RUTGERS

THE STATE UNIVERSITY
OF NEW JERSEY

Team Members:

Akhil Patil

Arjun Prakash

Palash Nandecha

CAPSTONE PROJECT

2019

MOTIVATION: -

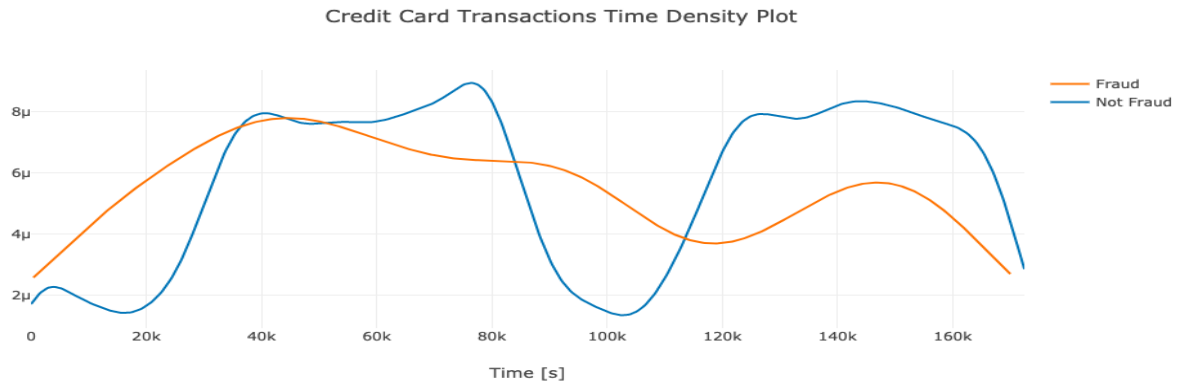
In recent years, topics such as fraud detection and fraud prevention have received a lot of attention on the research front, in particular from payment card issuers. A successful strategy for dealing with fraud can quite literally mean millions of dollars in savings per year on operational costs.

- The Federal Trade Commission estimates that 10 million people are victimized by credit card theft each year.
- Credit card companies lose close to $50 billion dollars per year because of fraud.
- These costs "trickle down" in higher interest rates and fees for all customers.
- As credit card becomes the most popular mode of payment, cases of fraud associated with it are also increasing.

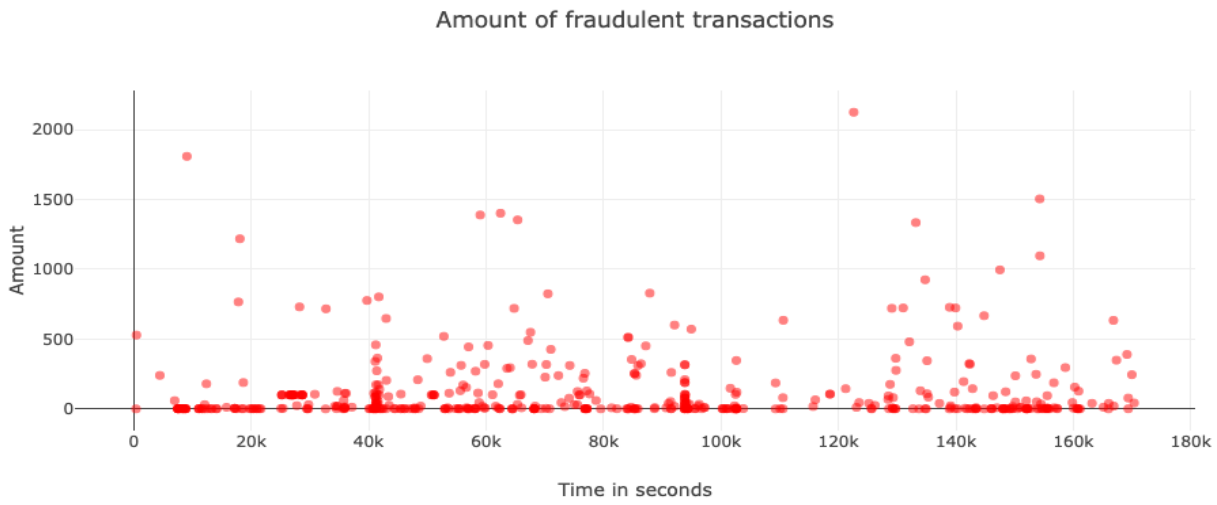
DATASET : -

The dataset has a transaction of credit card from September 2013. The dataset has transactions with 492 frauds out of 284,807 transactions in two days. It has numerical input data. Due to security purposes, most of the dataset have been subjected to PCA transformation, but the other columns "Amount" and "Time" have not been subjected to this transformation. Here Time is the number of seconds between the two first and every other transaction in dataset and Amount is the money in the dataset, then column named "Class" has two values "0" and"1" in which 0 means "Not Fraud" and 1 means "Fraud".

