CREDIT CARD FRAUD DETECTION

Agnel Aaron (2017010)
Raunak Srikant Mokhasi (2017085)
Anjanay Kirti Gour (2017021)
BTech in Computer Science and Engineering, IIIT Delhi

Abstract

There are countless transactions taking place around the world every second, and though the percentage of fraudulent transactions is less, there is still a possibility. In order to detect these fraudulent cases, we design Machine Learning models that detect the probability of whether a transaction is fraudulent.

1 Introduction

In light of the recent COVID-19 pandemic, it is not advisable to pay through cash as it helps spread coronavirus through physical contact. Most of the transactions are taking place through online mode or credit cards to avoid any physical contact. As the transaction through a credit card is growing in number, taking a considerable share of the world's payment system. As a result, the upsurge in stolen account numbers increases, leading to vast amounts of banks' losses. Therefore, an improved and robust fraud detection system has become an essential feature in Payment systems across different countries to reduce credit card transaction fraud.

With the advent of technology, movements for a cashless economy and the current pandemic situation, the credit card payments are on the rise. Credit Cards have immense advantages of their own, as they keep our wallets lighter, prevent issues with change and enable us to purchase things online from our couch.

Credit Card Fraud can take place by a variety of methods including getting your credit card stolen, your credentials getting hacked or even someone applying for a credit card in your name. Such problems are often unnoticed until we face this misfortune. In rare cases, this can even lead to identity theft. Thus, it is imperative to find out these frauds and tackle the problem seriously. Hence, this is a significant motivation for choosing the Credit Card Fraud Detection as our topic for the Machine Learning Project.

Using the concepts taught in the course, we study a vast dataset of credit card information to make some exploratory data analysis on the features. Then we develop models in order to find out the probability of a particular transaction being fraudulent.

2 Literature Review and Related Work

There are many different forms of credit card fraud, but there are several prominent and most common types of credit card fraud. Mainly, frauds due to stolen accounts or credit cards. Frauds due to fake cards are a growing problem despite having various security protocols, encrypting the chip on the credit card to add an extra layer of protection. Banks have started adding daily transaction limits to limit the transaction activity to avoid misuse. Yet, they are unable to control the fraud, and swift action is necessary.

Large scale machine learning algorithms can help banks and credit card companies to solve this problem. Various techniques to examine large datasets of transactions that can accurately compute the fraud detection in a short time. This task has multiple issues regarding the distribution of data for the training set. Many research papers have recently been published that discuss IEEE fraud detection dataset distribution of the transactions and various features.

The use of machine learning with fraud detection are listed as follows -

- Higher accuracy of detecting fraud: Machine learning algorithms provide higher accuracy. It returns better results compared to rule-based solutions as it considers multiple factors. Along with that Machine Learning algorithms can take more data points, even the slightest details about the behavioural patterns related to a particular account.
- Less manual work, more independent: as the machine learning algorithm returns high accuracy, it reduces the manual work to verify the solution.
- Identification of new patterns: ML algorithms can identify new ways and adapt to new patterns without changing the code. This helps us to identify new types of frauds from the transaction.

Using the concepts taught in the course, we study a vast dataset of credit card information to make some exploratory data analysis on the features. Then we develop models to find out the probability of a particular transaction being fraudulent.

In order to learn about related work, we went through the following papers –

- Chan, P.K., Fan, W., Prodromidis, A.L. and Stolfo, S.J., 1999. Distributed data mining in credit card fraud detection. IEEE Intelligent Systems and Their Applications, 14(6), pp.67-74.
- Ghosh, Sushmito, and Douglas L. Reilly. "Credit card fraud detection with a neural-network." In System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on, vol. 3, pp. 621-630. IEEE, 1994.

Some of the concepts in these papers were very advanced and beyond the scope of the course, but it provided wonderful insights on why this is such a prevalent problem and how we can solve it.

3 Dataset

In this project, we use the dataset jointly created by the IEEE Computational Intelligence Society and the Vesta Corporation, which is a leading payment service company. The main goal of the data is to seek the best solutions for fraud prevention. It contains details on real-world e-commerce transactions through a wide variety of features (590,540 rows and 434 features). These features are anonymised, and there is no Personally Identifiable Information. Some columns don't even have proper descriptions which certainly made our job more difficult.

We have to predict a binary attribute (isFraud) that denotes an online transaction's probability of being fraudulent. The dataset is disjoint into two parts, identity and transaction, which have TransactionID as the common attribute. The file names train_transaction and train_identity are the part of the training set, test_transaction and test_identity is part of the test set. The Transaction Table contains details of monetary transactions such as those being used for gifting goods or services like booking a hotel etc.

