Predicting Student Dropout

Amy Bui, Chung-Ying (Joey) Hsu, Joel Johnson, Benjamin Stone, Amy Yu
Github Repo: https://github.com/amy-ym-yu/dropout
ECS 171 - Group 32

I. Introduction and Background

Universities often have students who drop out and are not able to finish their degrees. The inability to complete higher education can impact an individual's employability and economic growth, in turn impacting the lives of the student and their family. Oftentimes, there are indicators that suggest that a student is at risk of dropping out. Machine learning models can help identify those students and notify the university to help them receive the resources they need.

The inability to continue one's education can be attributed to a myriad of factors. The experiment done in this paper will explore those factors to help create a model that predicts university dropout rates. Given that the model is successful at the university level, the model could be applied to lower levels of education as listed below.

- A high school diploma is often the minimum requirement for many jobs, thus dropping out of high school can be extremely detrimental to a student's future. Additionally, high school academics are often a major factor that determines whether a student pursues a university degree.
- Elementary school is where students learn vital academic skills such as reading comprehension, writing, and fundamental math. These lifelong skills not only impact the student's future academic success but also may be a detriment in their day-to-day life.

II. Literature Review

Identifying students who are at risk of failing out or dropping out is not a new research topic. Institutions and researchers have used various datasets, a multitude of models, and a diverse combination of factors to predict and identify these students. This section will highlight related works that guided the development of the models built in this paper.

Martins et al. used a dataset from the Polytechnic Institute of Portalegre, which focuses on student data from early academic careers, to build several machine learning models that aim to identify students proactively. These included Support Vector Machines (SVM), Random Forests (RF), and Boosted models, including Extreme Gradient Boosting (XGB), CatBoost, and LogitBoost. After 5-fold cross-validation and hyperparameter optimization through grid search and randomized grid search, the highest F1 and accuracy in their paper was found to be Extreme Gradient Boosting at an F1 score of 0.65 and accuracy of 0.73. Their paper also references the paper authored by Thammasiri et al. which aims to improve the poor performance of minority groups due to unbalanced data using 10-fold cross-validation, random over-sampling, random under-sampling, and SMOTE techniques to create more balanced data. They concluded that SMOTE on an SVM had the highest accuracy (0.905) and specificity (0.958) while oversampling on a decision tree had the highest sensitivity (0.885).

Gautam and Bansal explore how machine learning can detect cyberstalking using a Soft Voting Technique. They use SVM, Naive Bayes, and RF on three separate datasets of SPAM email, cyberstalking, and SPAM email subject lines dataset to obtain accurate predictions. The Soft Voting Ensemble method in which the model takes an aggregation of the probabilities and determines a final prediction from the aggregate max. The researchers concluded that the Soft Voting model had the highest accuracy (0.979) and precision (0.97).

III. Dataset Description & Exploratory Data Analysis

The dataset used in this experiment is the "Predict students' dropout and academic success" dataset created by Martins et al. at the Polytechnique Institute of Portalegre and funded by the SATDAP program. This set comes in the form of a CSV file and includes information from 4,424 students at the time of their enrollment in a higher education institution and includes 36 features. The target variable is the status of the student at the end of a normal degree period, and it is separated into three labels:

graduated, enrolled, and dropped out. For example, the expected degree period for an architecture student in the U.S. may be 5 years, thus this dataset would indicate the status of a given architecture student 5 years after their initial enrollment. Since there are several features with low correlations to the target variable, this section will only explain the attributes relevant to predicting student success and how the features are determined.

Since our project aimed to predict dropout rates amongst all higher education students, it was important that our dataset capture the nature of all specialties. The dataset was chosen because of its generalizability as it does not focus on a specific field of study. It also captures environmental factors possibly influencing students such as parents' qualifications and GDP. A limitation of the dataset is the class imbalance because of the large instance of graduated students. As a result, we have implemented the necessary steps to improve performance. Since there are several low-correlation variables, we only describe specific ones (see Appendix A).

