Project Report
DS502: Statistical Methods for Data Science
Topic: Bank Transaction Prediction

Team Members
Ayush Avinash Shinde
Snehith Datla Verma
Sree Likhith Dasari

Abstract

To serve customers better, banks face many problems which deal with binary classifications like whether the customer is satisfied or not, whether the customer will be able to repay the loan or not, and whether the customer will make a particular purchase. One such problem is predicting a future transaction of a customer. This prediction of the customer is irrespective of the amount of money transacted. The models developed using different machine learning models will effectively solve this problem with great accuracy. This helps segregate customers who make transactions frequently and those who don't.

Introduction

Our goal at Bank Transaction Prediction is to promote the success of both individuals and companies. New ways to assist their customers in understanding their financial situation and determining which goods and services might be able to help them reach their financial objectives. Banks struggle with a number of issues that deal with binary classifications, such as whether or not the client is satisfied, whether or not the customer will be able to repay the loan, and whether or not the consumer will make a specific purchase. Predicting a customer's future transaction is one of these issues. This consumer prediction is independent of the value of the transaction. This problem will be effectively and accurately solved by the models created by various machine learning methods. This makes it easier to distinguish between clients who transact frequently and those who don't. To better understand techniques for storing massive volumes of data, a case study on Kaggle was looked at in the early research for this work.

For our project, we used the Santander Bank dataset from Kaggle and the Santander Bank owns the data that was used in this study. Here, the test and training datasets are kept apart. Over 200,000 rows are present in 200 features. Order of transactions are represented by rows, while each customer's transactions are represented by a Row. A positive value indicates a credit, and a negative value indicates a debit. We'll clean up the data, type it, and take care of the missing values. The project will include exploratory data analysis of the dataset. The dataset is trained with logistic regression, linear SVC, and LightGBM. After building the model, the performance of the test dataset is done which has the same attributes as the training dataset. As the test dataset is separate, we'll use the test_train_split function to create a validation set and use it for checking the accuracy score of each algorithm. In the end, we use the Confusion Matrix and ROC_AUC score to evaluate the model. Finally, we discovered that the LightGBM model would be more appropriate for the issue.

Motivation

This is a financial domain problem where we have to classify a customer if he/she makes a transaction or not. For classification problems in general, the data is almost always categorical. But in this problem, the only data we had is transaction data which is numerical. We felt performing classification tasks on numerical data is challenging and excited us to solve the problem. So, we took up this as a challenge but failed in the beginning while trying a few algorithms to solve the problem. In this project, we got a better understanding of different algorithms and how to tune the parameters to make them work better for our given problem. We gave our best to come up with a proper machine-learning model to tackle this problem.

Methods and Methodology

Logistic Regression

In this project, we used a logistic regression algorithm for classifying the customer into 2 categories. Although the name has regression in it, it works perfectly with classification problems. Logistic regression is a modified version of linear regression which effectively works on binary classification problems. The logistic/sigmoid function classifies the target variable based on probability.

In this problem, we first didn't use any solver to train the model. Because of this, we only got an accuracy score of 74% which is very poor. After referring to a few resources online, we used an ‘l-bfgs’ solver. This is an optimization algorithm which belongs to the family of quasi-newton family. With this solver, we achieved a training accuracy score of 91.51 %. Also, the True Positive & True Negative values in the confusion matrix are high in number. The AUC/ROC Curve is 0.622.

Linear Support Vector Classifier (Linear SVC)

In this project, we used a linear support vector classifier. If the hyperplane classifies the data set linearly, the algorithm is an SVC. If the algorithm separates the dataset using a non-linear approach, then we call it SVM. Support Vector Machine (SVM) is a deep learning algorithm that performs supervised learning for classifications or regressions of groups. It can also be used to solve the classification or regression problem statement. A linear support vector classifier (linear SVC) is an algorithm that attempts to find a hyperplane to maximise the distance between classified samples. With more possibilities for penalties and loss functions, linear SVC should scale better to large numbers of samples.

For transforming the data, in order to make mathematics possible, SVM uses something called the kernel function to systematically find SVC in higher dimensions. The kernel function only calculates the relationship between every pair of points; if they are in higher dimensions, they don't actually do the transformation. This trick is called the kernel trick.

Each feature or variable is scaled to unit variance after the mean is removed using the standard scaler. Additionally, this is carried out independently based on features. Moreover, columns are subject to the conventional scaler. If the mean is equal to zero and the data is scaled to unit variance, the distribution is conventional. When the character has a normal distribution, it is helpful. This keeps the feature's distribution shape unchanged. Additionally, it works poorly on features with outliers. By calculating the pertinent statistic on the samples in the training set, scaling is done individually for each feature. Then, for usage with later data using transform, the mean and standard deviation are saved.

