Financial Fraud detection for Handling Imbalance Data

The main goal of this project is building analysis model for predicting financial fraud using this synthetic financial fraud dataset from Kaggle. In order to find solution for handling severely imbalanced data in classification models. Financial fraud data is typically very imbalance, as a very low proportion of transaction are actually fraudulent, the project aims at exploring different strategies for handling this imbalance and employing these strategies in building classification models.

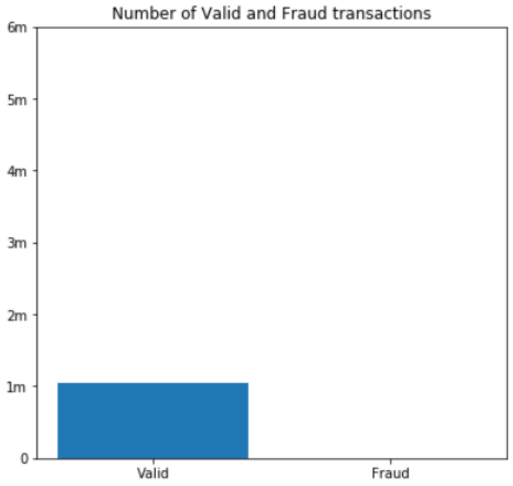
I will employ three strategies for handing the financial fraud detection for handling imbalance data; oversampling and undersampling, I apply these strategies to training data for model, essentially creating two different sets of training data for the model. I will then compare the results from the two different strategies. For each of the two sets of training data for the model.

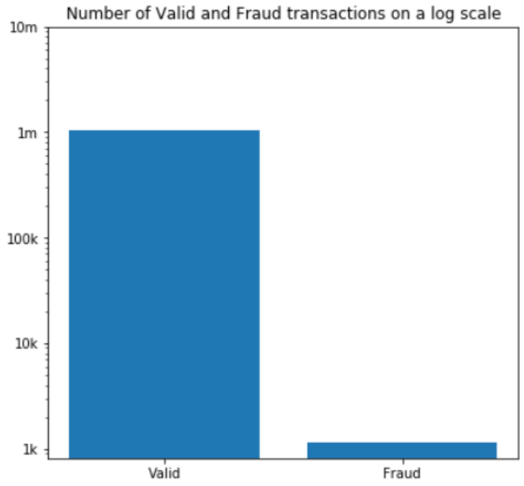
The data includes various fraud details, old balance, new balance detail as well as amount for involved in both positive and negative in the imbalance on financial fraud detection (in this case, the type of the transaction). This project achieves to find similar detail that encompasses a fraudulent in the financial, in for seek out the current feature are relevant enough in order to better spot fraudulent in financial for the management.

Dataset columns that are presented

```
Data columns (total 11 columns):
 #   Column          Non-Null Count    Dtype
---  ------          --------------    -----
 0   step            1048575 non-null  int64
 1   type            1048575 non-null  object
 2   amount          1048575 non-null  float64
 3   nameOrig        1048575 non-null  object
 4   oldbalanceOrg   1048575 non-null  float64
 5   newbalanceOrig  1048575 non-null  float64
 6   nameDest        1048575 non-null  object
 7   oldbalanceDest  1048575 non-null  float64
 8   newbalanceDest  1048575 non-null  float64
 9   isFraud         1048575 non-null  int64
 10  isFlaggedFraud  1048575 non-null  int64
dtypes: float64(5), int64(3), object(3)
```

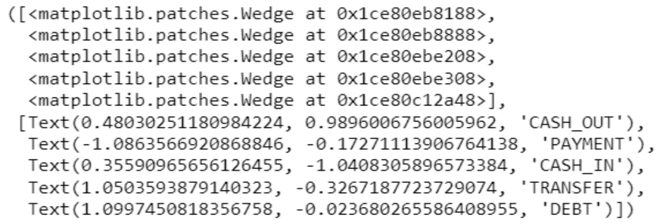
Finding the Number of Valid and Fraud Transaction



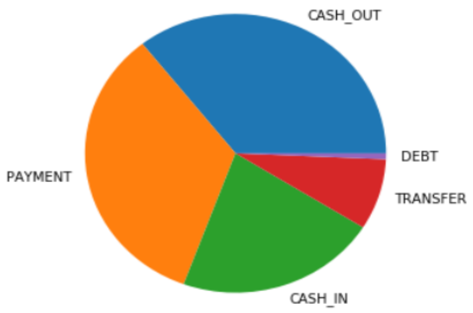


The number of valid and fraudulent transactions in the dataset was low in 1m for valid in the making up of 0.11% the entire dataset in fraud transaction. In this data show that was different imbalance data. This is important to keep in mind when choosing a strategy for building and deploying a KNN classification model.

