Contents

Fraud Detection on Bank Payments

K.Lalith Reddy

September 2021

1 Introduction

Now a days there are lots of ways coming up to cheat, Bank fraud is one of the top heard things, and this behaviour can be seen in different fields. Bank fraud is to obtain money from depositors by wrongly posing as a bank or other financial institutions. So i want to predict what percent of fraud is happening in different fields.

2 Data set

The data-set has 7 feature [1] columns and a target column. The feature columns are :

1. Customer: This feature represents the customer id

2. Age: Categorized age

0: less-than or = 18,

1: 19-25,

2: 26-35,

3: 36-45,

4: 46:55,

5: 56:65,

6: greater-than 65

U: Unknown

3.Gender: Gender for customer

E : Enterprise,

F: Female,

M: Male,

U: Unknown

5. Category: Category of the purchase. There are 15 different categories.

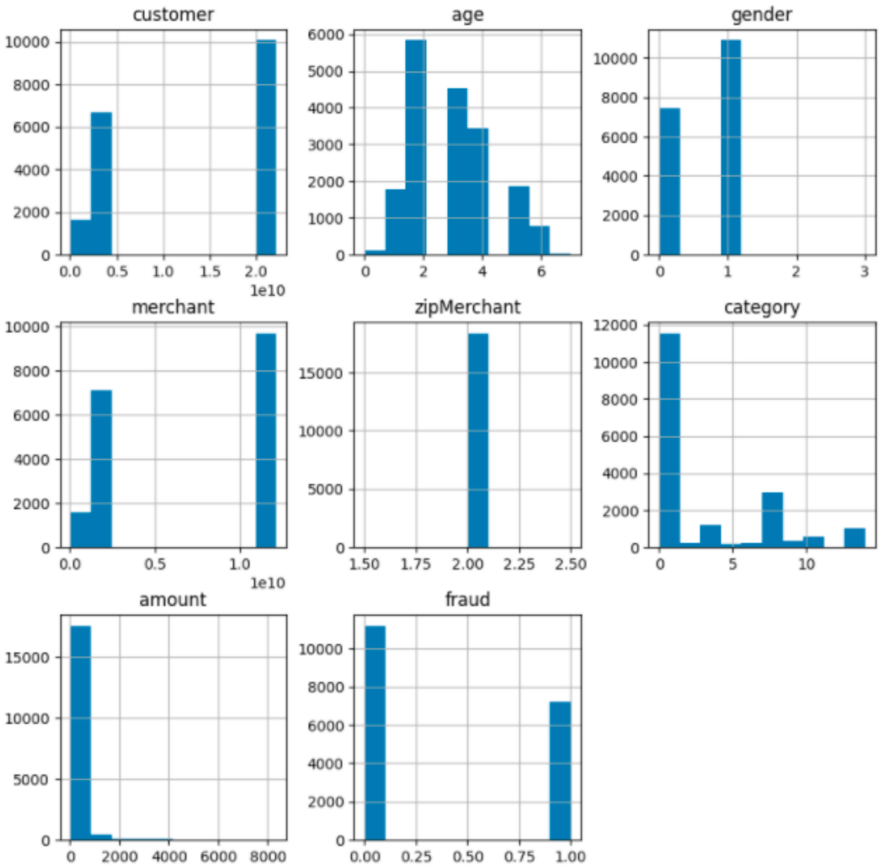
