Student Affairs Fortune Teller (SAFT): Predicting Student Persistence via Machine Learning

Faraz Moghimi[1], Michael C. Metzger, Fatema B Ahad

Jan 2022

1 Introduction

Student persistence serves as one of many objective indicators that help track and evaluate the university's progress towards its mission. The US news, for instance assigns 24% weight factor to retention and gradation in its overall university ranking [1]. Moreover, we could view the dropped out part of a persistence score in the form a loss, both monetarily speaking as well the human potential. To illustrate this point, assuming a 75% 1- year persistence and a $20000 annual tuition rate, UMass Boston is losing around $20,000,000 annually due to dropouts, in a very broad approximation. Naturally both scholars and practitioners are quite motivated to understand and improve student persistence/dropout rates.

Let's say that we could predict with reasonable validity which student would persist or drop out 6 months to a year from now. This task could help us both in improving persistence as well as the underlying factors associated student dropouts.

First, by knowing who would be more likely to leave our school, we could take on intervention practices such as consultation and targeted support to prevent certain students from dropping out. Second, knowing in advance could help us facilitate targeted data collection that would inform us with regards to underlying factors that cause students to leave in our school. This type of information is generally hard to

[1] Faraz.moghimi@umb.edu
University of Massachusetts Boston – College of Management

acquire once a student leaves the school, hence knowing in advance can help a lot here. Finally, knowing persistence in advance could help various planning procedures conducted by various university departments. For instance, the housing office might benefit from knowing in advance which of their residents are very likely to leave the next semester.

The suggested potentials aligned with the growing developments and capabilities in the machine learning literature, and the increasing data collection efforts within the university have motivated us to take on this project. In this project we seek to develop an in-house machine learning model/pipeline that would utilize student data to predict student persistence; a first step towards developing such a tool inhouse. The main data source used in this project is the bi annual census data and our main hypothesis is that there is useful information in the data that could help us predict student persistence.

As a side benefit, we hope to add some policy guidance to the organizational knowledge with regards utilizing machine learning tools for persistence prediction in the specific case of UMass Boston students. We should have in mind that various insights available from the literature concerning this problem may not be applicable to UMass Boston. The main reason being that our student population maybe inherently different from that of a systematically different school. This in fact is supported by one our results that illustrate prior student performance such as high school gpa , SAT, etc. are irrelevant factors in predicting UMB student persistence. This is contrary to what is proposed in the literature of the field.

2 Research Proposition

To summarize, our research proposition is as follows:

- Build the best viable machine learning model with using the available data that could be evaluated on the grounds of accuracy, true negative rate, and robustness.

- Generate Insight with regards to the underlying factors that could be used distinguish UMass Boston students who are more likely to leave the institution.
- Provide notes concerning policy guidance that could be used to help inform/improve the in house machine learning utilization and the persistence prediction endeavor as a whole.

3 Data Preprocessing

3.1 Data source

The main data used in this project is generated from the bi annual university wide census data. More specifically we work with the census data starting from spring 2018 and onward. The main reasoning behind limiting our data to this point is that the student affairs data collection started in spring 2018. So, it made sense to do our first project for this time period. That said all the procedures in this study can be extended to include prior data points in the future.

3.2 Raw data

The raw data included 160 features. The first cleaning step was to remove redundant features and remove any features that peek into the future beyond the semester time of the census data. As it is crucial to make sure no data from the future is fed to the ML algorithm during the training phase as that would make any prediction made invalid and useless. The raw data used for this project can be in the attached appendix. A list of deleted columns in the first step of preprocessing documents can be found in the appendix section this report(9.1).

3.3 Labeling

Each census semester data includes roughly around 10000-12000 data points. It is important to note, we can only use the data up to the spring 2020 semester and no later. At the time of this project, summer 2021, we cannot determine whether a student will drop out two semesters from any census date post spring 2020 as those semesters have not yet arrived.

Our labeling logic is quite simple using two of the future looking columns we deleted in previous section, last semester number and graduation date. We label any student who's either graduated or has the following logic condition : "last semester number > census semester number + 1" as positive label or 1. We assign our negative label or 0 to the rest of the data points. In other words, identifying which student will persist and which student will have left the institution two semesters in the future.

3.4 Three categories of features

At this stage of the project our features could be generalized to three main categories :

1. Fixed: These features are constant for each student throughout different census times. This category includes data points ranging from demographics to prior performance identification features such as high school and prior college gpa, level of entry upon enrollment (freshman, junior, etc.)
2. Variable: These features change from time to time and mainly include timely academic performance measures at a point in time such as term or cumulative GPA, or credits.
3. Student affairs: These features also vary with time and detect student involvement with a student affair metrics at a point in time. We will discuss this category in more depth in the following subsection

3.5 Handling student affairs data

The student affairs data are recorded with their code name in one of the ten "student_group_code" columns in the census data. They represent the answer to the binary question of "has the student ever participated in the activity x?". Features containing the student affairs related participation of students are formed based on the raw data. The overall statistics of binary occurrences of each student affair data over the semesters are shown in Table 1. As it can been, the majority of student affairs indicators are not captured in census data. We should have in mind the aggregated each semester census data contains roughly 10000-12000 data points. Hence, making the capture of meaningful variational information from student affairs features concerning persistence prediction very difficult. In other words, it is difficult to generate predictive power from a feature has a 0 value for more than 90% of the data.

Table 1. student affairs data, number of records in each semester

Code	Name	s2018	f2018	s2019	f2019	s2020	f2020	s2021
USG	Undergraduate Student Government	45	45	46	30	26	40	34
CADS	Police Cadets	7	7	2	15	11	11	8
CAS	Community Ambassadors	3	3	3	2	4	3	3
FLI	First-Year Leadership Institute	12	12	20	69	66	54	48
USES	Student Employees	906	906	1218	1149	1275	925	817
RLRA	Resident Assistants	0	0	30	22	19	5	6
RLDA	Desk Assistants	0	0	0	0	0	0	0
LLCB	LLC – Beacon Explorers	0	0	236	63	61	43	39
LLCS	LLC – Social Justice	0	0	87	44	42	35	32
LLCG	LLC Green Planet	0	0	200	22	18	14	12
ISP	Student Program Mentors 2018	0	0	0	7	7	5	2
	ISP Mentees	0	0	0	0	0	0	0
HRES	Resident Students	0	0	987	777	698	628	597
SASE	DSA Student Employees	0	0	204	231	251	196	180
MMED	Mass Media Executives	0	0	2	2	2	0	0

SPA	Student Activities Front Desk Staff	0	0	5	2	2	8	7
SAEC	Student Arts & Events Coordinators	0	0	1	1	1	3	3
UASC	U-Access Student Clients	0	0	86	65	126	100	84
UASL	U-Access Student Leaders	0	0	12	7	13	12	10
H4F1	Here4U Fall '19 submissions	0	0	0	0	0	0	0

To deal with this challenge, we decided to keep three of the more significant student affairs variables; Student Employees, Resident Students, and DSA Student Employees. For the rest of the student affairs related data we created an artificial aggregate variable "issta" where we track whether a student been involved with student affairs activities (combining all other student affairs features).

