IEEE FRAUD DETECTION

by

Thi Ha Nhi Bui

Capstone Project

Submitted in partial fulfillment of the requirements for the

Master of Science Degree in Business Analytics

Graduate Studies

California State University, East Bay

Hayward, Ca

May 2020

Abstract

The purpose of this paper is to a develop machine learning models to identify potential fraudulent transactions on Vesta's dataset. The project topic is from Kaggle competition. The dataset is obtained from Vesta, the world's leading payment service company. The large-scale dataset contains over 500,000 rows and 430 columns. Each row represents a transaction and each column indicates information about the identity of the customer and transaction information. The goal is to train a model that will take important features in those columns as an input then produce a target result whether that transaction is fraud or not.

Table of Contents

Chapter I: Introduction and Problem Description

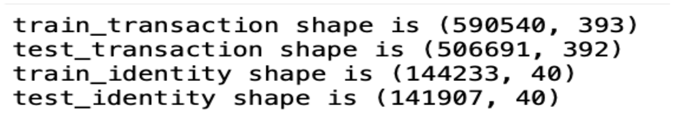
At some point in our lives, we run into the awkward moment of hearing the words, “I’m sorry, but your credit card has been declined.” Dealing with a declined credit card while there is a line of customers behind you is quite embarrassing. With the concern of fraud from your credit card company, there is likely a good reason your card has been flagged or declined. You ask the cashier to attempt again and hoping the outcome is different, but nothing has changed. As you step aside and allow the cashier to tend to the next customer, you respond to your banks text message asking you to confirm your purchase.

Don’t worry, there is some good news that stems from this. Declined transactions means your bank is doing their job. Credit cards are often flagged automatically when possible fraudulent changes are made. Researchers from the IEEE Computational Intelligence Society (IEEE-CIS) want to improve this figure, while also improving the customer experience. Today they’re partnering with the world’s leading payment service company, Vesta Corporation, seeking the best solutions for fraud prevention industry.

In this project, I will benchmark machine learning models on a large-scale dataset obtained from Kaggle. The data will come from Vesta’s real-world e-commerce transactions and will contain a wide range of features from device types to product features. This is a binary classification problem with a heavy imbalance which is an inherent property of such problems. My goal is to develop a highly accurate machine learning model that can detect potential fraudulent transactions, as denoted by the binary target isFraud.

Chapter II: Data Description

This data was extracted from Kaggle's competition. It provides a list of all the transactions from Vesta's customers. There are four separated excel files in the packages namely: train_transaction, test_transaction, train_identity, test_identity.



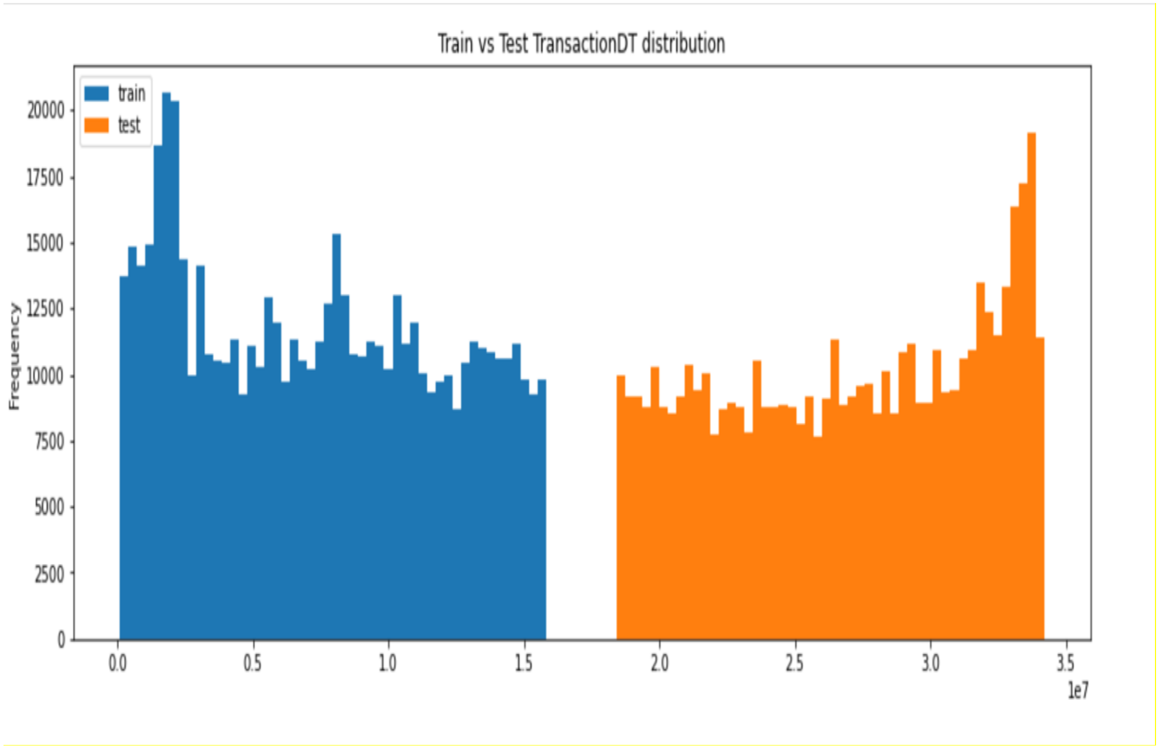
```
train_transaction shape is (590540, 393)
test_transaction shape is (506691, 392)
train_identity shape is (144233, 40)
test_identity shape is (141907, 40)
```

Train_transaction and Test_Transaction provide transaction information. Attributes in the files are defined as the following:

- TransactionDT: timedelta from a given reference datetime (not an actual timestamp).
- TransactionAMT: transaction payment amount in USD.
- ProductCD: product code, the product for each transaction.
- card1 - card6: payment card information, such as card type, card category, issue bank, country, etc.
- addr: register address of the card.
- dist: distance.
- P_ and (R__) emaildomain: purchaser and recipient email domain.
- C1-C14: counting, such as how many addresses are found to be associated with the payment card, etc. The actual meaning is masked.
- D1-D15: timedelta, such as days between previous transaction, etc.
- M1-M9: match, such as names on card and address, etc.
- Vxxx: Vesta engineered rich features, including ranking, counting, and other entity relations.

