

# **Predicting Student Dropout and Academic Success**

**ECS 171 Machine Learning**

**Group 19 Project Report**

**Group Members: Shuyu Cai, Leo Kim, Arya Ranadive, Srivatsan  
Srikanth**

**Github Repository:**

**<https://github.com/ranapp/ECS171Project>**

## Introduction and Background

College education is an important part of many people's lives. To many people it is one of the biggest opportunities that can jumpstart someone's career and livelihood. However, everyone that goes to college comes from different backgrounds and live in different ways while at college. This can have an enormous impact on the success of the student with respect to their college career and their graduation. It could be important to analyze these different factors and somehow relate them to the anticipated success of the students. Some of these factors include previous qualifications, parent's education, parent's occupation, and educational aspects like number of units. Although not a full causal relationship, a relationship can be made between these factors and the graduation rates.

Using a machine learning model to predict the graduation success of a certain student could be prone to error as there is a lot of uncertainty that comes with the type of factors and data. A 100% accurate prediction is near impossible to come up with because of all of this uncertainty. We humans would also find it hard to predict success rates of students based on these factors because people are different and their personalities and mentalities handle situations in different manners. To make this model useful we have to train it very well to really see what factors seem to be a potential cause of a student dropping out of college.

The machine learning model to predict the success rates of students can be very useful in a multitude of ways.

- One of these ways is that universities can better aid students that may need extra support in terms of academics. The model will zero in on some of the areas of the demographic where students might need the most help to graduate. With this information universities could help better prepare their students.
- Another way that these models could be used is that schools could analyze where their organizational structure is lacking in academics and sufficiently try to improve those areas to improve the university effectiveness in an overall manner
- Yet another way that this machine learning prediction model could be used is a guide for students. Using this model, students can best determine their college coursework in order to graduate on time without overloading with units. It would help destress students and make sure that their college life is not too strained.

As can be seen, the graduation predictor can be very useful as it can help improve education in areas where there are signs of struggle.

## Literature Review

The paper, [Predicting Student Dropout and Academic Success](#), worked with the same dataset that we are using. The authors of this paper are also the same authors of the dataset that we are using (Realinho V, Machado J, Baptista L, Martins MV). The paper identified, described, and grouped the attributes into classes: demographic, socioeconomic, macroeconomic, and academic data. The values of the categorical variables are also provided in this paper in Appendix A. The paper also provided us with a reference on how to determine which attributes were more important by looking at heatmaps and using XGBoost for feature importance.

The paper, [Data Balancing Techniques for Predicting Student Dropout Using Machine Learning](#), (Mduma N) presented approaches on how to predict student dropout by using machine learning methods. Some methods mentioned in this paper were Feed Forward Neural Network, Decision Tree, Support Vector Machines, and Naive Bayes models. However, the main methods used in this paper to predict student dropout were Logistic Regression, Random Forest, and Multi-Layer Perception. In the results section of the paper, it was found that Logistic Regression performed the best when used to classify the highest number of student dropouts and misclassify the lowest, followed by MLP and Random Forest.

Overall, both papers were used in our project by combining the techniques and methods used to

predict student outcomes for our project. Some techniques we used to explore data from the first paper were using heatmaps and feature importance using XGBoost. Also, while the second paper used different datasets and the objective was to predict only student dropouts, this paper was still useful because it gave references and inspiration in what models to use and how they could be used in our project to predict both outcomes of student dropout and graduation.

## Dataset Description and Exploratory Data Analysis

The dataset used in this project was created from a higher education institution and provides data from students enrolled in a variety of undergraduate degree programs. The dataset includes academic related information known at the time of enrollment, demographic, socioeconomic, and macroeconomic data. These factors or attributes are used to provide insight into student outcomes along with building models and predicting student outcomes which include dropout and graduated outcomes.

The dataset is in the format of a csv file with 4424 students/records/rows with 35 attributes/columns. The dataset contains no missing values. Grouped by academic, demographic, socioeconomic, and macroeconomic data, the 35 attributes are either numerical or categorical variables. The full list of the attributes and their descriptions can be found [here](#). The categorical attributes and the meaning of their values are provided [here in Appendix A](#). The last column in the dataset, Target, indicates whether the student has graduated, dropped out, or is still enrolled. The other attributes will be used to predict student target outcomes when creating the models.

When exploring the data, we looked at a basic overview of some attributes. Then, we did a deeper exploration into the attributes by looking at a correlation heatmap and a feature importance plot.