3.6 Standardizing prior student performance

We have a range of prior academic performance in our data, ranging from ACT/SAT to prior college gpa and high school GPA. Moreover, the missing data with regards to these features do not really follow a particular pattern. Making the use of raw data without preprocessing challenging. To address this challenge, we compose a new aggregate variable called "prior score". For this, first, all the scores are scaled in a range of 0 to 4 so that we can take averages of different scores with the same format. Then we compose prior score as the mean of all existing prior academic measures such as high school gpa, pre transfer gpa, ACT, SAT and so on. One student could contain all while another may only have two of these variables as non-missing either way the prior score is calculated for each data point.

3.7 The missing EOT data

Throughout our research, we came across on pattern occurring between the missing end of term (EOT) data and persistence. The number people who had missing EOT term/cumulative gpa or credits in our data was not that high. However, in our analyses, we came across an odd significant correlation between

the students missing EOT values and students dropping out. The following illustrates the correlation missing EOT and student persistence, and the significant p-value:

```
(corr = -0.17495272698092004, pvalue= 0.0)
```

As we can see there is an on odd negative significant correlation between the two. After following various leads that could possibly explain this significant relationship, we could not reach a conclusive explanation for this phenomenon. Some of our top explanations being : students dropping out halfway through a semester and human error. Hence, we decided to remove these data points all together to avoid having any future peeking features as this level of significance with time constrained data seemed improbable.

3.8 Other data cleaning tasks

Various other standard preprocessing and data cleaning task were conducted during this project. To name a few: Missing high school graduation year and age were combined to a single age factor to minimize missing data. Categorical with no logical sequence were one hot encoded and the sequenced categorical features were numerically mapped. Also, all the features were min-max scaled before being fed to the ML algorithms.

A few more preprocessing steps were conducted after heuristics experiments as we gained more insight about the relevance of our various features.

3.9 Handling class imbalance - sampling

Dealing with class imbalance has long been a hot topic of interest concerning machine learning classification problems [2] [3]. This occurs when class labels are not equally distributed and thus represented in the dataset. In a binary classification problem, which would be the case for our particular

problem, this problem becomes extra challenging when we care about predicting the underrepresented class more than the over represented class. That is indeed the case with predicting student persistence labels as predicting the students who leave the institution correctly is considerably more important that predicting the ones who stay. Naturally, we would relatively care more about the student who are likely to leave as it could provide us with intervention and data collection opportunities. We will discuss how this dynamic will influence the statistical measures we choose to evaluate our machine learning models in later sections.

After the initial preprocessing, our class distribution in the aggregated looks considerably imbalanced as presented in Table 2. We can notice that this imbalance is even larger than the average university persistence rate which hovers around 75% on average. We should have in mind that we are indeed double counting the students who persisted. As we would see those students in various census semesters while the students leaving the institution will no longer show in the later census semesters.

Table 2. Class distribution in the initially preprocessed data

Persisted (1)	42556 (86.7%)
Left(0)	6513(13.2%)
Total	49069

A common way to address class imbalance in the machine learning literature is sampling; the most popular approaches being over and under sampling. In other words, one would arbitrarily increase sampling from the under represented class or reduce the samples from over represented class to allow for the machine learning algorithm to better train for predicting the unrepresented class. Figure 1 illustrates how this practice will work. The main idea being that we do not want our model to be biased solely towards the overrepresented class.

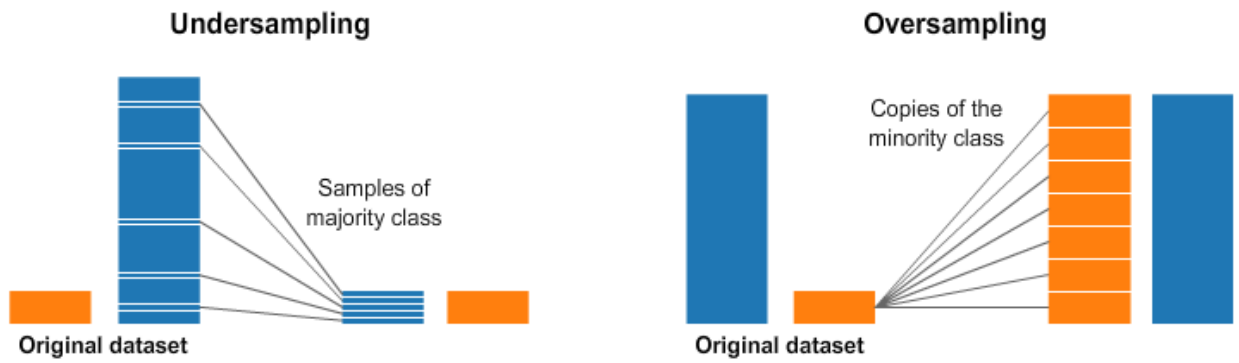

Figure 1.Over and under sampling

In this project, we experiment with three sampling methods:

1. Super Oversampling: In this approach, we arbitrarily increase the number our underrepresented class by creating copies of the existing data. This approach will results in an aggregated sample of 60,000 divided 50-50 between the two classes for our training sample. Note that we will not do this for the testing sample as it will bias the results. Even though we experiment with this approach, the results are not expected to be the best. As by double counting both classes here, one could argue that the data noise is increasing in this sample without adding any new information.
2. Super Under sampling: In this approach we randomly select an equal amount of data from each class. This will leave us with around 9000 data points distributed equally between the classes. Again the flaw of this approach is that perhaps we are loosing too much information just for the sake of equal class distribution. Plus, a 50-50 class distribution might not be fully desirable as that is rarely the case in reality.

3. **Two birds, one stone:** As mentioned before, we are indeed double counting persisted students that show up in various census semesters. Hence, the left(0) class is more underrepresented than it should be. So, in this approach we decide to sample via unique student ids, in this way we are no longer double counting and we increase the representation of left(0) class. Hence, reducing the bias towards the overrepresented class with losing too much information. Our final aggregate sample is as follows.

Table 3. Final aggregated sample class distribution

persisted(1)	12891	0.701551
left(0)	5484	0.298449
total	18375	1

3.10 Train-test split

In the last step of preprocessing, we randomly select 60% of our sample for training 15% for hyperparameter tunning/ validation of the machine learning model and leave 25% percent unseen sample for validating and testing the model performance. It should be noted that this train-test split is conducted while stratifying for our class label; meaning the class distribution between our train and test samples are the same.

