PROJECT DELIVERABLE 2

Project: CREDIT CARD FRAUD DETECTION

1. **Problem statement**: The aim of my project is to recognize the fraudulent credit card transactions by learning about how accurate in detecting whether a transaction is a genuine or fraud payment. The binary classes are “genuine” - denoted as 0 and “fraud” - denoted as 1

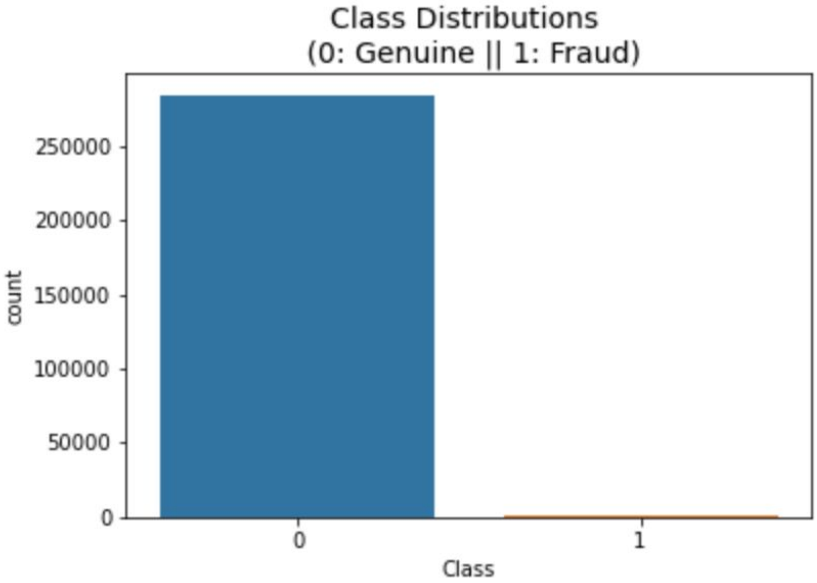
2. **Data Preprocessing:** I will be using this dataset:https://www.kaggle.com/mlg-ulb/creditcardfraud

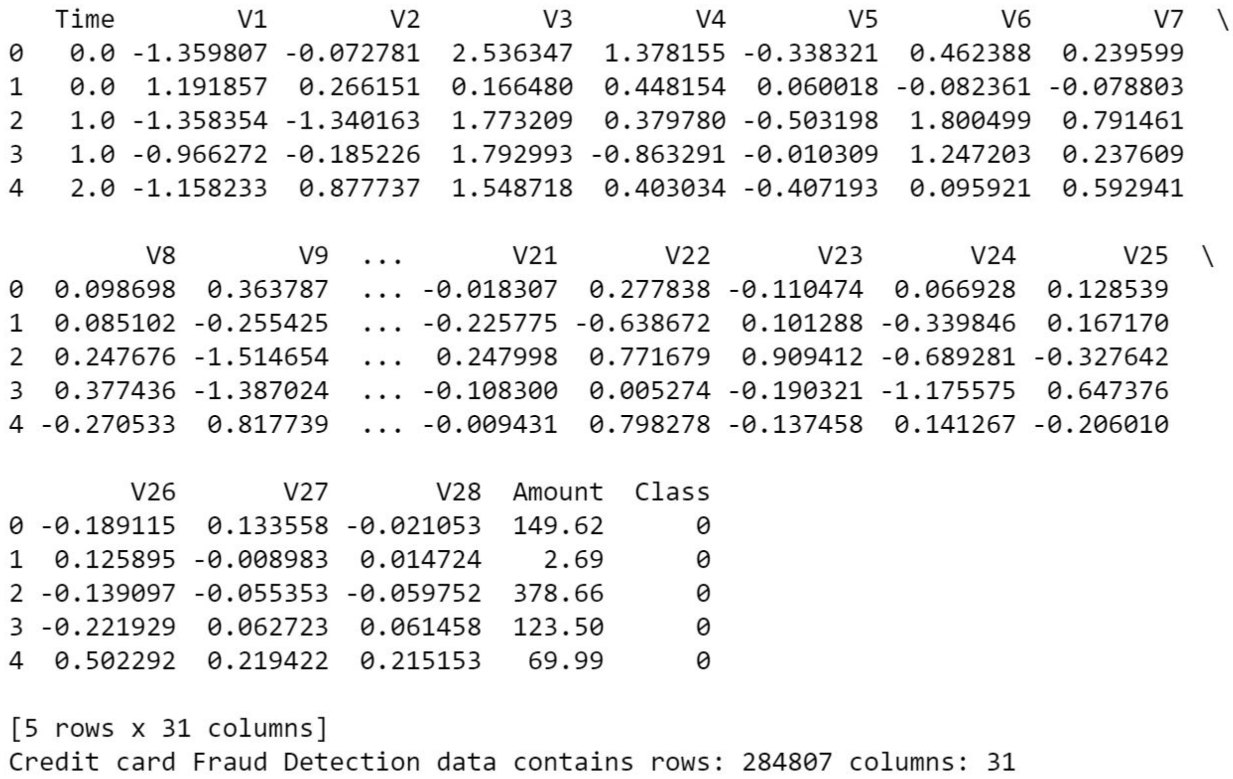
   Credit card Fraud Detection data contains rows: 284808 columns: 31.