Santander provided different training and testing datasets. In this approach, the model was trained on a training dataset and tested on a testing dataset. The test's accuracy score was 88.72%.

The confusion matrix can be seen above for the linear SVC model. This confusion matrix can tell us how many incorrect/true predictions were, and this information can be used to calculate false positives, true positives, etc. The performance metrics are illustrated in the graph above, which contrasts the true positive rate on the y-axis and the false positive rate on the x-axis. The Area under the curve (AUC-ROC) score for the linear SVC model was unsatisfactory, falling in here at 50.1%.

LightGBM

Gradient Boosting is a method during which weak learners continuously improve into strong learners. Eg: XGBoost, One of the most popular gradients boosting algorithms; one type of several Gradient Boosting Decision Trees (GBDT).

The following are the drawbacks of GBDT:

1. Trained in sequence by evaluating the residual errors of each iteration and improving the following one.
2. Computing the information gained across all instances and considering all possible split points – can be highly Time-Consuming in the case of Big Data.

One of the most innovative GBDT subtypes is LightGBM. It was created in 2016 by a group of Microsoft researchers. Simply said, Light GBM adds two innovative capabilities that are missing from XGBoost. They provide the function of aiding the algorithm with several variables and data examples.

- Gradient-based one-side sampling:

This sampling technique takes the gradient's magnitude (training error) into account. The situations when the mistake is still significant are retained. Before being included in the tree, cases with a slight mistake are randomly sampled. Each tree has to process less data as a consequence.

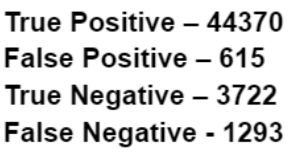
- Excessive Feature Bundling:

As implied by the name, this technique minimizes the number of attributes or variables. A significant portion of your dataset's features are frequently sparse (consisting primarily of zeros), especially if you use a lot of category variables. Many of these are concurrently mutually exclusive. This approach combines several remarkably comparable properties into a single feature.

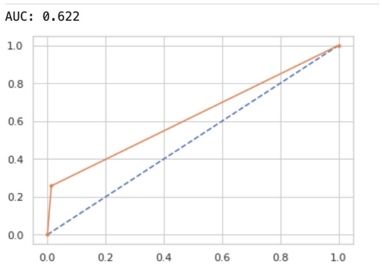
Results

The following shows the comparison between some of the performance metrics that we have computed:

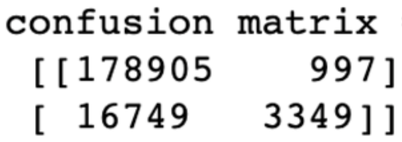
Linear Regression



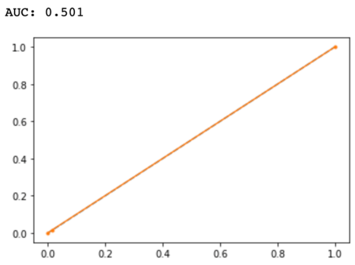
True Positive – 44370
False Positive – 615
True Negative – 3722
False Negative - 1293



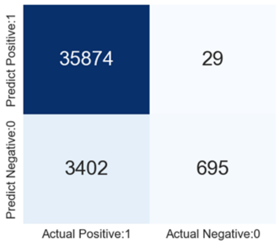
LinearSVC

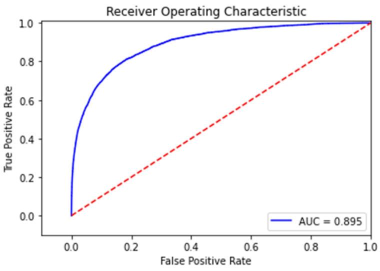


```
confusion matrix =
 [[178905    997]
 [ 16749   3349]]
```



LightGBM





AUC Score

	Linear Regression	LinearSVC	LightGBM
Training:	91.51	91.13	91.81
Testing:	91.33	88.72	91.42

Conclusion

In order to examine the data distribution and determine whether the dataset contains any null values, we implemented the EDA on the dataset. Three algorithms have been put into practice to forecast the desired variables. In accordance with the algorithm's requirements, we divided the dataset and utilized various pre-processing and optimization strategies. We excluded Linear SVC from the list of algorithms we implemented by comparing accuracy scores. We came to the conclusion that light GBM performs best with the aid of the AUC/ROC curve.