4 Data visualization insights and heuristics

In this section we use the training sample to conduct various data visualization task in order to gain useful insights about our data. Numerous visualizations were conducted, however, here we only report some of

the more significant ones. It should be noted that, we used the insights from this section to remove some additional features from our data as their distribution did not have a meaningful variation between our two classes. The final features used for the machine learning model building can be found in 9.2.

4.1 What matters? Your past or present?

Figure 2 illustrates the distribution for two of our variables among the two classes; prior score and cumulative gpa. We can observe that the students who left have a significantly different cumulative gpa distribution compared to the students who persisted. However, surprisingly, the prior score distribution does not vary between the two classes. This is contrary to what is commonly believed in the literature where it is often suggested that you could predict student persistence with a decent validity by solely analyzing the pre enrollment academic performance such as high school gpa, etc. We see that this is not the case for UMass students. This observation lend some support to two hypothesis that we touch on during the remainder of this paper:

1. UMass is a place where students can make it with their own performance regardless of were they come from.
2. The students who leave UMass are a bit different than general dropouts in that the number of people who transfer out to another perhaps more prestigious school is significant in relation to students who leave the school and higher ed all together. This is a recuring theme in the rest of our project, making the binary classification a difficult task as it is more difficult to cluster all the ones who leave into a single type.

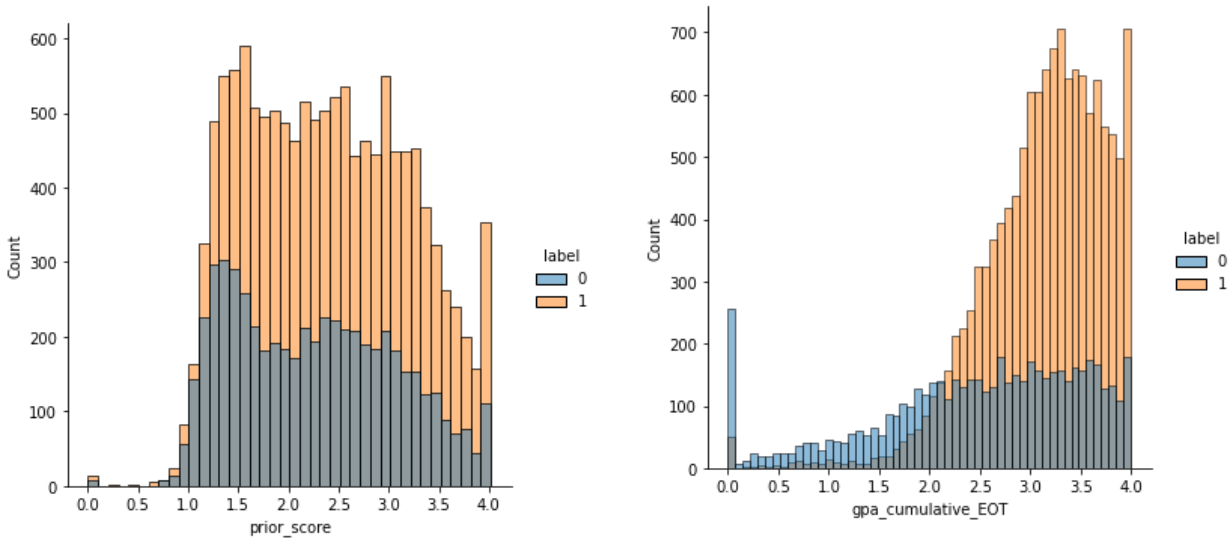

Figure 2. Class distribution between two classes: The figure on the left illustrates the prior score feature histogram among both classes. The figure on the right shows the histogram distribution of cumulative gap between the two classes. As it can be seen, cumulative gpa distribution varies significantly between the two classes while prior score distribution being relatively similar.

4.2 Gender

Figure 3 illustrates the gender distribution among the students who persist and leave. As commonly believed, we see that the female students are more likely to persist relative to our male students.

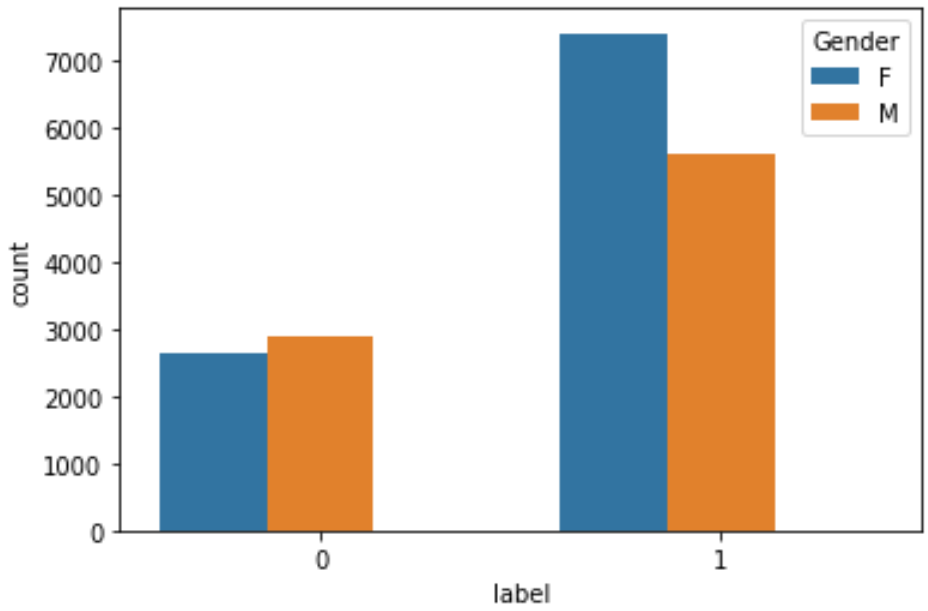

Figure 3. Gender distribution between the two classes

4.3 Census level

Figure 4 illustrates the class distribution among various student levels at the time of the census. As expected, it can be seen that the closer a student is to graduation, a senior for instance, the more likely he/she is to persist.

