Springboard - Data Science Track

Jacqueline (Xuan) Guo

First Capstone Project Report

June 10, 2020

Title:

Improve the efficacy of fraudulent transaction alerts

Introduction:

When doing some transactions with a debit card or credit card, we often face a situation where the card cannot work normally. Imagine there are a bunch of people waiting behind you and you tried your card once, twice, three times, but it's still not working. This is a very embarrassing situation when fraud detection went wrong. Therefore, improving the accuracy of fraudulent transaction alerts would help to avoid this embarrassing situation. Also, reduce the risk when the detection undiscovered leading people to lose money.

Companies such as JP Morgan Chase, Bank of America, and Credit Card Processing would love to use the improved model to predict fraud activities. If this project is successful, it will help hundreds of thousands of businesses reduce their fraud losses and increase their revenue. Customers may also want to choose banks or financial companies with high accuracy fraud detection.

Dataset:

The data comes from Kaggle Competition - IEEE-CIS Fraud Detection**,** Kaggle's data comes from Vesta's real-world e-commerce transactions and contains a wide range of features from device type to product features. The dataset contains 20 kinds of features in total such as *DeviceType* (transaction device type), *addr* (where this transaction was made), and *card* (what kind of card the customer used). Then using those features to analyze the relationship between

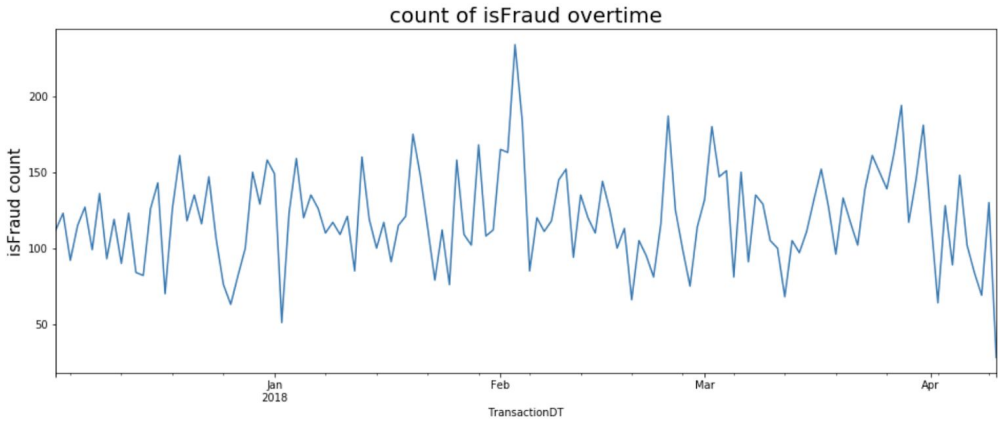
features and labeled fraud and build machine learning models to improve better accuracy fraud prediction.

Preprocessing

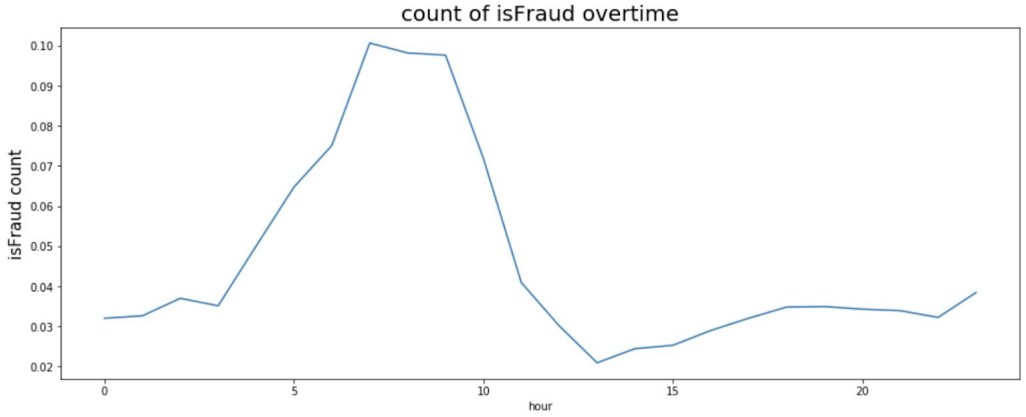
Since there are many available features in our samples, I decided to do feature selection first to see if there are some features that have strong correlations to fraud results. The link of coding notebook: [Capstone Project 1]

