Default Prediction: Predict if a customer will default in the future

An Lee, Nini Lin
New York University Predictive Analytics

Abstract

"The global credit card payments market revenue was valued at $138.43 billion in 2020 and is projected to reach $263.47 billion by 2028, growing at a CAGR of 8.5% from 2021 to 2028. "[1]

In the financial industry, credit cards are essentially important. Over the past few years, credit cards have been among the banks' most successful financial services. However, banks have been dealing with an increasing credit card default rate due to the rising credit card usage.

Therefore, data analytics can offer solutions for managing credit risks and addressing the current issue. In this report, we implemented Random Forest, LightGBM, and XGBoost to evaluate the variable in forecasting credit default; LightGBM and XGBoost have higher accuracy and area under the curve.

Keywords

Default prediction, modeling, analytics, Random Forest, LightGBM, XGBoost

Introduction

Big data is increasingly a part of our daily lives, and the financial sector is also aggressively utilizing this technology. Banks gather information about their consumers to change their monitoring from reactive to proactive.

The main goal of data-driven credit default monitoring is to make it possible to check on each customer continuously. The mission seems incredibly difficult to accomplish using the traditional standard credit rating process.

However, it is possible to carry out tasks in real-time by routinely utilizing big-data processing technologies. With the help of big data and machine learning, it is possible to take preventative action as soon as the client's situation changes.

In other words, regular evaluation of client credibility gives banks the ability to spot potential future problems and act as soon as possible, giving them additional alternatives for fixing the problem and preventing losses.

Automatic data-driven assessments that track client behavior serve as what we refer to as an early warning system. As a result, banks identify consumers with financial difficulties earlier rather than later, noticing problems before they arise and the customer defaults.

I. Business Understanding

An account will become delinquent before it enters default. After 30 days of late payments, this occurs. The credit card becomes substantially past due when six consecutive months go by without making at least the minimum payment required, at which point the default usually occurs. Since the constant risk of default is on the rise as a result of the expanding usage, the predictions that can reflect the likelihood of default are essential for credit card issuers.

II. Data Understanding

Collect and Describe the Data

The source used to establish the default prediction model is the user data from the American Express credit card service company. The data is desensitized, aggregated for each customer at each statement date, and does not contain the user’s identity information. Features are anonymized, normalized, and fall into general categories.

Furthermore, the data is imbalanced with the majority of paid data (negative class) and the minority of

default data (positive class), so the negative class (paid) has been subsampled for the dataset at 5%.

Explore the Data

In the dataset, aggregated features include general categories: Delinquency, Spend, Payment, Balance, and Risk. All features included in the above categories are all at each statement date. In addition to numerical features, there are eleven features being categorical: B_30, B_38, D_114, D_116, D_117, D_120, D_126, D_63, D_64, D_66, D_68.

The data source has two parts: training data and training label. For the first part, the size of training data is 16.39GB in CSV form. The training data includes data with multiple statement dates per customer_ID. Besides, the data has 190 features, 5,531,451 data rows, and 458,913 distinct customers, ranging from March 2017 to March 2018.

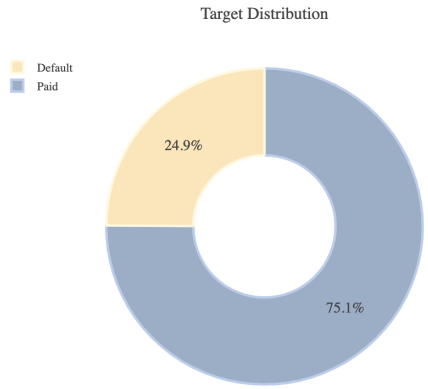
As for the training label, the size of label data is 30.75MB in CSV form. The data set includes the target label for each customer_ID.