6. Amount: Amount of the purchase

7. Fraud: Target variable which shows if the transaction fraudulent(1) or benign(0)

3 Data Pre-processing

3.1 Data Visualization

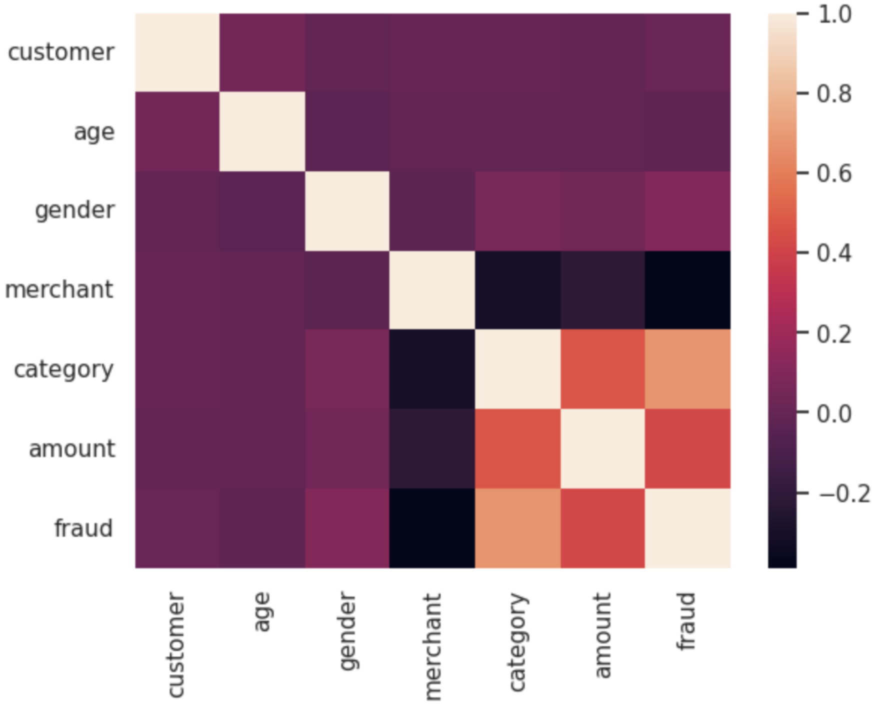
It is a graphical illustration of information and data.By using different elements like charts, graphs, and maps, it makes easier for human brain to understand and pull insights from. Because of this Visualization we can easily find patterns, trends and statistics in big data-sets and it is the effective way to show data. The below images show the representation of data visually between each factor.



Input Visualization

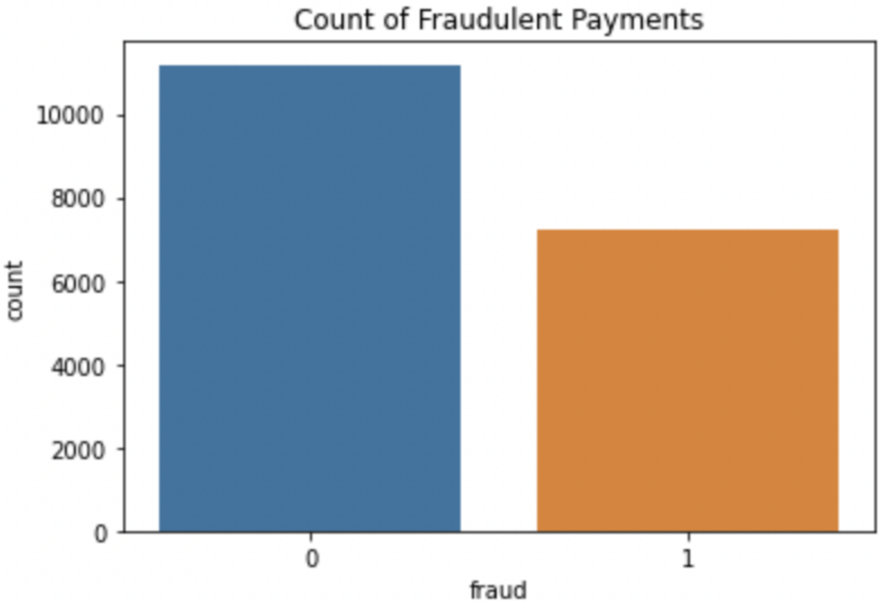
3.2 Correlation Between features

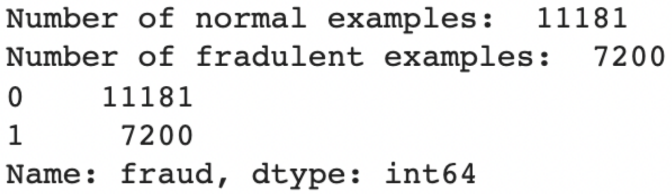
The co-relation between the features are more impotent to feature impotence as we can see category v/s fraud and amount v/s category are more important than other.