I. Date and Time

I've noticed that there are features about the date and time in the transaction data set. I want to measure if the fraud action is somehow related to the time of transaction. Therefore, I draw a fraud count graph with daytime and try to find if there's any obvious trends. First, I used per day as the unit of count, and draw the fraud action rate overtime:



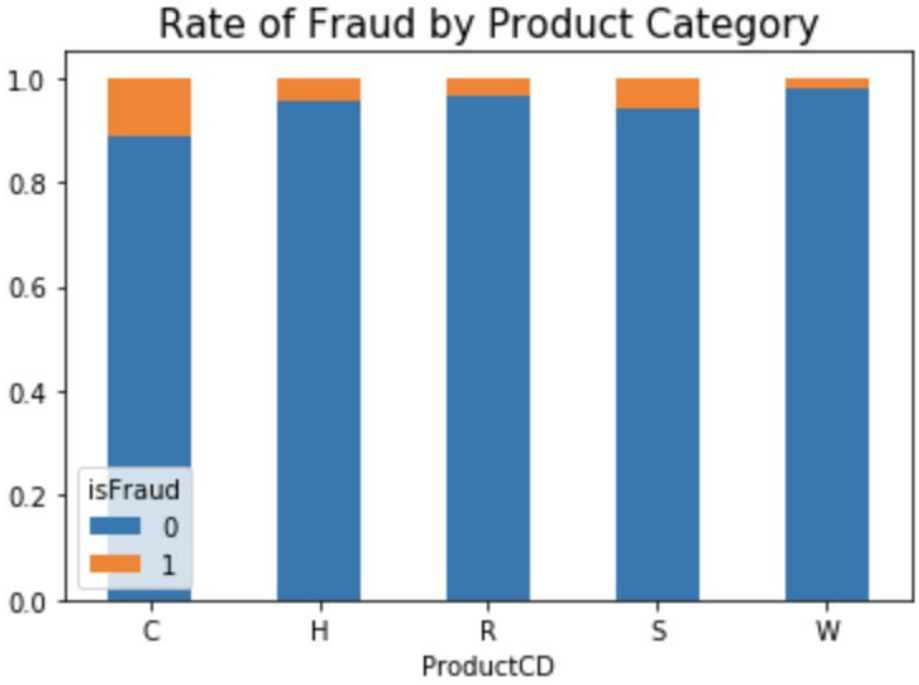
However, according to the figure above: no significant pattern visually based on days. Thus, I changed the unit to hours per day:



Then we can notice that there's a sharp increase between 5 am to 10 am. Thus we could conclude that the fraud action seems to peak during early morning between 5 am to 10 am. In conclusion, the day of the week doesn't seem like a very powerful feature compared to the hour of the day.

II. Product Types

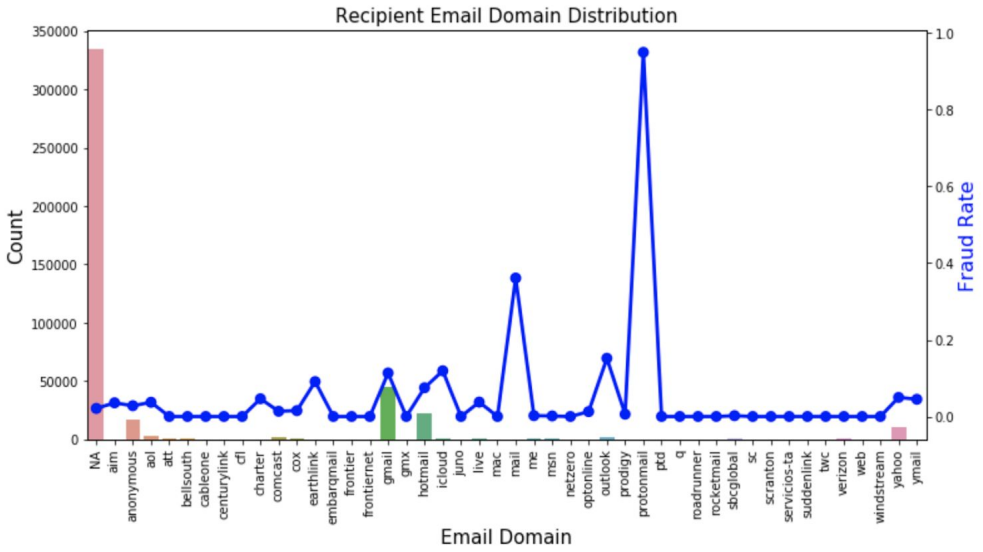
In addition, I've noticed that different product types may cause more or less frauds.