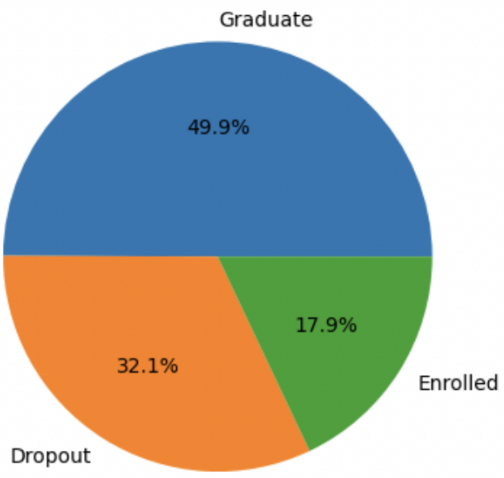

Fig. 1. Distribution of Classes

IV. Proposed Methodology

Due to the quantification of the target variables, this is a multiclassification problem. From the literature review, we decided to create models with a myriad of models based on their results including KNN, SVM, Random Forest, XGBoost, and Ensemble method. However, our best three models ended up being Random Forest, XGBoost, and Soft Voting Ensemble Method. Random Forest and XGBoost had been employed by Martins et al. in previous works, but the Soft Voting Ensemble method employed by Gautam and Bansal had not been applied to their dataset. In the end, our top 3 models are Random Forest, XGBoost, and Soft Voting. To prevent overfitting, we applied 10-fold cross-validation on all the sets to obtain the best results. Thammasiri et al. aimed to mitigate the poor performance of minority groups, thus we implemented some of their techniques. To increase accuracy for minority groups (e.g. dropout and enrolled), we tested random undersampling, SMOTE, and random oversampling (as a baseline to check for overfitting). To find the optimal hyperparameters, we created our own GridSearch. Although not as efficient, it provided hyperparameters with better performance. In total, we developed 9 trained models consisting of combinations of 3 types of models with 3 types of data balancing techniques.

Feature selection for all the models was done using the correlation coefficient given by the heatmap (see Appendix B) and dropping attributes with a significantly low correlation to the target. For any categorical values like major, we one-hot-encoded them so that SMOTE sampling would be more accurate. After selection, the dataset was standardized and partitioned into training and testing sets at an 80% to 20% ratio, respectively. For each of the data balancing techniques, a dataset was created for each of the models to be trained on.

V. Experimental Results

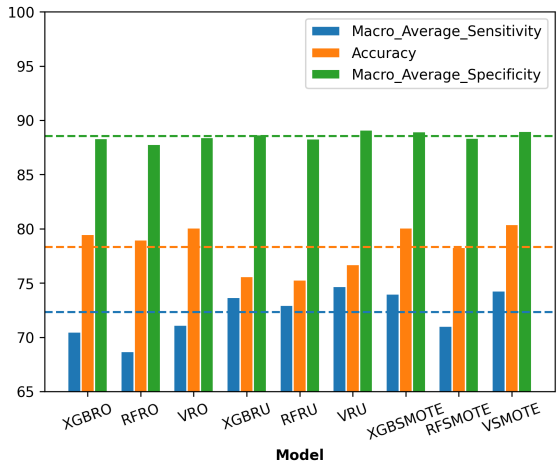

Fig. 2. All of the models with their Accuracy, Macro Average Sensitivity, Macro Average Specificity. The dotted lines illustrate the average for all models a given metric.
* Note: Short hand was used for random undersampling (RU) and random oversampling (RO).

Our Random Forest model was built using sklearn's RandomForestClassifier implementation which combines

multiple individual models trained on different subsets of the training data to make an accurate prediction. Initially, we attempted to hyperparameter-tune the model to various random states, trees, depths, and learning rates. We ultimately decided against random states since it can depend on the operating system's implementation or hardware differences. The best parameters we found are as follows:

- Undersampling: 150 trees, Max depth 8, Learning rate 0.4
- Random Oversampling: 150 trees, Max depth 8, Learning rate 0.4
- SMOTE: 100 trees, Max depth 5, Learning rate 0.4

It was interesting to see that RF had approximately the same graduate sensitivity as the other models for each data sampling technique, and even for undersampling, it was higher.

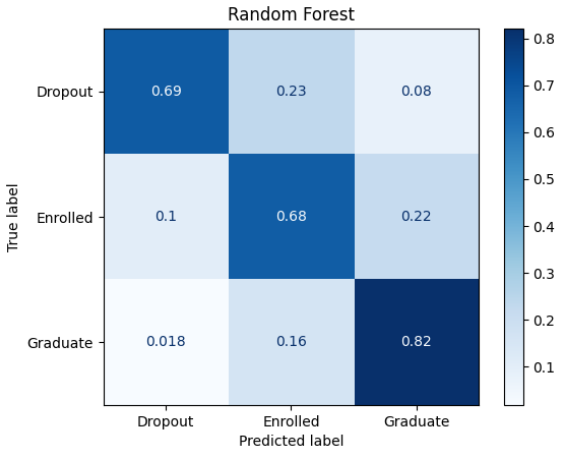

Fig. 3. Random Forest Random Undersampling Results

XGBoost is another ensemble learning method that gave us some of the best results. This model includes building on decision trees to correct errors and provide a more accurate model. We were able to help prevent overfitting with the L1 and L2 regularization offered by the algorithm. Some hyperparameter tuning was done on the number of trees used and the tree method. Both undersampling and oversampling used 150 trees while SMOTE samplings used only 100. We found that "hist" was the best tree method because of its overall reduced memory storage and training speed. The best results were obtained when doing SMOTE sampling giving us an accuracy of 80.14%.