   The dataset contains variables under PSA and the others that have not been transformed is “ Time”, “Amount”, “Class” where 0 is genuine and 1 as fraud. The data is quite unbalanced with only 492 frauds out of 284,807 transactions.

3. **Machine learning model**: The problem is an anomaly detection so I will be using Naive Bayes Gaussian as my dataset is continuous.

   a. I have split the dataset into three parts: 0.8 for the training set, 0.1 for validation and 0.1 for testing. The splitting process is done randomly with the condition that the fraud - genuine proportion is consistent between different sets.



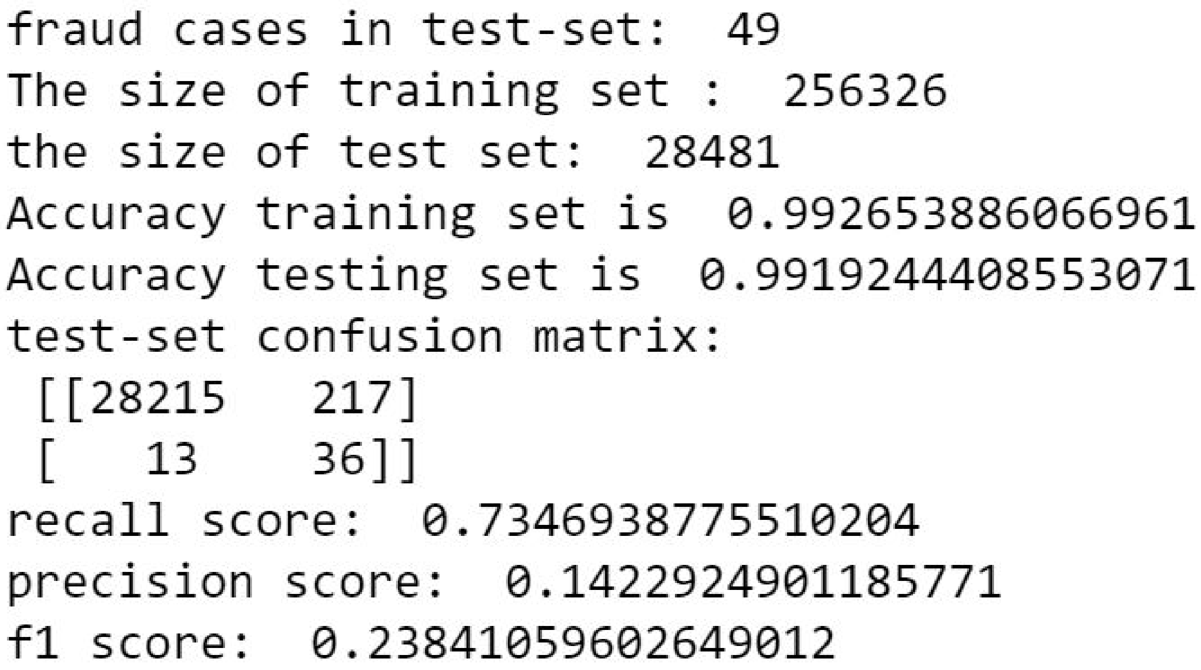


```
   Time        V1        V2        V3        V4        V5        V6        V7  \
0   0.0 -1.359807 -0.072781  2.536347  1.378155 -0.338321  0.462388  0.239599   
1   0.0  1.191857  0.266151  0.166480  0.448154  0.060018 -0.082361 -0.078803   
2   1.0 -1.358354 -1.340163  1.773209  0.379780 -0.503198  1.800499  0.791461   
3   1.0 -0.966272 -0.185226  1.792993 -0.863291 -0.010309  1.247203  0.237609   
4   2.0 -1.158233  0.877737  1.548718  0.403034 -0.407193  0.095921  0.592941   

         V8        V9  ...       V21       V22       V23       V24       V25  \
0  0.098698  0.363787  ... -0.018307  0.277838 -0.110474  0.066928  0.128539   
1  0.085102 -0.255425  ... -0.225775 -0.638672  0.101288 -0.339846  0.167170   
2  0.247676 -1.514654  ...  0.247998  0.771679  0.909412 -0.689281 -0.327642   
3  0.377436 -1.387024  ... -0.108300  0.005274 -0.190321 -1.175575  0.647376   
4 -0.270533  0.817739  ... -0.009431  0.798278 -0.137458  0.141267 -0.206010   

        V26       V27       V28  Amount  Class  
0 -0.189115  0.133558 -0.021053  149.62      0  
1  0.125895 -0.008983  0.014724    2.69      0  
2 -0.139097 -0.055353 -0.059752  378.66      0  
3 -0.221929  0.062723  0.061458  123.50      0  
4  0.502292  0.219422  0.215153   69.99      0  

[5 rows x 31 columns]
Credit card Fraud Detection data contains rows: 284807 columns: 31
```

