PREDICTING CREDIT CARD DELINQUENCY

By: Ryan C. Dallavia

ABSTRACT

For credit card issuers, credit card delinquencies can lead to billions of dollars in losses each year. Balances that are “charged off” make offering competitive interest rates and other incentives to prospective and current borrowers challenging. This begs the question: Is there a way to predict delinquencies before they happen and perhaps minimize them? In the following pages, we endeavor to predict credit card delinquencies using data supplied by American Express. 10,000 observations and over 150 features were included in our data set. After reviewing several choices, we selected the light gradient boosting machine as the preferred supervised classification methodology to successfully predict credit card delinquency. Computational constraints and sample size were limiting factors, however, we believe our study provides a sound foundation for further research.

Problem Statement

The New York branch of the United States Federal Reserve maintains consumer banking statistics. As of the third quarter of 2022, the seasonally-adjusted delinquency rate at the top 100 largest banks was, on average, 1.91% (New York Federal Reserve) The amount seems trivial until one considers $925 billion in credit card debt was held by Americans during the same time period. The figures imply ~$17 million in delinquencies.

Per Investopedia, a delinquency arises when a borrower misses two consecutive payments, "after which a lender will report to the credit reporting agencies . . . that the [borrower's payment] is 60 days late." Uncured delinquencies may continue to be reported to the credit agencies for as long as 270 days. (Investopedia). After 270 days, a delinquency becomes a charge-off and, thus, a matter to be handled in accordance with the U.S. Bankruptcy Code.

The distinction between delinquencies and charge-offs is an important one. A delinquent account may be "trued up," whereas a charged-off account represents money lost. Logically, delinquencies must precede charge-offs. This provides us with the motivation for our study. Specifically, can potential delinquencies be predicted before they occur, thus reducing the number off both bankruptcies and charge-offs? We created a prototype to do just that.

Using customer account data from American Express, we trained a light gradient boosting model to classify accounts that have the attributes of a delinquent account from those that do not. Model accuracy, AUC, recall, and precision were .8871, .962, .9153, .7168. We optimized for recall, knowing that all accounts classified as true and false positive accounts would be checked for potential delinquency, whereas having an overabundance of false negatives would render the prototype generally useless, as negative cases typically go reviewed. Going forward, the issue is one of scale. With cluster access, the number of observations we can contemplate will increase significantly, as will the number of models we can train and potentially combine.

Data & Data Wrangling

The data were provided by American Express, as part of a Kaggle competition. Per Kaggle, "the target binary variable is calculated by observing 18 months performance window after the latest credit card statement . . . . if the customer does not pay [the amount due] 120 days after their latest statement date it is considered a default event." 10 million accounts were provided in a training set, along with the corresponding target variables, while another 10 million accounts were provided in a test set. Due to resource constraints, we randomly selected 10,000 training and test set observations.

The feature space contained 192 columns. Feature data were anonymized and normalized in advance. Though the vector names are meaningless, the features can be grouped using the prefixes that follow: D_* = *Delinquency variables, S_* = *Spend variables, P_** = Payment variables, B_* = *Balance variables, R_** = Risk variables.

826 of the accounts were unique, suggesting multiple transactions per account in a minority of cases. For customers with multiple observations in our data set, the most common number of transactions was 13. A class imbalance was detected with ~34% of observations falling into the positive, delinquent, class.

To narrow our feature space, we took the average number of values missing per column (~15%) and removed all columns with less than, or equal to, 15% of their values missing. This approach to thresholding is a commonplace approach to winnowing useless features from a data set. Columns dropped from the training set were similarly dropped from the training set.

Feature Engineering

A date column was included in the data set. From this column, we extracted month and day data. Vectors containing continuous and categorical data were missing values. To remedy the situation, mean column values were inserted into empty slots in the continuous variable vectors, while the most frequently observed categorical data served to fill in missing categorical information. A binary encoder was then used to prep all non-numerical data for model training.

Model Training & Testing

We relied on PyCaret for model training and testing. We separated training and test data using the typical 70/30 split. Though we were prepared to use the test set provided by Kaggle, PyCaret splits one off from the training set, thus, we had two test sets at our disposal.