In this dataset number log scale, the high for in the valid and fraud transaction have less 1K in the fraud transaction this show the fraud imbalance. In this we need to build deploying the KNN classification model on different type of transaction.

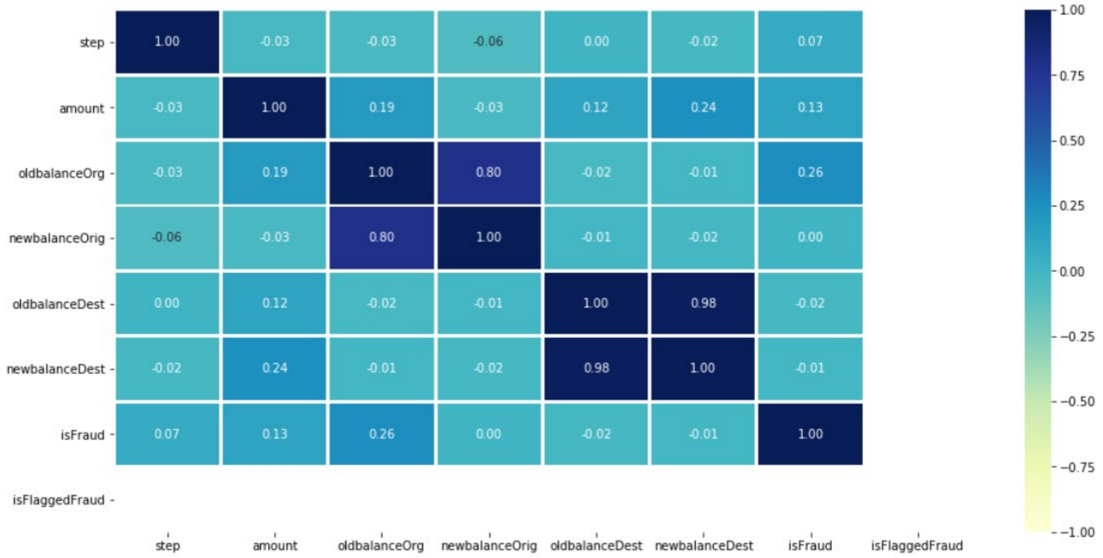


```
([<matplotlib.patches.Wedge at 0x1ce80eb8188>,
  <matplotlib.patches.Wedge at 0x1ce80eb8888>,
  <matplotlib.patches.Wedge at 0x1ce80ebe208>,
  <matplotlib.patches.Wedge at 0x1ce80ebe308>,
  <matplotlib.patches.Wedge at 0x1ce80c12a48>],
 [Text(0.48030251180984224, 0.9896006756005962, 'CASH_OUT'),
  Text(-1.0863566920868846, -0.17271113906764138, 'PAYMENT'),
  Text(0.35590965656126455, -1.0408305896573384, 'CASH_IN'),
  Text(1.0503593879140323, -0.3267187723729074, 'TRANSFER'),
  Text(1.0997450818356758, -0.023680265586408955, 'DEBT')])
```



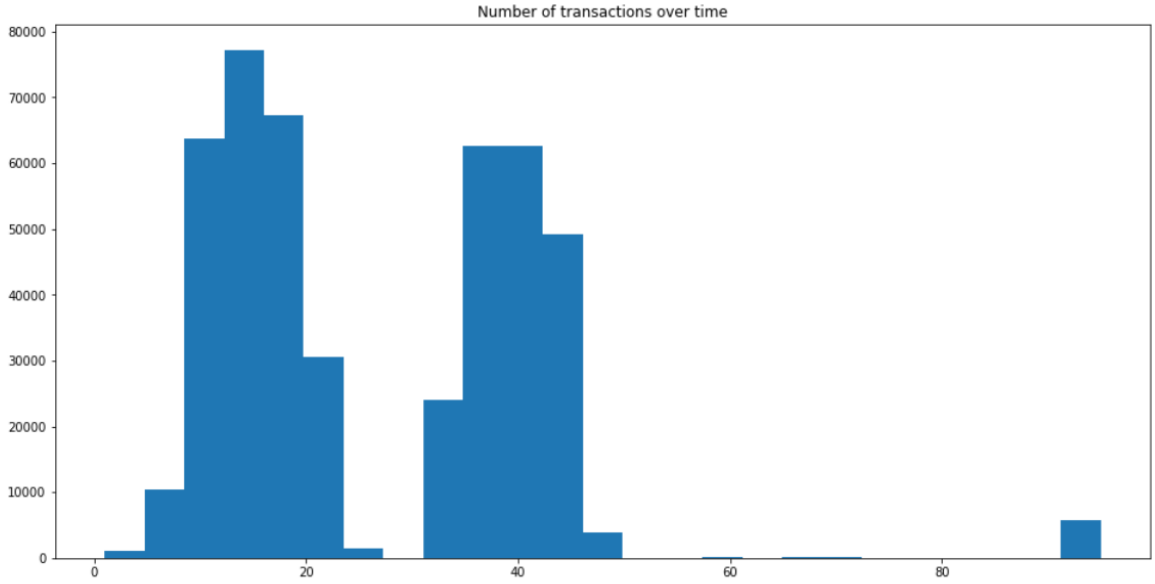
Fraudulent transactions only show up in two of the five transaction types, namely CASH_OUT and TRANSER. There are no fraudulent transactions in PAYMENT, CASH_IN or DEBIT. This means I should subset the data to include only the two transaction types which contain fraudulent transactions. This is done with the assumption that, even though this dataset is a sample of a larger population, fraudulent transaction will only ever appear in these two transaction types.

