Foundation of Data Science Project CSD 355-Report

FRAUD TRANSACTION DETECTION

By

Ayush Varma 1910110108

1. Introduction -

Since the advent of Online Payment Systems, people with bad intentions have come up with new ways to deceive the common masses and make them fall into the fraud traps. The most common type of fraud is found in the form of Card Frauds since the majority of the people worldwide hold one or other type of card which they commonly use for dispensing cash or making online transactions.

2. Business Problem -

2.1 Business Objective

The traditional approaches used for detecting transaction fraudulence involve manual monitoring through humans which also involves interaction with the cardholders. This approach is not very efficient as it not only consumes a lot of time but it is also quite expensive in terms of the resources it uses to detect the fraudulence of a transaction. Moreover, in the real world the majority of the transactions being manually monitored turn out to be legit and only a few of them turn out to be fraudulent which wastes a lot of time and energy. And, hence in this competition, we need to come up with an automated screening system which requires minimalistic human intervention to detect the legitimacy of transactions using Machine Learning.

We will be predicting whether a transaction is fraudulent or not and to predict this we will be building an ML model using a real-world e-commerce dataset.

2.2 Constraints

- The cost of predicting a legit transaction as fraudulent will lead to bad customer experience and predicting the fraudulent transaction as legit will lead to huge financial losses. And, hence the prediction must be as accurate as possible.
- Suppose, a fraudulent transaction occurs and the customer knows about it after hours or days then it is of no use and hence the prediction should be instant. Therefore, we need to build a model which predicts instantly about the state of a transaction.
- Interpretability is also partially important especially in cases where a transaction has been declared as fraud since one must know why a transaction has been declared as fraud.

3. Machine Learning Problem -

3.1 Data

The datasets can be downloaded from **here**.

The datasets provided consist of the following,

- **train_transaction.csv** : The transaction dataset comprising the transaction details to be used for training the model.
- **train_identity.csv** : The identity dataset comprising the additional details about the identity of the payer and the merchant between whom the transaction was performed and the details of transactions are present in the train_transaction.csv.
- **test_transaction.csv** : The transaction dataset comprising the transaction information to test the performance of the trained model.
- **test_identity.csv** : The identity dataset comprising the additional identity information about the transactions present in the test_transaction.

Description of Transaction Dataset

- **TransactionID** — Id of the transaction and is the foreign key in the Identity Dataset.
- **isFraud** — 0 or 1 signifying whether a transaction is fraudulent or not.
- **TransactionDT** — timedelta from a given reference datetime (not an actual timestamp)

- **TransactionAMT** — Transaction Payment Amount in USD.
- **P_emaildomain** — Purchaser Email Domain.
- **R_emaildomain** — Receiver Email Domain.

3.2 Mapping the real world problem to an ML Problem

Problem Type

Since, we need to classify a transaction as fraudulent or non-fraudulent and hence, this is a **Binary Classification Problem**.

Performance Metric

- The organizers decided to evaluate the submissions on the area under the ROC curve between the predicted probability and the observed target. Hence, **ROC-AUC will be our Key Performance Indicator (KPI)**.
- We will be added using the **Confusion Matrix** to add more interpretability to the models.