DATASET Preprocessing: -



Through time-density plot here we are analyzing how the fraud transactions are varying in the dataset with a blue line depicting not fraud and yellow line predicting fraud. In this x-axis is a time in seconds between first and every other transaction and y-axis is the probability per unit on the x-axis.

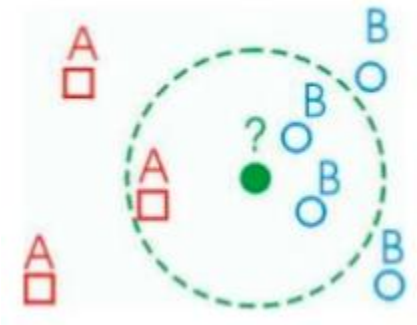


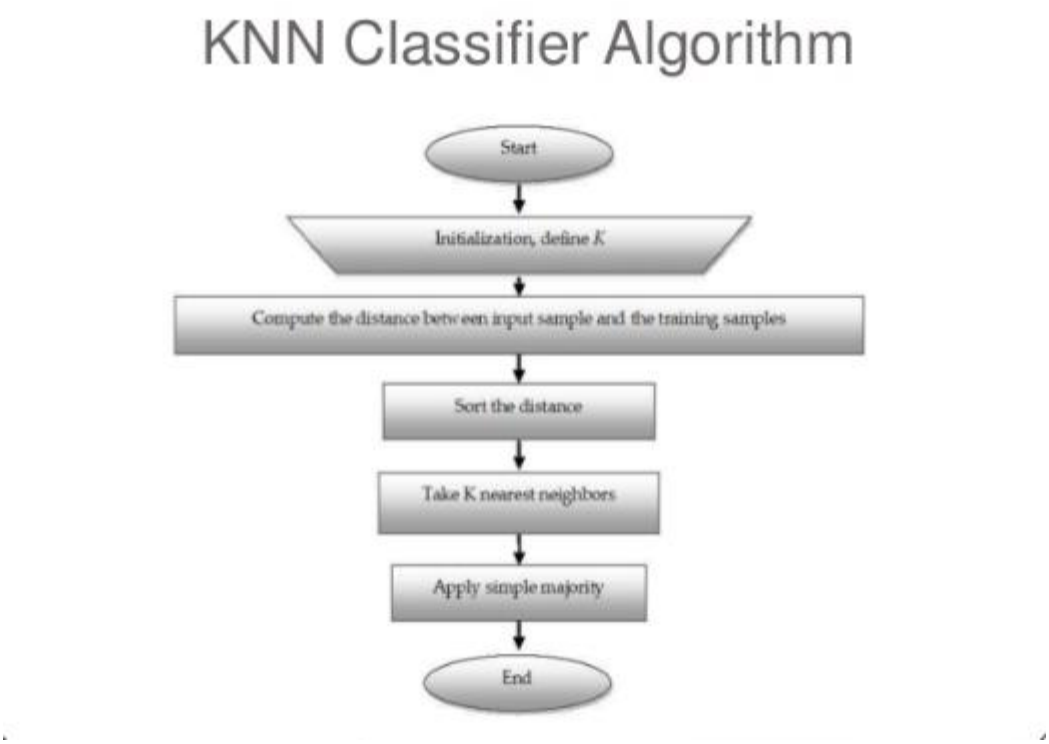
Through this scatter plot we have depicted the number of fraudulent transactions in comparison to time. In this dataset we have transaction of two days hence the 48 hours period is converted into seconds and the y-axis is the amount of money extracted while doing the fraudulent transaction.