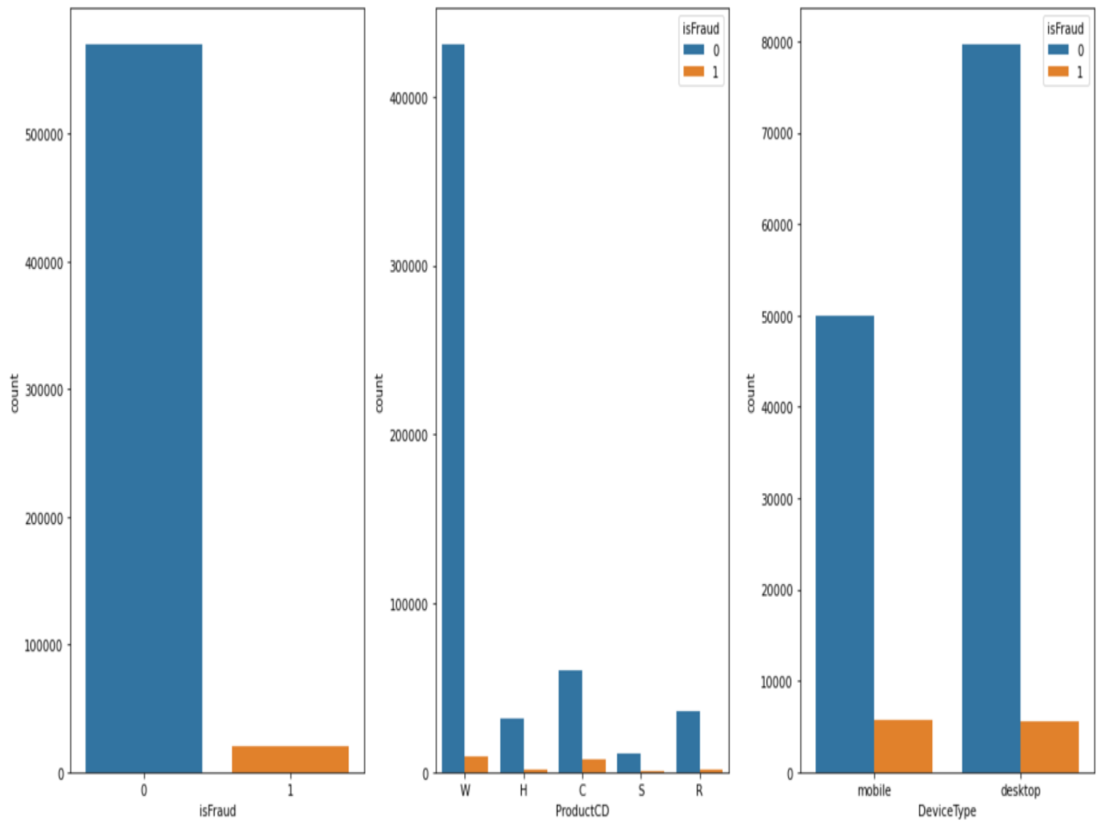
Whereas Train_identity and Test_identity provides information about the identity of the customers (such as network connection information: IP, ISP, Proxy, and digital signature: UA/browser/os/version associated with transactions).

They will be merge into train set (including train_transaction and train_identity) and test set (including: test_transaction and test_identity) by joined transaction_ID. It is important to evaluate whether the train and test set are overlap or not:



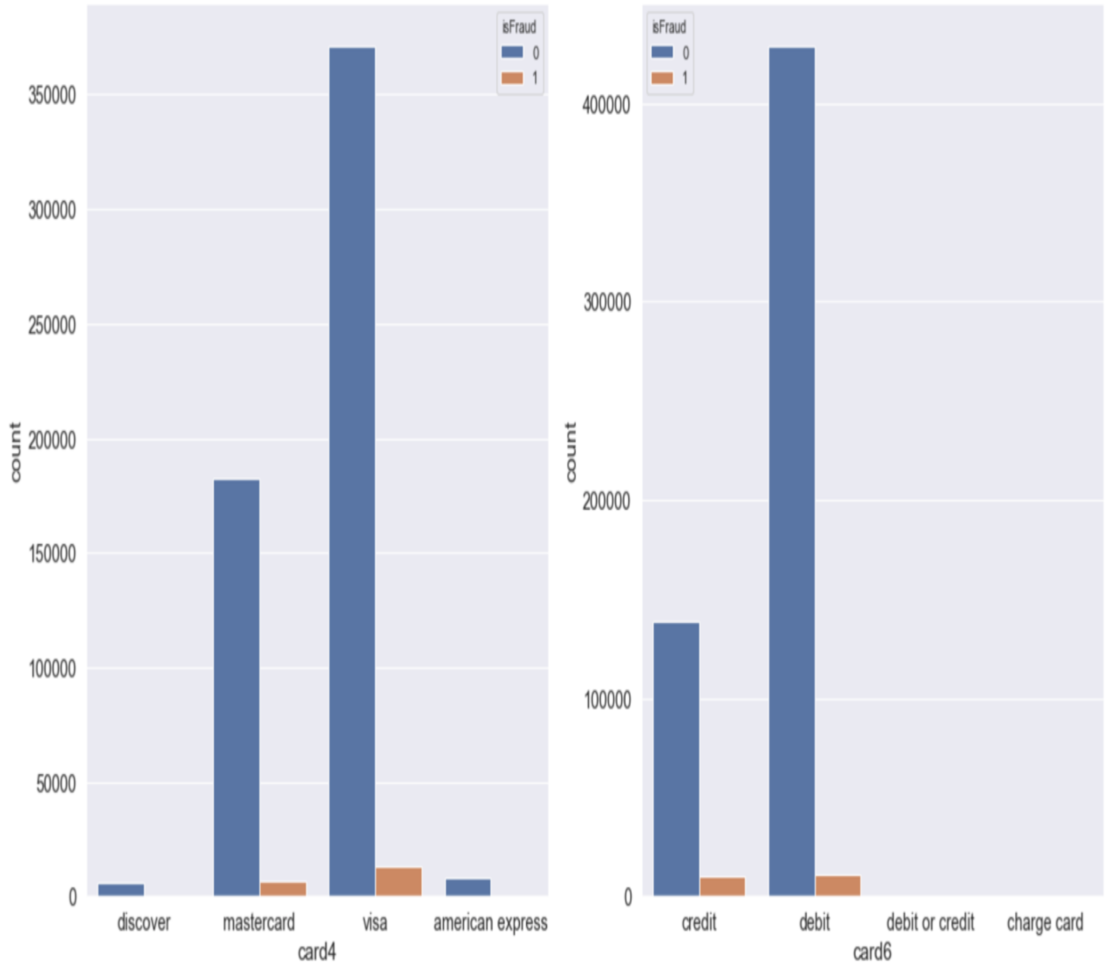
Based on the plot above, the train set and test set is actually time-series dataset. They are slightly separated by a small gap. It is important to split the dataset into training and validation data so that we could validate the accuracy of the model as well as detecting if there would be overfitting or underfitting problems.