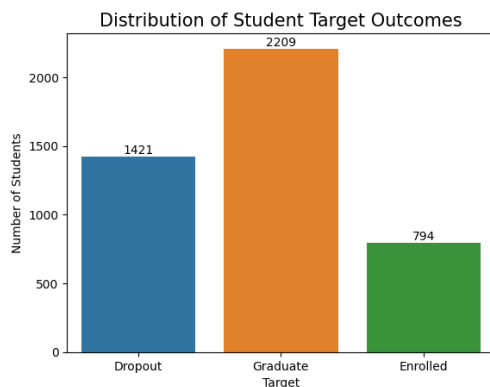


Figure 1: Overview of dataset's target outcomes

In Figure 1, with a total number of students at 4424, the majority of students in the dataset have graduated at 50% (2209 of 4424) and 32% (1421 of 4424) have dropped out. The rest at 18% (794 of 4424) are still enrolled. For the purpose of our project, we will drop the Enrolled rows from our dataset because we need to predict student outcomes of either graduated or dropped out.

Other attributes we looked at included a mix of demographic, economic, and academic data. For demographic information, most of the students in the dataset that have graduated are female and the average age of the students are in their 20s with the outliers starting at around 35 years of age and older. For economic information, most students that graduated had tuition fees that were up to date. Also, the majority of students that graduated and dropped out were non scholarship holders while the majority of scholarship holders graduated. For academic data, most of the students were enrolled in Nursing, Social Service, Journalism and Communication, and Veterinary Nursing.

After looking at a basic overview of the dataset, we explored the attributes deeper by using a heatmap and feature importance plot as seen in Figure 2. The heatmap was used to find correlation

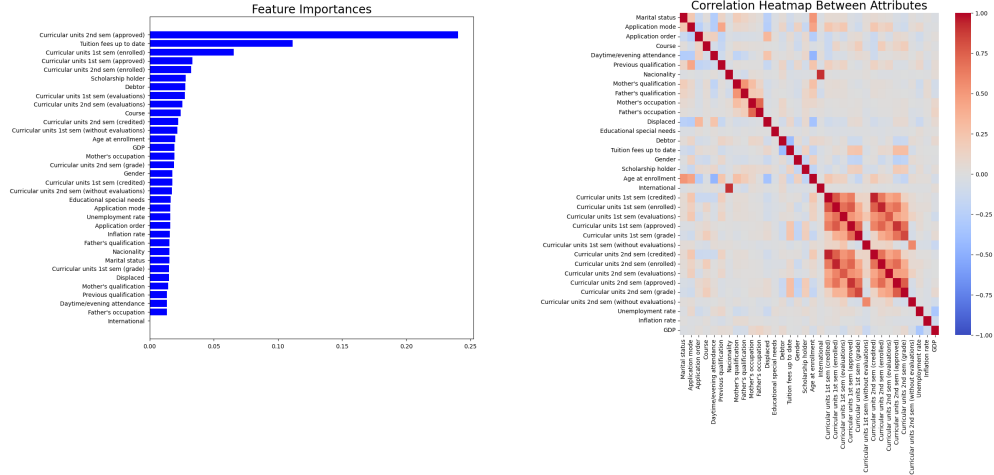


Figure 2: Heatmap and feature importance

between the different attributes. The feature importance plot was used to find the more important features or attributes in predicting student outcomes.

Using the heatmap, we found that there was more correlation between academic related attributes such as units that were taken during enrollment. There was also correlation between the attributes of international and national. We found that there was a dominant amount of Portuguese students at 4313 while the next biggest group was Brazilian students at 38. Hence, the correlation between international students and the nacional attribute with the large number of students from Portugal.

The feature importance plot was done by obtaining importances from a tree based model and by using XGBClassifier model from the XGBoost python module. In the feature importance plot, it was found that the more important features were academic attributes that were related to units and other important features included economic related attributes.

By using these two plots in Figure 2, it helped us in figuring out which attributes to drop or keep in predicting student target outcomes.

## Proposed Methodology

In this dataset, our objective is to predict students' academic performance using multiple attributes. Because this is a fundamental classification problem, we can use several traditional models that are commonly used for such scenarios. Within this project, we chose polynomial regression with sigmoid, logistic regression with K-fold cross validation, support vector machines (SVM), decision trees, Naive Bayes, and neural networks. These models have proven to be effective in classification tasks and offer diverse approaches to address the prediction of students' academic outcomes.

