.1094			0.1026	0.1057	0.0651	0.0679
STUDENT_BUDGET					0.0732		

Figure 21 - Transfer students w/o GPA and Units - Feature Importances

Reviewing the feature importance matrix we once again see a very familiar factor, as the average number of D's, F's, and withdrawals per term is the most significant feature in all but one of the targets. Notably relevant are the features representing percent change in GPA over time; these features are of some interest, as instead of measuring raw GPA scores they instead measure a change in GPA that might indicate that the trajectory of GPA may be an indicator of success.

6.5. Genetic Algorithm Performance

In order to review the effectiveness of the genetic feature selection algorithm I also inspected the plots of the various targets' Matthews Correlation Coefficients (MCC) over generations of the algorithm, comparing the MCC score of the training data used to direct the evolution of new populations against the MCC score against the holdout test data. As an MCC plot was generated for each of the 28 runs of the experiment (2 student populations, with/without GPA and unit information, 7 targets apiece) only several of the more interesting or illustrative charts are included below.

In many cases the genetic algorithm functioned as expected – as training data scores through successive generations of chromosomes, the holdout test data scores improved in conjunction as seen in Figure 22. However, in other cases training scores continued to improve while the test scores plateaued or even dropped somewhat, possible signs of overfitting the data as seen in Figure 23.

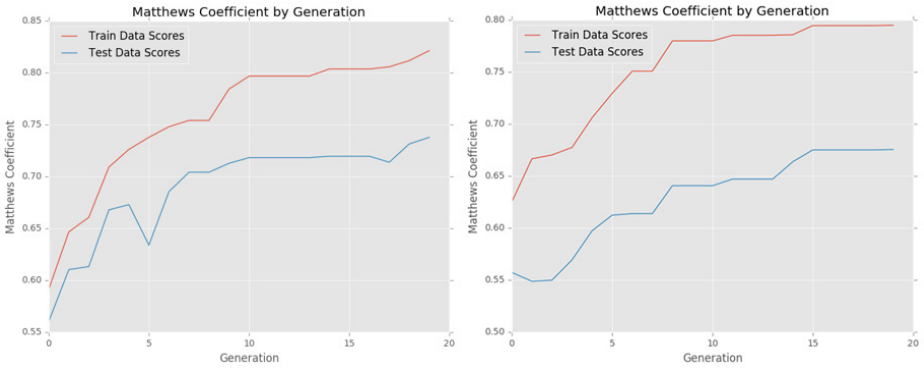

Figure 22 - MCC over generations – score improvement

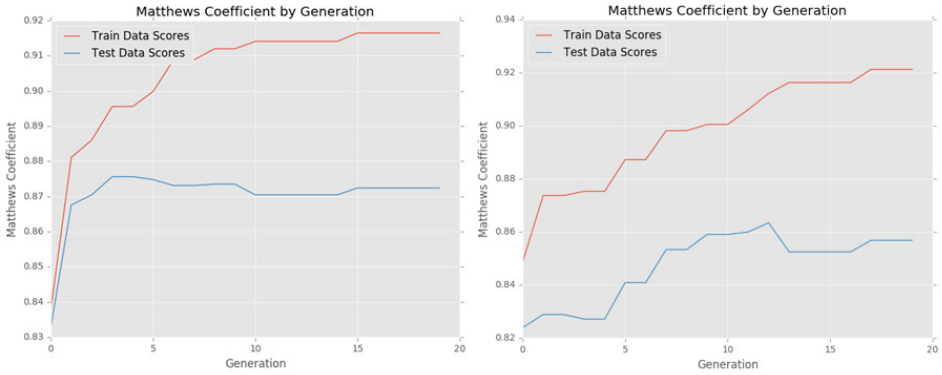

Figure 23 - MCC over generations - overfitting and plateauing

It should be noted that as we used the MCC as the scorer for the fitness function, each generation attempted to optimize towards this value. As such, other metrics sometimes showed declines over the same period.

7. CONCLUSION

Identifying factors that can help to identify students at risk of failing to complete their secondary education endeavors has taken on increasing importance in the last decade due to the rising costs of education and the increasingly grim prospects in the job market for those without a college degree. While machine learning techniques have been applied to this problem, a clear methodology for discovering the pattern of risk characteristics in student data is still lacking.

In this paper we present a system capable of predicting, with some success, which students will encounter difficulties and which will go on to achieve several common metrics of success based on a broad set of data encompassing both immutable and changing characteristics of the student. By using a non-linear ensemble method we are able to capture interactions between factors that might be otherwise missed in a linear system. And by use of genetic algorithms we optimize the feature selection process to strike a balance between recall and precision. The results of our efforts show strong predictive capability, particularly for 1 and 2 year retention periods and graduation outcomes. We also uncover several novel features such as trajectory of GPA and number of dropped classes as providing some importance to our models and warranting further investigation.