As the figure shows, product C takes up 67.5% of fraud cases for transactions that have identity. It also has the highest rate of fraud -- 12%, more than double any other class of product.

III. Email Domains

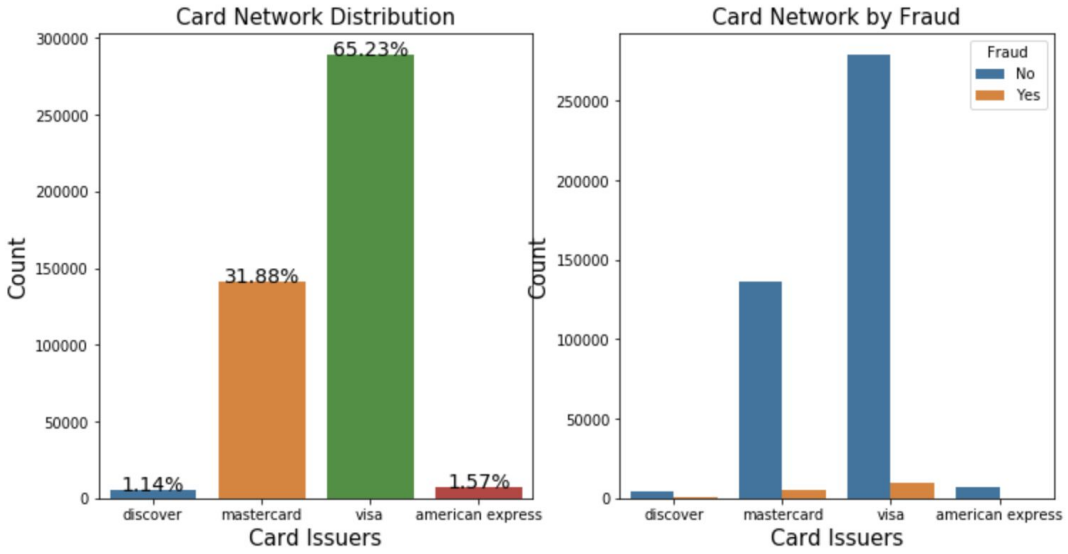
People using different email domains to make purchases. The fraud rate may vary by different email domains.



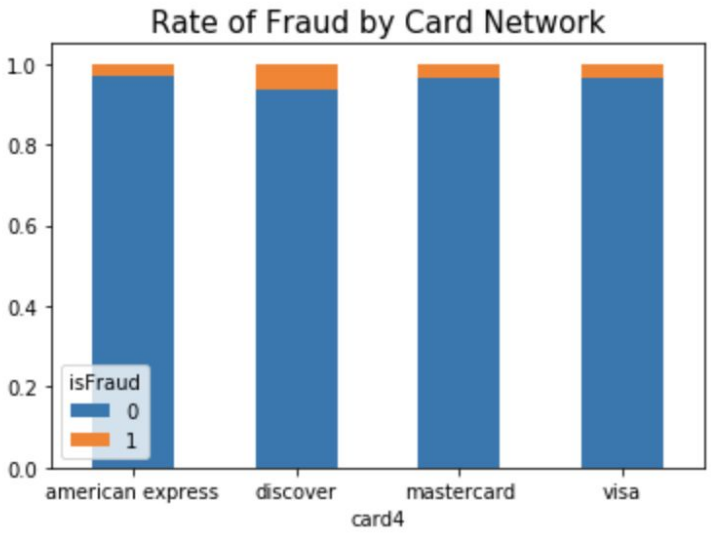
Even though gmail has lots of transactions, the fraud rate is in a normal range. We should pay more attention to those abnormal email domains such as mail, protonmail.