Finding from Correlation Matrix

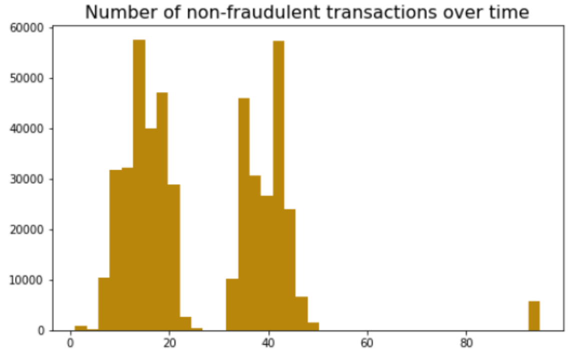


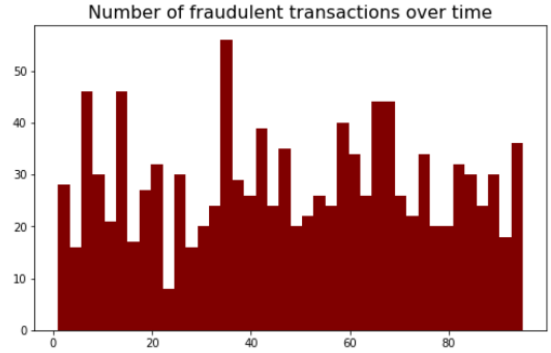
In the correlation matrix is present in between many of the feature available. The highest positive score of 0.80 between the old balance and new balance in the fraudulent transaction to suggest a familiarity trendline in between both features. The next positive score of 0.25 does not amount to much, but it can be assumed there is still a small correlation in between feature isFraud and isFlaggedFraud. As for negative score of -0.50 and -0.75 on the matrix, it seems to suggest an inverse relationship between isFraud and isFlaggedFraud.

Transactions by time step

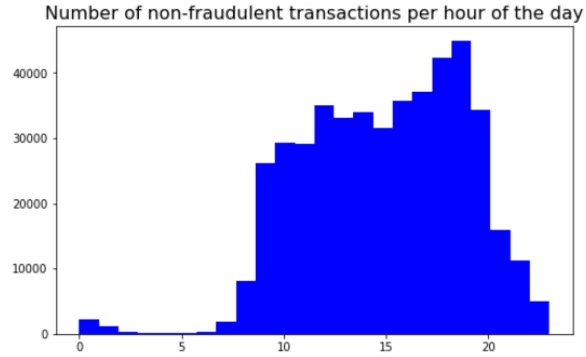


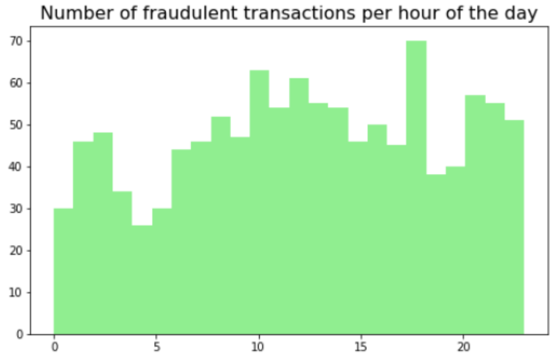
The step feature in dataset gives a timeline to data. Each step is one hours, and the total 95 steps represent one month of transactions (30 days). There anything in particular about the distribution of transactions throughout the month.





