# Prediction of Student's Performance and Academic Success

Ananya Mantravadi
Computer Science
North Carolina State University
USA
amantra@ncsu.edu

Nainisha Bhallamudi
Computer Science
North Carolina State University
USA
nbhalla@ncsu.edu

Pavithra Velumani
Data Science
North Carolina State University
USA
pveluma@ncsu.edu

## ABSTRACT

This study aims to predict the success of students in academic setting and also addresses the challenge of reducing academic failure of students in an institution using machine learning models. Leveraging a dataset encompassing academic, demographic, and socio-economic factors from a higher education institution, we formulate a classification tasks like dropout, enrolled and graduate.

**ACM Reference Format:**
Ananya Mantravadi, Nainisha Bhallamudi, and Pavithra Velumani. 2024. Prediction of Student's Performance and Academic Success. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION

Ensuring academic success in higher education is not only crucial for individual career prospects but also for fostering social justice and driving economic growth. For researchers and higher education institutions alike, the ability to predict whether a student is at risk of not completing their program or dropping out is of great value. This predictive capability enables the implementation of targeted strategies to support and guide these at-risk students towards successful completion of their academic endeavors. Given this backdrop, dropout remains a significant challenge plaguing higher education institutions worldwide. Defining dropout itself is a challenge, as it varies across studies due to differing definitions, data sources, and calculation methods. Although dropout analysis categorizes students based on the timing of their departure, whether early or late, meaningful comparisons across institutions are not easy due to disparities in reporting.

In this paper, we take a micro-perspective and define dropout as any departure from the institution or field, regardless of timing, before completion. By focusing on factors observable before registration and excluding internal assessments post-enrollment, it only relies on information available at the moment of enrollment. This is because the focus is to develop a system that helps to segment students as soon as possible from the beginning of their path at higher education. The problem is formulated as a three-category classification task (dropout, enrolled, and graduate) at the end of the normal duration of the course.


*Conference'17, July 2017, Washington, DC, USA*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

By leveraging machine learning to predict student performance, this project aims to equip institutions with tools to intervene effectively and support students along their academic journey, ultimately fostering greater success and retention rates in higher education.

The remainder of this paper is structured as follows. Section 2 presents a brief review of the literature. Section 3 describes the methodology including the novel aspects, rationale, and approach. Section 4 describes the dataset, states the hypothesis, and gives details of the experimental setup. Section 5 presents the the results and insights gained from the analysis. Section 6 attempts to interpret our final model through its feature importance and an explainability method, LIME. Finally, section 7 summarizes and indicates some directions for future work. The code for this project is available at https://github.com/ananya173147/Student-Performance-and-Academic-Success

## 2 BACKGROUND

Notably, recent years have witnessed a surge in interest in early prediction models for student performance. Related works have previously used several algorithms like Support Vector Machines, Naïve Bayes, Decision Trees, Random Forests, Bagging Decision Trees and Adaptive Boosting Decision Trees, obtaining the higher scores with Random Forests and Adaptive Boosting Decision Trees. Beaulac and Rosenthal [1] worked on large dataset comprising 38,842 students from a major Canadian university. By considering the initial courses attempted and grades achieved by students, the authors aimed to forecast program completion and, if applicable, the chosen major. They employed Random Forests to predict academic success. Their results demonstrated an overall 79% accuracy in predicting program completion, with 91% accuracy for students who completed their program and 53% for those who did not. Hoffait and Schyns [5] conducted a study utilizing a dataset comprising 6845 students, employing standard classification methods such as Random Forest, Logistic Regression, and Artificial Neural Networks to identify freshmen's profiles prone to encountering significant challenges in completing their first academic year. Despite employing various algorithms, the accuracy obtained for the majority class hovered around 70%, with less than 60% for the minority class. They also utilized RF to devise a strategy aimed at enhancing prediction accuracy for specific classes of significant interest although it did not consistently result in an increase in identifying the number of students at risk. The drawback of both of these approaches is that they do not present results with any other evaluation metrics apart from accuracy which can be very misleading in the context of classification.