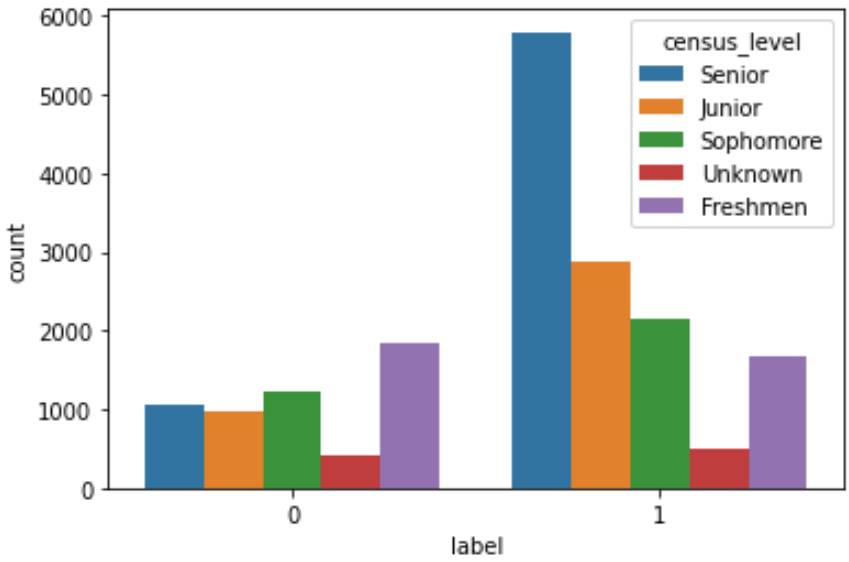

Figure 4.census level distribution among both classes

4.4 Residency

Figure 5 illustrates the persistence and not persistence distribution among students with different residency status. As expected, it is observed that in state student tend to persist more. That said, perhaps contrary to one would initially believe, we can see that the dropout ratio among international students is considerably large, a pattern that is somewhat different from the consensus literature when it comes to UMass students. Perhaps indicating that some of those international students are transferring out rather than dropping higher ed.

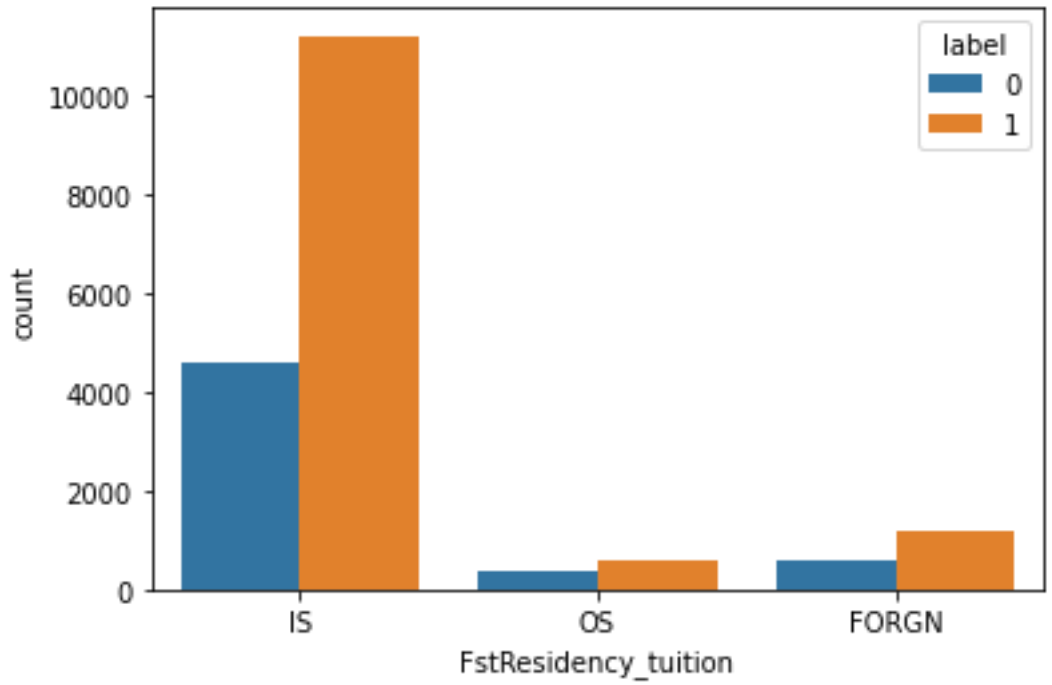

Figure 5. Residency distribution among the two classes

4.5 Overall correlation

Figure 6 provides the correlation matrix between our features and the classes. As we can see in this chart, the overall correlation between our features and the label column is relatively low. This in fact can be a sign that we are dealing with a difficult machine learning task since the majority of our features do not show a meaningful enough correlation with our classes. In addition, this table could inform us when dealing with certain algorithms that theoretically do not respond well to features being correlated with each other. Also, this led us to remove some of the more noisy and non-significant features from our dataset altogether. For instance, our experiments combined with a primary version of Figure 6 informed us that our predictive model works better color blind and student ethnicity showed no meaningful predictive power in general.

