Assignment 7

Credit Card Fraud Detection

Submitted by
Swathi Ganesan
12372237
swathiganesan@uchicago.edu

Problem statement - **Credit Card Fraud Detection case study**

The consulting team's approach in addressing Broom's Solutions' issue of credit card fraud and its impact on customer satisfaction rate is to design and build a credit card fraud detection model via Supervised ML that will not only help us identify fraudulent credit card transactions but also help Broom's Solution get a clear idea of the process behind the scenes thus achieving a more satisfied customer base by minimizing the number of denied transactions.

Step by step Predictive Modelling Process

a. Understanding the Data and Data Cleaning

We start the process with importing the relevant libraries and loading the Train and Test dataset of the Transaction data. After looking at a snapshot of the data using the head() method and understanding the datatypes of the columns using the info() method I was able to discern the following

Numerical feature:

amt
city_pop

Categorical features:

trans_date_trans_time
cc_num
merchant
category
gender
street
city
state
zip
lat
long
merch_lat
merch_long
is_fraud

A quick check for null or missing values showed that there were no null or duplicate values in the data.
We obtain the descriptive statistics on the numerical features of the data to understand the spread of data using count, mean, standard deviation, minimum, maximum and the percentiles.
We then understand the distribution of the categorical features using value_counts().

b. Data Pre-processing and Splitting

A quick look at the correlation matrix shows us that the **amt** column has comparatively higher correlation with our target feature **is_fraud.**

We plot some basic visualizations to understand the data and the features better before proceeding to modelling. Plotting histograms for the key feature as they are good indicators of the data distribution and spread along with helping us identify outliers and high-leverage points in the dataset.

We plot histograms on the **amt** column by splitting the data on our target feature. We can see that a lot of high value transactions are fraudulant. We can hence confirm our finding from the correlation matrix.

We perform some data-preprocessing by extracting the transaction date, year, month from the **trans_date_trans_time** column. We then compute the **age** at the time of transaction from the **dob** column. We then compute new calculated columns **avg_amt_60d** as the rolling 60 day average of the credit card transaction amount and **60d** as the rolling 60 day total of the credit card transaction amount.

In order to identify the effect of other categorical features, we can cross tab them to plot a frequency table of the factors. From the outputs we can infer that gender does not influence credit card fraud. However, we can see that **shopping_pos, misc_net, shopping_net and grocery_pos** categories contribute to about 69.3% of total fraudulant transactions.

is_fraud	0	1
category		
shopping_pos	165407	1056
misc_net	89472	1182
shopping_net	137103	2219
grocery_pos	173963	2228

Contrary to common assumptions that we could encounter more credit card fraud in online transactions, the data shows us that offline grocery POS and online shopping are the most susceptible to frauds. This could also be due to the fact that we still have a lot of consumers who prefer buying grocery offline but prefer shopping online.

Many machine learning algorithms cannot operate on label data directly. They require all input variables and output variables to be numeric. In general, this is mostly a constraint of the efficient implementation of machine learning algorithms rather than hard limitations on the algorithms themselves. To achieve this, we can do one-hot encoding or dummy/ indicator variables for the payment type column and the usage frequency columns. In our case we have created corresponding indicator variables for ***gender and category*** columns.

We can select the features for our model as : *'amt', 'trans_hour', 'avg_amt_60d', '60d', 'age', 'gender_M', 'category_food_dining', 'category_gas_transport','category_grocery_net','category_grocery_pos, 'category_health_fitness', 'category_home', 'category_kids_pets', 'category_misc_net', 'category_misc_pos', 'category_personal_care', 'category_shopping_net', 'category_shopping_pos', 'category_travel'*

Oversampling in data analysis is a technique used to adjust the class distribution of a data set when the amount of data collected is insufficient. When one class of data is the underrepresented minority class in the data sample, over sampling is used to duplicate these results for a more balanced number of positive results in training. Oversampling is a well-known way to potentially improve models trained on imbalanced data.

Since we know that in our data only **.5% of the entries are fraudulant**, we would need to oversample before proceeding with sampling.

We need a train-test split on the data to estimate the performance of our models in predicting fraud. We make a train:test split in the ratio 2:1.

c. Model Building

We are building several models to identify fraud/non fraud transactions using the following models:

a. Logistic Regression
b. Decision Tree
c. Random Forest
d. XGBoost

We fit the mode using our train data (X_train and y_train) and we predict the y_test_pred values using the X_test input.