Since there is also an added problem of class imbalance against the dropout class, sampling techniques like SMOTE [2] and ADASYN

[4] were used. Thammasiri et al. [9] proposed various class balancing strategies and standard classification methods employed to predict dropout within a dataset comprising 21,654 students. The approach evaluated class balancing techniques including random undersampling, random oversampling, and synthetic oversampling to discern their effectiveness in predicting dropout outcomes. The results indicated that the support vector machine combined with SMOTE data-balancing technique achieved the best classification performance with a 90.24% overall accuracy on the 10-fold holdout sample.

# 3 PROPOSED METHOD

Our project aims to predict students' likelihood of dropping out and their potential academic success based on a comprehensive set of features. The problem is formulated as a three-category classification task (dropout, enrolled, and graduate) at the end of the normal duration of the course.

## 3.1 Feature Engineering

Although the dataset is said to have already performed rigorous data preprocessing to handle data from anomalies, unexplainable outliers, and missing values, in closer inspection we have noticed a few problems in the way encoding was done for nominal features like parents' occupation and nationality. This may result in loss of information, as the model may interpret the encoded integers as ordinal, assuming a certain order or hierarchy that might not exist in the original data. There is a possibility of introducing bias in the model, which might unintentionally give more importance to certain occupations. We propose to reverse-engineer the processed dataset and perform domain-specific feature engineering that better represents the relationships between different categories. For example, we would like to create binary features indicating whether an occupation falls into certain broader categories or clusters.

Occupation-based feature engineering is a pivotal strategy in predictive modeling, especially where professions heavily influence outcomes. Individual occupations are mapped to predefined categories like 'Management,' 'Technical Workers,' and 'Service Workers.' This categorization balances reducing complexity with retaining essential information. Each binary feature indicates the presence or absence of a specific occupational category within an individual's profile, offering a concise yet comprehensive portrayal of their professional sphere.

Furthermore, we have also performed feature engineering on the qualifications of parents. Initially, the qualifications were encoded using individual codes, but these codes may not accurately represent the underlying relationships or hierarchies between different levels of education. To address this issue, we have grouped the individual qualification codes into broader categories such as 'Secondary Education', 'Higher Education - Bachelor's Degree', 'Higher Education - Degree', 'Higher Education - Master's', 'Higher Education - Doctorate', 'More than one Higher Education', and 'Other'. This categorization helps capture the essential information while reducing the complexity of the feature space. The new categorical features for parents' qualifications provide a more meaningful representation and may improve the model's ability to learn patterns related to educational attainment.

## 3.2 Approach

In our project, we propose going beyond conventional methods by not only utilizing the standard machine learning techniques. Specifically, given that boosting techniques gave higher performance, we plan to explore and implement voting ensemble models combining different classification algorithms like logistic regression, random forests, and XGBoost. By combining models, we aim to reduce overfitting and improve generalization performance. Bagging reduces variance by averaging or voting over multiple models. Boosting, on the other hand, reduces bias by emphasizing misclassified instances. Combining these two approaches can lead to a more balanced model that benefits from both the variance reduction of bagging and the bias reduction of boosting. Finally, we explored methods for interpreting the ensemble model's predictions through techniques like SHAP (SHapley Additive exPlanations) [6] values or LIME (Local Interpretable Model-agnostic Explanations) [7] that can help provide insights into how different features contribute to the model's decisions.

## 3.3 Rationale

Our rationale is that our algorithm is robust enough for us to outperform the current methods even without oversampling techniques. They may introduce noise or overfitting into the dataset, particularly if the synthetic samples are not representative of the true underlying distribution. Additionally, oversampling can inflate the size of the dataset, leading to increased computational complexity and memory requirements, which can be impractical for large datasets. By interpreting our results, we focus on making our algorithm more transparent and explainable which was previously not explored in this problem and dataset as far as we researched.

# 4 EXPERIMENTS

## 4.1 Dataset

The dataset [3] provided for this study originates from diverse sources within the Polytechnic Institute of Portalegre (IPP), Portugal, and encompasses students enrolled in various undergraduate disciplines, including agronomy, design, education, nursing, journalism, management, social service, and technologies. It consolidates information available at the time of student enrollment, encompassing academic trajectories, demographic profiles, and socioeconomic indicators. Additionally, it captures students' academic performance metrics at the culmination of the first and second semesters. It consists of 4424 instances and 36 features.

Given the nature of the problem, which involves addressing class imbalance, the dataset poses a significant challenge in model development and evaluation. Nevertheless, it presents an opportunity to explore sophisticated machine learning techniques tailored to mitigate class imbalance and enhance predictive accuracy across all classes.