3.3 Count of output variable

The output variable are as shown in the figure it shows count of fraudulent data. The normal examples are represented as 0 and fraudulent examples are represented as 1.
Number of normal examples: 11181
Number of fraudulent examples: 7200.

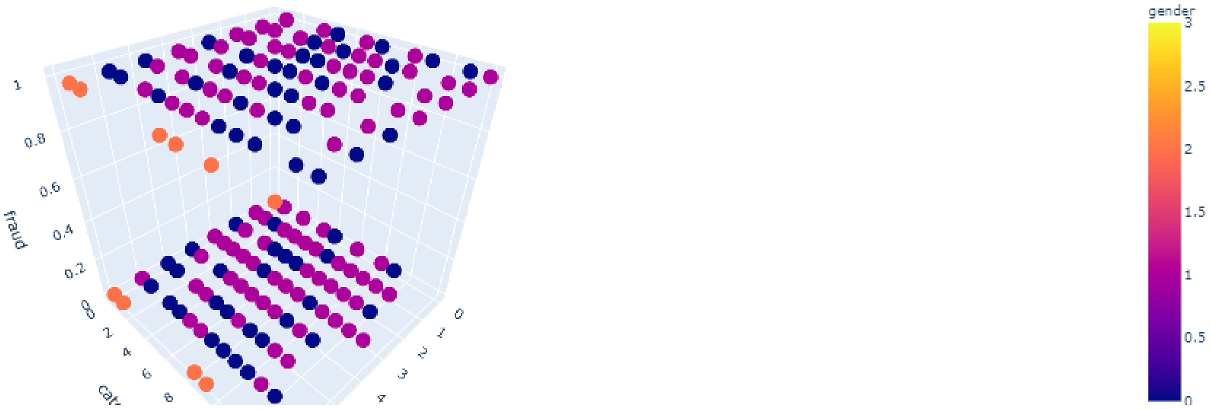




```
Number of normal examples:  11181
Number of fradulent examples:  7200
0    11181
1     7200
Name: fraud, dtype: int64
```

3.4 Data Analysis

Data Analysis is also called as data cleaning. This Analysis is used for inspecting, cleaning, transforming and modelling and also to find use full information from the data. It is the process of deleting, corrupted, inappropriate data and also to remove duplicate data. However data-cleaning strategies change depending on types of data we use.



DATA VISUALIZATION

3.5 Data Normalization

Next step will be Data Normalization and it plays a crucial role. As the data was not distributed uniformly. so we have to process the data by applying normalization techniques. So, we have to pre-process the data applying normalization techniques. Normalization makes the optimization problem more numerically stable and makes training less sensitive to the scale of features. When we normalize the data , all the values are scaled between ”0 and 1”, and the outliers will be eliminated from data-set, However they remain visible within our normalized data.

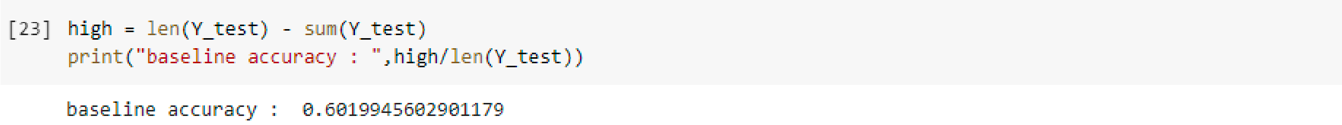
4 Data and Network Modelling

4.1 Logical Regression

Logistic regression is nothing but it is a Statistical analysis model which is used to prediction of a data value on the bases of previous observation that are made on data set. In this project i have used neural network which is one layer with an epochs(training the neural network with all the training data for one cycle)[2] of 256 where we can change to different numbers and that should be in the two to the power times(2power = 256, 512, 1024). ”An epoch means training the neural network with all the training data for one cycle. In an epoch, we use all of the data exactly once. A forward pass and a backward pass together are counted as one pass”.