Verify Data Quality

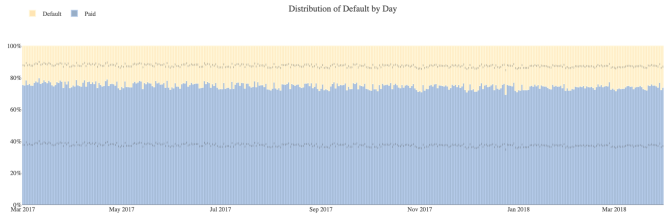
The purpose of predictive analysis is to find insights and patterns and let the company make informed decisions. However, if the initial step of collecting data cannot ensure the accuracy of the following steps, the analyst cannot trust the data to do any further analysis. Because data accuracy is a key attribute of high-quality data, a single inaccurate data point can wreak havoc across the entire system.

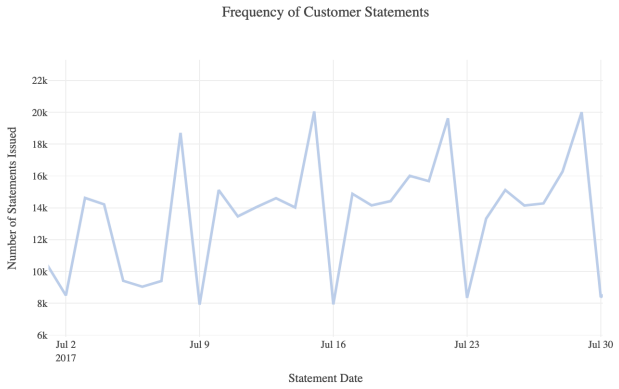
To avoid the circumstances, we first check such characteristics as missing values, blank fields, uniqueness, and the time that data was collected across.

We start by reading the label for the training data. There are neither missing values nor duplicated customer_IDs. Of the dataset, 75% of the data has the label of negative class which means the customer paid the bill and 25% of the data has the label of positive class which means the customer defaulted. Therefore, this is imbalanced data which is a challenge to handle.

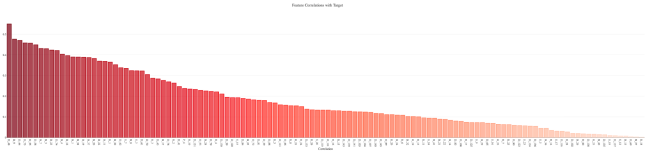


Besides, the distribution of target data is consistent across each day in the training dataset with a weekly seasonal trend in the day of the month when customers received their statements. S_2, one of the features, is the statement date and time. All statement dates in the training dataset are between Match 2017 and March 2018, a total of thirteen months, and no statement dates are missing.

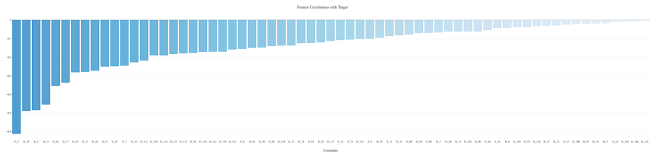




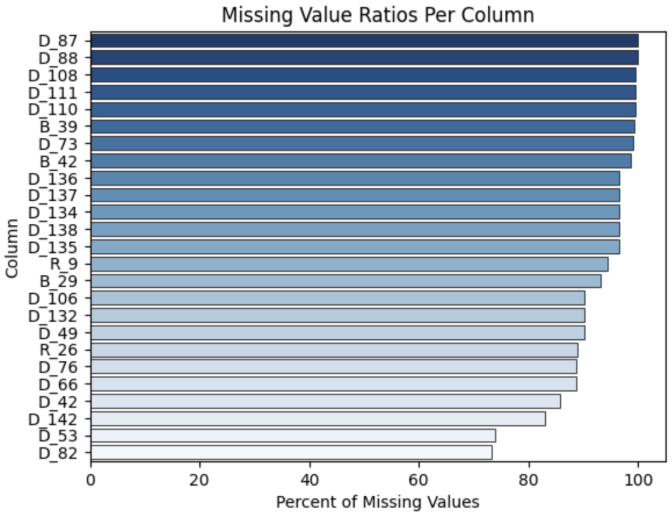
Also, the D_48 feature of the delinquency category is the most positively correlated with the target label at a probability of 0.61.