In addition, I will visualize some features to understand how the dataset is distributed so I can have a proper approach for the problems. Here below are the plots by target (‘isfraud), by ProductCD and by DeviceType:



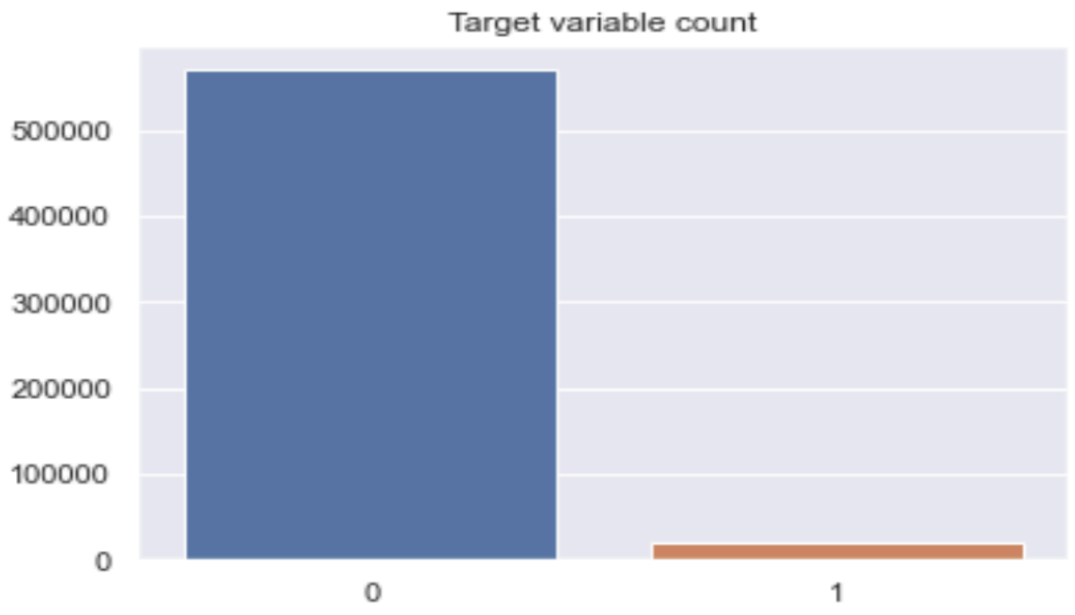
It is clear that the dataset has imbalanced targets where nearly 99% of transaction is detected as No-fraud. It is also interesting to point out that Fraud transactions are detected among different ProductCD(product code). Most transactions belong to product code W. An interesting point to point out is transactions related to product code C are only 1/6 of product W while their fraud transactions are identical to each other. It is also important to note that in the following chart, DeviceType’s, There are 2 types of devices, Mobile and Desktop with Mobile having relatively higher number of frauds reported. Which explains with most of the users using Apps have increased recently and which might have been targeted by the hackers.

In the fourth chart we also examine whether the type of cards has any effect on fraud transaction or not:



The table above shows a majority of transactions are conducted by visa and mastercard. There are only a smaller number of transactions conducted by American Express and Discover. Discover card has higher percentage of fraud transactions reported. Therefore, Discover card users might be a probable targets for the frauds. When we take debit and credit into comparison, Credit card has the highest proportion of fraud followed by Debit Card.

In order to further understand the data, I developed a more appropriate model to analyze the percentage of fraud to non-fraud transactions. Referring to the bar chart below, we had a highly imbalanced dataset where the number of isFraud(0) is overwhleming the number of NoFraud(1). Therefore, If we apply machine learning method before preprocessing data, the result will be heavily affected by the overwhelming one. For further analysis, it is a must to conduct preprocessing and clean the data.



Chapter III: Data Cleaning and Preprocessing

There is no doubt that data cleaning and preprocessing plays a crucial role in analyzing any dataset. This process helps us not only to eliminate noises that can affect the accuracy of the model but also involve the transformation of the raw dataset into an understandable format.

The dataset contains numerous missing values so I check for columns ‘isnull’ and drop all the columns with more than 90% missing data. I also check for unique value and exclude it from the population so that way it does not affect the accuracy of the models. Dividing the training dataset into two (independent variables & target), I then used LabelEncoder() to convert all text data into model-understandable numerical data because we cannot have text in our data if we’re going to run any kind of model on it. By reducing the memory usage, I would lower the risk of any memory errors while working on them as well due to such large datasets. I used a code from Kaggle which had condensed data frames by converting them into smaller units and increase efficiency. Then converting the NAs in the dataset to -999, it would not affect the learning results in the model. As mentioned previously, unbalanced data of fraud and notFraud, the model can produce biased results towards the majority class causing a reduction of classification performance and increasing the number of false negatives. By applying under-sampling method to improve the binary classification performance, we can overcome this issue. Under sampling is used to balance the majority class with the minority class. The main goal is to target potential overlapping characteristics within the collected data sets to reduce the amount of majority data and create an equal distribution of classes. Taking these steps accordingly will clean up the data and now the dataset is ready for use.

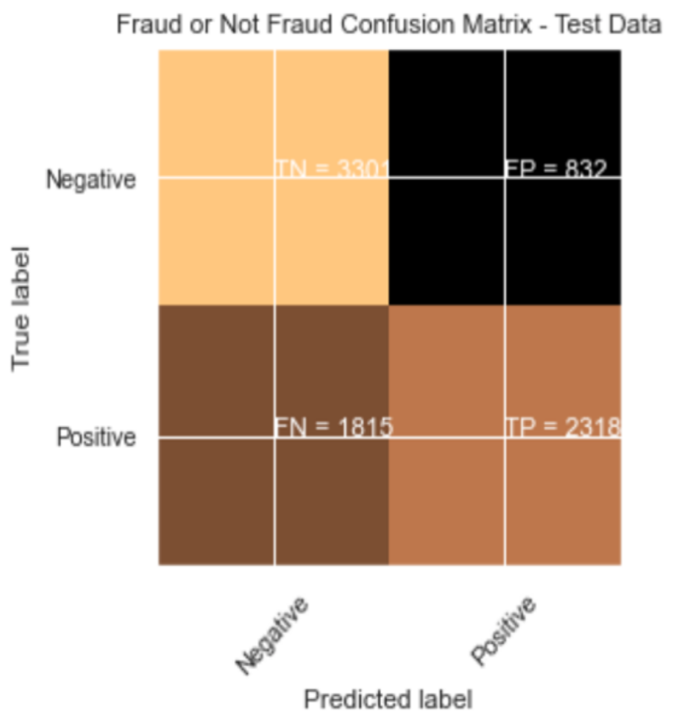
Chapter IV: Models and Analysis

Logistic Regression

To start with, a logistic regression model, the most common type for binary problem, is applied:

```
LogisticRegression(C=1000, class_weight='balanced', dual=False,
                   fit_intercept=True, intercept_scaling=1, l1_ratio=None,
                   max_iter=100, multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=2,
                   warm_start=False)
```