4.2 Baseline accuracy

The Baseline Accuracy of my algorithm is 60 percent which is very low. It is shown in Try Binary classification.

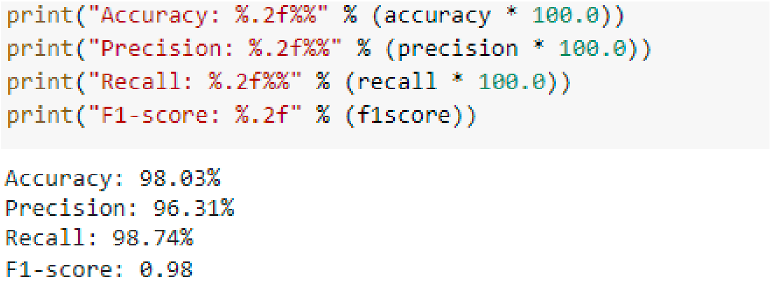


```
[23] high = len(Y_test) - sum(Y_test)
     print("baseline accuracy : ",high/len(Y_test))

     baseline accuracy :  0.6019945602901179
```

Baseline accuracy

4.3 Binary Classification (with Over-fitting)

Binary Classification is used to classify the elements of a set into two different groups on the bases of classification rule. The things i did in binary classification are first i have used random function to randomise the data and split the data into X and Y which contains input and output variables then normalizes the data. Created a neural network with three layers (8, 4, 1) and an epochs of 256 and the accuracy was 96 percent, after increasing the epochs to 1024 the accuracy is increased to 97 and got the accuracy of 97.5. This is one trial to over fit the model then changed the neural network into four layers (10, 8, 4, 1) and increased the epochs to 1024 it is increased to **98.03** percent, i have made lots of changes increasing and decreasing epochs and increasing neural network layer, i have tried with many different combinations but the accuracy is still in the same 98.3 percent. **This process is to over fit the model**

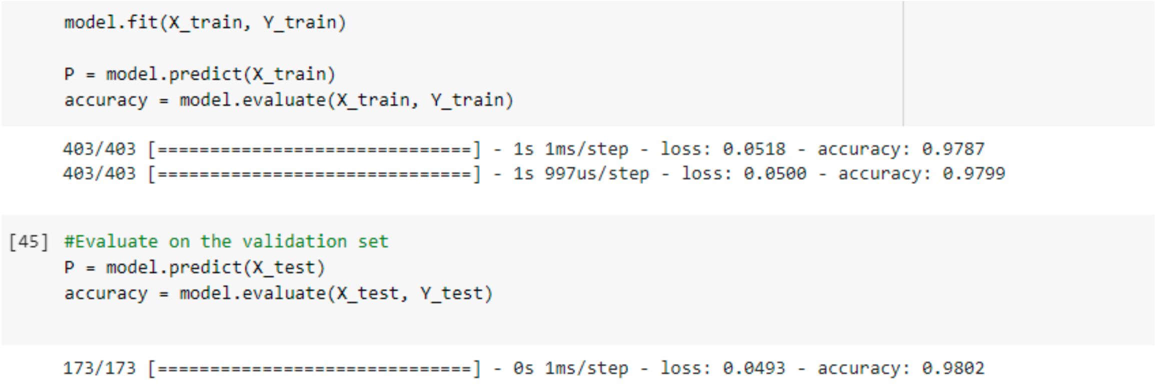


```
print("Accuracy: %.2f%%" % (accuracy * 100.0))
print("Precision: %.2f%%" % (precision * 100.0))
print("Recall: %.2f%%" % (recall * 100.0))
print("F1-score: %.2f" % (f1score))

Accuracy: 98.03%
Precision: 96.31%
Recall: 98.74%
F1-score: 0.98
```

Binary Classification Accuracy

4.4 Splitting Data-set

In this I have split the data-set in the 30:70 ratio, 30 percent of my data will go into test and the remaining data into train. after validating the data the output accuracy is 97.5 percentage.



