HOME CREDIT DEFAULT PREDICTION MEMORANDUM

DATE: July 21, 2018
TO: Dr. Myles Gartland, Chief Data Officer
FROM: Homer Kay, Data Scientist
SUBJECT: Analysis and recommendation for predicting credit default.

Purpose

The purpose of this analysis is to predict a target variable of defaulting/not defaulting on a home loan based on a number of features. Exploratory data analysis, application of machine learning models, and performance evaluation are included in this analysis. For the purpose of this analysis the main performance evaluator used was the ROC Area Under the Curve.

Data Cleaning

Data cleaning included: creating dummy variables for various categorical features, scaling data for appropriate models, and reducing the number of variables to a manageable size utilizing random forest feature selection (importance). Initially I dropped 45 (of the original 122) features due to the fact they were missing 50% or more of their data, and then by using a Random Forest feature selection I was able to further reduce the number of variables to 31. Each feature was ranked relative to one another on an importance scale. Due to the fact there was a few noticeable cutoffs I decided to make a cutoff at accepting features no lower than .01 on the scale. I felt the appropriate amount of variance was still being captured in the analysis with the variables left while also reducing the chances of running into the curse of dimensionality or multicollinearity. There were a number of NaN's in a few variables leftover, I decided to replace them with 0's since many of the values were already 0.

Exploratory Data Analysis

Evaluating features independently revealed the following results. 2/3's of the lessee's are female, don't own a car, but do own realty. The majority of lessee's have Secondary or Higher Education. The majority are married and over 90% live in homes (as might be expected as this is a dataset of home loans). The majority of workers are in the labor force followed closely by sales staff in frequency. By choosing the target variable (default) at chance you would only have an 8% chance of being correct; thus the dataset is very unbalanced. The defaults through each of the basic categorical variables is fairly well distributed, not pointing to one particular set of traits as a leading cause.

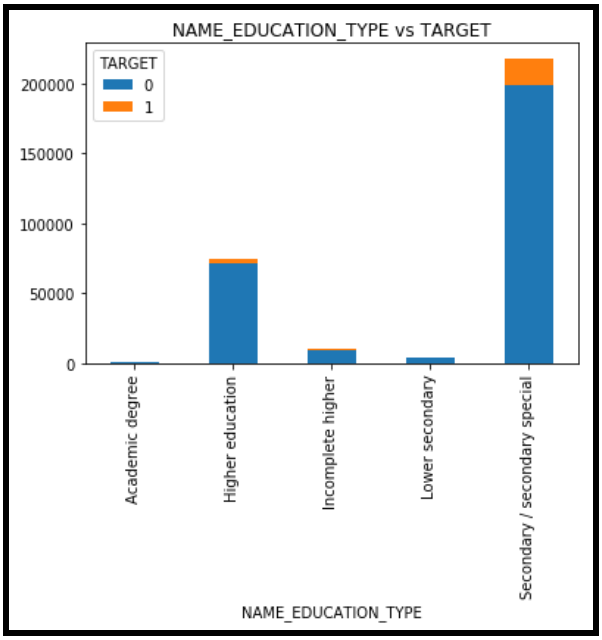

Figure 1: Education Type Frequency, broken down by default (orange), no default (blue).

I also found that the variables family members and number of children are 87% correlated, thus possibly pointing out a risk of multicollinearity, but this risk can be mitigated through cross validation during analysis.

Predictive Models

The training/test set was split 70/30; for the size of the data (300,000+ rows) this split is appropriate. On some occasions the data sets were normalized and/or reduced for model necessity or computational efficiency. The results of the best model from each type can be seen below.

Results

Model	ROC AUC	Accuracy	Precision	Recall	F1-Score	CV Stability
GBC	0.74	0.92	0.88	0.92	0.88	Stable
XGBOOST	0.74	0.92	0.92	1.00	0.96	Stable
Extra Trees	0.73	0.92	0.85	0.92	0.89	Stable
Random Forest	0.73	0.92	0.85	0.92	0.89	Stable
Adaboost	0.73	0.92	0.84	0.92	0.88	Stable
SVM Linear	0.72	0.68	0.89	0.68	0.75	Stable
ANN	0.72	0.92	0.92	1.00	0.96	Stable
Bagging	0.68	0.92	0.89	0.92	0.88	Stable
Decision Tree	0.66	0.79	0.88	0.79	0.82	UnStable
SGD	0.65	0.56	0.88	0.56	0.65	UnStable
KNN	0.57	0.92	0.87	0.92	0.88	Stable
Light GBM						

Models are ranked in descending order based on their ROC Area Under the Curve (Integral of Sensitivity as a function of (1-Specificity)). Boosting and bootstrapping models worked extremely well. In addition to the typical models I tried out XG Boost and got excellent results. I would have also like to try out Light GBM, but the version of Python installed and running were further up to date than the latest Light GBM files. I would like to predict a Light GBM once I can resolve this issue. To calculate the area under the curve, as well as the typical binary predictions, I also predicted probability targets, thus being able to create an ROC function.