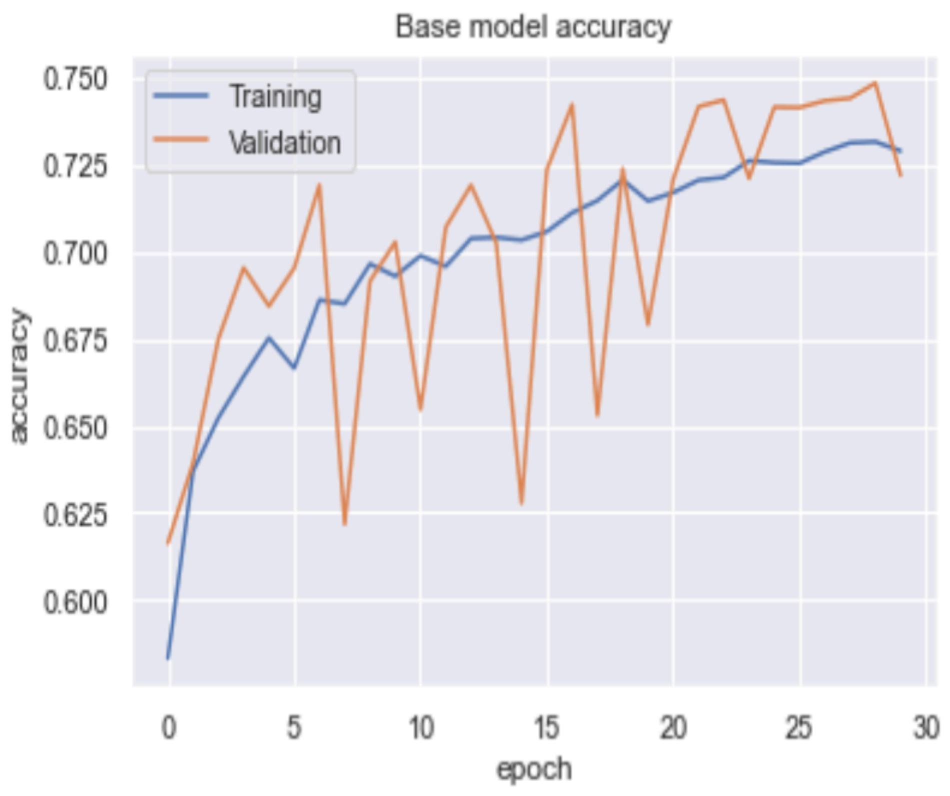
The model had a value of C of 1000, which indicated the inverse of regularization strength. The bigger the value the weaker the regularization. The ‘balance’ class weight mode used the values of y to automatically adjust weights inversely proportional to class frequencies in the input data to even out the imbalance data. The accuracy rate of this model was about 68.7%. The confusion matrix of the model is shown below:



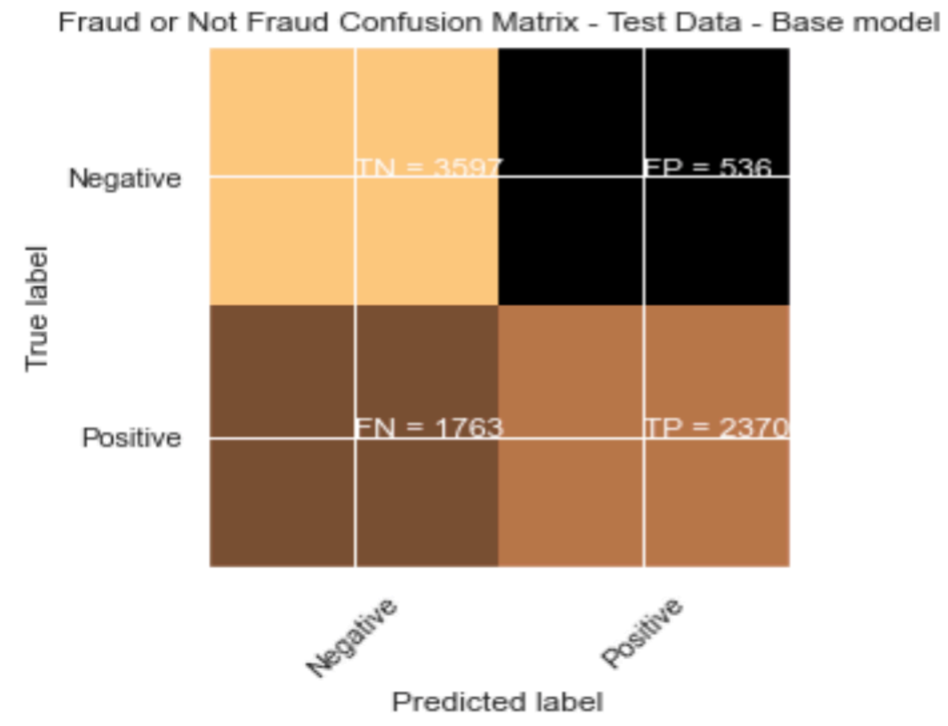
As we can observe, confusion matrix above tells us that we are able to detect the true frauds i.e. 2318 and we missed out of 1815. On the non fraud data, we ended up alerting 832 as Frauds and correctly detected 3301 as non frauds. Which means we have an 56% accuracy of detecting the frauds and 80% of accuracy in detecting non frauds. However, a tradeoff to this was that it mistakenly predicted lots of No Fraud transaction as isFraud. This would result in a high dissatisfaction of customers which might in turn hurt the company revenues. In an ideal situation, we would like to maximize the number of true negatives and minimize the number of false positives.

Neural Network – Base line model: Unweighted Neural Network

The model I will try next is a deep neural network model with 6 layers and each layer except the last one has 128 nodes in it. For loss function, as this is a binary classification problem, I utilize the binary_crossentropy function. For the optimizer, we tested out different optimizers and learning rates using hyperparameter tuning techniques and the adam optimizer had the highest accuracy rate. I want to try a basic model first to observe the results and work on enhancing the model from there. At the first few runs, the accuracy we obtained from this model is approximately 72% which was relatively acceptable. However, the plot of accuracy for training and testing data through each epoch was not ideal, the lines for testing data fluctuates a lot meaning the model performs pretty well on training dataset but it fails to demonstrate such good performance on testing dataset.