## 4.2 Hypotheses

Investigating the influence of parental occupation and qualifications on college dropout rates is crucial for comprehensively understanding the multifaceted factors impacting student success in higher education. Our hypothesis posits that students from diverse parental occupational backgrounds and educational qualifications may encounter varying levels of support and resources, significantly influencing their propensity to drop out of college.

To rigorously test this hypothesis, we have undertaken a meticulous feature engineering approach aimed at rectifying encoding anomalies and maximizing the representation of parental occupational and educational data. Specifically, we have reversed-engineered the processed dataset to create binary features that succinctly capture the presence or absence of specific occupational categories within individuals' profiles. These categories, including 'Management,' 'Technical Workers,' and 'Service Workers,' are derived from a thorough analysis of the original data and are meticulously designed to balance simplicity with information richness.

Moreover, in addressing the encoding issues of parental qualifications, we have performed comprehensive feature engineering by grouping individual qualification codes into broader categories such as 'Secondary Education,' 'Higher Education - Bachelor's Degree,' 'Higher Education - Degree,' 'Higher Education - Master's,' 'Higher Education - Doctorate,' 'More than one Higher Education,' and 'Other.' This refined categorization not only preserves essential information but also reduces the complexity of the feature space, thereby enhancing the model's capacity to discern patterns related to educational attainment.

Through the utilization of occupation-based and qualification-based feature engineering techniques, we aim to elucidate whether students from specific parental occupational backgrounds and educational levels exhibit divergent dropout rates compared to their peers. By analyzing these nuanced relationships, our research endeavors to identify disparities in college dropout rates among students from diverse socioeconomic backgrounds and inform targeted interventions aimed at fostering educational equity and improving college retention rates.

Ultimately, this study contributes to a deeper understanding of how parental occupation and qualifications shape students' educational experiences, underscoring the significance of creating inclusive and supportive environments in higher education.

## 4.3 Experimental Design & Settings

In the final phase of our project, we aimed to develop robust predictive models for assessing students' likelihood of dropping out and their academic success. by leveraging various machine learning algorithms tailored to our dataset. The dataset encompasses a rich array of features including parents' occupations, nationality, previous academic performance, and socioeconomic status.

*4.3.1 Data Partitioning & Cross-Validation:* In our experimental settings, we carefully delineated our approach to ensure robust model evaluation and parameter optimization. Initially, we partitioned the dataset into Training and Test sets using an 80% and 20% split, respectively, and employed Repeated Stratified K-Fold Cross-Validation to address the inherent class imbalance within the data. This technique involved partitioning the data into 10 folds, with each fold preserving class proportions, and repeating this process thrice to obtain a comprehensive understanding of model performance across various data subsets.

*4.3.2 Hyperparameter Tuning:* Hyperparameter tuning played a crucial role in refining model efficacy. We initially explored GridSearchCV but found its exhaustive nature to be time-consuming. Thus, we opted for RandomizedSearchCV, efficiently exploring a range of hyperparameters while striking a balance between thoroughness and computational efficiency. This approach facilitated parameter optimization within a reasonable timeframe, ensuring the robustness of our models.

| Hyperparameter | Value |
|---|---|
| `base_estimator__max_depth` | [3, 5, 7, 9] |
| `n_estimators` | [50, 100, 150] |
| `learning_rate` | [0.01, 0.1, 1] |

**Table 1: Parameter Grid for Adaboost Random Forest**

| Hyperparameter | Value |
|---|---|
| `max_depth` | randint(3, 8) |
| `n_estimators` | [100, 200, 300] |
| `learning_rate` | [0.1, 0.01, 0.001] |
| `gamma` | [0, 0.1, 0.2] |
| `subsample` | [0.8, 0.9, 1.0] |
| `colsample_bytree` | [0.8, 0.9, 1.0] |

**Table 2: Parameter Grid for XGBoost**

RandomizedSearchCV was utilized to search over these hyperparameters, utilizing F1-Macro scoring for evaluation. The selected hyperparameters were chosen based on their performance in cross-validation, aiming to strike a balance between model complexity and generalization. The above two table are the parameters which produced best results for Adaboost with random forest model: [1] and XGBoost model [2]

*4.3.3 Evaluation Metrics:* In assessing model performance, we employed multiple evaluation metrics to gain a comprehensive understanding of predictive capabilities. Primarily, we focused on accuracy, measuring the proportion of correctly classified instances. Additionally, to account for class imbalance and ensure a balanced evaluation, we utilized the F1-Macro score, which provides a harmonic mean of precision and recall across all classes.