I found tuning boosting and bootstrap models to give me increases anywhere from 5-10%. Basic models like the Decision Tree and KNN clearly lagged behind in performance. In bagging I used both KNN and Decision Tree base learners, but neither helped out immensely, overall bagging did not perform as well either.

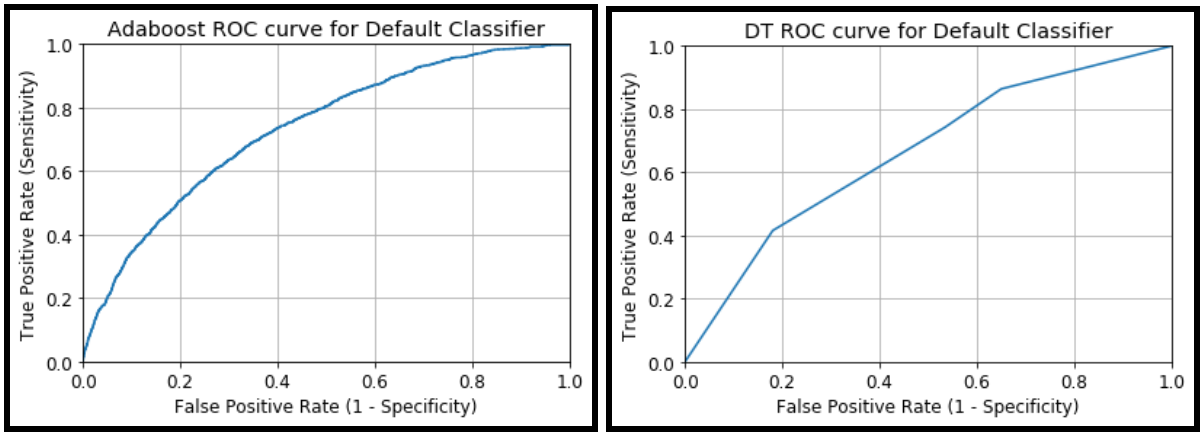

Figure 3: Visualizing the Differences in Adaboost (Left) & Decision Tree (Right) Prediction.

One more interesting note is the transformation of the ROC curve with the inclusion of more data.

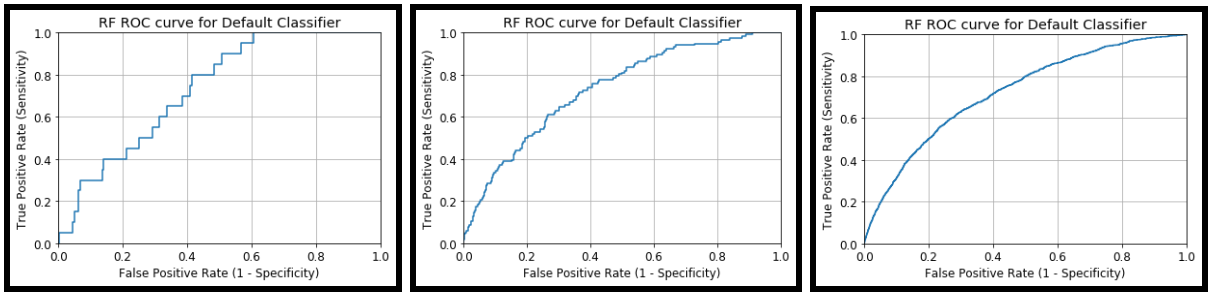

Figure 3: ROC Curves for Random Forest (Left = 1000 data points, Center = 5000, Right = Entire dataset)

Conclusion and Recommendations

From the results I can conclude that the predictions made from the best models in Gradient Boosting, XG Boost, and Extra Trees will produce the most desirable results for helping our clients receive the best and most accurate information when applying for loans. The evaluation criteria states only to maximize the ROC Area Under the Curve. Although there are other aspects one may want to look at. Since loan default risk is not a life threatening case (I would hope) it is not important to be overly conservative, however we may want to bias models based on the risk/reward cost tradeoff of missing a few defaults or potentially turning away to many clients because of high risk scores.

Competition

The most interesting part of this project has been the submissions on the Kaggle Competition. To begin with, just to understand the submission file type and process, I turned in a KNN, but the result was only 54% correct. However after the analysis was complete I chose a few of the best models to run a training on the entire dataset and testing it on the "unknown" target set given out by Kaggle. After turning in the submissions for Gradient Boosting and Adaboost my score dramatically improved. It was very satisfying to validate my results on a completely blind target dataset. As you can see the Score below for GBC and Adaboost are very close to the results above. Currently my rank online is in the 4100's. And the best models are currently pulling 80% scores, so considering I'm only 7% away doing my entire feature selection and modeling separately I find that to be pretty good. I would like to add in Light GBM though and also review what others are doing for feature selection and data munging. It seems most of the top ranked results are utilizing Light GBM.

Competition Results

Model	Score	Rank
GBC	0.730	4114
Adaboost	0.722	Lower
KNN	0.545	Lower

Reference

https://www.kaggle.com/c/home-credit-default-risk#description