REFERENCES

[1] J. Lorin , "College Tuition in the U.S. Again Rises Faster Than Inflation," 12 November 2014. [Online]. Available: http://www.bloomberg.com/news/articles/2014-11-13/college-tuition-in-the-u-s-again-rises-faster-than-inflation.

[2] B. Snyder, "Student loan debt has increased — again," 27 October 2015. [Online]. Available: http://fortune.com/2015/10/27/student-loan-debt-increase/.

[3] "Graduation Rates," December 2015. [Online]. Available: http://nces.ed.gov/programs/digest/d15/tables/dt15_326.10.asp.

[4] D. Shapiro, A. Dundar, X. Yuan, A. Harrell and P. Wakhungu, "Completing College: A National View of Student Attainment Rates – Fall 2008 Cohort," National Student Clearinghouse Research Center, Herndon, VA, 2014.

[5] "The Rising Cost of Not Going to College," 11 February 2014. [Online]. Available: http://www.pewsocialtrends.org/2014/02/11/the-rising-cost-of-not-going-to-college/.

[6] "Graduation Initiative 2025," [Online]. Available: https://www2.calstate.edu/graduation-initiative-2025.

[7] H.-R. Zhang and F. Min, "Three-way recommender systems based on random forests," Knowledge-Based Systems, vol. 91, pp. 275-286, January 2016.

[8] K. S. Kannan, P. S. Sekar, M. M. Sathik and P. Arumugam, "Financial Stock Market Forecast using Data," in Proceedings of the IMECS, Hong Kong, 2010.

[9] "PREDICTIVE ANALYTICS FOR STUDENT SUCCESS: Developing Data-Driven Predictive Models of Student Success," University of Maryland University College , 2015.

[10] M. N. Quadril and N. V. Kalyankar, "Drop Out Feature of Student Data for Academic Performance Using Decision Tree Techniques," Global Journal of Computer Science and Technology, vol. 10, no. 2, pp. 2 - 5, April 2010.

[11] U. K. Pandey, "Data Mining : A prediction of performer or underperformer using classification," International Journal of Computer Science and Information Technologies, pp. 686-690, 2011.

[12] D.-s. Liu and S.-j. Fan, "A Modified Decision Tree Algorithm Based on Genetic Algorithm for Mobile User Classification Problem," The Scientific World Journal, pp. 1 - 11, 2014.

[13] J. Bala, J. Huang, H. Vafaie, K. DeJong and H. Wechsler, "Hybrid Learning Using Genetic Algorithms and Decision Trees for Pattern Classification," in IJCAI conference, Montreal, 1995.

[14] L. Hansen, E. A. Lee, K. Hestir, L. T. Williams and D. Farrelly, "Controlling Feature Selection in Random Forests of Decision Trees Using a Genetic Algorithm: Classification of Class I MHC Peptides," Combinatorial Chemistry & High Throughput Screening, pp. 514-519, 2009.

[15] B. Marlin, "Missing Data Problems in Machine Learning," 8 April 2008. [Online]. Available: https://people.cs.umass.edu/~marlin/research/phd_thesis/thesis_defense_long_print_6up.pdf.

[16] T. Hauck, scikit-learn Cookbook, Packt Publishing, 2014.

[17] N. J. Horton and K. P. Kleinman, "Much ado about nothing: A comparison of missing data methods and software to fit incomplete data regression models.," The American Statistician, pp. 79-90, 2007.

[18] K. Lakshminarayan, S. A. Harp, R. Goldman and T. Samad, "Imputation of missing data using machine learning techniques," in KDD-96 Proceedings, Minneapolis, MN, 1996.

[19] L. Massaron and A. Boschetti, Regression Analysis with Python, Packt Publishing, 2016.

[20] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," Journal of Machine Learning Research , pp. 1157-1182, 2003.

[21] N. Soni and T. Kumar, "Study of Various Crossover Operators in Genetic Algorithms," International Journal of Computer Science and Information Technologies, pp. 7235-7238, 2014.

[22] J. R. Quinlan, "Induction of Decision Trees," Machine Learning, pp. 81-106, 1983.

[23] B. Sjardin, L. Massaron and A. Boschetti, Large Scale Machine Learning with Python, Packt Publishing, 2016.

[24] S. Gollapudi, Practical Machine Learning, Packt Publishing, 2016.

[25] D. M. W. Powers, "What the F-measure doesn't measure: Features, Flaws, Fallacies and Fixes," China & Flinders University, 2015.

[26]D. M. W. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," School of Informatics and Engineering, Flinders University of South Australia , Adelaide, South Australia , 2007.

ACKNOWLEDGEMENTS

The author would like to thank Dr. Ahmad Hadaegh and Dr. Xiaoyu Zhang for their guidance and assistance in completing this research.