IV. Card

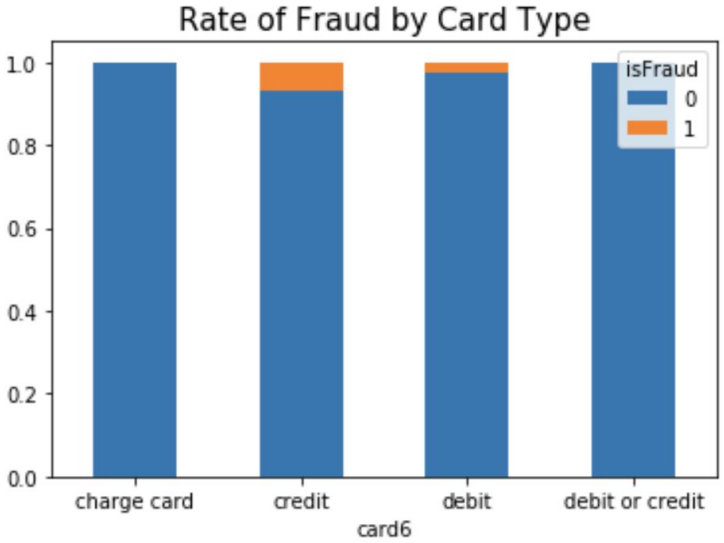
Firstly, I was determined to see the categorical card features. I used data visualization to figure out the relationship between card issuers, card types and fraud actions as the following figures show:



As we can see, although visa has the highest counts of "isfraud", this is because visa is most used type of card. If we divide it by its population then we can see that American Express has a lower fraud rate.



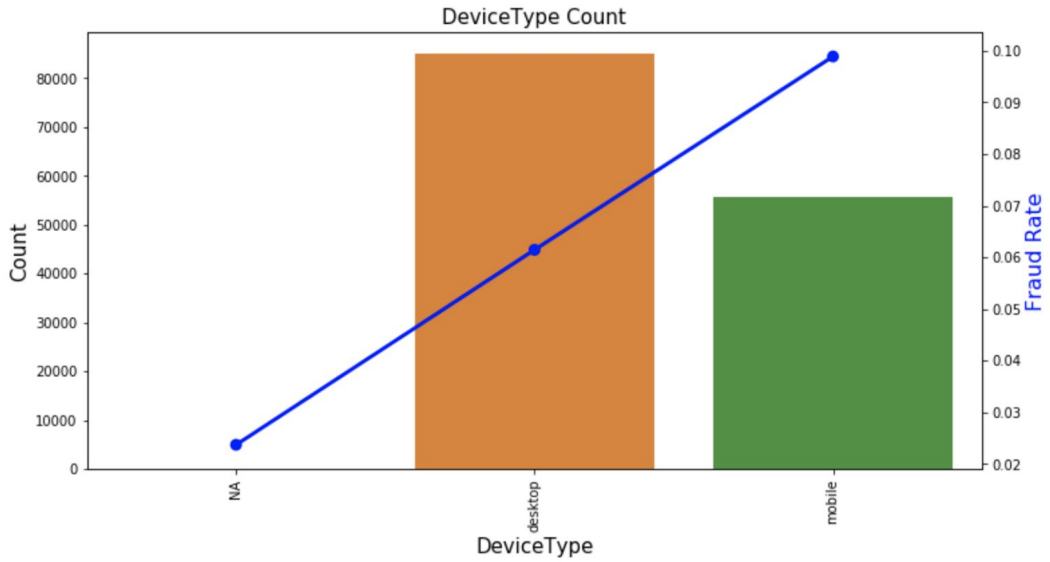
Then looking at the card networks:



The above figure indicates that credit cards tend to have a higher fraud rate.