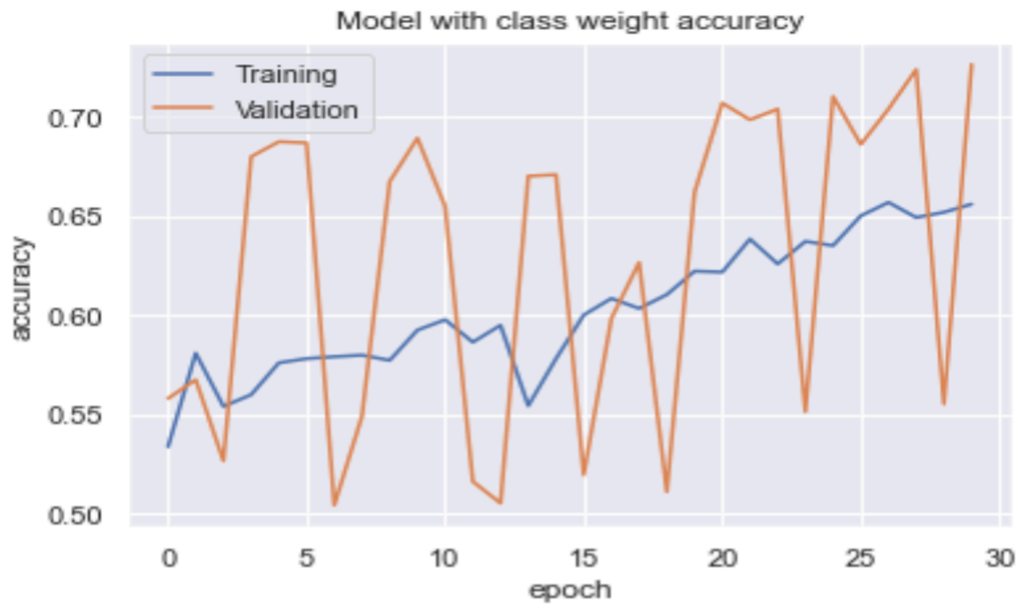
Apart from the accuracy of the model it is also important to know whether we are able to detect frauds or non frauds better. For which we relied on confusion matrix and ROC curves. As we can clearly see from the result chart below, although the model has a rather high accuracy rate, the model just performs relatively low at 57% correctly reporting isFraud(TP) transaction. Therefore, I will continue to perform improvement techniques for the model to boost the accuracy rate. Confusion matrix below tells us that we are able to detect the true frauds i.e. 2370 and we missed out of 1763. On the non fraud data, we ended up alerting 536 as Frauds and correctly detected 3597 as non frauds. Which means we have an 57% accuracy of detecting the frauds and 87% of accuracy in detecting non frauds.



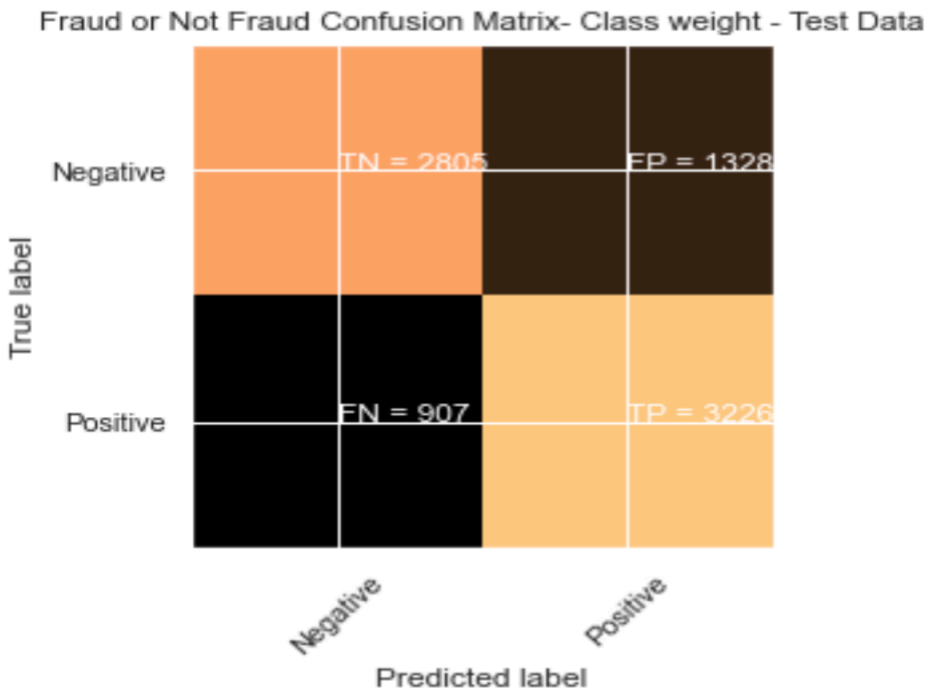
Neural Network – Improved models:

a. Improve model 1: Class weight

Class weights is a technique by which we can assign our priorities and adjust the weights to ensure the preference goes to which side we want. We used class weights and tried to give a preference for detecting Frauds over non frauds using below logic: `,class_weight={0:1, 1:2}`. The accuracy chart is illustrated as below. The accuracy rate of the model is 72.96% which is pretty good. However, the model does not perform well in test set with a lot of fluctuation in test set. Next, The confusion matrix will also be examined.

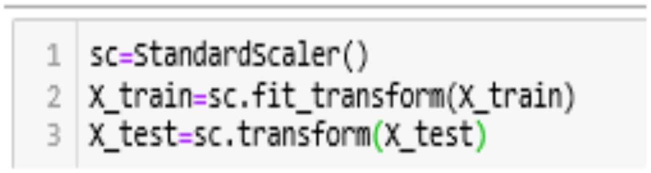


Confusion matrix below tells us that we are able to detect the true frauds i.e. 3226 and we missed out of 907. On the non fraud data, we ended up alerting 1328 as Frauds and correctly detected 2805 as non frauds. Which means we have an 78% accuracy of detecting the frauds and 67% of accuracy in detecting non frauds.



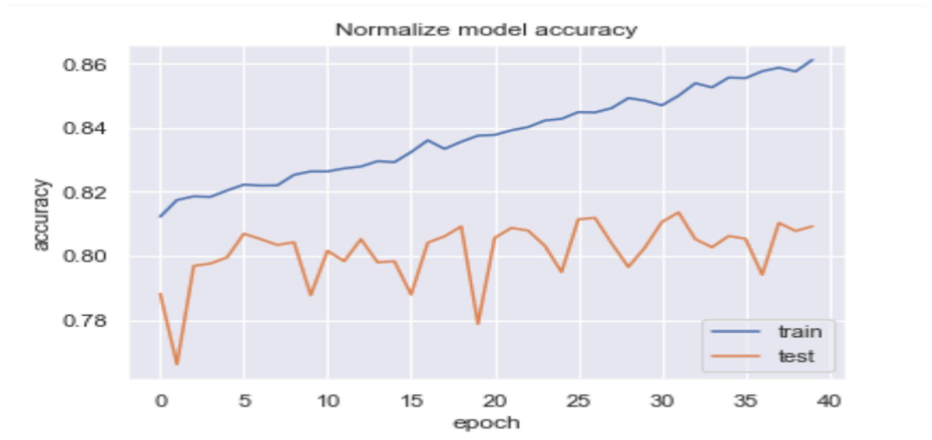
b. Improve model 2: Normalization

The performances of two models above is pretty well. However, the models seem to perform well in train set but still do not perform well in test dataset. Further data preprocessing is still necessary. The remaining columns are on different scales (e.g. hundreds vs hundreds of thousands, etc) which would affect the accuracy of the model, so normalization was employed to improve accuracy and reduce learning time.