*4.3.4 Baseline Models:* We commenced our modeling journey with fundamental algorithms such as Logistic Regression, Decision Tree, and Random Forest. These models served as benchmarks, providing insights into initial predictive performance and guiding subsequent experimentation.

**Logistic Regression:** A classic fundamental algorithm known for its simplicity and interpretability. By fitting the data to this model, we sought to establish a baseline performance measure. Logistic Regression provides a clear understanding of how each feature contributes to the predicted outcome.

In our initial exploration, Logistic Regression yielded a testing accuracy of 0.71 and an F1-Macro score of 0.61. While this model

**Table 3: Baseline Model Performance Metrics**

| Model | Dropout | | | Enrolled | | | Graduate | | |
|---|---|---|---|---|---|---|---|---|---|
| | Precision | Recall | F1-score | Precision | Recall | F1-score | Precision | Recall | F1-score |
| Logistic Regression (Acc = 71.4) | 0.82 | 0.67 | 0.74 | 0.43 | 0.21 | 0.28 | 0.70 | 0.93 | 0.80 |
| Decision Tree (Acc = 73.7) | 0.84 | 0.68 | 0.75 | 0.47 | 0.32 | 0.38 | 0.74 | 0.93 | 0.8 |
| Random Forest (Acc = 76.0) | 0.83 | 0.74 | 0.79 | 0.51 | 0.26 | 0.35 | 0.75 | 0.94 | 0.84 |

demonstrated decent performance, it struggled particularly with recall for the 'Enrolled' and 'Dropout' classes (Refer table: [3]), indicating room for improvement in identifying students at risk of dropping out early.

These insights from Logistic Regression provide a solid foundation for further exploration with more complex algorithms and feature engineering techniques. The next steps in our model selection process will involve experimenting with Decision Trees, Random Forests, and ensemble methods to enhance predictive performance and better support student success in higher education.

**Decision Tree:** Our modeling journey continued with Decision Tree, a powerful algorithm known for its ability to handle complex interactions in the data. Decision Trees offer a transparent view of how features are split to make predictions, aiding in interpretability.

The results from Decision Tree modeling were enlightening. The model exhibited a testing accuracy of 0.73 and an F1-Macro score of 0.65. While the model exhibited strengths in predicting the 'Graduate' class, it faced challenges with 'Enrolled' and 'Dropout' classes, particularly in precision and recall, suggesting areas for refinement in identifying at-risk students.

**Random Forest:** Exploration continued with Random Forest, which is an ensemble learning that joins multiple Decision Trees to improve predictive performance and reduce overfitting. The Random Forest model yielded promising results indicating it correctly predicted the class of a student's outcome 76.0% of the time. A detailed examination of the classification report provided valuable insights into its performance.

The Random Forest model demonstrated notable improvements over Decision Tree, particularly in precision and recall for all classes where the previous models failed to do well. This suggests that the ensemble approach effectively addressed the challenges faced by the individual Decision Trees, leading to enhanced predictive performance. Table 3 shows initial model results.

*4.3.5 Ensemble Methods:* Building upon the performance of baseline models, we explored ensemble techniques to further enhance predictive capabilities. Bagging and Boosting methods were employed with both Decision Tree and Random Forest classifiers, leveraging their strengths in handling complex interactions within the data.

**Bagging:** Bagging with Decision Tree and Random Forest yielded testing accuracies of 0.75 and 0.75, respectively, with corresponding F1-Macro scores of 0.67 and 0.65. These ensemble methods showcased improvements over standalone Decision Tree and Random Forest models, particularly in precision and recall across all classes.