V. Device Type

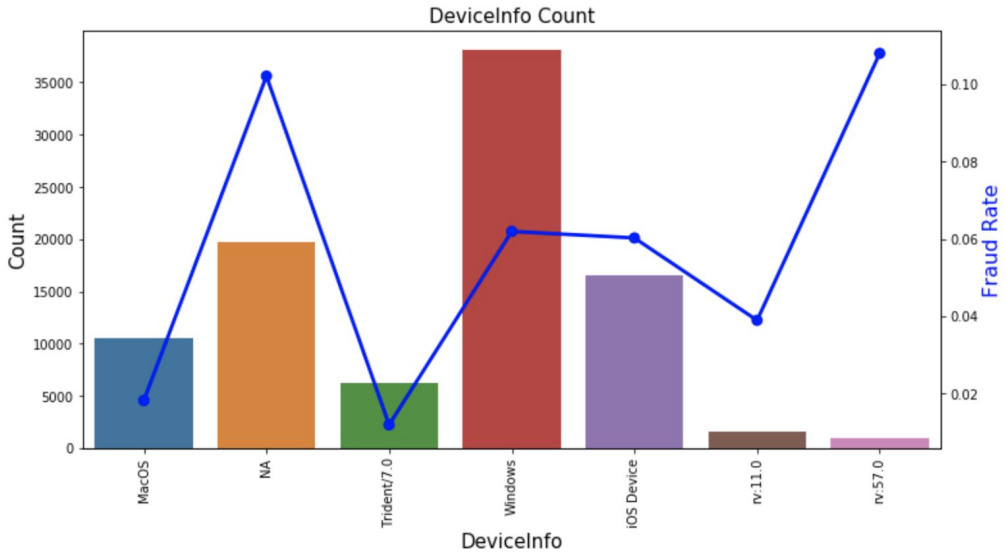
Moreover, there are features about different device information. It's worth finding out if there are some correlation between different device types and fraud rates.



We've noticed that most people pay their bill through a desktop. Although it has a higher fraud rate than null type, it is because the transaction amount of this pay device is very high. On the contrary, the fraud rate is very high on mobile payments. It's probably more risky in mobile payment service.

VI. Device System

There are 7 different device systems such as ios, windows, ect. It's necessary to find out if the different device system has strong correlation to fraud rate:



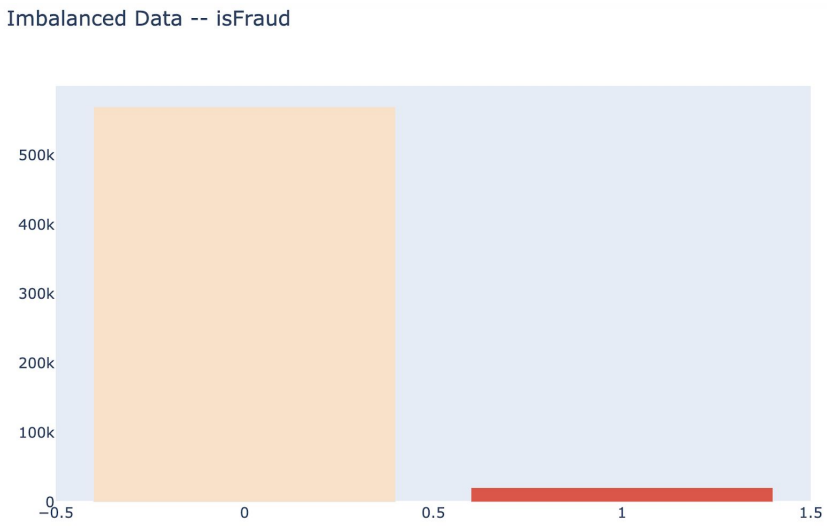
The Macos system has relatively low fraud rate, and the financial company should pay more attention to undefined device systems and "rv" kinds of deceive systems.

In the future research and analysis, it is worth to focus on those features that have a strong relationship with the fraud rate and improve the detection from those aspects.

Limitations:

I. Imbalanced Data

Our labeled data is imbalance cause the fraud action is only a small part of the whole sample size.



II. Unknown Data

I found some of the features are hard to analyze because we could not know the feature's meaning due to security reasons. For example, the card information, 4 of them are numeric data and we cannot know what's the meaning of them. However, we couldn't drop them because the ROC score became smaller after dropping those features.

III. Outliers and Missing Data

There are lots of outliers and missing data in our data set. It's hard to determine if the outliers are due to real world phenomena or some human mistakes. What I did is include those outliers into our analysis, and replace the missing value with median value.

Feature Engineering and Models:

I. Objective:

Utilize supervised learning techniques to build predictive models for the credit card transaction data. So far I have gone through the data, did some exploratory analysis on the dataset. With the understanding gained, the next step is feature engineering, selecting models, and evaluating models.