We would now need to evaluate our model by comparing the y_test_pred and y_test values.

d. Model Outcomes and Validation

In order to classify transactions as fraud or non-fraud we would be using several models and comparing their performances using the following methods:

a. Confusion Matrix

Confusion matrix is a method used to summarize classification algorithm on set of test data for which the true values are previously known. Sometimes it also refers as error matrix. To interpret the model performance from the confusion matrix we can compute Accuracy as follows:

Accuracy = (True Positive + True Negative / Total)

b. Classification Report

A Classification report is used to measure the quality of predictions from a classification algorithm. It gives us the precision, recall, f1-score and support metrics which are explained below

Precision - Accuracy of positive predictions: TP/(TP + FP)

Recall - Percent of the positive cases identified correctly: TP/(TP+FN)
F-1 score - 2(Recall * Precision) / (Recall + Precision)*

c. ROC curve

ROC- Receiver operating characteristic curve will help to summarize model's performance by calculating trade-offs between TP rate and FN rate and it will plot these 2 parameters. To classify this term AUC (Area under the curve) is introduced which gives summary of ROC curve. We can conclude that higher the value of AUC better its ability to distinguish between positive and negative classes.

The higher the AUC, the better the performance of classifier.

We can interpret the AUC values as follows:

1) If AUC =0 then classifier is predicting all the positive as negative and negative as positive.

2) If 0.5< AUC < 1 means classifier will distinguish the positive class value from negative class value because it is finding a greater number of TP and TN compared to FP and FN.

3) If AUC = 0.5 it means classifier is not able to distinguish between positive and negative values.

Logistic Regression

A logistic regression model is a statistical model that models the probability of an event taking place by having the log-odds for the event be a linear combination of one or more independent variables.

Model Accuracy: 84.12%
AUC: 92.53%

Decision Tree

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.

Model Accuracy: 88.38%
AUC: 94.45%

Random Forest

Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a

multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees.
Model Accuracy: 89.5%
AUC: 95.9%

XGBoost
XGBoost, short for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.
Model Accuracy: 94.52%
AUC: 98.67%

Recalling our findings from the EDA visualizations as follows:

- We can clearly see that there are more fraudulant instances in higher valued transactions.
- Data seem to suggest that females and males are almost equally susceptible (50%) to transaction fraud. Gender is not very indicative of a fraudulent transaction. But we have more female shoppers than male and this could mean that male shoppers could be more involved in fraudulant transactions.
- In fraudulent transactions, the age distribution is a little smoother and the second peak includes a wider age group from 50-65. This does suggest that older people are potentially more prone to fraud.
- Fraud tends to happen more often in ‘shopping_net', 'grocery_pos', and 'misc_net' categories while 'home' and 'kids_pets' among others tend to see more normal transactions than fraudulent ones.
- While normal transactions distribute more or less equally throughout the day, fraudulent payments happen disproportionately around midnight.
- We can see a spikes in frauds around December which is the holiday season, as a greater number of shopping transactions happen around the holidays.
- We can infer that New York, Texas, California, and Ohio are most susceptible to credit card fraud

Findings from our model output:

The Decision tree model shows that the top 3 most important features are **amt, trans_hour, category_travel.** This reinforces our findings from the EDA where **transaction amount** and **transaction hour** most definitely seem to have a huge impact on detecting fraudulant transactions.

Insights and inferences for our client – Broom's Solutions:

We can see that XGBoost is the best model for detecting fraud as it has a high model accuracy as well as the greatest area under the curve amongst the selected models. *XGBoost* would be the perfect solution if our client was focused on accurately detecting fraud and preventing such fraudulant transactions. However, the CEO mentioned that they wanted to focus on customer satisfaction as it is pivotal to their business.

Therefore, in order to detect transaction fraud but also maintain customer satisfaction to an extent, we can use the **Logistic Regression** or **Decision Tree** models to detect fraud as they have less than 90% accuracy in detecting fraud and this will help preserve customer relations for Broom's Solutions.

MO suggestions for client to operate vigilantly:

- During holiday season most people tend to go for shopping and other recreational activities. Thus the retailers in these domain categories can be notified to be more cautious.
- We would need to monitor transactions made during nights more actively.
- Double check transactions made by credit cards owned by old age people are they might be targeted by fraudsters to take advantage of their lack of financial knowledge
- We would need to alert our retailers and ecommerce vendors in NY, TX, CA and OH about the concentration of fraud in these states
- Vendors in shopping and grocery need to be educated on safe practices to avoid frauds.