**Adaptive Boosting:** AdaBoost was applied with both Decision Tree and Random Forest classifiers. AdaBoost with Decision Tree achieved a testing accuracy of 0.74 and an F1-Macro score of 0.67, while AdaBoost with Random Forest exhibited the highest performance among all ensemble methods, with a testing accuracy of 0.77 and an F1-Macro score of 0.70. These results highlight the benefits of ensemble methods in enhancing predictive capabilities, with AdaBoost with Random Forest emerging as the top-performing ensemble technique. Additionally, XGBoost demonstrated promising results with a testing accuracy of 0.76 and an F1-Macro score of 0.69, further emphasizing the effectiveness of ensemble learning in improving model performance.

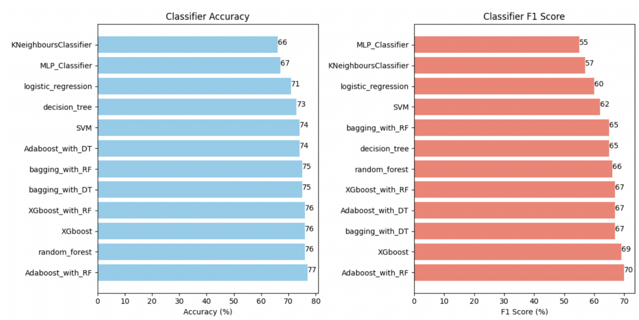


**Figure 1: Model Results**

*4.3.6 Model Selection & Final Ensemble:* Following extensive experimentation, we identified Adaboost with Random Forest and XGBoost as top-performing models in individual and ensemble settings. To maximize predictive performance, we employed a soft voting ensemble, combining the strengths of these models. This ensemble approach yielded the highest accuracy of 77.8% and an F1-Score of 71%, showcasing its effectiveness in predicting students' academic outcomes.

| **Model** | **Testing Accuracy** | **F1-Macro** |
|---|---|---|
| Logistic Regression | 0.71 | 0.61 |
| Decision Tree | 0.73 | 0.65 |
| Random Forest | 0.76 | 0.66 |
| Bagging with DT | 0.75 | 0.67 |
| Bagging with RF | 0.75 | 0.65 |
| Adaboost with DT | 0.74 | 0.67 |
| Adaboost with RF | **0.77** | **0.70** |
| XGBoost | **0.76** | **0.69** |
| Voting Classifier (Final) | **0.78** | **0.71** |

**Table 4: Comparison of Model Performance**

*4.3.7 Optimization Strategies:* To expedite model training and hyperparameter tuning, we implemented several process optimization techniques. This included leveraging cloud-based computing resources, such as Google Colab's TPU runtime, to reduce processing time. Additionally, parallel processing was employed to maximize computational efficiency, utilizing all available cores for hyperparameter optimization.

*4.3.8 Future Directions:* While our current approach has demonstrated promising results, there remain avenues for further exploration. Future experiments may involve refining ensemble methods, exploring additional feature engineering techniques.

## 5 RESULTS

The exploration into Decision Tree and Random Forest models provided valuable insights into their strengths and areas for improvement in predicting student outcomes. Decision Tree offered transparency in feature importance but struggled with precision and recall, especially for 'Enrolled' and 'Dropout' classes. Random Forest, on the other hand, showcased significant improvements in these metrics, leveraging ensemble learning to provide more robust predictions.

Further, Bagging and Boosting techniques provided valuable insights into their impact on predictive performance. Bagging, in combination with Decision Tree and Random Forest, showcased improvements in precision and recall, particularly for classes with lower performance in standalone models. AdaBoost demonstrated mixed results, with AdaBoost with Random Forest showing promising accuracy. XGBoost was among the top models with an accuracy of 0.76 and F1-score of 0.69.

After evaluating multiple combinations of both voting and stacking ensembles with about the same level of performance, we found out that the voting classifier combining XGboost and Adaboost with Random Farost gave a slightly higher edge in performance, resulting in 77.8% accuracy, 70.8 % average F1, and 77% weighted F1. Hence our rationale of combining models was validated. We aimed to reduce overfitting and improve generalization performance. Bagging reduced variance by averaging or voting over multiple models. Boosting, on the other hand, reduced bias by emphasizing misclassified instances. Combining these two approaches led to a more balanced model that benefits from both the variance reduction of bagging and the bias reduction of boosting.