The dataset contains the following attributes:

- TransactionDT which is the time delta from a given reference date and is not an actual timestamp.
- TransactionAMT which is the transaction amount paid in Dollars.
- ProductCD is the product code of the specific product.
- Columns card1 - card 6 have the payment information of the card like bank details.
- Addr contains the address of the purchaser, and that includes Addr1 with billing region and Addr2 with billing country.
- Dist contains the distance between the billing addresses or zip codes.
- P_emaildomain is the email domain of the purchaser.

- R_emaildomain is the email domain of the recipient.
- Columns c1-c14 count how many addresses are found to be associated with a particular card.
- Columns d1-d15 include the time delta or days between transactions
- Columns m1-m9 contain Boolean values on whether the names of the card and address match.
- Vxyz are Vesta contrived payment features.

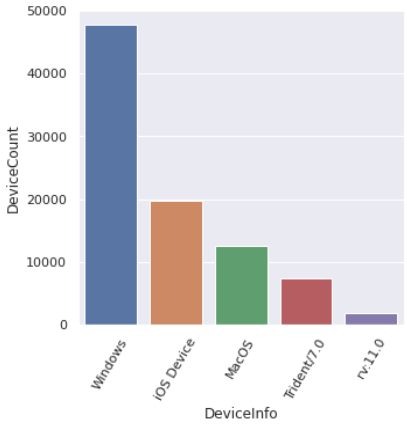
The Identity Table contains information about the transaction's distinctiveness or a digital signature related to a particular transaction. The columns id01 - id11 are essentially numerical attributes of the identity. It mainly has network connection information (IP, ISP, Proxy) and digital signature (OS, Browser version) associated with the transactions. It also contains information like the number of times account is logged in or failed to login or duration of the time the account was active.

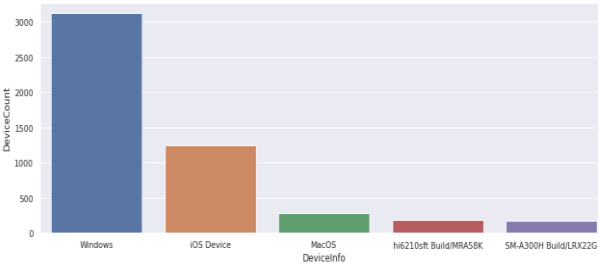
4 Methodology and Analysis

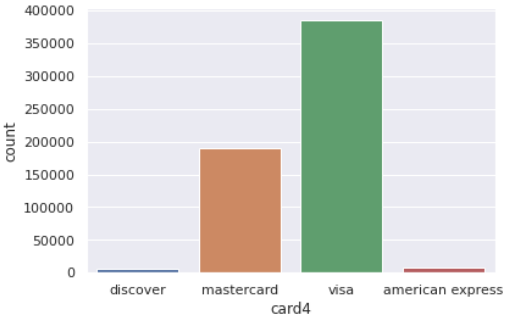
Preprocessing: The major difficulties that we noticed in the dataset right away was that it was extremely skewed. Now, this is realistic because only a very small percentage of all credit card transactions are fraudulent. Thus we had to use random undersampling techniques in order to solve this issue. There were a lot of outliers and NaN values, so those had to handled as well. We also tried the Z-Score approach to remove outliers and also normalisation, but that did not work out well in our case. (Though to remove outliers we feel our undersampling techniques work well).

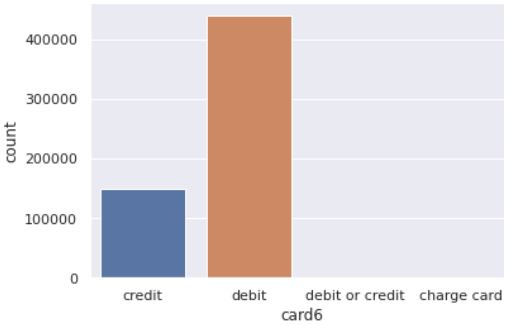
Exploratory Data Analysis: On examining the features and plotting all the relationships as follows:

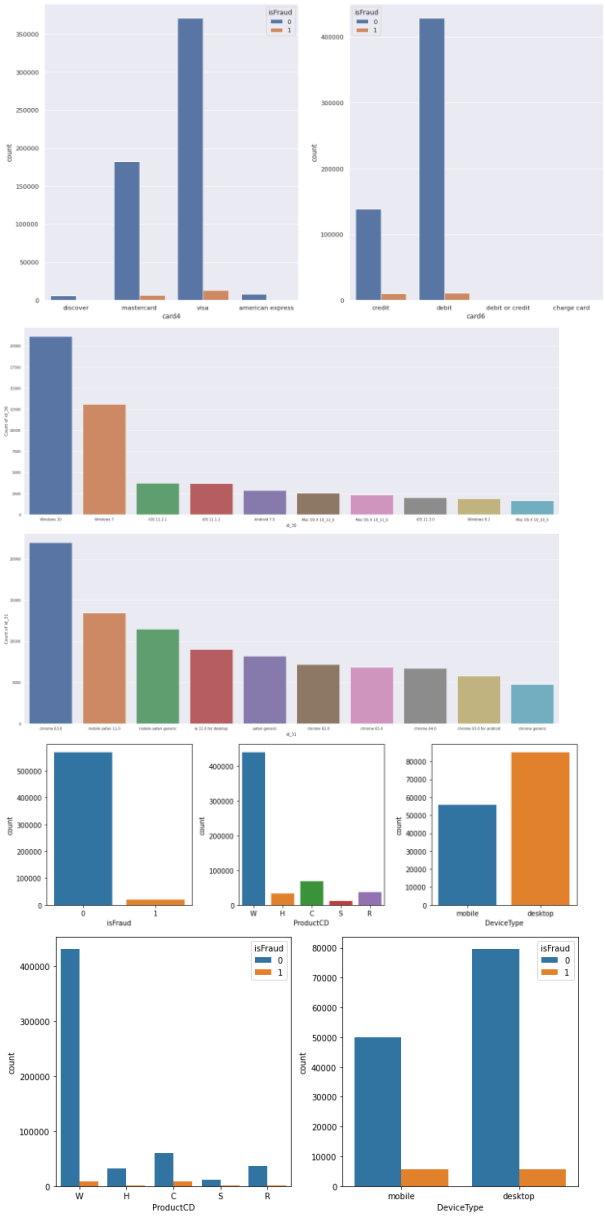
1) Desktops are being used more than Mobiles for Transactions. Mobiles and Desktops have the relatively same number of Fraud Transactions, but the proportion is more in Mobiles.
2) C and w Products have the highest number of Fraud Transactions.
3) Most transactions and frauds take place from debit cards.
4) The plots below include these points and also show the count of Operating Systems, Fraud per Operating System, Counts and Types of the card used in the transaction, Purchaser and Recipient email domain.
5) More plots have been attached in the submission.











Model Number 1 (Decision Tree):

At first we used a Decision Tree in order to classify the Transaction as Fraudulent. It chose 4 particular features (for example "TransactionAmt","ProductCD","card4","isFraud") and then we used Stratified Sampling i.e Splitting the data set to ensure that the train and test sets have approximately the same percentage of samples of each target class as the complete set. We also changed all the categorical attributes to numerical values. After that we ran the Decision Tree Classifier Algorithm, and it gave us a 69.45% Test Accuracy.

Model Number 2 (Decision Tree):

We had decided to use the DecisionTreeClassifier (69% Accuracy) or the K Nearest Neighbours (55% Accuracy) initially, but then due to the sheer complexity of the dataset and also the poor accuracy, we had to move further use to tree ensemble techniques like RandomForestRegressor and RandomForestClassifier to combine the multiple models and give a better AUC accuracy. We planned to use RandomForest as each tree in the forest is trained on a random subset of the data points with replacement (bagging / bootstrap aggregating). The RandomForestRegressor gave a higher AUC than RandomForestClassifier due to its probabilistic nature, given values between 0 and 1, and also due to the fact that we have to predict the probability of whether the transaction is fraudulent and not exactly if it is fraudulent or not (i.e. it is more of a regression than a classification). The best AUC accuracy we got for RandomForestRegressor was 76.65% for 4 Features used in the previous decision tree model and 88.99 for all features in the dataset.

Model Number 3 (XGBoost):

Looking at the positive response from the ensemble technique of RandomForestRegressor we decided to try XGBoost (eXtreme Gradient Boosting). Using it on the Fraud Credit Card Dataset, it worked by combining the predictions from several models into one by taking the predictors sequentially and modelling it based on the predecessor's error. Then it gave a few features (like TransactionDT) a higher weight based on whether it performed better. We used GridSearchCV in order to hypertune the parameters to get as high accuracy as possible.

Here we got an accuracy of 91% due to the appropriate combination of features and model. After this, we were going through research papers on the detection of fraudulent transactions using Machine Learning. Here we came across a paper by Ge, D., Gu, J., Chang, S. and Cai, J [4] that used a new model called LightGMB for the detection of credit card

frauds and thus we decided to proceed with that model next.