The Soft Voting Ensemble model is built from sklearn's voting classifier which takes in both of the models mentioned above. It uses the prediction from

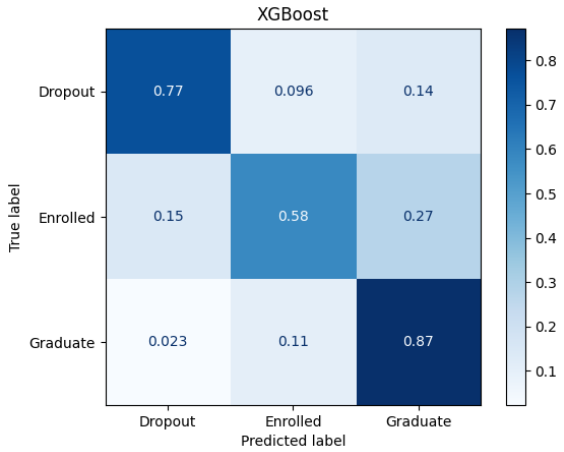

Fig. 4. XG Boost SMOTE Results

XGB and RF, sums them up, and outputs a prediction based on the class with the highest label. We tested hyperparameter tuning by changing XGB parameters, but there was no difference as the model depended on the previously developed models. Overall, the Soft Voting models outperformed the RF and XGB models in terms of average model sensitivity as expected due to their dependencies on the other models.

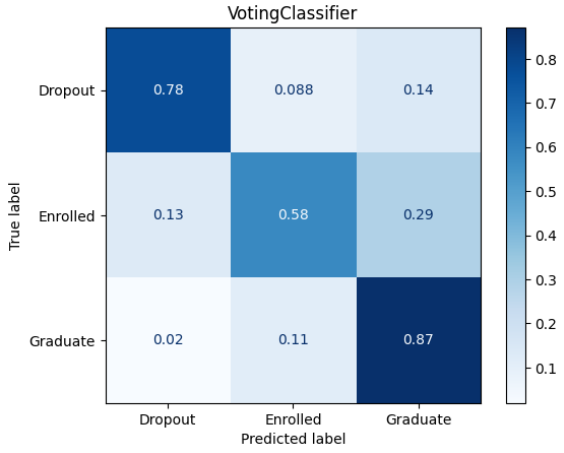

Fig. 5. Soft Voting SMOTE Results

All the models trained on randomly undersampled datasets had high specificity predicting dropout and enrolled students compared to oversampled and SMOTE sampled datasets. Due to the nature of random undersampling, the models trained on this dataset will capture less of the overall trends in the data and be overly pessimistic about predicting graduation rates. This can be a strength since enrolled students can quickly become dropout students if not given sufficient support.

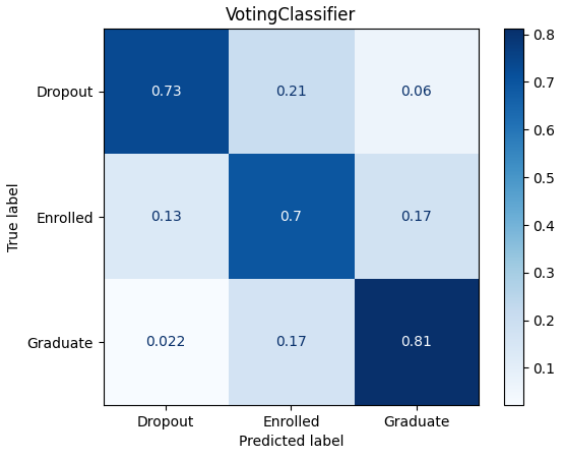

Fig. 6. Soft Voting Random Undersampling Results

Oversampled models capture overall trends better than undersampled models but the replication reduces diversity in the set, making dropout students look more like a specific type of student. Oversampled models have a higher specific precision and overall precision for each label, but these models are more likely to be overfitted. However, a strength is that the computational complexity decreases since synthetic samples do not need to be created. If a use case has a large dataset and insufficient computational power is a concern, an oversampled model may be a better fit with a tradeoff of accuracy.