b. I will be testing my model based on the accuracy_score, F_1 score, which is the harmonic mean of Precision and Recall. The precision measures how many times my model can correctly predict fraud divide by how many times it gives the prediction of fraud. The recall measures how many times my model correctly predicts fraud given that it is actually a fraud transaction. Also, I will be analyzing the confusion matrix and see how many transactions are False Negative, the goal is to train my model so that the False Negative cases occur as low as possible.

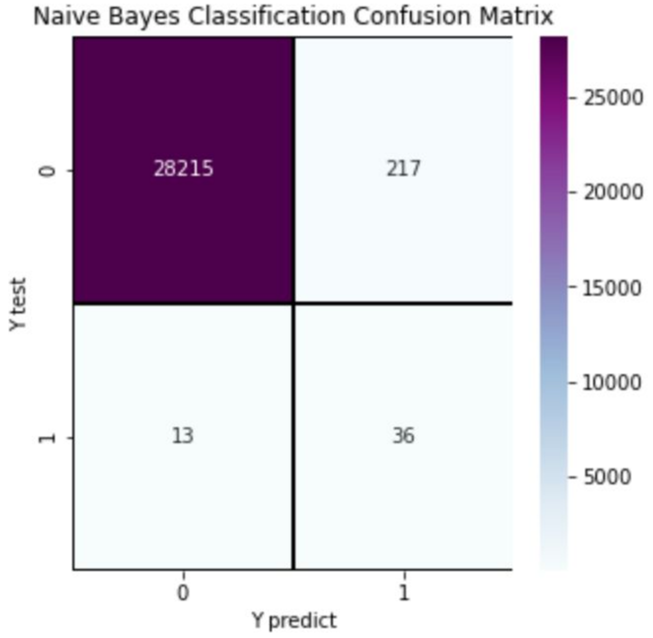
c. My dataset has many variables transformed under PSA so it is hard for me to work on those. Also because the dataset is unbalanced, the F_1 score received is not as high as I wanted. My current F_1 score is 0.23, i.e if there is a fraud then my model will catch it 23% of the time. However as the features are from PSA, I can make an assumption that they are probably ordered based on the variance of each feature. This means that I can try to do the feature selection.

4. **Preliminary results**: After training my set using Naive Bayes Gaussian, here is my accuracy_point and F_1_score



```
fraud cases in test-set:  49
The size of training set :  256326
the size of test set:  28481
Accuracy training set is  0.992653886066961
Accuracy testing set is  0.9919244408553071
test-set confusion matrix:
 [[28215   217]
 [   13    36]]
recall score:  0.7346938775510204
precision score:  0.1422924901185771
f1 score:  0.23841059602649012
```

My f_1_score is pretty low here as I mentioned that my dataset is quite unbalanced. My confusion matrix is



28215 is the number of transitions that y_pred is 0 and the actual transaction in y_test is genuine. 217 is the number of False Positive, i.e the transaction is a genuine but we predict fraud and 13 is the number of False Negatives - the one we want to avoid since our y_pred is genuine but the actual transaction is a fraud, i.e the thief has managed to stole money from that transaction. The number of False Negative here is pretty low compared to the set size.

5. **Next step**: My goal in the next stage is to rise up my f_1_score. I will start with feature selection, such as dropping V23-V26 variables. I will also work on testing my data with logistic regression, xgboost or lightgbm. Also about the PSA variables, I want to study more about the correlation between those variables.