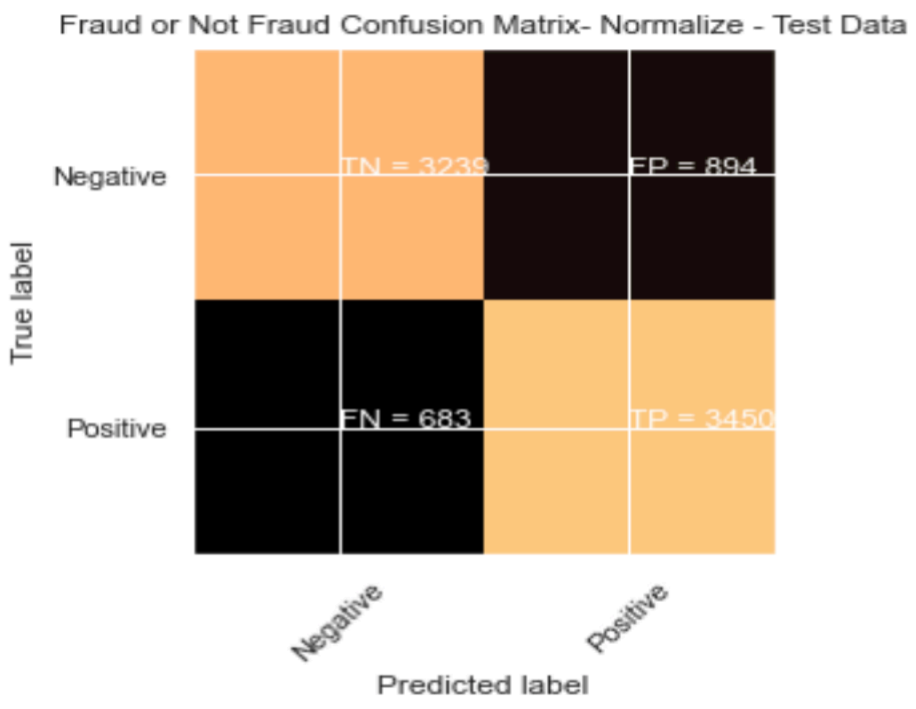


```
sc=StandardScaler()
X_train=sc.fit_transform(X_train)
X_test=sc.transform(X_test)
```

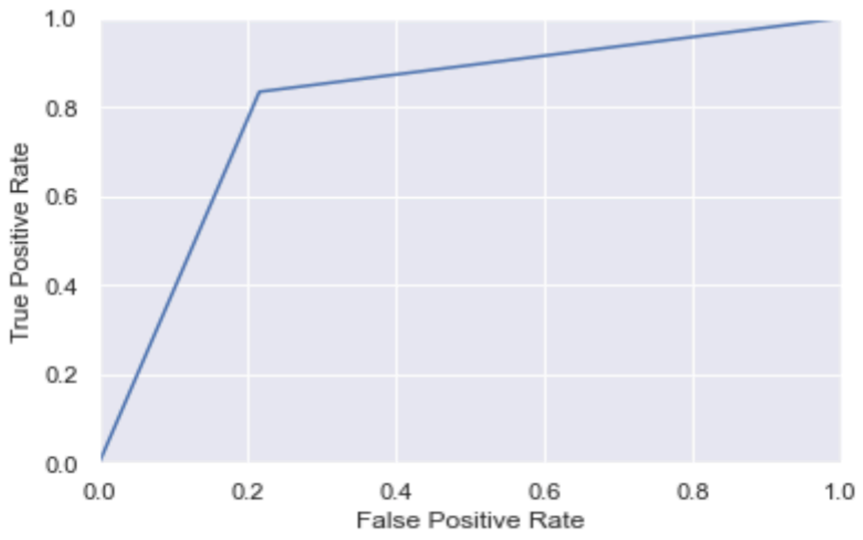
As we can see from the accuracy chart below, the model slightly improves on both training set and test set without too much fluctuations in the test line compared to previous models. The accuracy rate of the model is also very high at around 80%. We have an AUC score of 0.8 which is better and means that we are able to detect the true positive of Frauds which was our intended usage of the model. We would want to capture all the Frauds and we are fine with capturing False Frauds also as Frauds but not vice versa. The accuracy of the model also shows that as the epoch increases we are getting lesser loss:



When it comes to the confusion matrix, the model is able to predict around 83.4% isFraud records. Confusion matrix below tells us that we are able to detect the true frauds i.e. 3450 and we missed out of 683. On the non fraud data, we ended up alerting 894 as Frauds and correctly detected 3293 as non frauds. Which means we have an 83.4% accuracy of detecting the frauds and 76.6% of accuracy in detecting non frauds. This model provides the best performance so far.



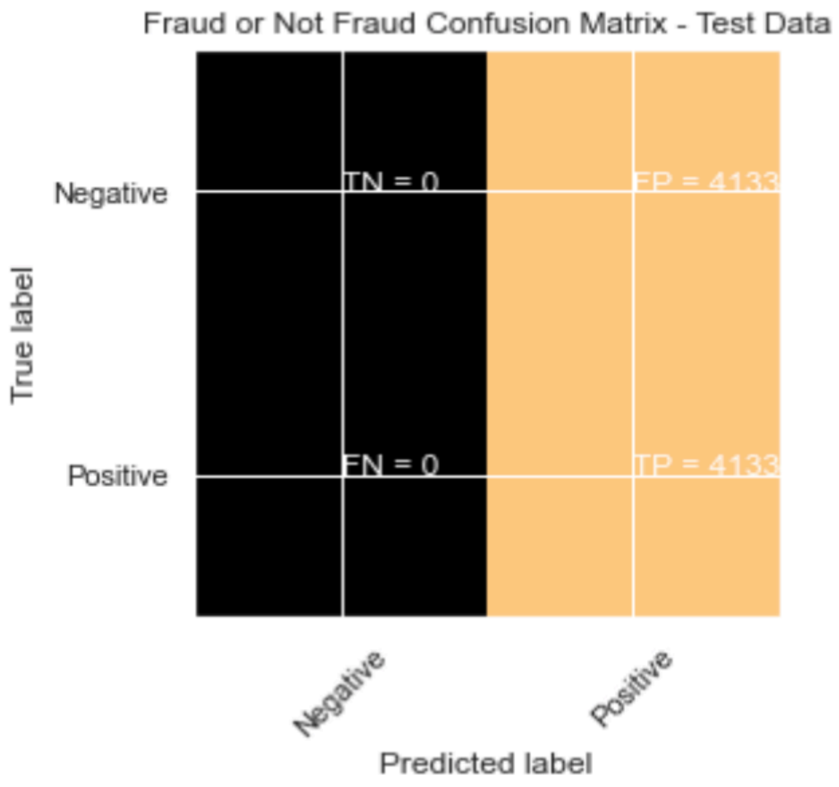
If we consider our purpose to detect the frauds correctly, then we are able to detect : ROC Curve and AOC:



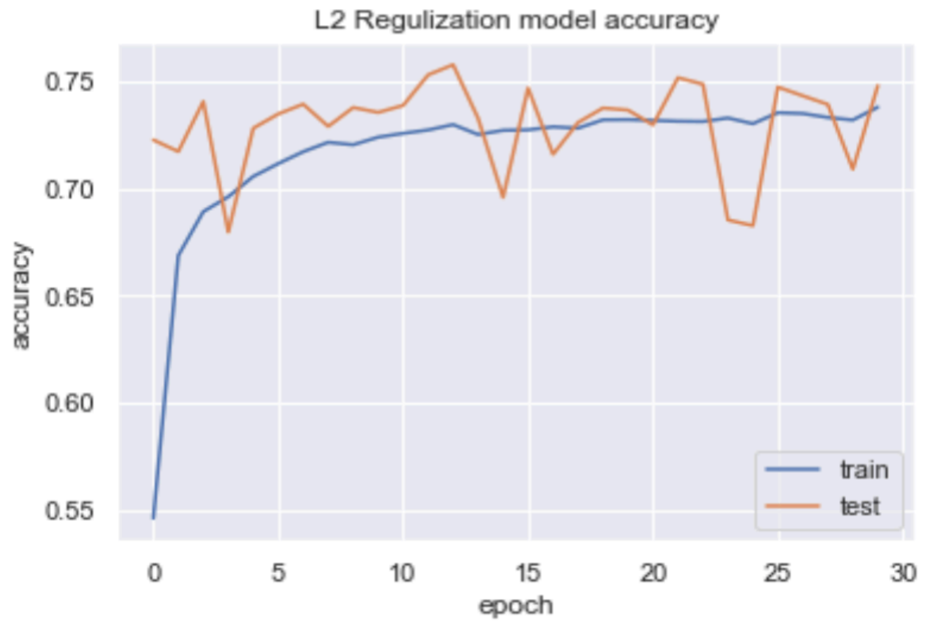
c. Improve model 3: L1&L2 Regularization

Regularization helps in reducing the overfitting errors. We wanted to ensure that if there is any overfitting we would want to remove and tried using Regularization techniques to confirm the same.

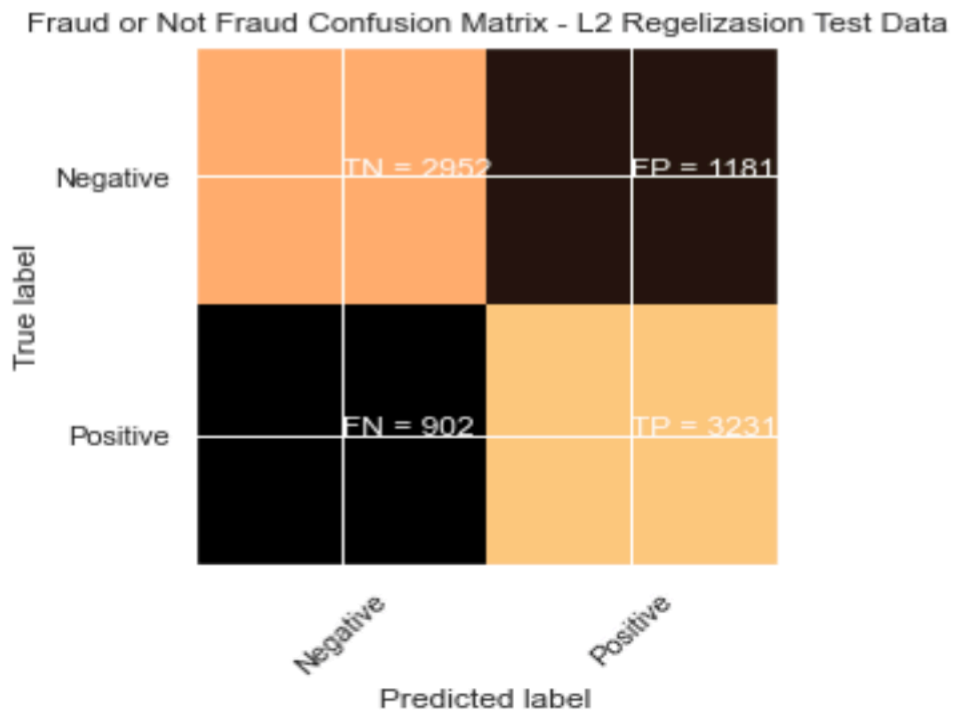
- L1 Regularization: The confusion matrix of L1 model is shown below. The L1 model seems to refuse to learn anything. ROC curve stays between 50:50 which means the model's performance does not get any better when we apply the L1 regularization techniques. Therefore, L1 model is not a good model to apply.



- L2 Regularization provide a better performance since its accuracy rate is 74.8%

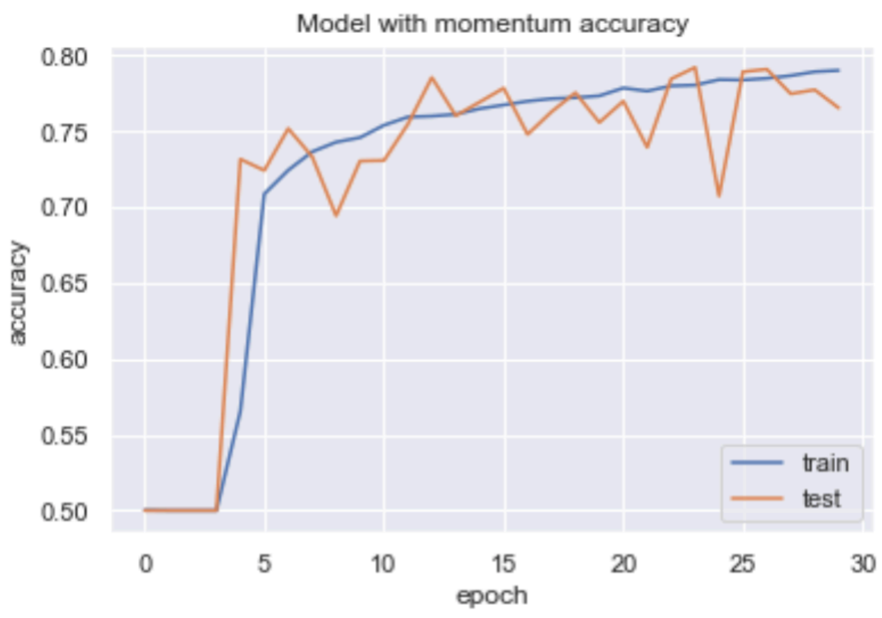


Confusion matrix below tells us that we are able to detect the true frauds i.e. 3231 and we missed out of 902. On the non fraud data, we ended up alerting 1181 as Frauds and correctly detected 2952 as non frauds. Which means we have an 78% accuracy of detecting the frauds and 71% of accuracy in detecting non frauds.