Initially, we conducted data cleaning and preprocessing procedures. This applies one-hot encoding to categorical attributes and normalizes numerical attributes. Additionally, we manually dropped certain columns based on the visualization of feature importances, as these attributes had a comparatively lower impact on predicting graduation or dropout compared to others. These preprocessing steps were applied to polynomial regression with sigmoid, logistic regression, SVM, and neural networks. Notably, decision tree models were not subjected to one-hot encoding since they can effectively handle both numerical and categorical attributes. For Naive Bayes, we separate the data into two cases based on numerical and categorical attributes, enabling independent testing.

Subsequently, we split the datasets into training and test sets with an 80:20 ratio, ensuring consistency across all models. By setting the random seed to 42, we guarantee that the same datasets are used for training and testing each model. This approach facilitates accurate and equitable comparisons based on model accuracy.

For the model selection, we carefully considered the advantages and suitability of various models to the characteristics of our dataset. Polynomial regression with a sigmoid transformation allows us to predict probabilities, facilitating the testing of different thresholds to optimize accuracy. Logistic regression emerged as the most popular choice for binary classification, as it models the probability of events and offers interpretability. So it is definitely suited for our binary classification datasets. To obtain a more accurate estimate of the model’s performance on unseen data, we applied k-fold cross-validation with logistic regression, partitioning the data into multiple folds for iterative training and testing.

Moreover, the decision tree is also a powerful model for classification, and it can handle both numerical and categorical features without one hot encoding. Also, it allows us to understand the decision-making process. Since our dataset has above 200 attributes after one-hot encoding, Support Vector Machines (SVM) proved helpful due to their ability to handle high-dimensional data and effectively capture complex decision boundaries. Naive Bayes, with its simplicity and efficiency, allowed us to apply the appropriate variant for both numerical and categorical attributes, facilitating accurate comparisons of model accuracy. Furthermore, neural networks demonstrated their capability to learn complex patterns, and we applied hyperparameter tuning to optimize the performance of MLP classification. By considering and evaluating these models, we aim to select the most suitable approach for accurate prediction of graduation and dropout.

## Experimental Results

### Polynomial Regression with Sigmoid

	precision	recall	f1-score	support
0	0.96	0.75	0.84	296
1	0.85	0.98	0.91	430
accuracy			0.89	726
macro avg	0.90	0.87	0.88	726
weighted avg	0.89	0.89	0.88	726

**Confusion Matrix:**  
[[223 73]  
[ 10 420]]

Figure 3: Testing result of polynomial regression with sigmoid when degree = 1

We used Sklearn’s linear regression function and polynomial features to implement polynomial regression models for our prediction task. We experimented with two cases: polynomial degree 1 and polynomial degree 2. After obtaining the predictions from the polynomial regression, we applied the sigmoid function to transform the continuous outputs into probabilities, representing the likelihood of a student’s graduation. This enabled us to establish an optimal threshold for classification by iterative tests, ensuring the highest accuracy possible.

Upon evaluating the results, we found that the model with a polynomial degree of 1 exhibited superior performance with an accuracy of 89%. This suggests that a linear relationship, captured by a first-degree polynomial, is more effective in predicting graduation outcomes. On the other hand, the model with a polynomial degree of 2 has an accuracy of 58% and demonstrated signs of overfitting, indicating that it may have captured noise or irrelevant patterns from the training data. As a result, we decided against testing higher polynomial degrees to prevent overfitting issues. Overall, the polynomial regression with the sigmoid function provided a good methodology for predicting graduation and dropout, with the degree 1 polynomial achieving the highest accuracy in our experiments of polynomial regression with the sigmoid.

### Logistic Regression

We initially applied logistic regression, a popular model for binary classification, to train our model. Logistic regression provided an accuracy of 90%, which is a very good model. However, to further enhance the accuracy and evaluate the model’s robustness, we chose k-fold cross-validation. By setting the number of folds to 5, we aimed to ensure consistent and reliable comparisons among different models.

```

Accuracy scores: [0.9146005509641874, 0.9104683195592287, 0.9077134986225895, 0.9008264462809917, 0.9104683195592287]
Average accuracy: 0.9088154269972453

Confusion Matrix:
[[251  52]
 [ 13 410]]

```

---

Figure 4: Testing result of logistic regression with k-fold cross validation

Through logistic regression with k-fold cross-validation, we achieved a notable improvement in accuracy, with the model yielding a higher accuracy of 91%. This enhancement of approximately 1% demonstrates the effectiveness of k-fold cross-validation in refining the model's performance. This is the highest accuracy we got for our project. We think this is an ideal model for binary classification. And, the k-fold cross-validation allowed us to obtain a more comprehensive assessment of the model's accuracy by considering various combinations of training and test sets. This approach helps to mitigate the potential biases that may arise from using a single training and test split. By the averaged performance across multiple folds, we got a more reliable estimate of the model's accuracy and its generalization capability.

