CREDIT CARD FRAUD DETECTION REPORT

Hiranmai Karedla
AITS
VIJAYAWADA
hiranmai.k15@iiits.in

ABSTRACT

Now a days cyber crimes have been increased.The problem we are dealing is finding a credit card transaction is fraudulent or not using machine learning. This is binary classification problem solved using 3 layered neural networks.

Exploratory Data Analysis, Data Preprocessing, Modeling, Evaluation tasks are performed for the data.

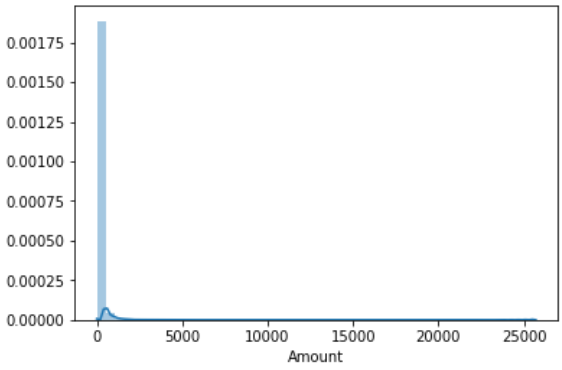
INTRODUCTION

Dataset:

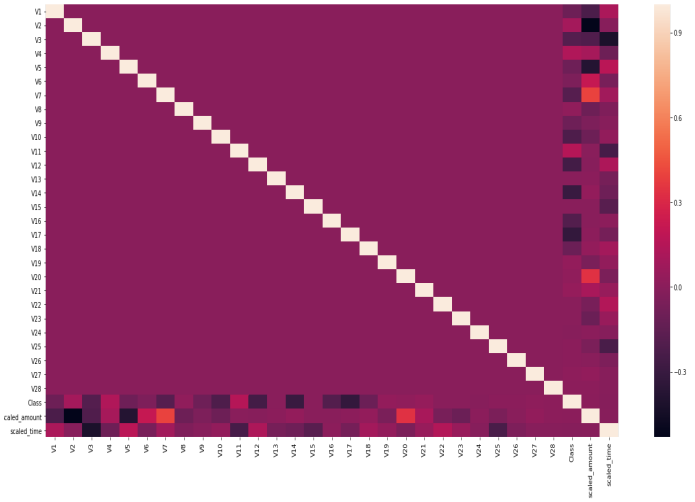
- The dataset is of size 284807 rows with 30 independent variables and 1 target variable.
- The data set contains 285315 non fraud cases and 492 fraud cases.

EXPLORATORY DATA ANALYSIS

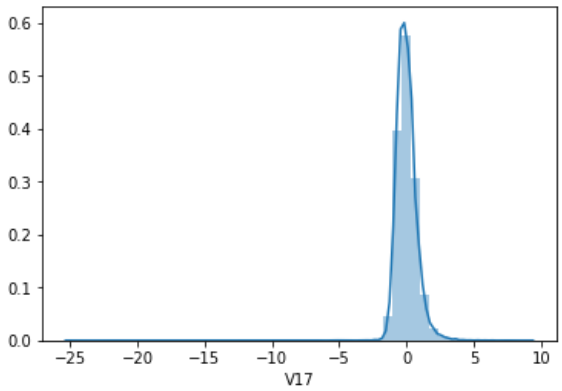
- Distribution of Amount of Transaction



- Normalization of the Data
- Correlation between the variables

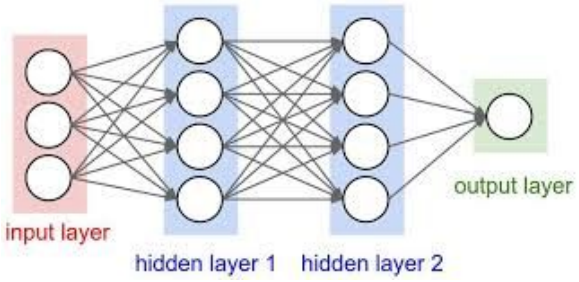


Distribution of one of the important variable

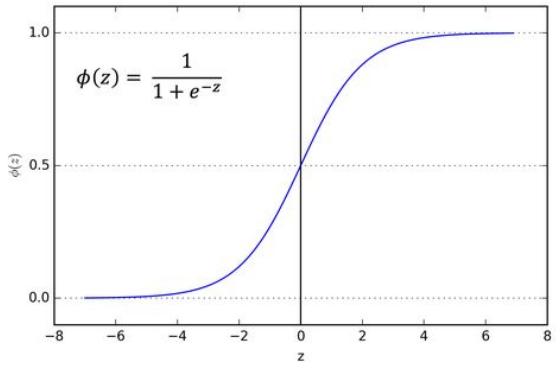


MODELING

- The Dataset is split into (80-20) train and test sets.
- Create a three layered neural network with nodes [10 -5 -1]
- Input dimension is 30 (No of features)



- Layers are Dense(fully connected)
- At the output Sigmoid Activation function is used.It gives probability between 0 and 1. Probability above 0.5 as one class and below 0.5 as other.



- Adam Optimization function is used for updating weights through backpropagation

$$For\ each\ Parameter\ w^j$$
(j subscript dropped for clarity)

$$\nu_t = \beta_1 * \nu_{t-1} - (1-\beta_1) * g_t$$
$$s_t = \beta_2 * s_{t-1} - (1-\beta_2) * g_t^2$$
$$\Delta\omega_t = -\eta \frac{\nu_t}{\sqrt{s_t + \epsilon}} * g_t$$
$$\omega_{t+1} = \omega_t + \Delta\omega_t$$

$\eta : Initial\ Learning\ rate$
$g_t : Gradient\ at\ time\ t\ along\ \omega^j$
$\nu_t : Exponential\ Average\ of\ gradients\ along\ \omega_j$
$s_t : Exponential\ Average\ of\ squares\ of\ gradients\ along\ \omega_j$
$\beta_1, \beta_2 : Hyperparameters$

EVALUATION METRICS

used or classification problems

- Confusion Matrix:

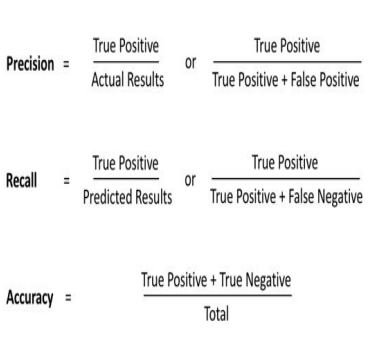
Actual value \ Prediction outcome	positive	negative	
positive	TP	FN	$TP + FN$
negative	FP	TN	$FP + TN$
	$TP + FP$	$FN + TN$	

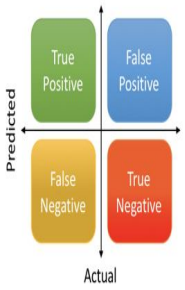
```
target_names = ['class 0', 'class 1']
print(classification_report(y_test, y_pred, target_names=target_names))
```

```
              precision    recall  f1-score   support

     class 0       1.00      1.00      1.00     56864
     class 1       0.00      0.00      0.00        98

    accuracy                           1.00     56962
   macro avg       0.50      0.50      0.50     56962
weighted avg       1.00      1.00      1.00     56962
```





- F1 score :
  2 * (Precision * Recall)
  (precision + Recall)

RESULTS

Training Accuracy : 99.944 %
Test Accuracy : 99.827%
Hence there is no overfitting and no under fitting.

CONCLUSION

From the above results the three layered neural network model performed well for the data in both train and test set with satisfying results.