4 Data preprocessing and exploratory data analysis

4.1 Initial Processing

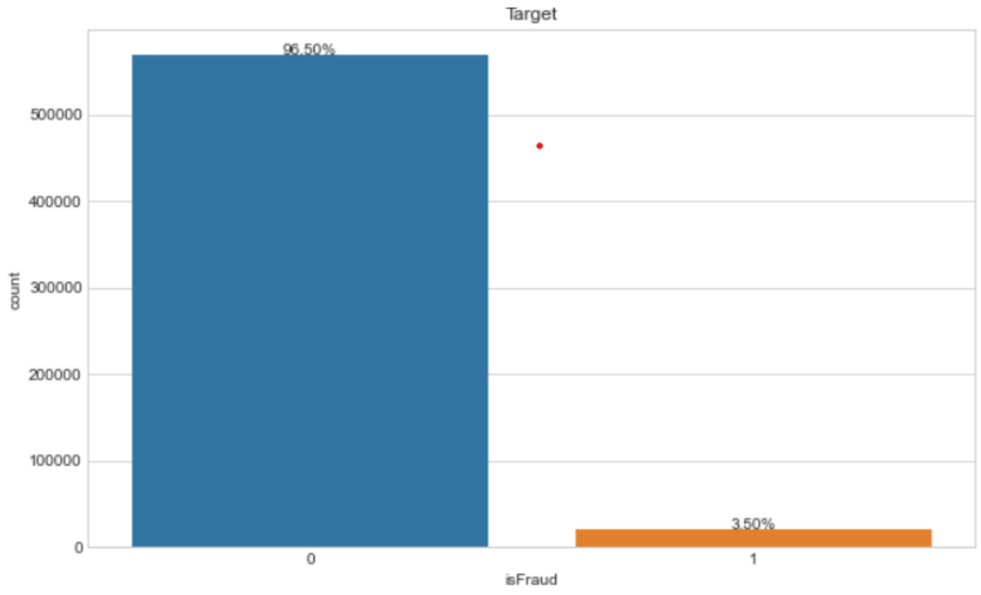
The train_transaction dataset had a total of 590540 rows/data points and 394 columns/features, train_identity had a total of 144233 rows/data points and 41 columns. *The difference in the number of rows in the train_transaction dataset and train_identity dataset made clear that the identity information is present for very few transactions and the majority of transactions have no identity information.* The test_transaction had a total of 506691 rows and 393 columns, test_identity had a total of 141907 data points and 41 rows.

We started by merging train_transaction and train_identity datasets, test_transaction and test_identity datasets using the TransactionID column which was a foreign key in the identity table referencing the transaction table. So, after merging we had two datasets namely train_combined and test_combined

4.2 EDA

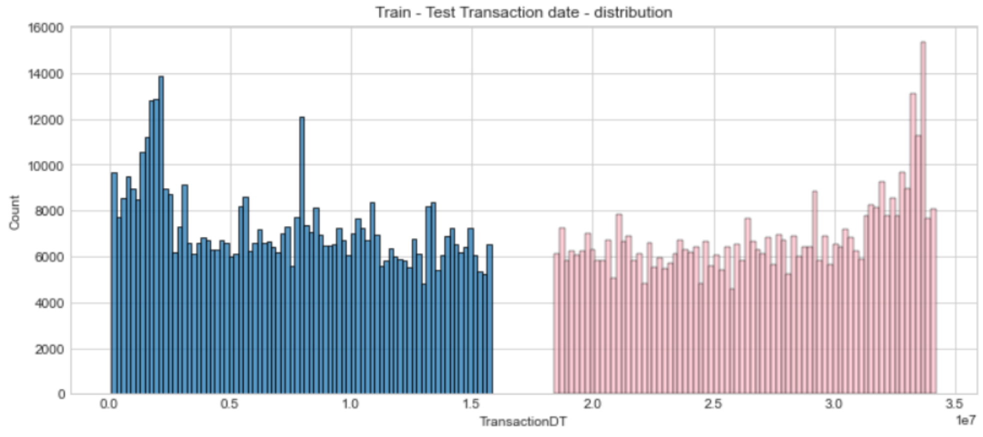
Based on fraud transactions

First, we checked for the number of transactions where data was found fraudulent.