ALGORITHMS: -

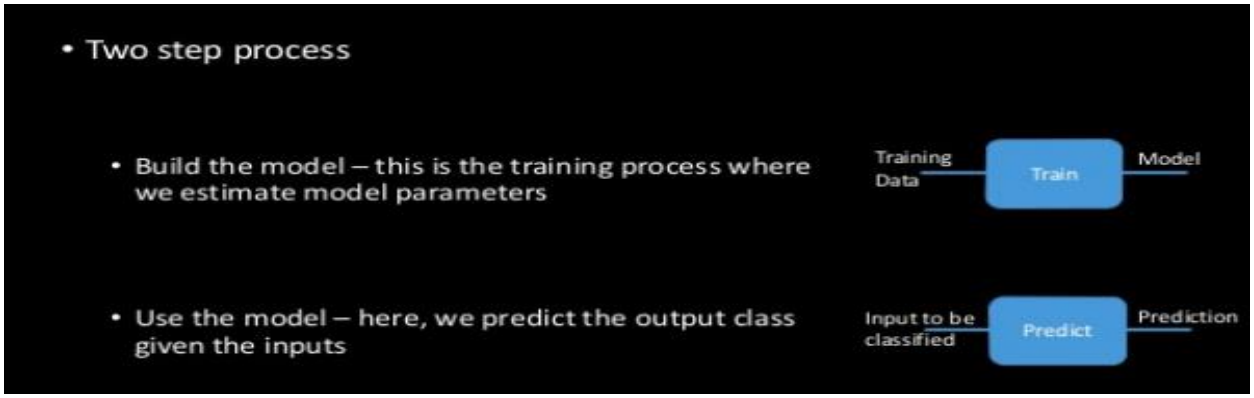
Classifiers: -

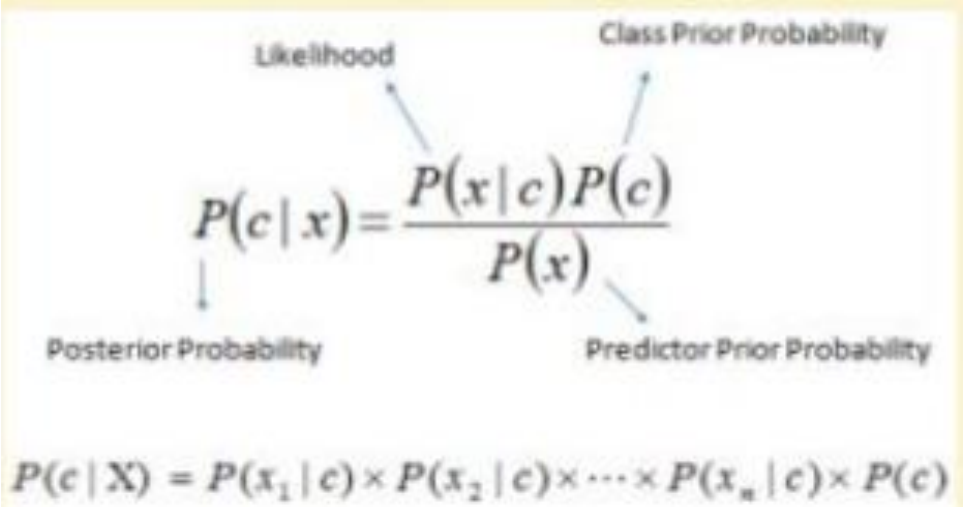
k-means Clustering: - In this, we will create a group of data that is fraudulent, then using our data we will train in this model and if after training the clustering that we have formed matches the one we trained earlier then fraud detection will become easy.In this we have used Minkowski Distance for checking the accuracy and similarity between multiple transactions and on this basis we have categorized the transactions as fraud or not fraud.





Naive-Bayes Classifiers:- Naive- Bayes are probabilistic classifiers that are based on the application of Bayes theorem and also there is an independent assumption between the variables.In this, we have used Gaussian naive Bayes.





where:

1. P(c|x):-is the probability of hypothesis c given the data x.
2. P(x|c):-is the probability of data x given that the hypothesis c was true.
3. P(c):-is the probability of hypothesis c being truly independent of data. This is known as the prior probability of c.
4. P(x):-is the probability of the data which is also independent of data.

In this dataset, for each transaction, two separate posterior probabilities are calculated one with x = fraud and one with x = Not Fraud.