The distribution of transactions of transaction overtime is markedly different between fraudulent and non-fraudulent transactions. Fraudulent transactions are happening at an unchanged rate regardless of time, while non-fraudulent transactions have a very uneven distribution. This could make the step feature useful for the modelling.

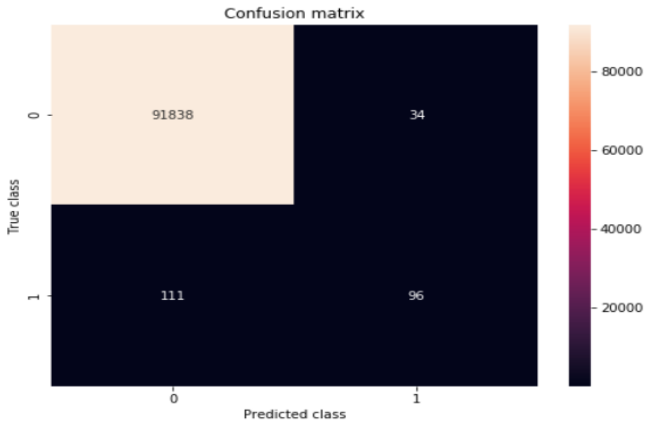




In this timestep the first timestep the first hour of the first day (ie. 0.00 to 01.00), but in the distribution of non-fraudulent transactions, it seems to be likely theory, with few transactions happening the first few hours (assuming this is the night-time), and most of the transactions happening between time step 9 to 24. The distribution of fraudulent transactions is again very different from the non-fraudulent. They are evenly distributed throughout the day.

Model Performance

Model with oversampled data



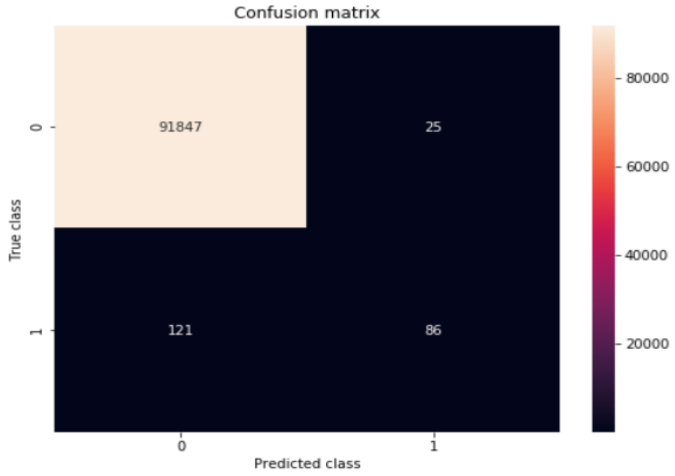
```
The precision is 0.8125
The recall is 0.41935483870967744
The F1-Score is 0.5531914893617021

Training Accuracy with KNN:  0.9983781983521337
Testing Accuracy with KNN:  0.9983579353426181
```

The results of the baseline model on the full oversampled training data, ie with no parameter tuning. In compare the tuned model with this base mode to assess improvement. In the results of baseline oversampled model are quite encouraging. In the training and testing accuracy with KNN is very high, and the number of false negatives and false positives are high in 0.99837 to 0.99835 as not easy to predicate the imbalance for non-fraudulent transactions.

In this KNN was classification for report for the precision score was precision is 0.8125, recall is 0.42, F1-Score is 0.553 was the best parameters for the subset of the oversample data.

Model with undersampled data



```
The precision is 0.7747747747747747
The recall is 0.41545893719806765
The F1-Score is 0.5408805031446542
```

```
Training Accuracy with KNN:  0.9984198308513094
Testing Accuracy with KNN:  0.9984144050217748
```

The results of the baseline undersampled model are not different in the oversampled model. In KNN accuracy in training and testing was higher. The number of false positives is much higher. In defined false negatives as the most important metric, could argue that this model would create too many the false positives to be worth the this can the testing have high accuracy in the imbalance data.

In the using the KNN was classification for report for the precision score was precision is 0.77, recall is 0.42, F1-Score is 0.54 was the best parameters for the subset of the undersample data.

Summary and Finding

In financial fraud predication, what you absolutely want to avoid is classifying a fraudulent transaction and non-fraudulent transaction. As all models performed well on this metric, they all pass the most important test in the using KNN algorithm to address the issue. In doing KNN model was able in training and testing accuracy on the imbalance data.

- Using the KNN KNeighbors classification to predict future for financial fraud detection and handing imbalanced data.
- The KNN model performed the best in this case.
- For further model improvement, we can obtain more information regarding in financial fraud detection in transaction in different ratio which can also added to have higher predication accuracy.