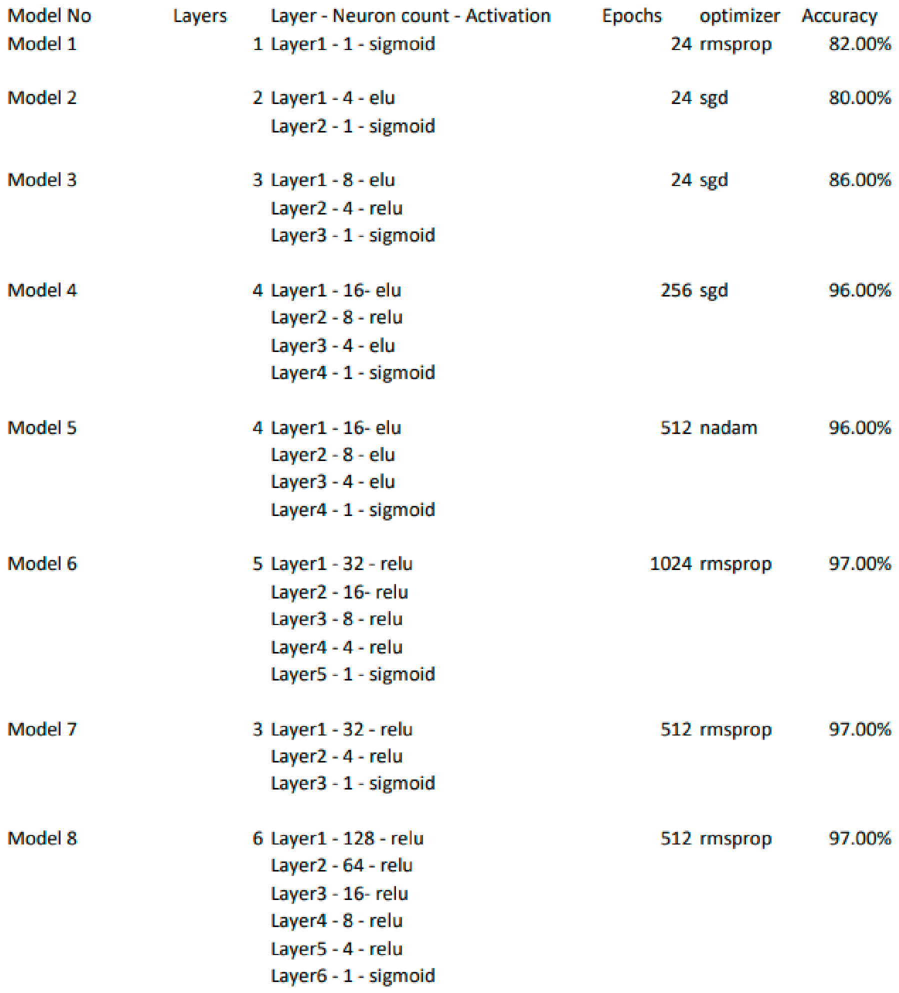
```
     model.fit(X_train, Y_train)

     P = model.predict(X_train)
     accuracy = model.evaluate(X_train, Y_train)

     403/403 [==============================] - 1s 1ms/step - loss: 0.0518 - accuracy: 0.9787
     403/403 [==============================] - 1s 997us/step - loss: 0.0500 - accuracy: 0.9799

[45] #Evaluate on the validation set
     P = model.predict(X_test)
     accuracy = model.evaluate(X_test, Y_test)

     173/173 [==============================] - 0s 1ms/step - loss: 0.0493 - accuracy: 0.9802
```

Accuracy

5 Finding the best model

After splitting the data-set i have tried different models with different combinations of neural layers with changing and adding layers and changing different epochs, and changing different optimize-rs and activation functions. The below table shows the different models with different layers, epochs, optimizer and accuracy.