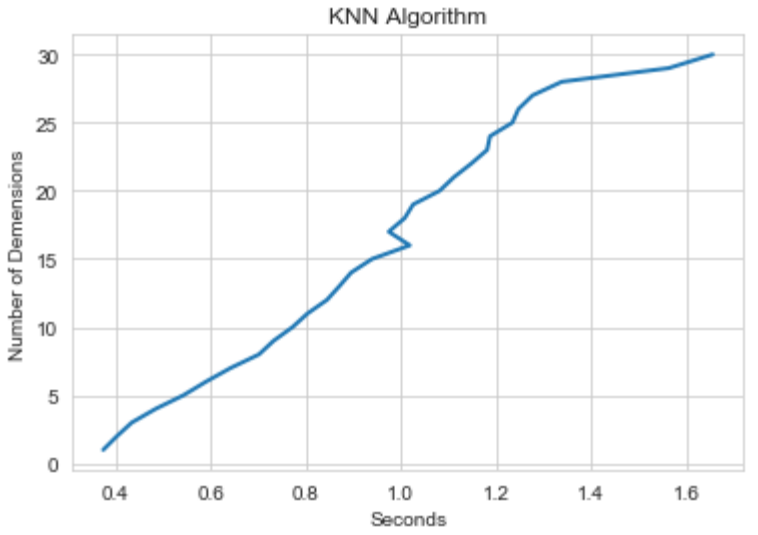
EXPERIMENTAL CONFIGURATION: -

In this, we have divided the data into training data and testing data, where we have prediction by applying naive-Bayes and knn model on training data and then we have checked for validation and accuracy of predictions and models.For this project the operating system used was Windows 10 64-bit, the processor was Intel(R) Core(™) i5 CPU @1.60GHz.For coding we used Jupyter notebook, under Anaconda IDE.

Result and Analysis:

Time Complexity Analysis of Knn Classifier Algorithm:

The plot of the number of dimensions with time shows that they are a direct proportional i.e. increase in dimensions results increase in time. This depicts that the algorithm depends on the dimensionality of the data and the Number of records (N). So, Overall time complexity comes out to be O(Nd).



Evaluation of Knn Classifier :

Calculation of KNN scores.

```
In [28]: print_scores(y_test0,preds1)
```

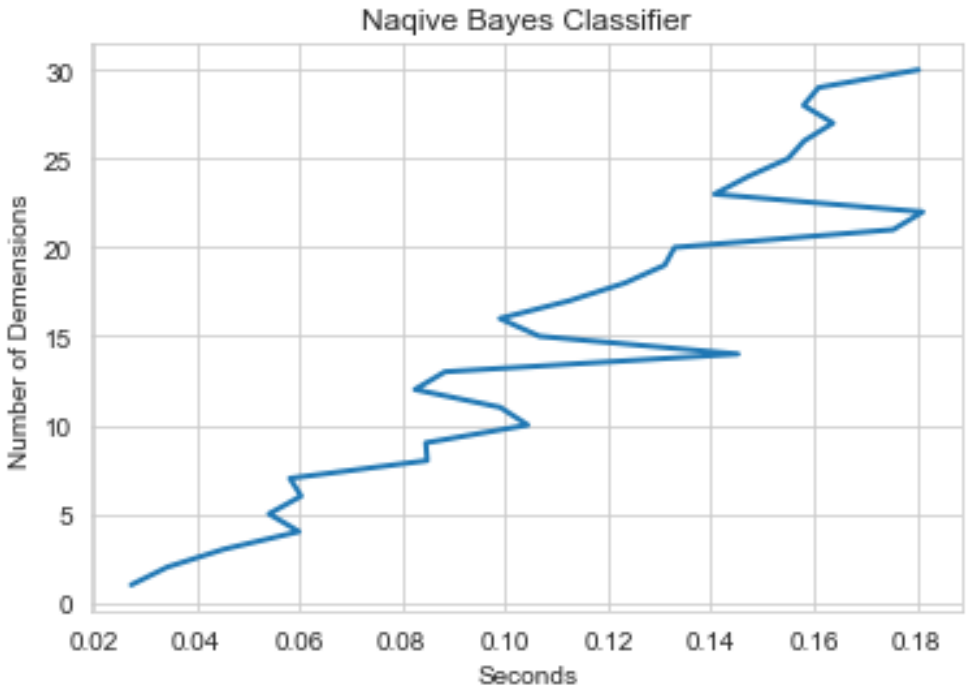
```
test-set confusion matrix:
 [[56856     0]
 [   97     9]]
recall score:  0.08490566037735849
precision score:  1.0
f1 score:  0.15652173913043477
ROC AUC score:  0.5424528301886793
```

From Confusion Matrix, we can see that 97 Actual Fraud that Knn Algorithm predicted not frauds which seems to be the reason of low 54% ROC AUC-Area under Curve score.
Precision is very high 1.0 as it calculates true positives by total predicted positives while Recall is low to 8% because it calculates true positives by total actual positives. Finally, F1 score calculated on the basis of precision and recall has a low score of 15 because of imbalance in Precision and recall.