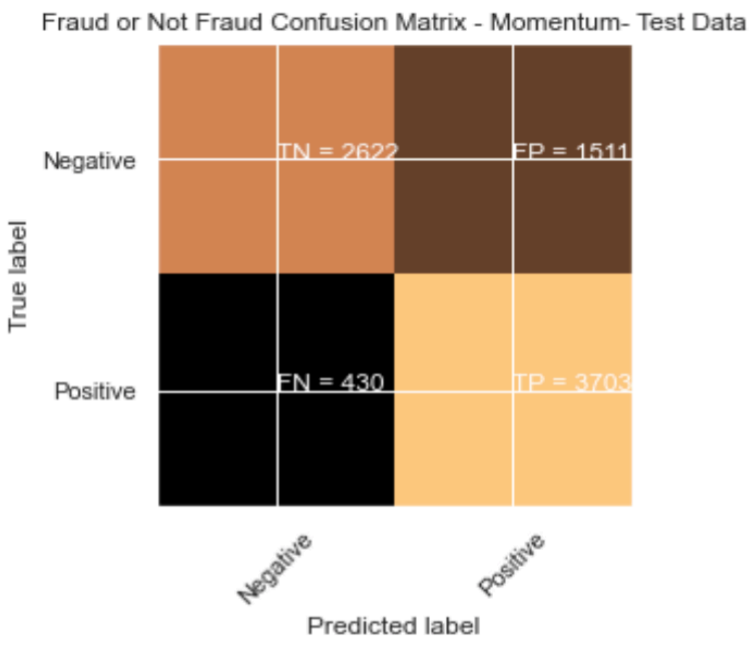


d. Improve model 4: Momentum

We wanted to check if there was a problem with local minima and tried to use momentum to ensure it accelerates SGD in the relevant direction and we have updated it in our logic using SGD. Here below is the result. The model has an accuracy rate of 0.76.

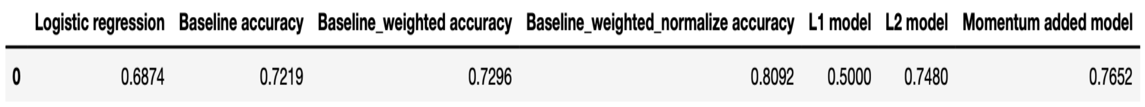


As regard to the confusion matrix: Confusion matrix below tells us that we are able to detect the true frauds i.e. 3703 and we missed out of 430. On the non fraud data, we ended up alerting 1511 as Frauds and correctly detected 2622 as non frauds. Which means we have an 89% accuracy of detecting the frauds and 63% of accuracy in detecting non frauds.



Chapter V: Comparisons and Results

In comparision, the Neural Network with class weight, normalization and momentum model provides a better performance in predicting actually Fraud in this data set than the baseline model as you can see we have detected more frauds than in the baseline model i.e 3703 reocords. The accuracy of this model would be 89% accuracy for detection of frauds and 63% of detection of non frauds. We have an AUC score of approxiamately 0.77 which is better and means that we are able to detect the true positive of Frauds which was our intended usage of the model. We would want to capture all the Frauds and we are fine with capturing False Frauds also as Frauds but not vice versa.

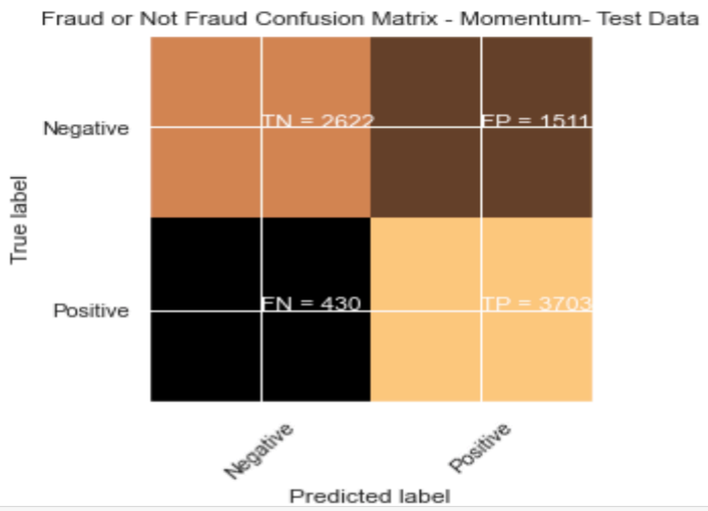


	Logistic regression	Baseline accuracy	Baseline_weighted accuracy	Baseline_weighted_normalize accuracy	L1 model	L2 model	Momentum added model
0	0.6874	0.7219	0.7296	0.8092	0.5000	0.7480	0.7652

Models	Accuracy rate	True Positive (correctly predict is Fraud)	True Negative (correctly predict Non fraud)
Logistic Regression	68.74%	56%	80%
NN- base line	72.19%	57%	87%
Weighted NN	72.96%	78%	67%
Weighted_Normalize NN	**80.92%**	**84%**	**77%**
L1 Regularization	50%	1	0
L2 Regularization	74.8%	78%	71%
Weighted_Normalize Momentum	**76.52%**	**89%**	**63%**

Chapter VI: Conclusions

In conclusion, the Neural Network with class weight, normalization and momentum model has been the best model so far with the below confusion metrics:



```
sgdmm_score=roc_auc_score( y_test, y_predbaselinesgdmm)
sgdmm_score
0.7651826760222599
```

With this we should be able to detect with 89% accuracy on fraud detection which is very valuable for the customer to ensure they mitigate the frauds and help them in their mission of better customer experience. This fraud detection applies to any organization which deals with the financial transactions. As we saw in this data we saw that fraud applies to mobile devices and accounts associated with the customers who use their credit card. With these kind of hints from exploration business can do probable predictions. Even though the Neural networks is a kind of black box which doesn't give us the coefficients of various variables involved in the transaction however as we have seen in this project it helps us in deriving the proper model which can be used in the real word and we are able to validate the data which gives us confidence to implement them in large enterprises and to ensure they achieve their customer satisfaction.

References

https://www.kaggle.com/c/ieee-fraud-detection