Model Number 4 (LightGMB):

On our 4th model, we were keen on improving our accuracy even further. So in order to do that, we had to improve on preexisting feature extraction and try to make them more meaningful. As of yet, the features were a bit vague to understand. To achieve that we used feature engineering and we used LightGMB as boosting algorithm.

LightGBM is better than XGBoost in terms of accuracy, faster computation, lesser memory requirement, handling large scale data. LightGBM uses leaf wise split the trees and it fits better in comparison to other methods which uses depth wise or level wise split. From various testing and in theory, using Exclusive Feature Building and Gradient-based One Side Sampling LightGBM is able to perform much better in terms of XGBoost.

First we did some Feature Engineering in order to create new and meaningful features from the previously vague features.

Feature Engineering

1. After going through the dataset, we noticed many rows have the same card1 - 5 values. For example, there are 93000 entries for which card3=150. The maximum value_counts for these columns are huge, and card1-5 basically contains details about the card used during the transaction. We wanted to see if we can break down the high-value counts, maybe even isolate/ uniquely identify individuals/cards by combining card details. So in order to do so, we made uid1-4 each of which combines different card details. We see that even uid's have a high maximum value count, for example, Uid_1 has a value of 28000 counts, Uid2 has 26000.
2. Since we have uid's now, we decided to group the dataset by uid1-4, and we applied the 'mean' function to the 'TransactionAmt' column. Since we are interested in outliers, maybe the transaction amt of the rows we are interested in might be away from the mean. Using aggregate function on pre-existing columns in the dataset is very common.
3. We split the P and R_emailDomain column into 2 new columns each. The values in P_emailDomain are of the form gmail.com, msn.net etc. So we split by '.' and keep domainName separate in a new column and the extension '.com',' .org', etc. in a new column. We also combined different aliases of the same domain for example live, MSN is just Microsoft etc. Reason for this is fraud transactions might have a connection to the extension or the domain they can use.
4. TransactionDT is the time delta in seconds, in fact, the very first value is 24*60*60 =86400, which is the number of seconds in 1 day. From this we calculated the hour of the day, the day number (out of 365), the month, the week number (out of total weeks in 1 year). No year because data spans over 6 months. We feel fraud transactions can be linked to the time of day or maybe fraud transactions could be grouped in day number or done in the same week etc.
5. Since we have numerous categorical columns, we decided to use some sort of encoding for those columns. We used frequency encoding as we feel frequency has some kind of connection with isFraud, as the Fraudsters might use their exploit more than once.

We used GridSearchCV to get the values of the of the parameters. The reason for using parameters is that for higher accuracy the learning rate is very small close to 0.006. Since the classes are imbalanced we use the AUC score as the metric. The AUC Score gives the probability that a random positive sample will have a higher score than a random negative sample. According to LightGBM documentation, for high accuracy min_Data_in_Leaf should be in 100’s and num_leaves should be high.

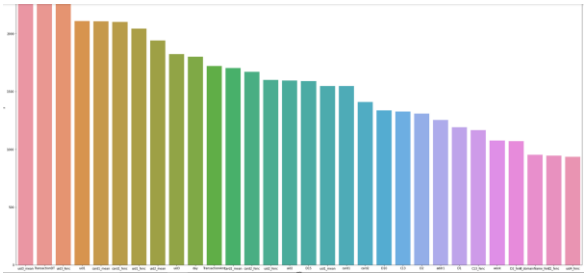
```
params = {'num_leaves': 600,
          'min_data_in_leaf': 100,
          'learning_rate': 0.006,
          "boosting_type": "gbdt",
          "metric": 'auc',
          'random_state': 2020,
}
```

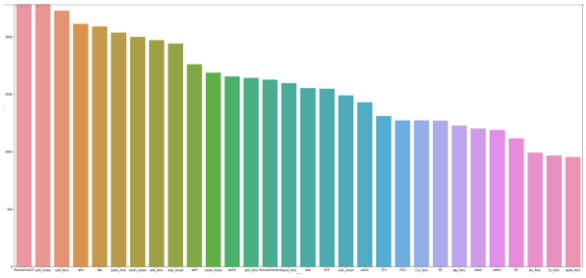
After completing our feature engineering, we use lightGMB boosting algorithm with K-fold cross-validation to measure the AUC score of our model. The LightGBM function returns the feature importance for each fold as it can be seen on the bar graphs.

