# ## STAT 4185 Final Project Analysis

## Purpose

This project seeks to create predictive models that identify whether a given student falls into the category “Dropout”, “Graduate”, or “Enrolled” based on cross validation with a training and test dataset.

Process:

- Data Collection: load dataset

- Data Preprocessing and Cleaning: Splitting with Pandas, scaling

- Data Visualizations with matplotlib and seaborn (will come back to this)

- Model Construction and Performance Evaluation: decision tree/random forest

## Data Collection

I went about seeking datasets on Kaggle because it serves as a hub of repositories containing various Jupyter notebooks and projects, and stumbled upon a study based on data from a Portuguese educational institution. This dataset consists of student information such as their course (major, area of study), how they applied to the school, their marital status, and more. Other attributes listed included socioeconomic conditions such as GDP, unemployment rate, and inflation rate.

This data consisted of numeric data (either quantitative or qualitative but numerically encoded per Appendix A in the original paper) and one Target column that specified if the student fell into the category “Dropout”, “Graduate”, or “Enrolled”. I noticed that in the original paper, many of the variables that were numerically encoded as integers were interpreted in such a way that implied a sense of ordinality and/or magnitude comparison, which is very misleading. For example, the course “Agronomy” associated with the integer encoding of 4 has no purpose to be valued more than course “Biofuel Production Technologies`” which is associated with the integer encoding of 1. In the preprocessing phase, I attempt to reencode these variables in a way that would not lead to misinterpretation.

## Data Preprocessing and Cleaning

I primarily relied on the pandas Python package to clean the data. First, I checked the data types of each variable to verify that most variables were indeed quantitatively encoded, and examined for any values that needed to be imputed or replaced. I decided to use one-hot encoding for each value in most categorical variables so that each attribute is given equal weight, this was done using the pd.get\_dummies() method. The original paper used summary statistics with an “average” for these variables, which I found to be a direct misapplication. I dropped variables pertaining to student parent information since there were a ton of different encoded values because each value in a variable would have its own column; for the purpose of this assignment, I wanted to conserve computational power and work with a more manageable size data frame. The filtered and processed data frame has 104 columns.

After reencoding the data, I then split the dataset into training and test sets for cross validation and measuring predictive performance later on. I also scaled the quantitative columns using ColumnTransformer() from the sklearn package so that they are standardized for interpretability. The data was split 25% for the testing and 75% for model training, since there are a small number of observations.

## Data Visualization

My intention was to just graph attributes of the training data set because the model should be constructed based on that information, but ended up using the test and training set combined to get a holistic picture. The value\_counts() method was used to categorize based on the original integer-encoded values from the raw dataset. The matplotlib package was used to generate stacked bar graphs and box plots of various features of the dataset. Visually we can see that the proportion of each target group remains relatively the same across categorical variables, so we can guess that the predictive model may not perform with excellent accuracy.

Outlined in the visualizations are only some of the features described in the dataset; some features from the training data are not included above. If given more time and more data, we would only look at features of a training dataset to construct our models from. It appears that the proportion of dropout, graduated, and enrolled students appear to be relatively consistent across most categories; I am guessing that it will be relatively difficult to create a classification model given the lack of distinction between classification groups.

The distribution of first and second semester units look roughly the same, I expect them to be highly correlated. As mentioned earlier, in the original study, regression and correlation analysis was done on label-encoded qualitative variables as if they were quantitative, so I had changed them to one-hot encoding for more interpretable results.

## Model Construction and Performance Evaluation

Finally, I used decision tree and random forest algorithms to cross validate between training data and test/holdout data, and assessed overall prediction performance. The decision tree model was able to correctly identify true positives and true negatives about 66% of the time, whereas the random model was able to do so 76% of the time. This indicates that these models perform moderately well in identifying the status of student enrollment based on the provided training and test data.

## Conclusion

Overall, this was an interesting process where I got to interact with each step of the data science pipeline. Different prediction models were constructed to predict student status with adequate accuracy based on preprocessed data.

Because I use Microsoft Edge as my main browser, I was not able to use a Chrome CSS selector extension for webscraping, so I opted to find a csv file from an online database. There are numerous ways to perform data collection, but if data scientists have a goal in mind, they should ensure that the method by which they collect their data is in line with their objectives.

I struggled most with preprocessing and getting the certain function packages to work, especially with encoding, preprocessing, and scaling. Luckily, most would agree that this is the hardest part of the data science pipeline. Taking initiative with similar projects like these will help me become more fluent with packages like Pandas and using these tools in my data science career.

Works Cited

GeeksforGeeks. “Create a Stacked Bar Plot in Matplotlib.” *GeeksforGeeks*, 24 Aug. 2022, www.geeksforgeeks.org/create-a-stacked-bar-plot-in-matplotlib.

“Matplotlib Different Size Subplots.” *Stack Overflow*, stackoverflow.com/questions/10388462/matplotlib-different-size-subplots.

“One-Hot Encoding of Label Not Needed?” *Stack Overflow*, stackoverflow.com/questions/62911538/one-hot-encoding-of-label-not-needed.

“Predict Students’ Dropout and Academic Success.” *Kaggle*, 3 Jan. 2023, www.kaggle.com/datasets/thedevastator/higher-education-predictors-of-student-retention/data.

Realinho, Valentim, et al. “Predicting Student Dropout and Academic Success.” *Data*, vol. 7, no. 11, Oct. 2022, p. 146. https://doi.org/10.3390/data7110146.

“Rotate Axis Tick Labels.” *Stack Overflow*, stackoverflow.com/questions/10998621/rotate-axis-tick-labels.