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **Dropout** | 0.85 | 0.76 | 0.80 | 316 |
| **Enrolled** | 0.56 | 0.40 | 0.47 | 151 |
| **Graduate** | 0.79 | 0.93 | 0.85 | 418 |
| **Accuracy** | | | **0.78** | 885 |
| **Macro Avg** | 0.73 | 0.70 | **0.71** | 885 |
| **Weighted Avg** | 0.77 | 0.78 | **0.77** | 885 |

**Table 5: Voting Classifier Result**

## 6 DISCUSSIONS

In this section, we discuss and analyze how our final model, the voting classifier consisting of Adaptive Boosting with Random Forests and eXtreme Gradient Boosting performs. From the Figure 2, we can understand which features in the dataset contribute the most while making predictions. We can see that the top features include the number of curricular units approved, enrolled, and number of evaluations to curricular units in the first and second semesters. Whether the tuition fees is up to date or not also is important. One more observation we can make is that features like 'Educational special needs', 'International', and 'Marital Status' do not take up a lot of importance suggesting that this model is socially biased which is a good sign. It reinforces the idea that student success depends only on factors related to their educational progress and not their background.

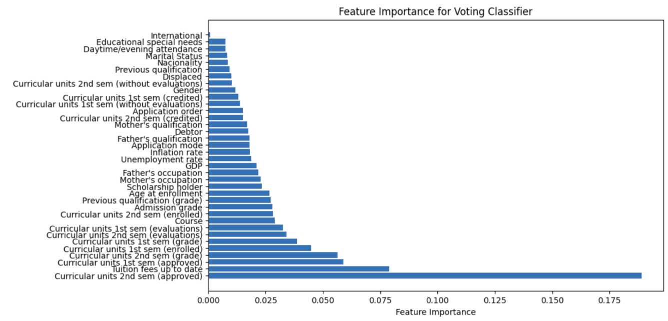


**Figure 2: Feature Importance of VotingClassifier**

### 6.1 Local Interpretable Model-agnostic Explanations

Data-driven decision-making presents a challenge in terms of a trade-off between accuracy and interpretability. While complex algorithms boast higher accuracy rates, they often operate as black-box methods, obscuring the reasoning behind their predictions. In contrast, business settings frequently prioritize simpler, more interpretable models, even at the expense of some accuracy. However, bridging this gap between accuracy and interpretability is essential. Techniques like Local Interpretable Model-agnostic Explanations (LIME) [7] offer a solution by providing a means to explain individual predictions from any classifier or regressor. By computing a local surrogate model, LIME faithfully elucidates the underlying rationale behind specific predictions, thus reconciling the need for accuracy with the imperative for interpretability in practical applications. In our paper, we take two instances, one of a correct prediction, and the other of an incorrect prediction and try to understand how LIME explains these results predicted by our model.

The correct prediction by the model as Graduate on an instance is shown in Figure 3. The prediction probabilities show that Graduate has a majority probability of 0.81. There are binary classification tables (color-coded) for each of the classes that show the rule importance along with weights that the local surrogate model evaluated. Each of these tables indicate whether the instance would belong to that class or not. As we can observe, 'Graduate' side of the first table has higher weighted rules, similar to the 'NOT Dropout' side of the third table. Hence, another perspective of explanation can be

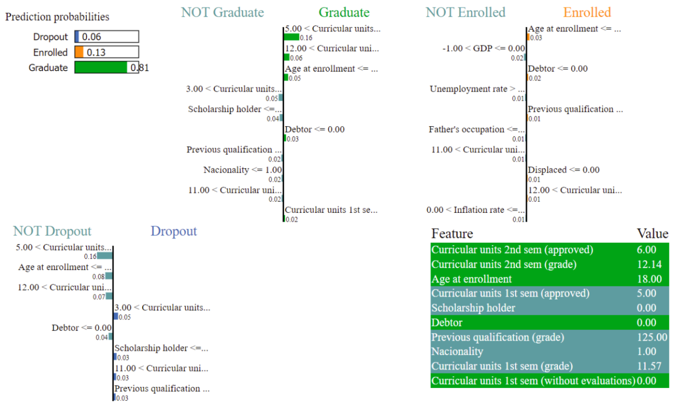


**Figure 3: True: Graduate, Predicted: Graduate**

gained through LIME on this instance, along with understanding which rules contribute to making this prediction.