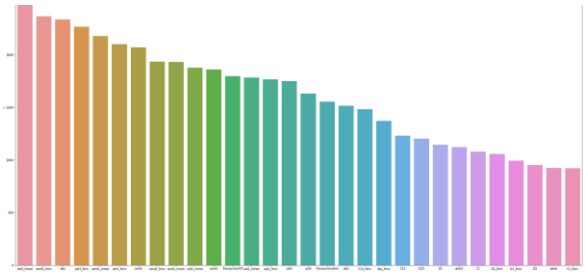
Our highest accuracy is 92.32%.

```
Training until validation scores don't improve for 100 rounds.
[100]   training's auc: 0.944724        valid_1's auc: 0.882809
[200]   training's auc: 0.955086        valid_1's auc: 0.890316
Did not meet early stopping. Best iteration is:
[200]   training's auc: 0.955086        valid_1's auc: 0.890316
Training until validation scores don't improve for 100 rounds.
[100]   training's auc: 0.943788        valid_1's auc: 0.908853
[200]   training's auc: 0.95463 valid_1's auc: 0.915153
Did not meet early stopping. Best iteration is:
[200]   training's auc: 0.95463 valid_1's auc: 0.915153
Training until validation scores don't improve for 100 rounds.
[100]   training's auc: 0.945509        valid_1's auc: 0.904002
[200]   training's auc: 0.95615 valid_1's auc: 0.911763
Did not meet early stopping. Best iteration is:
[200]   training's auc: 0.95615 valid_1's auc: 0.911763
Training until validation scores don't improve for 100 rounds.
[100]   training's auc: 0.943044        valid_1's auc: 0.916925
[200]   training's auc: 0.953886        valid_1's auc: 0.924409
Did not meet early stopping. Best iteration is:
[200]   training's auc: 0.953886        valid_1's auc: 0.924409
Training until validation scores don't improve for 100 rounds.
[100]   training's auc: 0.945415        valid_1's auc: 0.894773
[200]   training's auc: 0.955855        valid_1's auc: 0.901574
Did not meet early stopping. Best iteration is:
[200]   training's auc: 0.955855        valid_1's auc: 0.901574
Mean FOld auc = 0.9086429347030253
```

Feature Importance –
Features like TransactionDT (time delta from reference date) and Uid3 are the most important. The following plots showcase them in decreasing order.







5 Conclusion

Though Credit Card Frauds account for only about 0.1% of all card transactions, they are of extreme importance as the losses can be disastrous. It is challenging to detect such frauds and also challenging to build models for such a problem due to huge imbalance in data. In this project, we proposed and experimented with various models in order to get accurate results in the Credit Card Fraud Detection problem.

Through this project, we learnt various theoretical components of the course in the practical format. We learnt how to preprocess the data to remove anomalies and outliers and learnt how to handle class imbalance by undersampling or oversampling. We also learnt how to use models such as Decision Trees, KNN, Random Forest, XGBoostClassifiers, LightGMB and why one model is better than the other. Moreover, we are learnt how to improve our models' accuracy through various techniques such as hyperparameter tuning using GridSearchCV and data mining techniques like feature engineering. For this, we even had to learn the theory behind each parameter.

Our highest accuracy of 92.32% was satisfactory and we were quite happy with the result. This project gave us a scope of improving our skills and learning new concepts to go beyond the concepts that were taught in class. The project was an enjoyable experience considering the given

pandemic situation and also taught us to work well in a team irrespective of the location we are in.

6 References

1) All the animations on the first page have been taken from the Creative Commons License website.
2) Dataset Credits - https://cis.ieee.org/ and https://trustvesta.com/
3) Chan, P.K., Fan, W., Prodromidis, A.L. and Stolfo, S.J., 1999. Distributed data mining in credit card fraud detection. IEEE Intelligent Systems and Their Applications, 14(6), pp.67-74.
4) Ge, D., Gu, J., Chang, S. and Cai, J., 2020, April. Credit Card Fraud Detection Using Lightgbm Model. In 2020 International Conference on E-Commerce and Internet Technology (ECIT) (pp. 232-236). IEEE.
5) Chan, Philip K., Wei Fan, Andreas L. Prodromidis, and Salvatore J. Stolfo. "Distributed data mining in credit card fraud detection." IEEE Intelligent Systems and Their Applications 14, no. 6 (1999): 67-74.
6) Ghosh, Sushmito, and Douglas L. Reilly. "Credit card fraud detection with a neural-network." In System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on, vol. 3, pp. 621-630. IEEE, 1994.
7) Libraries used - Sklearn, Matplotlib, Seaborn, imblearn, Pandas, Numpy.