Given the sizable feature space, we relied on PyCaret to correct for multicollinearity (threshold = 70%), as well as use the SMOTE approach to correcting class imbalances. We created three models using PyCaret's k-fold cross validation (k=3): 'light gradient boosting machine', 'gradient boosting machine', and a 'dummy' model to provide a basis for comparison.

The training process indicated the light gradient boosting machine was the clear winner across all metrics. The results are presented below.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
lightgbm	Light Gradient Boosting Machine	0.9400	0.9828	0.9047	0.8646	0.8841	0.8437	0.8441
gbc	Gradient Boosting Classifier	0.9065	0.9649	0.8796	0.7793	0.8264	0.7628	0.7654
dummy	Dummy Classifier	0.7473	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000

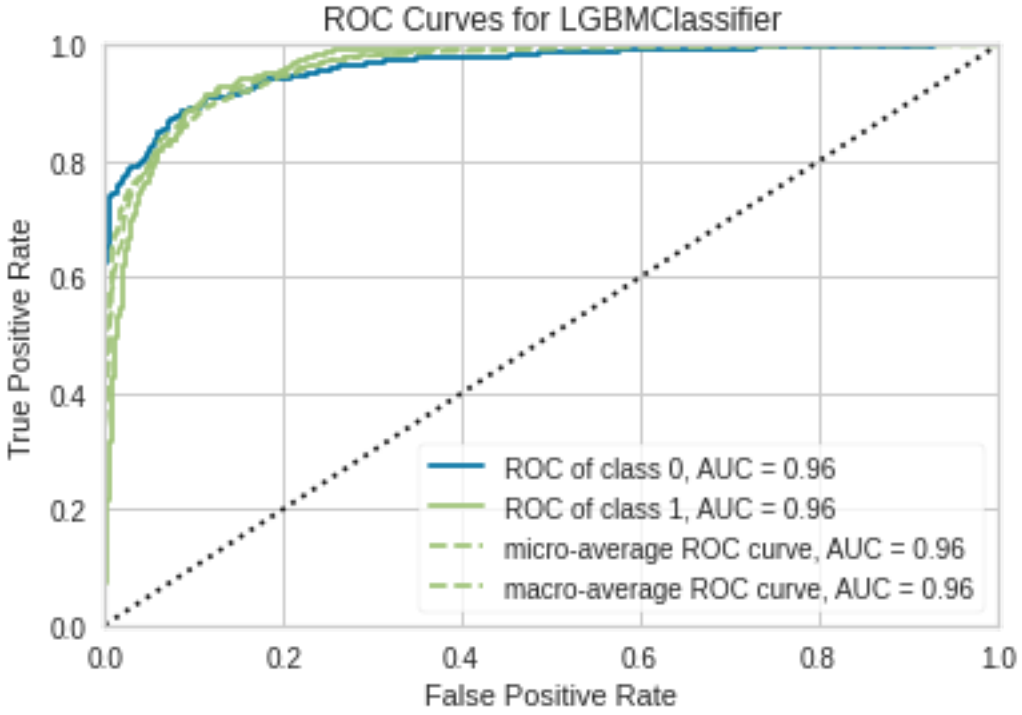
After selecting the light gradient boosting machine as our model of choice, we cross-validated again at k=5. Three had been used initially for the purpose of efficiency and resource constraints. The results of the second cross validation were just as promising as the first (see below).

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9367	0.9810	0.8992	0.8577	0.8780	0.8353	0.8357
1	0.9408	0.9852	0.8790	0.8862	0.8826	0.8430	0.8430
2	0.9612	0.9919	0.9556	0.8977	0.9258	0.8996	0.9004
3	0.9469	0.9879	0.9190	0.8764	0.8972	0.8615	0.8619
4	0.9449	0.9849	0.9190	0.8697	0.8937	0.8565	0.8571
Mean	0.9461	0.9862	0.9144	0.8776	0.8955	0.8592	0.8596
Std	0.0083	0.0036	0.0254	0.0137	0.0167	0.0223	0.0224