## Neural Network

```

Case 1: logistic 100 batch size
Accuracy : 0.8512396694214877
Mean Square Error : 0.1487603305785124
[[1 0]
 [1 0]
 [0 1]
 [0 1]
 [1 0]]
Confusion Matrix for each label :
[[[188  27]
  [ 27 121]]

 [[121  27]
  [ 27 188]]]
Classification Report :

```

	precision	recall	f1-score	support
0	0.82	0.82	0.82	148
1	0.87	0.87	0.87	215
micro avg	0.85	0.85	0.85	363
macro avg	0.85	0.85	0.85	363
weighted avg	0.85	0.85	0.85	363
samples avg	0.85	0.85	0.85	363

Figure 5: Testing result of neural network

We decided to use a Forward Feed Neural Network as one of the models to predict the graduation rates of students based on their background. After normalizing the data and preprocessing data, we started making a default mlp classifier. The mlp parameters we used initially were the SGD solver, a logistic activation function and a batch size of 100. A sample run of the mlp predictor yielded a 0.85124 accuracy with around 0.85 precision and a mean square error of around 0.14876. After the initial run we wanted to see if tuning the hyperparameters could help our accuracy and yield better more efficient results. We initially altered the batch size to experiment with the accuracies. We changed the batch size to 70 which slightly lowered the accuracy. We then experimented with changing different parameters. We initially changed the activation functions from logistic to the tanh h function and then we proceeded to change the solver from SGD to lbfgs. We trained and tested the model many times but the changing of the parameters still kept the accuracies around the same. In any simulation run the accuracy was around 85-90%, which we considered to be a relatively high accuracy rate.

## Support Vector Machine

Using sklearn's SVC function, we constructed multiple SVM models with two kernel options. We observed that the Linear kernel achieved the highest accuracy of 90%. This result suggests that a linear decision boundary was more effective in separating the graduation and dropout classes

	precision	recall	f1-score	support
0	0.93	0.82	0.87	277
1	0.89	0.96	0.93	449
accuracy			0.90	726
macro avg	0.91	0.89	0.90	726
weighted avg	0.91	0.90	0.90	726

Confusion Matrix:

```
[[226  51]
 [ 18 431]]
```

Figure 6: Testing result of SVM with linear kernel

in our dataset. And, this is the second-highest accuracy rate based on a comparison with other models. We think SVM is well suited to solve our data. Additionally, the RBF (Radial Basis Function) kernel obtained slightly lower accuracy, achieving an accuracy rate of 87%. Because the RBF kernel is more flexible and capable of capturing complex non-linear relationships in the data, it might easily cause overfitting in this dataset. Thus, the linear kernel has better performance, indicating its suitability for our prediction task.

## Decision Tree

	precision	recall	f1-score	support
0	0.78	0.83	0.81	277
1	0.89	0.86	0.87	449
accuracy			0.85	726
macro avg	0.84	0.84	0.84	726
weighted avg	0.85	0.85	0.85	726

Confusion Matrix:

```
[[230  47]
 [ 64 385]]
```

Figure 7: Testing result of the decision tree



Figure 8: The graph of the decision tree

For our dataset, which consists of both numerical and categorical attributes, we chose sklearn's decision tree algorithm. One advantage of decision trees is their ability to handle mixed data types without the need for one-hot encoding. This made decision trees a suitable choice for our prediction task. After training and evaluating the decision tree model, we got an accuracy of 85%. This accuracy score demonstrates the model's ability to correctly classify instances into their respective classes, thereby providing valuable insights into predicting graduation and dropout outcomes. However, by comparing with other accuracy, it's relatively low. So, it might not be an ideal model for this dataset.

## Naive Bayes for Categorical and Numerical Attributes

The Naive Bayes Classifier model was used for categorical and numerical attributes. When splitting the dataset, changing test size and random state values allowed accuracy numbers to change. However, accuracy numbers did not change significantly. In addition, in order to keep the Naive Bayes model consistent with previous models, test size and random state values were kept at 0.2 and 42 respectively.