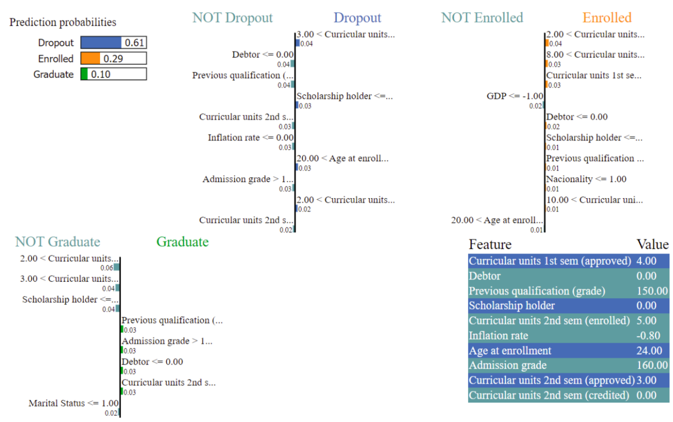


**Figure 4: True: Enrolled, Predicted: Dropout**

The second prediction is an interesting case where the model incorrectly predicted the instance to be a dropout when the true label was enrolled. The model predicted with a probability of 0.61, but LIME tells a different story. Most of the higher weighted rules fall onto the 'Enrolled' side, and by this it must mean that the instance should be classified into the same. Therefore, LIME can be another point of consideration for possible explanations when we evaluate new individual predictions. Having said that, LIME needs to be used with caution [8]. The LIME model is not very 'stable' - small variations in the input data can completely change the LIME interpretations. Moreover, the right definition of the neighboured the algorithm looks into for creating the local model still remains an unsolved challenge.

## 7 CONCLUSIONS

In conclusion, our study has addressed the pressing challenge highlighted in the problem statement regarding the prediction of student dropout rates in higher education. Despite our comprehensive approach, including meticulous feature engineering and leveraging an ensemble Voting Classifier, we found that our method yielded only modest improvements over previous methods. While we achieved an accuracy boost from 73% to 77.8% and an average F1 score enhancement from 65 to 70.8, the gains were not as substantial as anticipated.

Interestingly, our analysis suggests that the feature engineering efforts aimed at capturing parental occupation and qualifications did not significantly contribute to the model's predictive performance. This observation raises questions about the presumed influence of parental occupation and qualifications on academic success. It implies that these factors might not play as substantial a role in determining student dropout rates as previously assumed. This finding underscores the complexity of factors affecting student outcomes in higher education and highlights the need for further exploration beyond traditional socioeconomic indicators.

Furthermore, our evaluation indicates that the model's performance is not biased towards any particular class or demographic group. Despite the imbalanced nature of the dataset, with the minority class "Enrolled" suffering from a lower F1-score compared to the other two classes, our model demonstrates fairness and impartiality in its predictions. This reassures the integrity of our classification framework and underscores the importance of equitable predictive modeling in educational contexts.

Looking ahead, there is a need to explore state-of-the-art methods to address the challenges posed by imbalanced datasets and potentially rethink the role of parental occupation and qualifications in academic success. While deep learning architectures may not be suitable for the relatively small dataset size, techniques such as sophisticated feature selection, one-class classification, and resampling warrant further investigation. Additionally, alternative goals such as anomaly detection or change detection could broaden the scope of future research endeavors, offering promising avenues for innovation and refinement in our classification framework.

## REFERENCES


[1] Cédric Beaulac and Jeffrey S Rosenthal. 2019. Predicting university students' academic success and major using random forests. *Research in Higher Education*, 60, 1048–1064.

[2] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321–357.

[3] Dheeru Dua and Casey Graff. 2019. UCI machine learning repository. https://archive.ics.uci.edu/dataset/697/predict+students+dropout+and+academic+success.

[4] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. 2008. Adasyn: adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. Ieee, 1322–1328.

[5] Anne-Sophie Hoffait and Michaël Schyns. 2017. Early detection of university students with potential difficulties. *Decision Support Systems*, 101, 1–11. DOI: https://doi.org/10.1016/j.dss.2017.05.003.

[6] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

[7] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

[8] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. Fooling lime and shap: adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 180–186.

[9] Dech Thammasiri, Dursun Delen, Phayung Meesad, and Nihat Kasap. 2014. A critical assessment of imbalanced class distribution problem: the case of predicting freshmen student attrition. *Expert Systems with Applications*, 41, 2, 321–330.