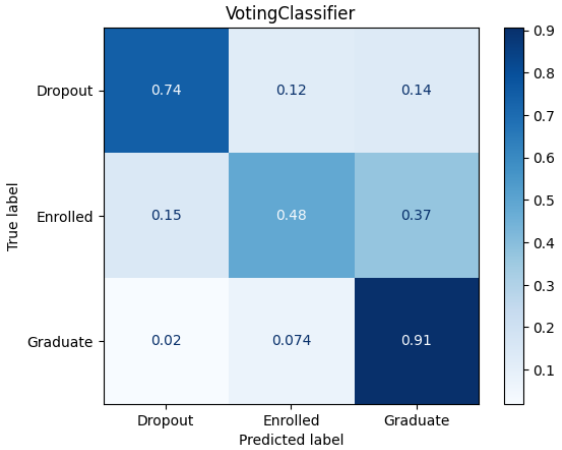

Fig. 7. Soft Voting Random Oversampling Results

The SMOTE Model produces the highest overall accuracy and specificity out of the three balancing methods. Due to its creation of synthetic samples, the model can maintain the underlying trends in the original data while simultaneously keeping diversity. It was interesting to see that between the oversampled and SMOTE techniques, the enrolled label had the most accuracy improvement, probably due to the slight data diversification of the label from SMOTE. For the graduate label, it looks like the sensitivity for SMOTE models dropped compared to the oversampled models; however, this is most likely due to overfitting in the oversampled models. For general use cases, the SMOTE models produce the most accurate predictions with the least likelihood of overfitting.

	Dropout Sensitivity	Enrolled Sensitivity	Graduate Sensitivity	Average Sensitivity
RF Undersampling	0.69	0.68	0.82	0.730
XGB Undersampling	0.74	0.68	0.79	0.737
Voting Undersampling	0.73	0.70	0.81	0.747
RF Oversampling	0.74	0.42	0.91	0.690
XGB Oversampling	0.75	0.47	0.89	0.703
Voting Oversampling	0.74	0.48	0.91	0.710
RF SMOTE	0.75	0.52	0.86	0.710
XGB SMOTE	0.77	0.58	0.87	0.740
Voting SMOTE	0.78	0.58	0.87	0.743

Fig. 8. Sensitivity Table

VI. Conclusion & Discussion

In our study, we were able to train 3 ensemble tree models and train them with 3 data sampling techniques. This resulted in the development of two outstanding models with varying use cases. Our results showed that the overall most accurate model was a Soft Voting Ensemble model that utilized SMOTE data sampling which had an accuracy of 80.36%. This is the model that we developed for our web-based front end because of its high generalizability for all use cases.

Other models we developed could be helpful in other use cases. One notable model is the Soft Voting Ensemble model trained on an undersampled dataset. Due to the model's bias toward predicting given students as a minority group (dropout or enrolled), this model would be particularly useful for institutions that want to identify more at-risk students. In the dataset, students in the enrolled category will graduate after the average course period, but when this model is applied to the real world, these students are at moderate risk of dropping out. Thus, this model is helpful if an institution has either plentiful resources or desires to keep a high graduation rate. Conversely, if an institution does not have enough resources and must allocate them strategically, the general use case model would be more than sufficient.

A strength of this study is that we combined ensemble tree methods paired with various data sampling techniques. As illustrated by altered use cases, we were able

to illustrate that undersampling, although prone to bias, can be helpful in certain situations.

In order to improve the versatility of our project for future application, we could investigate risk factors throughout a student's entire academic journey rather than only at the time of enrollment as our dataset does. Our current dataset is limited by an absence of variables that could potentially influence a student's academic success over 3-4 years. An important attribute to track would be the accumulation of student debt which can influence a student's academic persistence and access to resources.

VII. Roadmap

Project Management Throughout (Amy Y)

Part 1

- Data Management - creating graphs & analyzing data [Amy B, Joel]
- Data Visualization - understanding data & writing up data [Amy Y]
- Machine Learning & Algos - find relevant research papers to include in the paper and inspire the project & developing procedure [Ben & Joey]

Part 2

- Machine Learning & Algos - Building & Training the model [Ben & Joey]
- Quality Assurance - Come up with ways to test & improve the model [Ben & Joey]
- Write up the methodology, developing accurate models of prediction, evaluation of models [Amy B & Amy Y]
- Research and experiment with best frontends [Joel]

Part 3

- Software development - create frontend [Joel]
- Machine Learning & Algos + QA (3-4): Finalize model & Results [Ben & Joey]
- Researcher - Explain experiment results & write up conclusion (1-2) [Amy B + Amy Y]
- Prepare Presentation & Questions [Amy Y]

VIII. References

Dataset: https://archive.ics.uci.edu/dataset/697/predict+students+dropout+and+academic+success

Realinho,V., Martins, M., Machado, J., and Baptista, L. (2021). Predict students' dropout and academic success. UCI Machine Learning Repository. https://doi.org/10.24432/C5MC89.