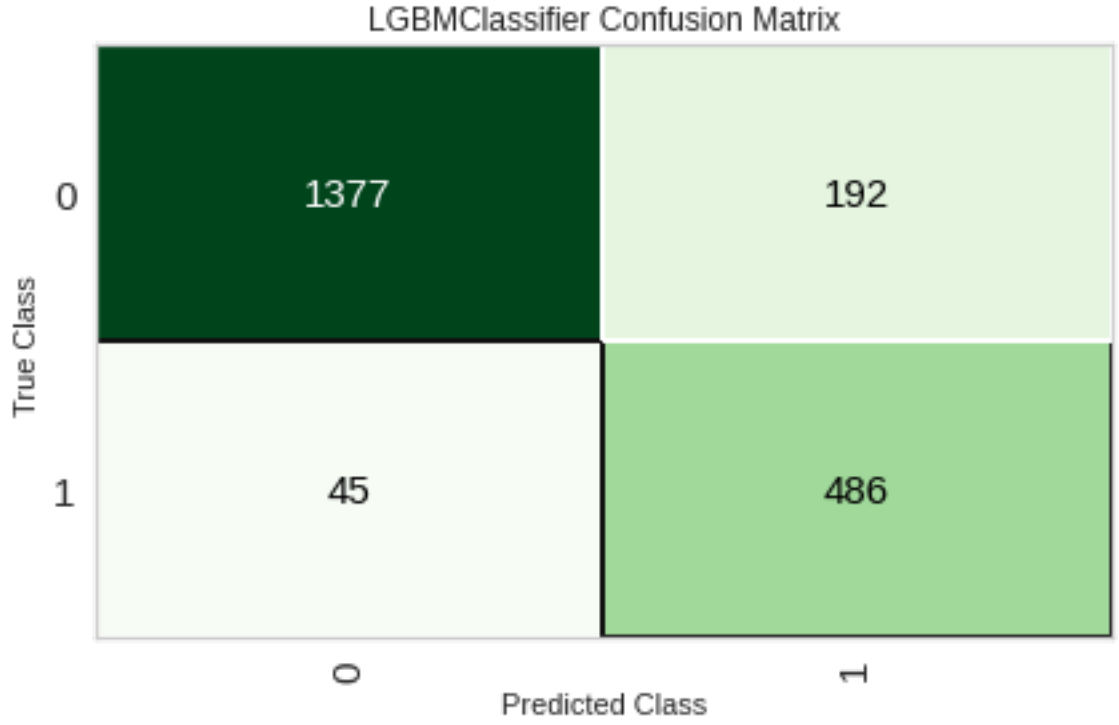
We further tuned our model to optimize for recall at the expense of other metrics. Naturally, some numbers declined, as seen on the following page.

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.8765	0.9544	0.8952	0.7003	0.7858	0.7009	0.7113
1	0.8959	0.9582	0.8831	0.7500	0.8111	0.7399	0.7446
2	0.9112	0.9770	0.9677	0.7524	0.8466	0.7855	0.7978
3	0.8990	0.9665	0.9555	0.7284	0.8266	0.7572	0.7710
4	0.9031	0.9586	0.9312	0.7468	0.8288	0.7623	0.7714
Mean	0.8971	0.9629	0.9265	0.7356	0.8198	0.7492	0.7592
Std	0.0115	0.0081	0.0330	0.0195	0.0204	0.0282	0.0293

As one can easily see, the AUC for all options remained relatively high.



The confusion matrix was impressive, with most cases classified correctly and with an emphasis on positive observations (upper right box) versus false negatives (lower left box).



Performance on separate testing data was similarly promising, as shown in the output below.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Light Gradient Boosting Machine	0.8871	0.9620	0.9153	0.7168	0.8040	0.7264	0.7371

Conclusion

We believe the positive output generated by the application of our light gradient boosting machine to customer data suggests this is the path forward, if we truly wish to model consumer credit card delinquency. Clearly, there were limitations to this study. The sample size consisted of a mere 10,000 transactions. Multiple models were not stacked due to space constraints. Only data from American Express was considered. A more robust model would include data from other providers. That said, gradient boosting seems to be particularly well suited for the task at hand. Thus we consider this an initial brick to use in building out our future analysis.