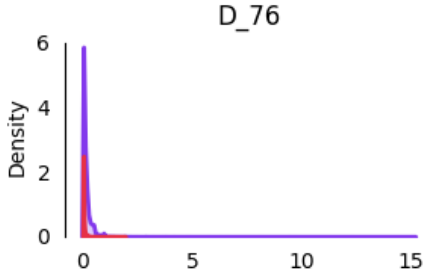
And the P_2 feature of the payment category is the most negatively correlated with the target label with the defaulting probability of -0.67.



Besides, we explore the remaining features. First thing first, we check the proportion of missing values in the features. We found that there are lots of features that include missing values and we list the top 25 columns with the highest percent of missing values.

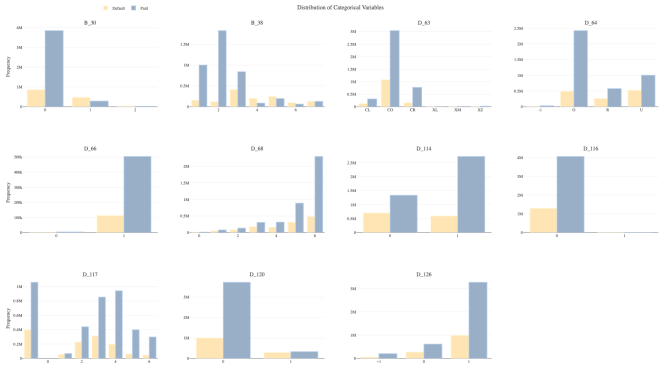


Then, we dive into the distribution of other features, including numerical and categorical data (See Figures 1 to 5 in Appendix A folder). For numerical features, histograms with white space at the left or right end may indicate that data contain outliers. Yet, those data could be characteristics of rare events or default data.



For categorical features, every feature has at most eight categories, including NaN value category.

Therefore, we would use on-hot encoding to convert categorical features after. As shown in the following figures, we can see the target distribution differs in each feature, meaning every feature gives some information about the target.



Among correlations between the target label and other features, there are several highly positively correlated pairs of features but also negatively correlated ones (See Figures 1 to 5 in Appendix B folder). We can tell in the following correlation heatmap that the D_87 feature in the delinquency category is missing due to the large proportion of missing values.

To prepare final data for modeling, based on the above explorations, we know there are some issues to deal with: imbalanced data, a large number of missing values, and big data size. And in the data preparation section, we will talk about how we reduce the data size; in the modeling and evaluation sections, we will continue talking about how we handle the missing values and imbalanced data.

III. Data Preparation

In the training dataset, it is already down-sample the negative class label, which is paid customers, at 5%, so we do not resample the data section again. In addition, the data source is relatively clean and does not need further cleaning, so our focus is on reducing the data size because of the memory and disk efficiency.

Integrate and Format Data

To access the features and the corresponding target labels easier, we merge the target class with the training data by distinct customer_IDs. And we start by reducing the data type to a smaller size. In detail, we convert the feature customer_ID from type object to type int64; the feature S_2 from type object to

type datetime, which is only 4 bytes; the numerical features from float64 to float32, which is 4 bytes per row; and the categorical features from object to int8 because they all have up to eight categories.

After reducing the size of data types, we store our data in parquet format files to save memory and disk space. Parquet format is a column-based format: files are organized by column, rather than by row, saving storage space, speeding up analytics queries, and providing efficient data compression to handle complex data in bulk.

Eventually, compared with the original size of 16.39 gigabytes, the size of the training dataset in parquet format is 4.79 gigabytes, making reading the data more efficient.

IV. Modeling

Select the Modeling Technique

Consequently, we come up with some solutions to fix the missing value. Nevertheless, should we fix it? There are many columns with missing values, not only the above-listed 25 columns. Hence, dropping all columns with missing values is not a sensible strategy. On the other hand, it is also not rational to drop all rows with missing values instead because it might lead to a problem of lack of representative sample.