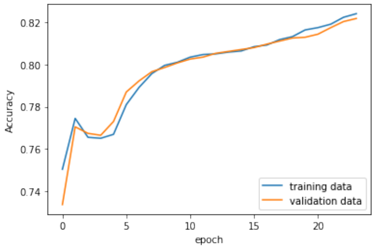
5.1 Models with different Neural Networks



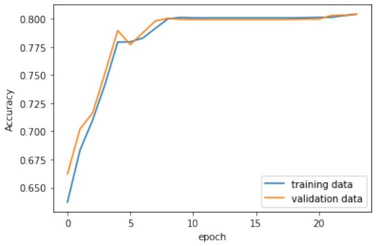
Model No	Layers	Layer - Neuron count - Activation	Epochs	optimizer	Accuracy
Model 1	1	Layer1 - 1 - sigmoid	24	rmsprop	82.00%
Model 2	2	Layer1 - 4 - elu Layer2 - 1 - sigmoid	24	sgd	80.00%
Model 3	3	Layer1 - 8 - elu Layer2 - 4 - relu Layer3 - 1 - sigmoid	24	sgd	86.00%
Model 4	4	Layer1 - 16- elu Layer2 - 8 - relu Layer3 - 4 - elu Layer4 - 1 - sigmoid	256	sgd	96.00%
Model 5	4	Layer1 - 16- elu Layer2 - 8 - elu Layer3 - 4 - elu Layer4 - 1 - sigmoid	512	nadam	96.00%
Model 6	5	Layer1 - 32 - relu Layer2 - 16- relu Layer3 - 8 - relu Layer4 - 4 - relu Layer5 - 1 - sigmoid	1024	rmsprop	97.00%
Model 7	3	Layer1 - 32 - relu Layer2 - 4 - relu Layer3 - 1 - sigmoid	512	rmsprop	97.00%
Model 8	6	Layer1 - 128 - relu Layer2 - 64 - relu Layer3 - 16- relu Layer4 - 8 - relu Layer5 - 4 - relu Layer6 - 1 - sigmoid	512	rmsprop	97.00%

Performance summary of different models

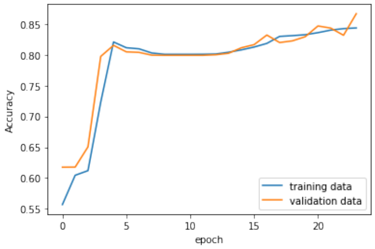
5.2 Learning Curves for Neural Network Models