II. Feature Engineering

Based on exploratory data analysis, here are the feature engineering I did (aside from minor data cleaning such as lowercase all device info):

- ❏ Oversample the fraud class.
- ❏ Fill missing values with median and create a new column that counts the missing value.
- ❏ Create new features representing transaction hours in a day.
- ❏ Create two new aggregated features using card1-2 and addr1-2 to represent user id.
- ❏ Change device version into latest and outdated where outdated is represented in a numerical scale.
- ❏ Convert categorical values into a numerical representation.

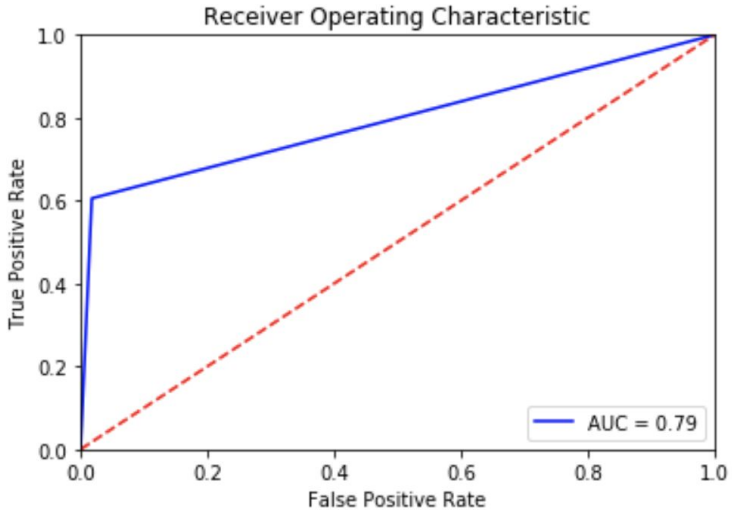
III. Model Selection

I have selected three models to perform our fraud rate prediction which are: Decision Tree, Random Forest, and Support Vector Machine.

Before jumping into our models, as I mentioned in the previous section that we have unbalanced classes which is the most of our fraud action count are “not fraud”. At this point, we could not use the general accuracy score of the model to determine whether our result is accurate. In a dataset with highly unbalanced classes, if the classifier always “predicts” the most common class without performing any analysis of the features, it will still have a high accuracy rate. This is obviously something I want to avoid. For example, simply predicting everything true would have ~92% accuracy. Since our problem is fraud detection, what we really care about is to detect as many frauds as possible while it’s ok to classify non-fraud action as fraud. In other words, we should be focusing on minimizing the false negative rate instead of false positive. Therefore, I decided to use the Receiver Operating Characteristic curve as the metric. An ROC curve is a graph showing the performance of a classification model at all classification thresholds (Google's Machine Learning Crash Course). Then, let’s talk about our models:

1. Decision Tree

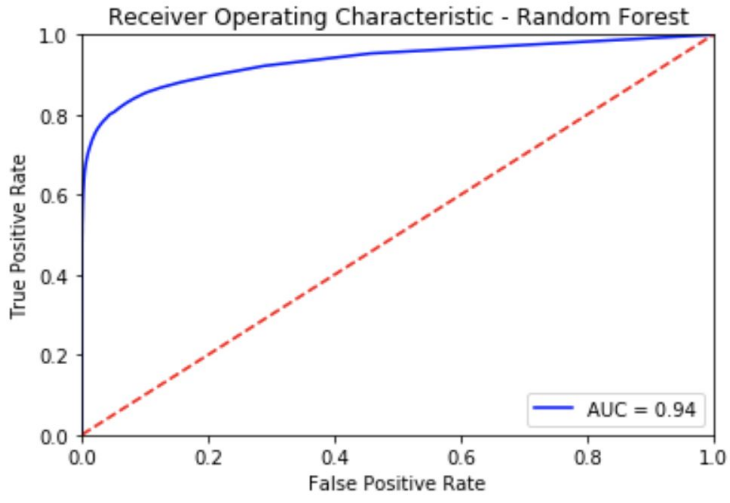
   Decision Tree Learning is one of the predictive modeling approaches used in statistics, and it uses a tree-like model of decisions to category our data. I've applied the decision tree machine learning model of Scikit-learn package after the data engineering and got the AUC-ROC score of 0.794. The AUC-ROC curve as the following:



   As the figure states, the decision tree seems like not a perfect fit for our data.