For the above reasons, we choose to use decision-tree-based algorithms because they can handle missing values in nature. Another popular and common-used modeling technique is Neural Network. Yet, we do not consider applying such a modeling method because Neural Network modeling cannot deal with missing values naturally. In other words, we would have to use the "Imputation" to replace missing values.

Here comes the problem, we do not know the exact meaning of each feature due to the feature aggregations. Therefore we cannot determine which value to substitute is the most representative without skewing the characteristics of the original value.

Build and Assess the Model

1. Random Forest Model

The random forest model is an extension of the bagging approach since it uses both bagging and feature randomness to generate an uncorrelated forest of decision trees. These decision nodes in the tree are questions, and they serve as a way to separate the data by functioning as decision nodes in the tree.[2][3] In other words, the random forests train each tree independently with a randomly selected sample of the data. Compared to a single decision tree, the model is less likely to overfit the training set according to its randomness, making it more robust.

In short, the random forests technique provides advantages for our prediction, including handling classification and regression issues, managing missing values by its classifier, and preserving the correctness of a substantial portion of the data.

In our experiment, setting "n_estimators" as 200 and the "max_depth" as *None* yields the best results for random forests. The variables cause the forest to grow 200 trees and the nodes in each tree to spread out until all leaves are pure or until all leaves contain fewer samples than are necessary to split an internal node.[4]

2. LightGBM

LightGBM belongs to the family of gradient boosting decision trees (GBDTs) and adopts two cutting-edge techniques: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). The GBDT improves a single weak model by merging it with several other weak models to produce a model that is more robust as a whole. GOSS allows LightGBM to train each tree using a small portion of the total dataset. EFB allows LightGBM to handle high-dimensional sparse features considerably more effectively.[5]

Unlike the random forest, the LightGBM gives us benefits with a faster speed and uses less memory when training data since it utilizes a decision tree technique based on a histogram. Additionally, LightGBM supports GPU-accelerated distributed training with reduced communication costs.

When "n estimators" is set to 50,000 (LightGBM contains 50,000 trees) and "extra trees" is set to True, LightGBM will only check one randomly selected threshold for each feature when analyzing node splits. This option yields the best results.

3. XGBoost Model

Although XGBoost is a member of the GBDT, it finds split points using a pre-sorted version of an

exact greedy algorithm, unlike LightGBM, which employs a histogram to speed up training while using less memory. The main goal of the development of XGBoost was to enhance the performance and computational speed of machine learning models. It is a scalable and highly accurate gradient boosting implementation that pushes the limits of processing power for the "boosted tree" algorithm.

As opposed to GBDT, which builds trees sequentially, XGBoost constructs them in parallel. Utilizing a level-wise approach, it assesses the quality of splits at each potential divide into the training set by scanning through gradient values and using these partial sums.[6]

Compared to the random forest, XGBoost provides a better solution for unbalanced datasets because when the model fails to predict the anomaly for the first time, it gives it more preferences and weightage in the following iterations, increasing its ability to predict the class with low participation.

Implementing XGBoost to grow 50,000 trees(Setting "n estimators" to 50,000) will produce significant results for our testing.

V. Evaluation

The datasets in domains like fraud detection and default prediction have one thing in common that is these positive cases are so rare that the datasets are often imbalanced. However, most machine learning algorithms do not work well with imbalanced data, so utilizing techniques such as evaluation metrics, resampling, and K-fold cross-validation is helpful to evaluate the generated modeling.

Model Evaluation Metric

As mentioned in the previous section, the negative class (paid) has been subsampled for this dataset at 5%, and thus adds a 20 times weighting to the down-sampled examples in the scoring metric.

The evaluation metric M, having a maximum value of 1.0, is the mean of two measures of rank ordering: Normalized Gini Coefficient, G, and default rate captured at 4%, D.

$$M = 0.5 * (G+D)$$

AUC (area under the curve), the relationship between true-positive rate and false-positive rate, is calculated using the metric G, the normalized Gini coefficient, and has a value between 0 and 1. Between -1 and 1, the normalized Gini coefficient falls. The score increases with area size.