CLASSIFICATION REPORT:

	precision	recall	f1-score	support
0.0	0.69	0.64	0.66	277
1.0	0.79	0.82	0.80	449
accuracy			0.75	726
macro avg	0.74	0.73	0.73	726
weighted avg	0.75	0.75	0.75	726

Figure 9: Naive Bayes for categorical attributes classification report

Dropping or keeping categorical attributes did not seem to have any effect on the accuracy. However, to keep it consistent with the other models, the categorical attributes dropped were: Marital status, Nationality, Educational special needs, and International. As can be seen from Figure 9, accuracy is 0.75 or 75% which is lower when compared to previous models.

CLASSIFICATION REPORT:

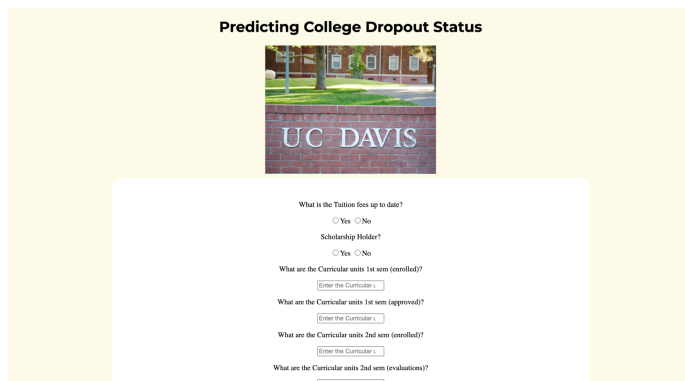
	precision	recall	f1-score	support
0.0	0.82	0.76	0.79	277
1.0	0.86	0.90	0.88	449
accuracy			0.84	726
macro avg	0.84	0.83	0.83	726
weighted avg	0.84	0.84	0.84	726

Figure 10: Naive Bayes for numerical attributes classification report

All numerical attributes were kept. As can be seen from Figure 10, accuracy is 0.84 or 84%.

Overall, Naive Bayes for categorical attributes had an accuracy of 75% while the numerical attribute model had an accuracy of 84%. Hence, for our dataset, Naive Bayes seems to work better for numerical attributes. However, it should be noted that the Naive Bayes model had lower accuracy when compared to our other models. As a result, Naive Bayes was not ideal for our dataset.

## Software Implementation



The image shows a web application titled "Predicting College Dropout Status". It features a header with a photo of a UC Davis building. Below the header is a form with several questions and input fields:

- What is the Tuition fees up to date?
  - ☐ Yes ☐ No
- Scholarship Holder?
  - ☐ Yes ☐ No
- What are the Curricular units 1st sem (enrolled)?
- What are the Curricular units 1st sem (approved)?
- What are the Curricular units 2nd sem (enrolled)?
- What are the Curricular units 2nd sem (evaluations)?

Figure 11: Frontend Interface



To display our results on a frontend, we used HTML, CSS, and Python’s Flask for the backend. The HTML contained a form, asking users to input eight features identified as significant in making predictions during our exploratory data analysis. These eight features are shown in order in the chart below. The model that we chose to use for our frontend was the logistic regression with k-fold cross validation since it had the highest accuracy among all our models and thus the best one to make predictions.

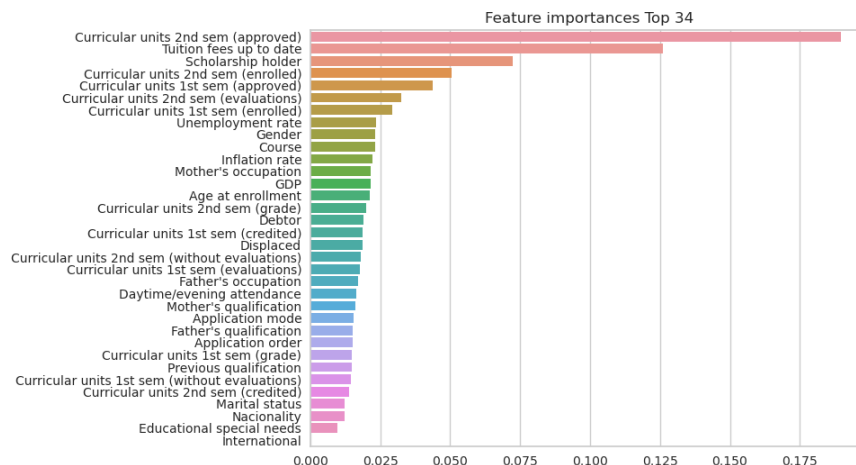


Figure 12: Eight Important Features

After the form is filled out, clicking the submit button sends a post request to the backend, where the user data is pre-processed to match the format of the training data for the model. The user data was one-hot encoded using the same processes used in the training data. It was also transformed to contain the same columns as the training data. Then, it was fed into the model, which was saved in a .pkl file using pickle, to make the prediction. The prediction was then mapped to the correct prediction label and sent back to the frontend to display the results, either graduate or dropout, in a different page, results.html.