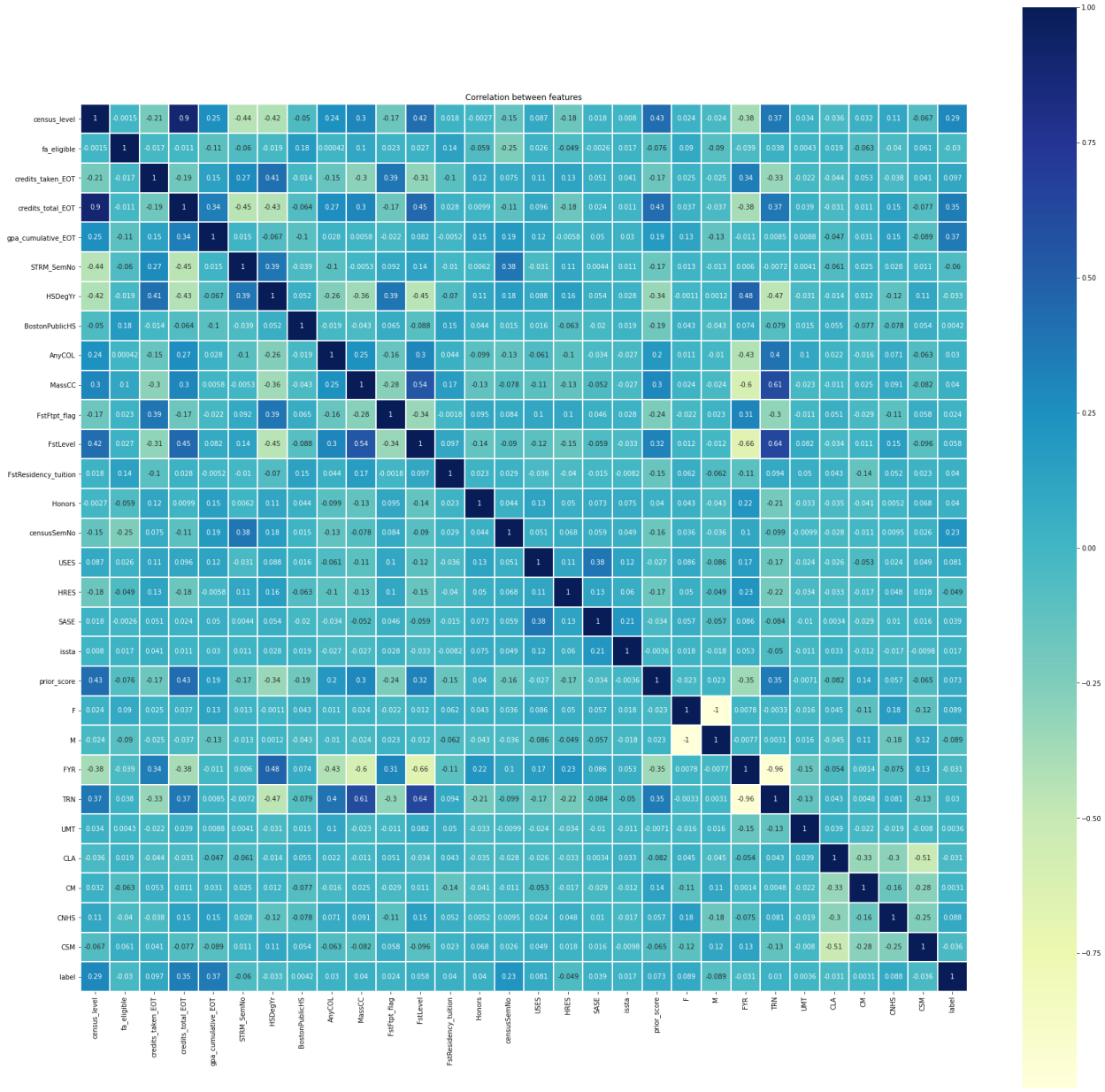

Figure 6.correlation matrix among both features and the labels.

5 Methodology

5.1 Machine Learning algorithms

In this project we experiment with various machine learning algorithms and structures to find the best viable approach for our particular problem. Moreover, this process can add empirical evidence to any future endeavors in this realm with regards to why certain models work better with low variation, imbalanced, and small sample data.

Our machine learning task has a few major challenges that we must take into account when utilizing a predictive algorithm. Meaning, we have to examined these challenges with the theoretical frame of any model we experiment with to avoid certain pitfalls. First, our sample size is quite small relative to what major machine learning algorithms deal with. This fact forces us to heavily favor model regularization as our training data is limited and chances of overfitting specially with more complex models becomes very significant. Overfitting is what occurs when a model becomes biased towards the training sample. In this situation, the model would fit the training sample quite well, while not fitting the test sample or the general problem that well. In other words, the training model is fitting a pattern that does not exist in reality. This is in fact what happens in practice when we experimented with relatively "deeper" neural network designs. Another factor to consider is the other end of this trade off. As we saw in the previous section, our overall feature variation with regards to our classes is not that obvious, meaning we need a complex enough model that would capture any hidden useful information. So, simpler models are not expected to work that well here, something that we see in our results. Last important thing to have mind for our experiments is of course the concept of class imbalance in each algorithm. We have already conducted sampling procedures to mediate this challenge. However, each algorithm generally has a way

of adjusting weights based on which class we care more about. So, that is something we need be conscious of as it might help our model development.

A variety machine learning algorithms and concepts were utilized in this project. We avoid explaining each algorithm as it would make this paper extremely longer than needed. However, references are provided for interested readers that hope familiarize themselves more with certain concepts and algorithms. The list of machine learning algorithms is as follows:

1. Support Vector Machines(SVM) : Various forms of supported vector machines were used in our experiments including all the common linear and non linear kernels(liniear, rbf, poly) [4]
2. Bagging classifier [5]
3. Decision Trees [6]
4. Various forms of semi deep neural networks: it should be notes that after conducting numerous experiments, our final neural network design consisted of 4 dense layers with sigmoid and relu activation functions respectively [7]
5. XGBoost (eXtreme Gradient Boosting with Decision Trees) [8]: This fairly new algorithm has shown a lot of promise in recent in solving the type of problems that we in fact deal with in this project. There is a fair amount evidence in the literature that XGboost is indeed well equipped to deal problems with correlated features, complex relationships, essential need for generalization.

5.2 Validation

There are various statistical measures that are often used to evaluate and validate machine learning classification tasks ranging from accuracy to precision, recall and so on. In our case of course, we are dealing with two important factors; class imbalance, and class importance. As mention before a simple accuracy measure for this type of problem can often be misleading as a naïve predicter would reach very

high accuracy without any model training. In other words, if we say every student will persist, we will have a more than 70% accuracy. This is very important as accuracy can never be sufficient measure on this level of class imbalance. Still, even some well published documents in the literature seem to make the mistake solely reporting accuracy or other measures concerning the positive class. Moreover, it is quite reasonable that we care about predicting our negative class (the students who left) considerably more than the ones who stay. As any intervention or data collection effort would seek to target the students who are more likely to drop out. With that in mind we use two main statistical measures for our final evaluation and validation; accuracy and True Negative Rate (specificity)(Equation 1). This way by setting our students who left class as our negative class, we would be able to measure how well we can predict that negative class as well as gaining general performance measures from accuracy.

$$Accuracy = (TP+TN)/(TP+FP+FN+TN)$$

$$TNR = TN/(TN+FP)$$

Equation 1 Accuracy and TNR formula. Where TP is True Positive, TN is True Negative, FP is False Positive, FN is False Negative.

Moreover, we care about model generalizability or in simpler terms we do not want a model that overfits. Therefore, we introduce another performance measure we refer to as the TNR train/test spread. Which simply reports the absolute difference between the TNR on the train sample and the test sample. Intuitively, the lower the TNR train/test spread, the better and more generalizable the model is.

Equation 2. TNR train/test spread defition.

$$TNR\ train/test\ spread\ = |TNR_{train}\ - TNR_{test}\ |$$

Finally, there is a tradeoff between the performance of our model with regards to the persisting class and the non-persisting class. As mentioned before, we care more about identifying the underrepresented class of non-persisting students, which we also refer to as the negative class in this study. That said, a model can theoretically achieve the maximum TNR of 100% by simply predicting every class as negative. Needless to say that model would be biased and completely useless. Therefore, ideally, we would want a well-rounded model that would have the highest TNR whiles demonstrating an overall high accuracy. Naturally, there is a tradeoff there and we quantify this trade off as another performance metric by taking absolute difference of test accuracy and test TNR. Intuitively, the lower this value is the more well rounded our model should, which is a desirable feature.