The true-positive rate (recall) for a threshold set at 4% of the total (weighted) sample count is measured using the metric D, the default rate caught at 4%. It is always between 0 and 1. Higher intersection points result in better scores.

The metric M is the average of these two components. In other words, it concurrently seeks to maximize both the actual positive rate and the area under the curve (recall). [7]

Evaluate Results

The following table is the evaluation results of all three models:

Model	Evaluation Method	Evaluation Score
Random Forest	Baseline	0.7552
LightGBM	Baseline	0.7832
	K-fold	0.7866
	Stratified K-fold	0.7867
XGBoost	Baseline	0.7774
	K-fold	0.7867
	Stratified K-fold	0.7869

K-fold Cross-Validation

With so few positives relative to negatives, the training model will spend most of its time on negative examples and not learn enough from positive ones.

However, K-fold might work fine for data with a balanced class distribution, but when the distribution is severely skewed, one or more folds will likely have few or no examples from the minority class. That means that some or perhaps many of the model evaluations will be misleading, as the model need only predict the majority class correctly.

Thus, we also use Stratified K-fold Cross-Validation, which is similar to normal K-fold Cross-Validation but preserves the imbalanced class distribution in the dataset of each fold, to evaluate the predictive model. In each fold of the training process, we calculate the evaluation score to know the performance of each model in each fold. We also save the prediction of examples out of folds for each fold to learn the overall cross-validation evaluation score.

VI. Conclusion

In this default prediction project, we can say the following in summary:

For the prevailing forecast: Since customers' aggregated profile attributes at each statement date are included in the collection. Additionally, characteristics are standardized and anonymized, and we know that some aspects are more correlated with delinquency than others, such as P_2 and some of its features.

We have a lot of missing values in the delinquent feature, and we are unable to determine what these missing values mean. If we can offer the bank any advice, we will advise them to conduct some research to determine which features have missing values. If they do this and can identify the missing values, they may decide that the missing values are the result of rare occurrences or are otherwise meaningless.

For the model: Due to the previously discussed factors, the lightGBM achieves a better outcome. Which method should we implement when putting the k-fold or stratified k-fold into practice? It should depend on the situation. A stratified k-fold is not always a desirable thing.

Acknowledgments

We would like to show our gratitude to the Kaggle platform for making the notebooks available where we mainly developed our project and its community of data scientists and machine learning practitioners for sharing brilliant ideas. Also, we would like to thank American Express Credit Card Service for making the available data. Finally, we are immensely grateful to Prof. Bari and Assistant Christopher for the insightful comments and invaluable guidance this semester.

References

[1] Credit Card Payments Market By Card Type (General Purpose Credit Cards and Specialty & Other Credit Cards), Application (Food & Groceries, Health & Pharmacy, Restaurants & Bars, Consumer Electronics, Media & Entertainment, Travel & Tourism and Others), and Provider (Visa, MasterCard, and Others): Global Opportunity Analysis and Industry Forecast, 2021–2028: https://www.alliedmarketresearch.com/credit-card-payments-market-A11836#:~:text=The%20global%20credit%20card%20payments,8.5%25%20from%202021%20to%202028.

[2] IBM Cloud Learn Hub-What is Random Forest?: https://www.ibm.com/cloud/learn/random-forest

[3] Y.Sayjadah, I. Abaker Targio Hashem, F. Alotaibi, K. A. Kasmiran, “Credit Card Default Prediction using Machine Learning Techniques”, *IEEE(2018)*

[4] sklearn.ensemble.RandomForestClassifier: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[5] Microsoft Research-LightGBM: https://www.microsoft.com/en-us/research/project/lightgbm/

[6] NIVIDA-XGBoost: https://www.nvidia.com/en-us/glossary/data-science/xgboost/

[7] America Express-Default Prediction evaluation: https://www.kaggle.com/competitions/amex-default-prediction/overview/evaluation