## Conclusion and Discussion

Through using our dataset from ‘Predicting Student Dropout and Academic Success’, which contained 4424 student records and 35 features before data preprocessing and one-hot encoding, and building six different models (Polynomial Regression with Sigmoid, Logistic Regression, Neural Network, SVM, Decision Tree, and Naive Bayes), we were able to construct a model with 91% accuracy using Logistic Regression with K-Fold Cross Validation.

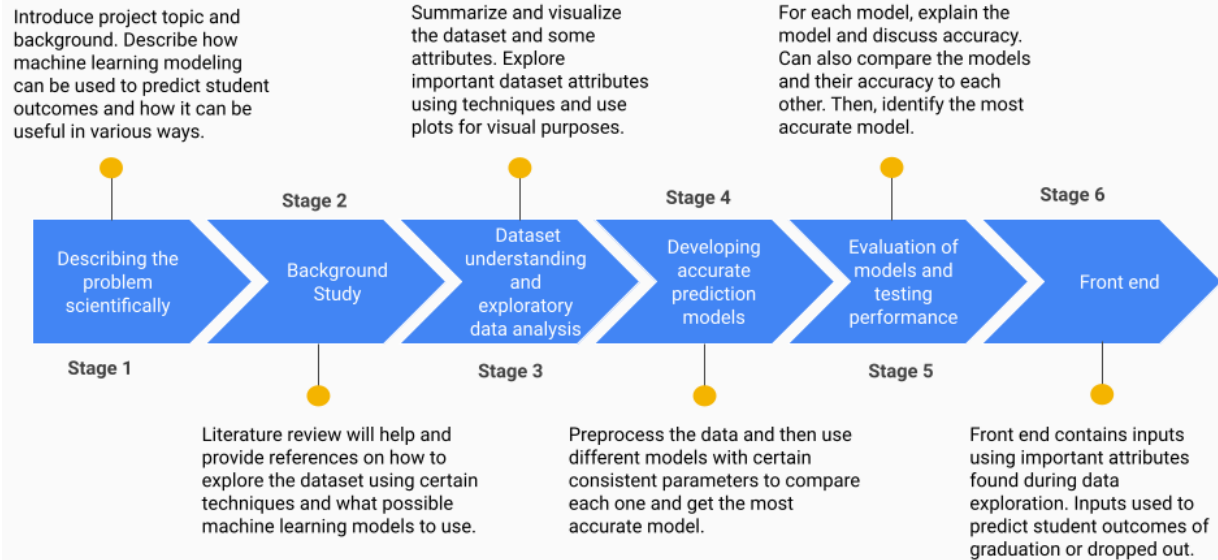
We could have further improved the accuracy of our model by also adding grid search to tune the hyperparameters of our model. However, this method could be computationally expensive and take a long time to run with a dataset of our size. It would still be worth to explore the hyperparameters that lead to the highest accuracy.

Our software implementation was done with HTML, CSS, and Flask, providing a simple frontend interface for users to input data and explore predictions. The interface displays a form that asks users questions to obtain the features and sends the data collected through a POST request. Since our logistic regression model had the best accuracy, it was the model used to make predictions in the frontend. We could further improve the frontend by adding more models and resources for users to explore.

# Roadmap

## Predicting Student Dropout and Academic Success

### Group 19 Project Roadmap



## References

### Dataset:

<https://www.kaggle.com/datasets/thedevastator/higher-education-predictors-of-student-retention>.

### Literature Review:

Mduma N. Data Balancing Techniques for Predicting Student Dropout Using Machine Learning. Data. 2023; 8(3):49. <https://doi.org/10.3390/data8030049>.

Realinho V, Machado J, Baptista L, Martins MV. Predicting Student Dropout and Academic Success. Data. 2022; 7(11):146. <https://doi.org/10.3390/data7110146>.

### Other Sources:

<https://betterdatascience.com/feature-importance-python/>.