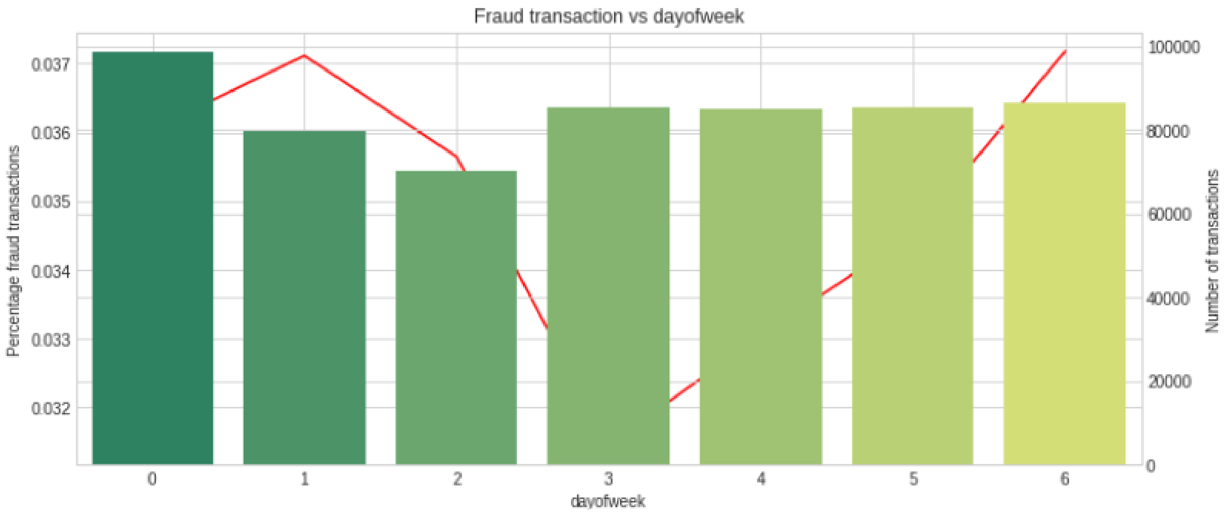
Data is highly imbalanced and since we can see 96.5% of transactions are fraudulent and rest 3.5% are not fraudulent at all. So we chose area under ROC curve for our detection model.

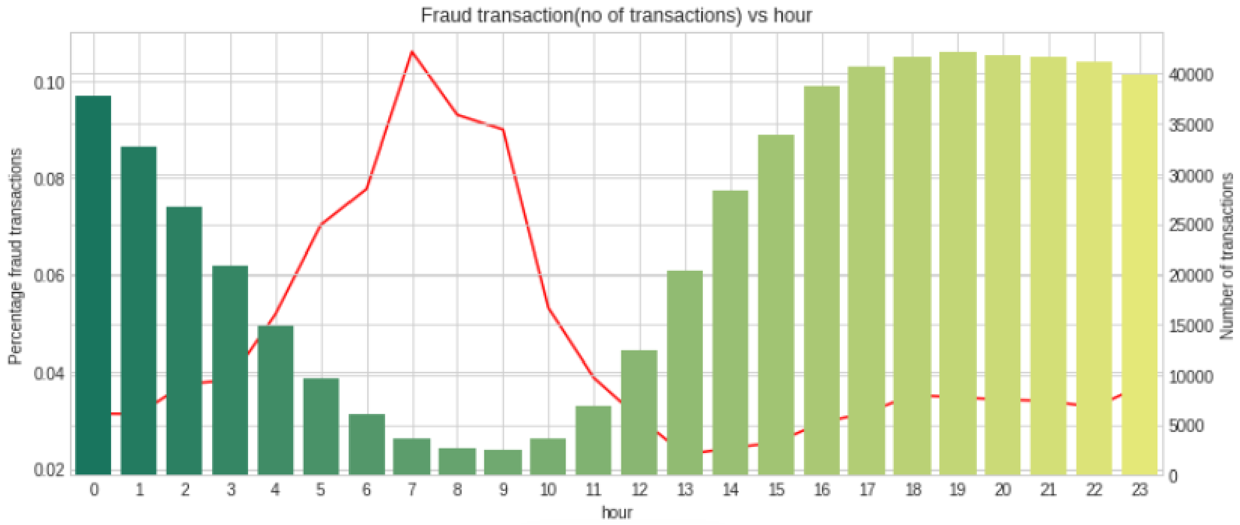
Based on Time



Here we took a time delta reference, which means that the time 86900 is a data at the time interval 86900 seconds after a reference point of time. This shows us that the training data is observed before the testing data. The gap between both the observations is the one month gap between the training data and testing data.