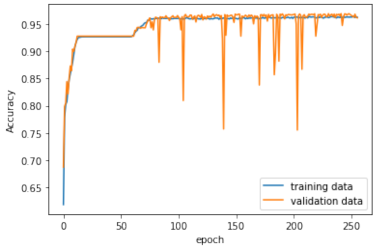
Model1



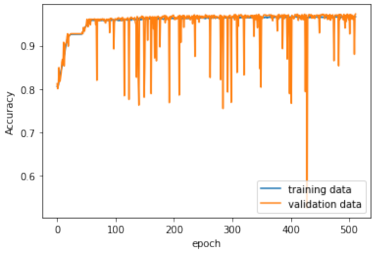
Model 2



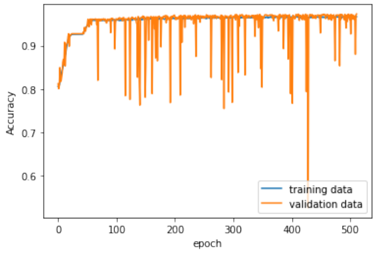
Model3



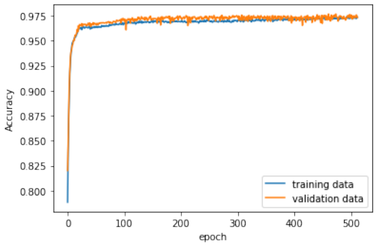
Model 4



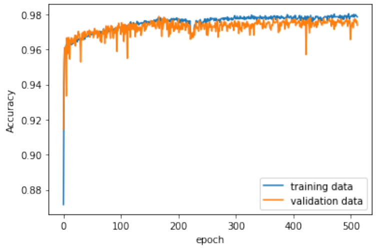
Model5



Model 6



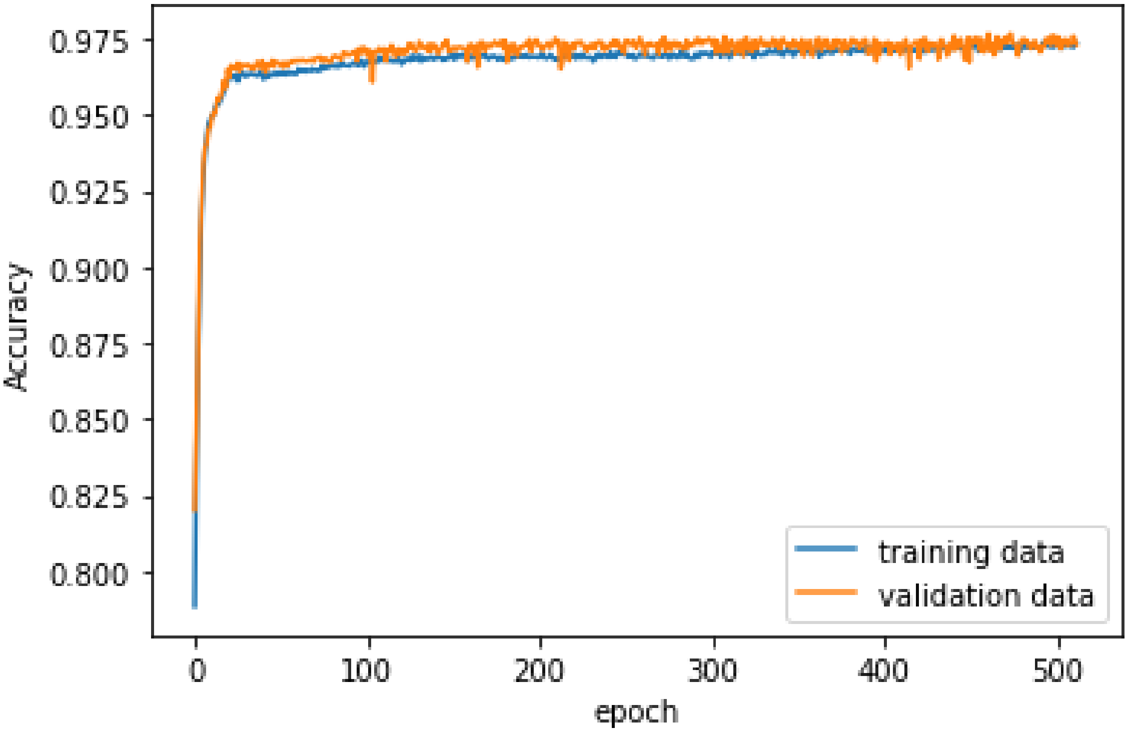
Model7



Model 8

5.3 Best Neural Network Architecture(still working on this)

My best Neural network architecture is my model 7(i am still working on this) this is the best model because it has few layers of neurons and still able to give the accuracy of 97 present with the best learning curve as shown below. Yes this accuracy is very high but i am figuring it out where the mistake is and i will rectify this in next phase.