Literature:

Gautam, A. K., & Bansal, A. (2023). Email-based cyberstalking detection on textual data using multi-model soft voting technique of machine learning approach. Journal of Computer Information Systems, 63(6), 1362–1381. https://doi.org/10.1080/08874417.2022.2155267

Martins, M. V., Tolledo, D., Machado, J., Baptista, L. M., & Realinho, V. (2021). Early prediction of student's performance in Higher Education: A case study. Advances in Intelligent Systems and Computing, 1365, 166–175. https://doi.org/10.1080/08874417.2022.2155267

Thammasiri, D., Delen, D., Meesad, P., & Kasap, N. (2014). A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition. Expert Systems with Applications, 41(2), 321–330. https://doi.org/10.1016/j.eswa.2013.07.046

IX. Appendix

Appendix A: Attribute Table

Attribute	Type	Description
Marital Status	Discrete, categorical	1—Single 2—Married 3—Widower 4—Divorced 5—Facto union 6—Legally separated
Application Mode	Discrete, categorical	Student's type of application consists of 18 labels. The labels are categorized into general groups for ease of understanding below. • Transfer from another institution or change of course • Applicant from which application cycle, includes information if the student is from a Portuguese territory • Alternate path degree student (eg. over 23 years old, Technical degree)
Application Order	Discrete, numerical	Applicant's desirability to their enrolled university, options from 0 to 9 • 0 – first choice • 9 – last choice
Course (Major)	Discrete, categorical	Course / Majors: Biofuel Production Technologies (33), Animation and Multimedia Design (171), Social Service (evening attendance) (8014), Agronomy (9003), Communication Design (9070), Veterinary Nursing (9085), Informatics Engineering (9119), Equinculture (9130), Management (9147), Social Service (9238), Tourism (9254), Nursing (9500), Oral Hygiene (9556), Advertising and Marketing Management (9670), Journalism and Communication (9773), Basic Education (9853), Management (evening attendance) (9991)
Daytime / Evening Attendance	Discrete, binary	Time of day for classes • 1 – day time • 0 – evening
Previous Qualification	Discrete, categorical	Education Status at the time of enrollment, consisting of 16 labels. The labels are categorized into general groups for ease of understanding below. • Complete or incomplete secondary education (12th year of schooling) • Complete or incomplete basic education (9th - 11th year) • Complete basic education (6th-8th year) • Higher education (bachelor's, master's, doctorate, general degree) • Specialized or general technical degree
Previous Qualification (Grade)	Continuous, numerical	Qualification for student's education at the time of enrollment on a 0 to 200 scale
Admission grade	Continuous, numerical	Equivalent to GPA in the US
Displaced	Discrete, binary	Any student who is seeking sanctuary (e.g. refugee, seeking asylum, resettlement program) • Yes (1), No (0)
Debtor	Discrete, binary	• Yes (1), No (0)
Tuition Fees Up to Date	Discrete, binary	• Yes (1), No (0)
Gender	Discrete, binary	• Male (1), Female (0)
Scholarship Holder	Discrete, binary	• Yes (1), No (0)
Age at Enrollment	Discrete, numerical	Student age at enrollment
Curricular units 1st sem (enrolled) Curricular units 1st sem (credited) Curricular units 1st sem (approved) Curricular units 1st sem (evaluations)	Discrete, numerical	Number of curricular units enrolled in the 1st semester Number of curricular units credited in the 1st semester Number of curricular units approved in the 1st semester Number of evaluations to curricular units in the 1st semester (graded units)
Curricular units 1st sem (grade)	Continuous, numerical	Grade average in the 1st semester (between 0 and 20)
Curricular units 2nd sem (enrolled) Curricular units 2nd sem (credited) Curricular units 2nd sem (approved) Curricular units 2nd sem (evaluations)	Discrete, numerical	Number of curricular units enrolled in the 2nd semester Number of curricular units credited in the 2nd semester Number of curricular units approved in the 2nd semester Number of evaluations to curricular units in the 2nd semester (graded units)
Curricular units 2nd sem (grade)	Continuous, numerical	Grade average in the 2nd semester (between 0 and 20)
GDP	Continuous, numerical	GDP at time of enrollment

Appendix B: Heatmap