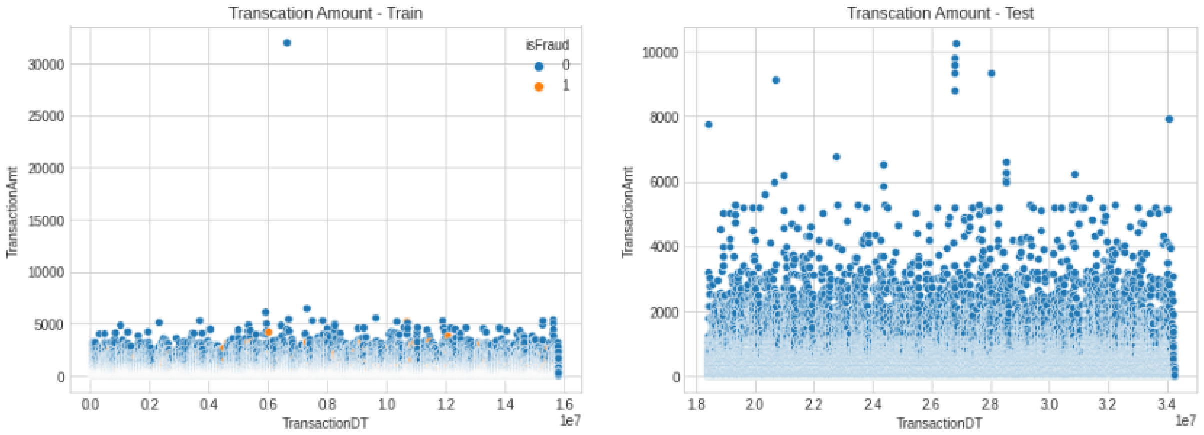
Now let us see fraudulent transactions on different days of the week as well as among different hours.