Equation 3. TNR/Accuracy spread definition.

$$TNR/Accuracy\ spread\ = |TNR_{test}\ - Accuracy_{test}\ |$$

5.3 Hyperparameter Optimization

Hyperparameter optimization is an important task in designing machine learning pipelines. The objective being to find the model modification that would work best with our problem while maintaining model robustness. For instance, setting the right neural network learning rate or XGBoost's regularization lambda or tree depth. We shy away from digging too deep in explaining the role of each hyperparameter in each algorithm here as it would be a lengthy discussion if done right. Provided sources can be referenced by the interested readers to acquire more information. In this process we use a guided grid search approach to optimize our hyperparameters. The main evaluation criteria being set with measuring the Area under the ROC Curve (AUC), often suggested by the literature as the right measure for model selection in imbalance classification problems.

Overall, along with super over sample and super under sample and main datasets, more than 3000 experiments we conducted. This helped us identify the best models to implement while increasing our confidence in the robustness of the models as a lot of the top models showed very similar performances.

6 Results and Discussion

In this segment, we report a series of results to evaluate and interpret our machine learning models. First, we report and discuss the results generated from different machine learning algorithms. Next, we expand our analyses by focusing on understanding the road through which we achieved the best model. Finally, we discuss the relative feature importance in our model to improve the interpretability[2].

6.1 Summary of performances

We report the performance of each optimized model as demonstrated in Table 4. Each row in this table represents a performance metric; reporting the accuracy and True Negative Rate (TNR) results on both the training and testing samples, as well as the spread metrics. As discussed before, we are mainly concerned with the True Negative Rate as any interventionist policy would be directed towards the students who are likely to not persist. We can see that overfitting the negative class is a recurring theme on multiple algorithms as captured by the spread between TNR rate of test and train samples. For instance, we can observe that SVM fails to generalize the underlying pattern in identifying the negative class. Therefore, whilst the TNR on the training sample is 75% , it drops to 47% when evaluated on the test sample. Moreover, we can see that the more complex algorithms such as NN and XGBoost demonstrate an overall better and more generalizable performance. This leads us to hypothesize that the underlying pattern that can predict student non-persistence is complex. In other words, based on this

[2] It should be noted we further check the robustness of our results with several non-reproducible random seed experiments to mediate any potential biases raising from the stochastic nature the algorithms. We similar results, in the extended experiments.

evidence, we believe that although simpler model structures such as linear and logistics regression may reduce the over fitting problem, they would not be able to detect the underlying pattern through which we can predict student non-persistence. Our additional robustness checks confirm this notion. Overall, we can see that XGBoost model performs significantly better all across the board, having a 64% TNR on our test sample while maintaining a 82% overall accuracy. This result is also reinforced with the lowest TNR Train/Test Spread and Test Accuracy/TNR Spread values in our sample; 14% and 18% respectively. Therefore, in addition to having the best well rounded performance, we can infer that the XGBoost model is the most generalizable and robust model in this experiment. Overall, the TNR achieved by our XGBoost model here is a significant improvement to the naïve 30% benchmark as well as other machine learning modeling approaches. Similarly, our other performance indicators support the notion that the XGBoost model is also most generalizable and robust approach in our experiments.

Table 4. Summary of optimized results

	SVM	**BG**	**DT**	**NN**	**XGBoost**
Accuracy_Test	76%	84%	87%	82%	82%
TNR_Test	47%	53%	50%	56%	64%
Accuracy_Train	92%	90%	88%	86%	88%
TNR_Train	75%	90%	74%	70%	78%
TNR Train/Test Spread	28%	37%	24%	14%	14%
Test Accuracy/TNR Spread	29%	31%	37%	26%	18%

6.2 Road to performance improvement (XGBoost)

Now that we have identified the best model, we seek to demonstrate the underlying mechanism of achieving the best model. In this section, we report the confusion matrices for our initial and optimized XGBoost model as shown in Figure 7, and Figure 8, respectively. The initial model correctly predicts that 3712 (out of 3868) students would persist and 887 (out of 1645) students will not persist. The initial model is miss classifying 46% of the students who end up leaving the institution, having a TNR of 54%. On the other hand, the model has a significantly good performance with regards to our positive class (the students who persist) having True Positive Rate of 96%. Therefore, we can see that the initial model over predicts the positive class (persisted students) and consequently underpredicts the negative class. In other words, the model appears to be biased towards the positive class due to their over representation in this problem. As mentioned before, the TNR is our main performance indicator of interests alongside having a well-rounded model as captured by accuracy, TNR Train/Test Spread and Test Accuracy/TNR Spread. Hence, in our optimized model, we seek to improve the TNR while maintaining well rounded model that is not overly biased towards a single class. The optimized XGBoost model correctly identifies 1053 (out of 1645) of the students who will leave the institution resulting in a 64% TNR which is a significant improvement from the initial model. In addition, the model also maintains 90% True Positive Rate which is a still quite good. Therefore, resulting in the best well rounded performance while maximizing our main indicator of interest.

