Santander Customer Transaction Prediction
Kelly Field | Sy Lax | Derartu Dinku | Kishan Patel

1. Overview

The objective is to build a model that identifies which customers will make specific transactions in the future regardless of the amount of money transacted. The dataset contains 200 anonymized numeric feature variables that have the same structure of the company's real data that would be used to solve the problem, a binary target column, and an ID column.

2. Notebook Critique

The top-rated notebooks have similar structures and details that set them apart from the rest. They lay out their objectives and thought processes in a way that makes them not only easy to follow, but replicable. Oddly enough, the team that had the highest scoring model with an ROC AUC of 0.92 did not have a highly ranked notebook because it only contained their final code with very little comments within it.

The two most common problems in approaching this problem is feature selection and dealing with an imbalanced target column. The highly ranked notebooks extensively examined that data to see how the features interacted with each other and if some had larger impacts. The most common methods were to start off graphing their distributions and correlations. We also saw some unique visuals such as a permutation importance graph that ranked the features from biggest to smallest impact. Feature selection is an important aspect of this problem because there are 200 features and weighting them all the same may not produce the best outcome. The most common ways to handle feature selection were ridge and lasso, either eliminating features all together or weighting them to allow for the appropriate ones to have the biggest impact.

All of the top notebooks noted that the binary target column was unbalanced in the training set, specifically that there were more 0 instances than 1's. This imbalance can result in poor results because the model will have a hard time generalizing and it could be biased towards one response. The most common ways seen in the notebooks for handling this issue is with undersampling, oversampling, and SMOTE. Undersampling involves discarding observations in the majority class to keep underlying information of the sample but have a balanced target variable. Oversampling is essentially the opposite - you randomly sample observations in the minority class to balance the target variable, but don't skew the underlying characteristics of the data. Both ways can be done with random sampling, but the outcomes are better if they are done

systematically. Unlike oversampling and undersampling, SMOTE systematically creates new observations instead of making copies from the original data.

Several different methods were used in creating the final model across notebooks. The most common were Stratified K-Fold, modified Naïve Bayes, Random Forest Classifier, and Decision Tree Classifier. The notebook with the highest accuracy implemented a stratified K-Fold. This method varies slightly from the traditional k-fold because it takes an extra step to rearrange the data before splitting it into groups so that each group is a better representative of the whole.