Finding Fraudulent Transactions

Financial Statistics & Risk Management

Jinwoo Lee, Kunal Jangam

Summary / Abstract

We are a team of data analysts aiming to figure out a way to detect fraudulent bank transactions so they can be dealt with preemptively. We use the features of the transactions in order to predict if the transaction is fraudulent or not.

We plan to do data preprocessing and exploratory data analysis. Then we will try methods such as Naive Bayes, KNN, Logistic Regression, and Random forests to see which methods fit our data the best. We will tune those methods and use the resulting models for prediction.

Problem

We live in a world where many services are becoming increasingly reliant on computers and digital technology. While this does make them much more convenient to use, it also opens the doors to new forms of fraud and theft.

While the number of bank robberies decreases slowly each year, credit card fraud incidents are on the rise. While banks do reimburse victims, it would be beneficial to stop fraudulent transactions before they occur instead of having to deal with the aftermath. In order to do so, we need to be able to detect fraudulent transactions first.

Data

The dataset is a simulation of bank transactions. A small amount of these transactions are fraudulent: that is, there are some agents within the simulation who aim to gain control of a customer's account, transfer all of the funds to a different account, and cash out. Our goal is to build models to predict fraudulent transactions. The response variable in this dataset is the binary variable isFraud.

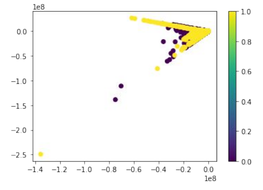
The dataset has 8 predictors to consider: the type of the transaction(cash in, cash out, debit, payment, or transfer), the amount of the transaction, the ID of the customer who started the transaction, their initial balance and new balance, the ID of the recipient, and their initial and new balance. Of these predictors, we dropped the IDs of the customers, as using those predictors would most likely lead to overfitting. In addition, we separated each of the 5 types into binary predictors, for a total of 10 predictors.

Methodology

We balanced the data by isFraud classes, ending with ~16000 rows.
First, we performed principal component analysis to visualize the data. We split the data into two components and graphed them against isFraud.

CASH_IN	CASH_OUT	DEBIT	PAYMENT	TRANSFER	amount	oldBalanceOrig	newBalanceOrig	oldBalanceDest	newBalanceDest
-0.27594004	0.03303204	0.01642419	0.22954011	-0.03569681	-0.37811953	-0.57130991	-0.48434842	-0.25643007	-0.31434715
0.15904145	-0.45268743	-0.00634558	0.13571149	0.27224326	0.02504121	0.22147076	0.26520767	-0.51397057	-0.54132101

From this result we can see that the regular transactions have a few more outliers than the fraudulent transactions, most likely because all of the fraudulent transactions were used while only a sample of regular transactions were used. There is also some class overlap.



Next, we went over the different ML algorithms to compare each model against the others to determine which methods we should use. We compared Naive Bayesian, LogisticRegression, KNN, and Random Forest models.

We can see from our results that KNN and Random Forest have the best accuracy score out of the methods.
So we decided to use the methods for training.

Model	Accuracy
Random Forest	0.99
KNN	0.95
Logistic Regression	0.83
Naive Bayes	0.65

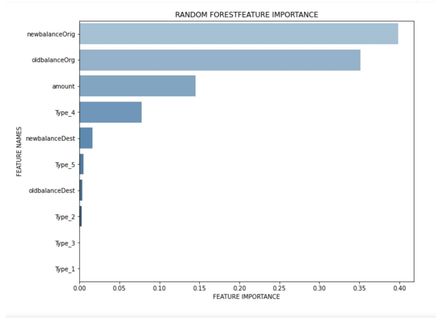
To tune our models we conducted optimization for the hyperparameters. We performed a GridSearch Cross-Validation to find the best hyperparameters for each model. We took the number of cross folds to be 5 and the validation set was within the training set.

Model	With Hyperparameter Optimization Accuracy
KNN	0.94
Random Forest	0.99

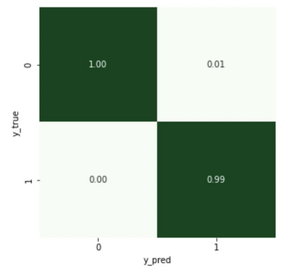
Findings and Evaluation

As our dataset is imbalanced, we transformed our dataset into balanced dataset. First, we extract feature importance from random forest model and plot it below. Second, we plot the confusion matrix for each model so that we can visualize our model performance including precision and recall.

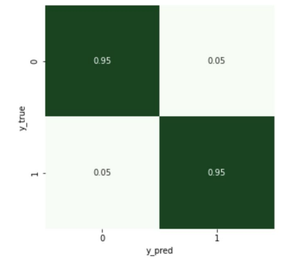
Random Forest Feature Importance



Confusion Matrix for Both Logit and RF model



Random Forest Model



KNN Model

Conclusions

Through our experiment, we found out that the Random Forest model performs the best among other models, so we used that model to do hyperparameter optimization to get the best model. Our hyperparameters are 'criterion': 'gini', 'n_estimators': 30 and 'max_features': None. Our accuracy score is 0.99 which is almost close to 1 and our precision and recall scores are 0.99 and 0.99. Also, from the Random Forest model, we can extract feature importance. And we get top3 feature importances for this dataset which are 'newbalanceOrig', 'oldbalanceOrg' and 'amount'.