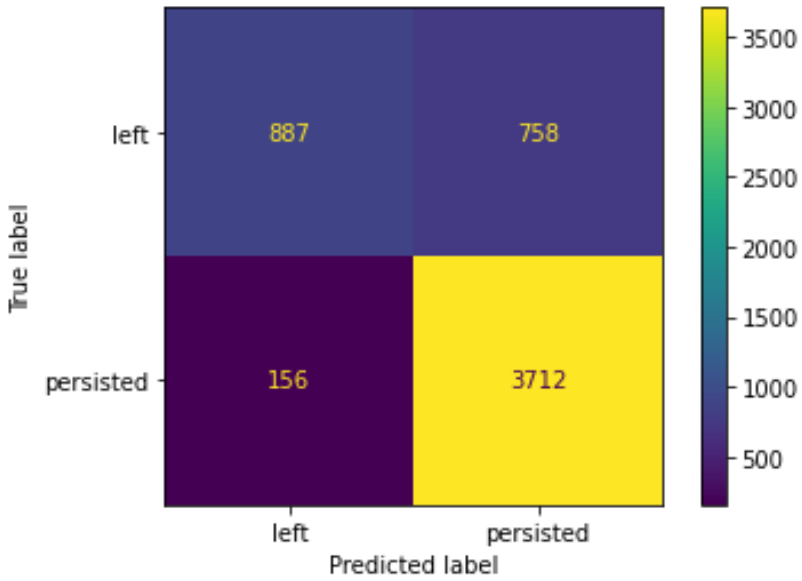

Figure 7. Test Sample Confusion matrix for the initial XGBoost model

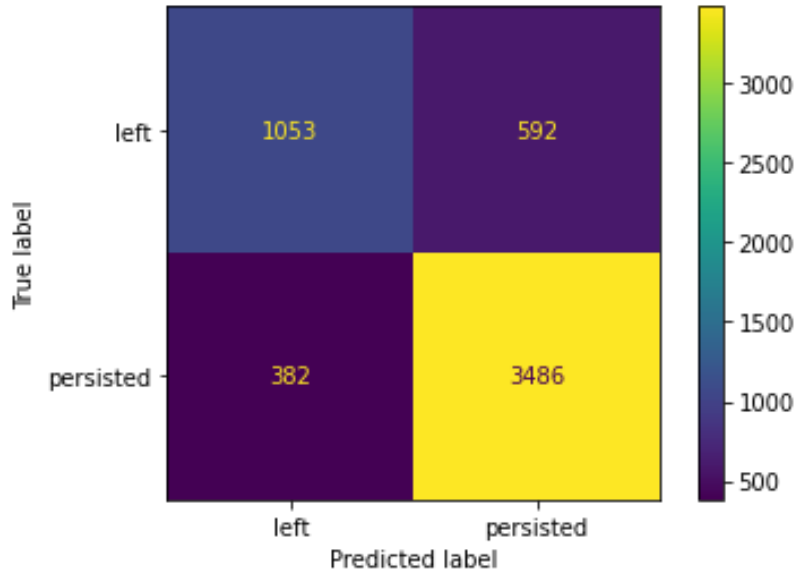

Figure 8.Test Sample Confusion matrix for the optimized XGBoost

6.3 Feature importance

In this section, we seek to add another level of depth to our analyses by examining feature importance in informing the predictions of our optimized model. This step naturally informs the interpretability of our model. Moreover, this step could inform data collection policies and data preprocessing steps for future work. The relative step is calculated following "A working guide to Boosted Regression Trees" [9]. The "relative importance" measure represents the number of times a variable is selected for splitting branches in the XGBoost algorithm, weighted by the squared improvement to the model as a result of each split, and finally averaged over all trees. The relative importance is scaled from 0 to 1, where the larger values infer the higher relative importance of a feature. Figure 9 reports the relative feature importance for our optimized XGBoost model. Interestingly, we find that the continuously-changing category of our features such as "end of term GPA" and "end of term credits taken and total" have the strongest effect in informing our model's predictions. Interestingly, contrary to the guidance from the prior literature, we find that the pre-enrollment static data points such as SAT scores, and High School GPA captured by our prior_score feature, do relatively very little to inform our model's prediction. This evidence could mean that the students at University of Massachusetts Boston are different from other samples used in the prior literature. In other words, it is likely that they make the leave or stay decision mainly based on what happens after their enrollment rather than features from their past. We suspect that this could be due to the relatively large population of transferring students among the population who end up leaving UMass Boston. Therefore, further classifying our non-persisting class to transfer student and dropouts could improve our understanding of the problem and consequently improve our predictive performance. New evidence presented in the recent literature highlights the importance of student engagement in predicting student persistence; arguing that more engaged students are more likely to stay. The student affair segment of our features which captures a dimension of student engagement does contribute to our statistical learning endeavor. However, their relative importance is quite low. We suspect that this is due

to the sparsity of our student affairs data as the majority of the data collection effort started post 2018-2019. Moreover, we only have access to semester-by-semester data; therefore, we hypothesize that adding more frequent and less sparse student engagement data will improve the model performance. Finally, we find that consistent with consensus knowledge that the in state / out of state tuition plays an important role in predicting student persistence. To put this in perspective, at UMass Boston, the out of state tuition is more than double in-state tuition. Naturally, students from out state will have more reasons to leave the institution if they are unsatisfied.

