Project 3
Topic: **Classifying Whether A Client Will Default On Credit Card Debt**

Issuing credit to a client comes with its risks. Research shows that for the typical bank, credit default rates range from 1-3%. This fraction may appear insignificant, but when issuing hundreds of millions, or even billions of dollars, defaulting clients can result in a substantial loss of money. Ideally, you would like to mitigate such loss and identify which clients are likely to default. This project aims to binarily classify a sample of 30,000 clients as either defaulting or not by implementing various supervised learning classification algorithms. The end goal is to train an adequate model such that it can be applied for future situations.

**Data**

The first step was to acquire relevant client data pertaining to their credit history. Information was pulled from UCI's Machine Learning Repository where a nice dataset is available. The data was sampled from April to September 2005 in Taiwan. It is important to note that although the data was sampled from clients in Taiwan, the type of data gathered is exactly the same as that of information gathered in the United States. Where there might be concern for variability between the relationship of the type of data collected and the model's results, with the fundamental level of modeling done one can assume variation to be insignificant. 30,000 clients were sampled and the features (type of information) include age, education level, marital status, bill amount for each month, amount paid in a month, payment status, and default state in the following month of October. In general, there are a total 23 features, most related to fiscal data.

During 2005 Taiwan was in economic downturn, so a higher than usual default rate is expected. Nevertheless, 22% of the sample had defaulted. This indicates that the data is highly unbalanced, and only a small proportion falls into one class. The labels for the target variable are 0 for ok, and 1 for defaulting.

**Tools**

Python and relevant libraries such as sci-kit learn, pandas, and matplotlib were used. In addition, given the large size of data and processing time of each algorithm, AWS EC2 was used to expedite the computational process. Lastly, GitHub was used for version control of files.

**Modeling**

Given a variety of supervised learning classification algorithms, a baseline model was generated for each of the following algorithms: KNN, SVM, Decision Tree, Random Forest, Gradient Boosting. After comparing each model's respective ROC curve and AUC scores, *Gradient Boosting* was determined to be the significantly better performer. Subsequent modeling thus was based only on this algorithm.

In training the model, *Cross Validation* was used along with a *GridSearchCV* in order to sample through several tuning parameters to determine the most optimal ones. Parameters of interest are: max_depth, n_estimators, and max_features. Evaluation of the model's performance was judged on several metrics. In this case, *recall* was the primary metric of interest. As a reminder, default rates are relatively low. As a result, it is more important to be able to capture most of the sub-sample of clients that will default than it is to capture a few clients at a higher precision. With this methodology, it initially minimizes the amount of lost money, but also it ensures that all our model requires further is perhaps more observations or an additional feature to increase precision. Analogously, it is better to know that the police has put all criminals in jail at the cost of wrongly convicting a few people than it is to only catch one criminal, but let loose the rest.

After optimizing said parameters, further improvements to the model was made by setting a probability threshold for classification. For instance, when predicting classification based on a test set, python returns a

probability that a client is either 0 or 1. By setting a minimum threshold, above 0.8 will be defaulting for example, the model could aim to capture more of missed defaulters, and thus increase recall.

**Conclusions**

The max AUC score of the tuned gradient boosting model was 0.83. This is a notable + 0.05 increase from 0.78 from the baseline model. In addition, precision was 0.72 and recall increased steadily to 0.4.

Take note that although our model does not result in a perfect recall, I believe that 0.4 recall is an acceptable mark for the status of the model so far. The default proportion is 22%, so to capture close to half of a small proportion is still adequate.

It is important to mention that features available primarily dealt with fiscal data. To improve recall, additional features should be included. Job type, salary, residency, and family size are good features that can be included. Overall, this model shows that classification algorithms can be implemented to reduce risk and loss in financial practices such as issuing credit. By improving the model with new types of data, it shows promise that it can be applied to similar situations as well.