2. Random Forest

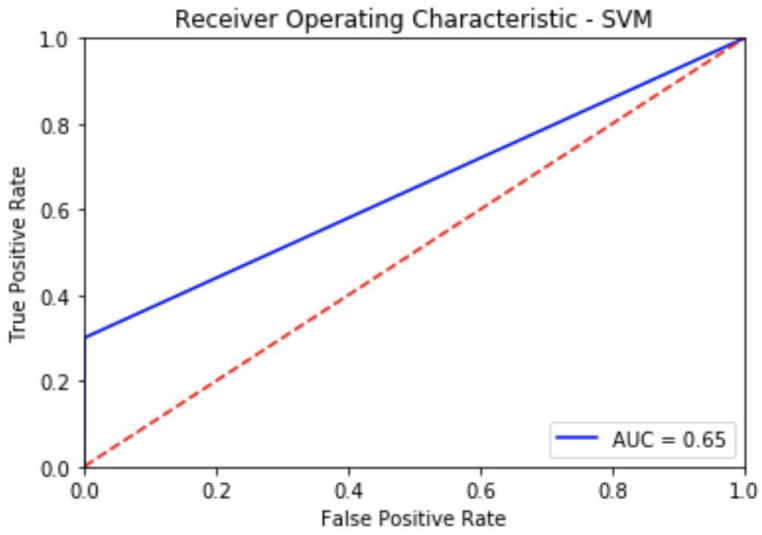
   The Random Forest is a classification algorithm consisting of many decision trees to avoid overfitting. After applying random forest to the data, our roc score is 0.935, and the roc curve as the following:



The roc score is much higher than the decision tree, and it's close to 1. Thus we can conclude that the random forest is a better model to predict the fraud rate compared to the decision tree.

3. Support Vector Machine

   I've tried one more model to find the best model to predict. Support Vector Machine is based on the idea of finding a hyperplane that best divides the data into two classes (fraud or not fraud). I got the roc score of 0.650, and the roc curve is:



Thus we know the SVM method is not a good fit for this problem.

Discuss and Conclusion:

In conclusion, as this is a classification problem – classify if a data point is fraud or not. I've tried three models to find the best model for our prediction. According to the roc score of each model, we can conclude that Random Forest is the best model to use in this problem because it has a 0.935 roc score. The prediction result is much more accurate of the random forest model compared to others. I've also found that feature engineering does not always improve the model. There are various cases that I make modification on features based on statistical inference, for example I remove all date information except hour because only hour shows slightly stronger correlation with the label, but then the model performs worse.

In this capstone, I spent most of the time on exploring data and feature engineering. I've learnt a lot of new ways to gain insight among features and reinforced my previously learnt knowledge. Furthermore, I realized how visualization can be more helpful in large data sizes. I have a better understanding of some of the techniques I've learnt, for instance, when reducing the dimension of features, I reviewed details of PCA and found out that I can look at the eigenvalue of each principle component to determine the dimension to reduce without losing too much information on original data. Through evaluating the model, I've learnt all sorts of metrics such as precision, recall, f1 score, AUC ROC, etc.

Future:

This Kaggle competition data set has lots of features for credit card transaction and identity information. To make the prediction more accurate, sometimes we should abandon some weak features and only use features that are strongly related to fraud rate. However, I could not completely understand the relations between features, so it may cause overfitting.

In addition, each feature may have more aspects for us to understand this problem and do our analysis. Take the device information as an example, we've already found different devices may cause the different level of fraud rate. However, on the other hand, the device system version may also indicate the probability of fraud action happening. More detail on this point, if the consumer's device system is up to date (newest version), then it means new techniques may already applied on this consumer's device to protect his/her from fraud. On the contrary, if a consumer didn't pay any attention to their system or any updates, then they might be too lazy to update their security protection methods such as password. This also could lead to more fraud. Therefore, there are still a lot of insights to find out of this dataset. We could never stop to explore the insights from our data.

Deliverables:

1. Code notebooks

2. Report on the capstone project
3. Presentation on the capstone project

Reference:

1. Kaggle.com. 2019. IEEE-CIS Fraud Detection | Kaggle. [online] Available at: <https://www.kaggle.com/c/ieee-fraud-detection/overview
2. ROC Curve and AUC | Machine Learning Crash Course. (n.d.). Retrieved from: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc