# Final Project

## Team Member - Peter Sanchez

## Project Topic - Build Model to predict whether a Loan will be charged off based on features available at loan origination.

This topic was selected because I am employed by a $2.2 billion financial institution in the area of credit risk management. In addition to the utilitarian purpose of the model, I am observing artificial intelligence emerging in lending. In my environment, non-learning models are in use today in predicting the probability of losses. We have just implemented a learning model in the consumer loan origination process. I desire a greater and deeper understanding of the logic behind a model's "decision making" and how and what it is learning.

At a minimum, the model will consider the features available at the time of loan origination. These features are obtained primarily from the loan applicant.

Data Exploration

The search for appropriate data was primarily driven by the project objective of building a model to predict whether a Loan will be charged off. Accomplishing my objective would require a dataset that included not only the features relevant to loan decisioning but also the determination of whether the loan had been charged-off. I determined that I could obtain actual historical loan data for loans originated during a selected period which included an array of features assessed at loan origination. I also would be able to obtain historical loan data of loans charged off to the present day. I decided to limit the scope of my analysis to consumer loans (unsecured personal, automobile, and unsecured lines of credit). These types of loans account for the majority of my institution's charge-offs.

Based on my observations, the majority of loans that charge-off do so within the first 24 months after origination. I therefore created a dataset of loans originated from 11/01/2015 to 10/31/2019. I also obtained a trial balance of all loans charged off as of 10/31/2021. By joining these tables, I had a table of loan originations that had sufficient time (24 months) to charge-off. This resulted in a table of 22,508 loans, of which 1,175 had been charged off.

Other Pre-Processing

Pre-processing included the following steps.

These steps were accomplished using Excel.

1. Removed personal information (name, account numbers). A KEYID had already been created to accomplish the join in PostgreSQL.

2. Because I used a FULL OUTER JOIN, I had Charge-offs that had no match to the Loan Origination set. These rows were deleted. This resulted in a table of 22,508 loans, of which 1,175 had been charged off.

These steps were accomplished in Jupyter Notebook.

3. Removed features that were objects. Fortunately, all features were available in both numeric and string form so no relevant features were lost.

4. Clean data where data was inconsistent or blank. In most cases, I was able to correct or replace that data. However, 5,865 loans had blanks for Debt Ratio. I believed that Debt Ratio would be significant to prediction and therefore deleted all such loans. This resulted in a table of 16,643 loans, of which 731 had been charged off.

5. Binned certain features with low value counts.

Machine Learning Model

Due to the desire to view the ranked importance of features, I decided to utilize Balanced Random Forest Classifier model. The performance of this model was second to Easy Ensemble AdaBoost Classifier in the Module 17 Challenge, and as mentioned, offered the feature\_importances\_ function. Re-binning may also help.

Initial Results

The model's initial performance was disappointing. Its balanced accuracy score was 0.65389. More disappointing was its recall score of 0.59. Based on its confusion matrix, the model tended to predict a high number of false charge-offs. I am hopeful that my decision to include all of the features available in the data is the cause of noise that is leading to its lack of performance. I plan to use trial and error in eliminating less relevant features in hopes that the model's performance will improve.

Further Model Refinement

The Balanced Random Forest model was adjusted using various configurations (see Results\_Tracking). Among the changes tested were:

1. Changing bin cutoff of loanpurposecode

2. Changing n\_estimators at various intervals

3. Using various combinations of features based on feature\_importances\_

The best result was a configuration that yielded a 0.6584 balanced\_accuracy\_score. More importantly it produced what subjectively was the optimal balance with the highest number of "True Good" loans compared to "False Charge-offs" loans.

User Input Version

A simplified version was created that would accept user input. In this version, scaling was eliminated so that user could make inputs which were more familiar to them in the loan origination process. The input model functioned, and while it had the highest "False Good" count by 33, it also had the highest "True Good" count by 10. Some quick arithmetic shows that the interest earned on 33 loans is not enough to cover the loss of principle of 10 loans in event of charge-off, even if the loans paid for half their term.

Easy Ensemble AdaBoost Classifier

While the Balanced Random Forest model's performance was satisfactory, an attempt was made to use the Easy Ensemble AdaBoost Classifier. The features from the best Balanced Random Forest model were used. The results were expected to be better based on the comparative performance of these models in Challenge 17. Surprisingly, the Easy Ensemble AdaBoost Classifier model was outperformed by the Balanced Random Forest model.