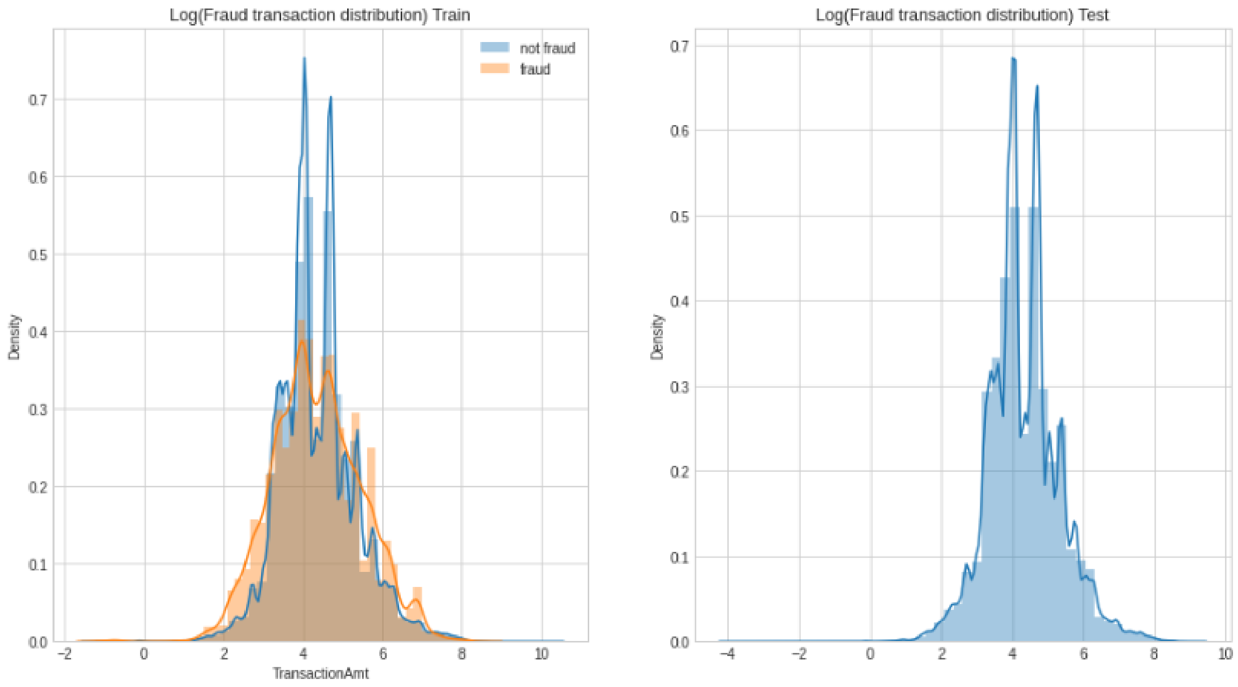




We can see that fraudulent transactions were the highest during the day 0 and 19th hour and least on day 2 and 9th hour.