Accuracy

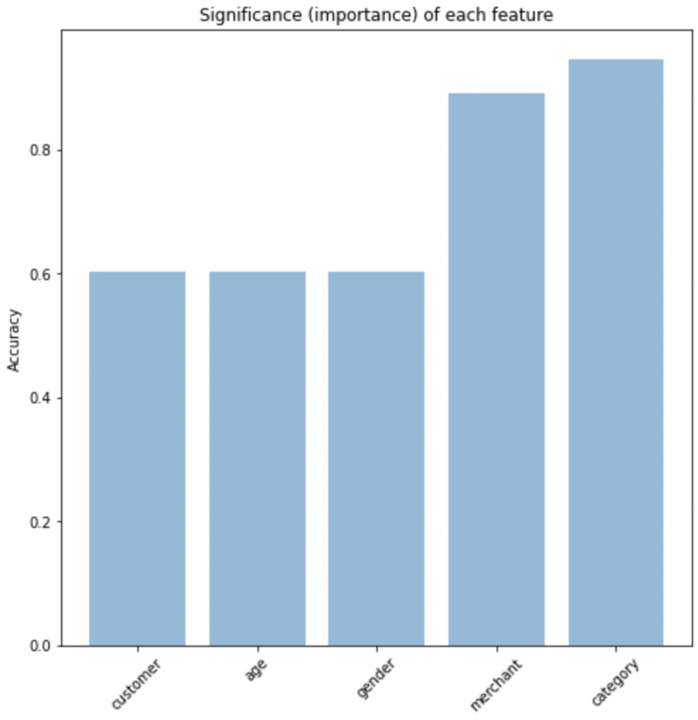
5.4 Early Stopping

I have used early stopping for my Model 7 because i see some fluctuations in the graph and my value-loss did not improve from 0.06357 (6 percent), and in this i have used call back function of my model7 and used it to draw graph of loss/epoch.

6 Feature Importance and Reduction

6.1 Feature Importance

The data-set has 6 independent input features and one output feature. The importance of each feature is obtained by training the best model with feature of interest(one at a time)and by evaluating the model's performance on the validation set. By training our model with one feature at time will help us to determine how strong the influence of that particular feature is there on deciding the output. The below figure shows the relation between single input feature and output in terms of accuracy which is obtained by that feature.

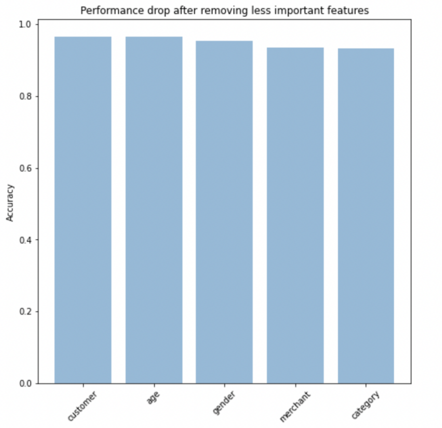


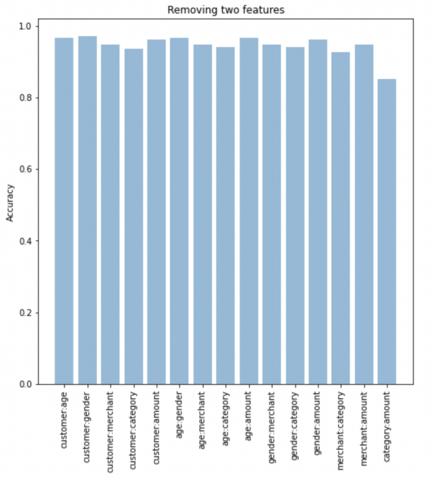
Accuracy

From the above figure we can see the accuracy remained almost constant around 60 percent, for all the features.However accuracy at **"Merchant and category"** features has been increased and to more than 80 percent and this clearly tells us that these are most important features and have strong influence in deciding the output class label.

6.2 Feature Reduction

After understanding the importance of each feature I experimented on removing one feature at a time and observed how the value of the accuracy is affected in the absence of that respective feature. As the below images gives the Performance increment after removing less important feature and the next image gives the performance after removing two features.





6.3 Reference

[1] Data set source https://www.kaggle.com/turkayavci/fraud-detection-on-bank-payments/data
[2] https://www.baeldung.com/cs/epoch-neural-networks