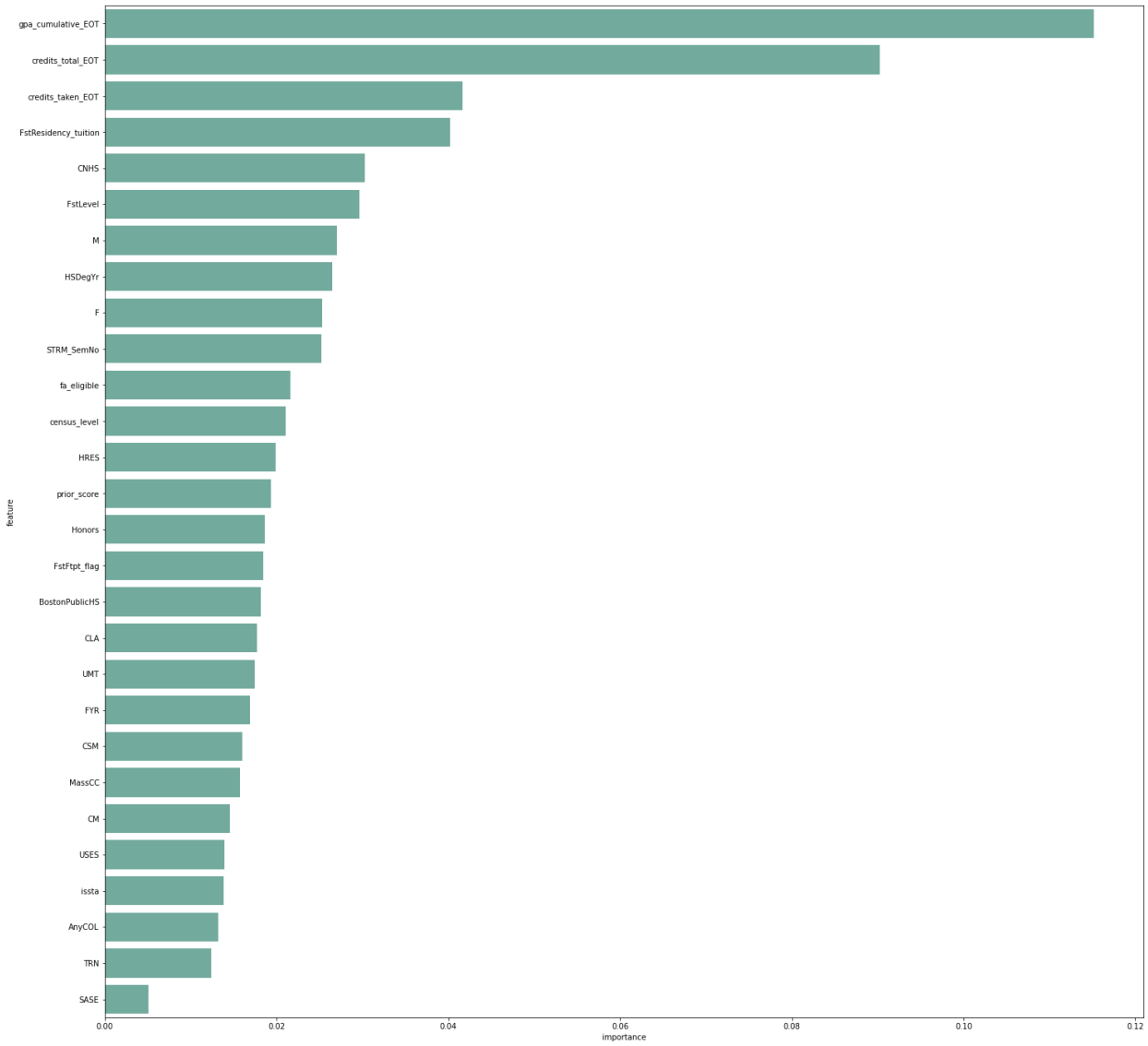

Figure 9. Relative feature importance for model prediction

7 Conclusion

In this study, we leverage student data from the University of Massachusetts Boston to develop a machine learning model for predicting student persistence. Our optimized model which results in a 64% TRN and a 82% overall accuracy demonstrates the merits of this approach to support the decision making process at an academic institution. An effort that could mediate intervention policies, more efficient future

planning, and better institutional knowledge acquisition (i.e. understanding our students better). During this process, we propose several frameworks ranging from a sampling procedure as well as training, model selection, and evaluation mechanisms that could be utilized for similar projects and studies. Moreover, we find several interesting empirical evidence that could add to our understanding of student persistence and retention. For instance, we find evidence supporting the notion that the main indicators for predicting student persistence at UMass Boston is related to what they have done since they have joined the institution, rather than their prior history before joining. Meanwhile, we observe that prior academic backgrounds such as high school GPA and SAT scores do not contribute as much to the predictive power our model. A pattern that is somewhat contradictory with the findings of the prior literature. On the other hand, some of our features' predictive power are quite consistent to the findings of the prior literature such as the in/out of state tuition status. This leads us to believe that while there are commonalities in the behavior of students in different institution, certain predictive features and behaviors concerning persistence prediction maybe institution specific. In other words, while students who pay higher tuition rates are classified as more likely to leave, SAT scores might be a relevant predictive indicator only some institutions and not for others. This might be due to the unique characteristics of an institution's student body. For instance, a large portion of UMass Boston's non-persisting students actually transfer to another college rather than dropping out from higher education all together. Our study suggests that incorporating this into our now binary classification approach might improve our performance; therefore, justifying the effort of making this distinction in our data collection endeavors. However, high rate of transferring out may not be a relevant part of the prediction problem at another institution with a very low population of students transferring out of the university. Therefore, we would argue that there still a lot to explore in understanding the institution specific features and student behaviors that are relevant in predicting student persistence in the future. Overall, we would summarize our contributions to the literature as follows: 1. We provide a comprehensive framework for utilizing machine learning algorithms for student

persistence prediction. 2. Our approach, provides a generalized sampling procedure, model selection, and evaluation framework that improves the work in the prior literature. Specifically, conduct hyperparameter sampling separation which mediates various biases of overfitting, provide sampling mechanism for addressing class imbalance in these types of problems, and utilize evaluation procedures such as the TNR and the robustness spreads that are arguably more relevant for the decision makers in this problem. 3. We provide additional insights regarding to identify the features that are important to the student persistence prediction task. 4. Relying on our feature experiments we are led to hypothesize that while some features are universal, others could be institution specific. 5. Therefore, we believe that our work provides a new unique perspective on dealing with the student persistence prediction problem in institutions similar to University of Massachusetts Boston . For future work, there are several interesting avenues worth pursuing. Naturally, additional data could help improve the model and add further confidence to our evaluation procedures. Moreover, we could explore labeling students differently based on what they do after they leave the institution. For instance, we could label the data based on employment, transfer status to other institution, and persisting at the institution. Another avenue that seems interesting is to include higher frequency data points. The COVID 19 pandemic accelerated the digitalization of education. This of course creates more frequent alternative data sources that can be leveraged for student persistence prediction. For instance, the login counts, or times spent on various online channels of education, event logs, online text communications, etc. could improve our understanding of the students and consequently help us predict future leaving students better. There are many possibilities to drive various forms of student engagement proxies from the mentioned source that could prove to be promising.

8 References

[1] U. News, "US News," [Online]. Available: https://www.usnews.com/education/best-colleges/articles/how-us-news-calculated-the-rankings.

[2] N. a. S. S. apkowicz, "The class imbalance problem: A systematic study," *Intelligent data analysis 6.5,* 2002.

[3] A. e. a. Guzmán-Ponce, "DBIG-US: A two-stage under-sampling algorithm to face the class imbalance problem," *Expert Systems with Applications 168 (2021): 114301,* 2021.

[4] I. a. A. C. Steinwart, "Support vector machines.," Springer Science & Business Media, 2008.

[5] L. Breiman, "Bagging predictors," *Machine learning.*

[6] L. a. O. M. Rokach, Decision trees, Data mining and knowledge discovery handbook.

[7] W. e. a. Liu, "A survey of deep neural network architectures and their applications," *Neurocomputing.*

[8] T. e. a. Chen, "Xgboost: extreme gradient boosting," *R package version.*

[9] J. J. R. L. a. T. H. Elith, "A working guide to boosted regression trees," *Journal of animal ecology,* 2008.

9 APPENDIX

9.1 The initially deleted columns

'FstDegree_Type','FirstGen','NatAM_tribe' , 'CHINESE', 'AnyCC', 'previous_deg1', 'FstDegree', 'race_federal', 'BachDegree'

,'BachCollege', 'BachCollege', 'BachMajor1', 'strm_demo' , 'athlete_flag', 'HawPac', 'FirstEnrollSemNo',

'Sem', 'age', 'FstSemMajor', 'LastGPA','LastCollege', 'LastMajor2', 'LastMajor1', 'OneYrRetention', 'TwoYrRetention',

'BachCohortYear', 'census_term_short', 'Cohort', 'STRM', 'ThreeYrGradRate', 'FourYrGradRate', 'FiveYrGradRate', 'SixYrGradRate',

'census_term_demo', 'LastSTRM', 'BachSTRM', 'LastYear', 'LastTotCredits'

9.2 Final features

```
census_level', 'fa_eligible', 'credits_taken_EOT', 'credits_total_EOT',
       'gpa_cumulative_EOT', 'STRM_SemNo', 'HSDegYr', 'BostonPublicHS',
       'AnyCOL', 'MassCC', 'FstFtpt_flag', 'FstLevel', 'FstResidency_tuiti
on',
       'Honors', 'censusSemNo', 'USES', 'HRES', 'SASE', 'issta', 'prior_sc
ore',
       'F', 'M', 'FYR', 'TNR', 'UMT', 'CLA', 'CM', 'CNHS', 'CSM', 'label'
```