Based on transaction Amount

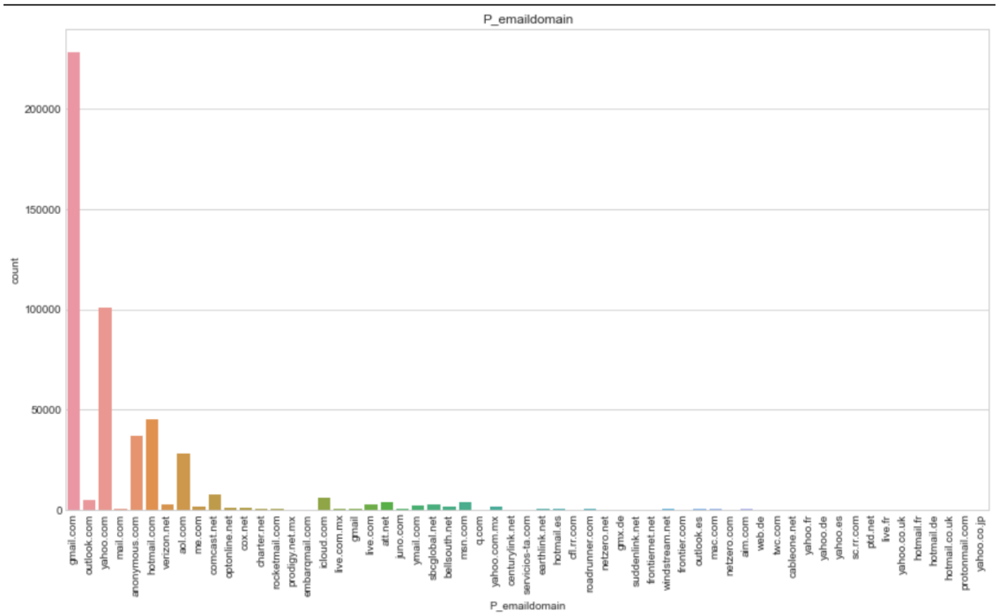


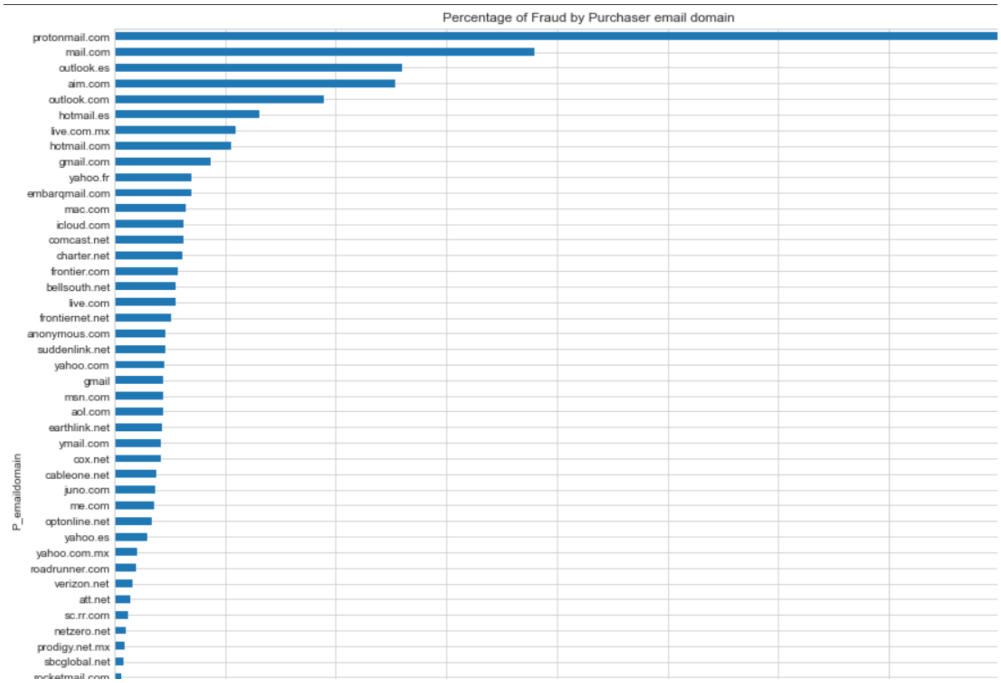


We can observe there are some transactions >30000. We removed this kind of transaction as these behave like outliers and sometimes lead to overfitting of the model.

Also since we have taken the log of amount of the transaction we can say the amount of the transaction whose log falls between 3.3 to 5.5 has a higher chance of being fraudulent

Based on Domain





As we can observe even though the number of transactions which happened with the domain @gmail.com is the highest followed by @outlook and @yahoo, where in terms of fraudulent transactions these domain stays way behind than others. Most of the fraud transactions happened with @patronmail.com or @mail.com.

5.0 Preprocessing

- **Handling of missing values:**
    - Numerical data: Median of the column
    - Categorical data: ‘missing’ as a new category
- **Encoding of Categorical Data:**
    - Used factorize function to for encoding

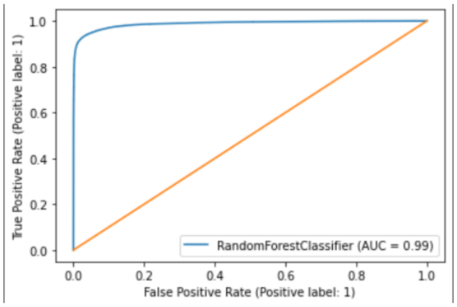
We have checked the rows where the percentage of missing data was quite high. Then we removed all the NaN values, in the case of numerical data, to reduce the influence of outliers we used median instead of mean and in the case of categorical data where the number of unique values was less we used mode meanwhile wherever the number of unique value was high we factorized them and converted the categorical data to numerical one.

6.0 Limitation

The computational requirements of the dataset were quite high because of its large size. We couldn’t use hyperparameter tuning because of the large dataset and the kernel kept on crashing while trying a Randomized grid search. So, we couldn’t perform hyperparameter tuning this time.

7.0 Conclusion

In this project, we were able to see how the dataset we received was unbalanced and we were able to see how that dataset is useless for prediction since if even it predicts all cases as non-fraudulent we get accuracy of 97%. Then we explored the method known as SMOTE for oversampling the dataset which helped us to balance the dataset and predict better for both the categories


True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)
RandomForestClassifier (AUC = 0.99)