We can say that Knn classifier is not good for classifying Fraudulent transactions as it has high precision, low F1 score, and incorrect prediction.

Time Complexity Analysis of Naive Bayes Classifier Algorithm:

The plot of the number of dimensions with time shows that although it is not increasing sharply over a period of time it increases. This depicts that the algorithm depends on the dimensionality of the data and the Number of records (N). So, Overall time complexity comes out to be O(Nd).



Evaluation of Naive Bayes Classifier :

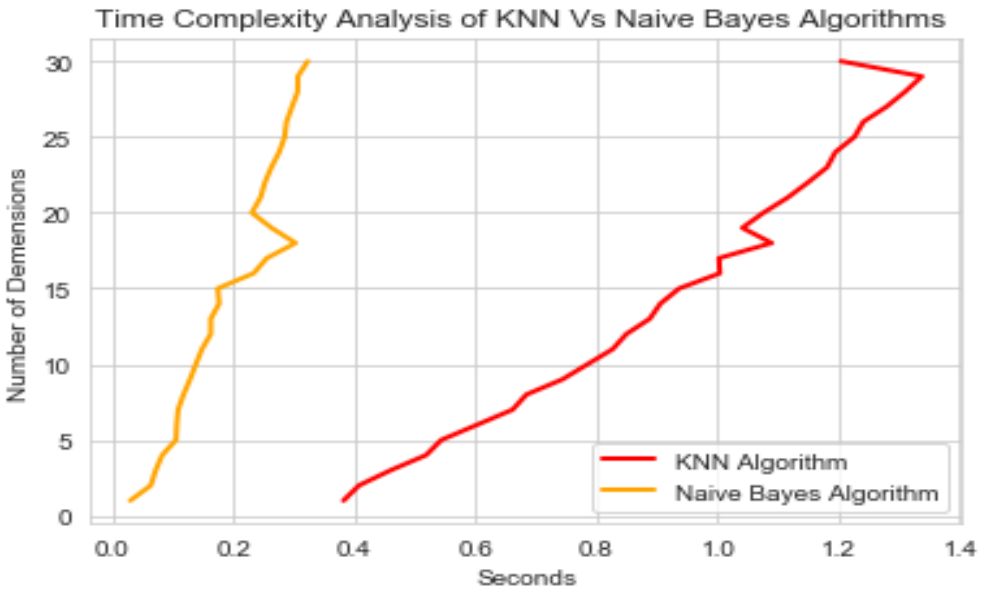
Naive Bayes Test Scores.

```
In [35]: print_scores(y_test0,model.predict)

         test-set confusion matrix:
          [[56442   414]
          [   39    67]]
         recall score:  0.6320754716981132
         precision score:  0.1392931392931393
         f1 score:  0.22827938671209538
         ROC AUC score:  0.8123969591500275
```

From Confusion Matrix, we can see that only 39 Actual Fraud that Naive Bayes Algorithm predicted not frauds which seems to be the reason of high score of 81% ROC AUC score. Precision is low at ~14% as it calculates true positives by total predicted positives which are a good indicator of the model while Recall is high at 63% because it calculates true positives by total actual positives. Finally, F1 score calculated on the basis of precision and recall has a good score of 22 because of better indicators of Precision and recall.
Here we can see that Naive Bayes does a better job of classifying Fraudulent transactions with just 39 frauds predicted as not frauds.

Comparison of Complexities of KNN and Naive Bayes Algorithm:



We can observe that Naive Bayes outperforms Knn classifier in predicting fraudulent transactions and is much faster in execution. Naive Bayes has far better ROC area under curve score of 81% then knn which as a score of 54%.

Conclusion:

Through this project, we thoroughly analyzed the fraud detection during the transaction of credit card and how using two different approaches of KNN and Naive-Bayes classifiers we can take on the problem head-on and solve it.In both the algorithms we used confusion matrices, ROC-AUC, F1 scores and recall values for better accuracy and predictions and also to determine which of the two algorithms perform better and present us with better results which resulted in naive-Bayes classifiers giving us better insights about the fraud detection.Hence we conclude that Naive-Bayes is a pretty good predictor and can be used for detecting fraudulent transactions in scenarios to prevent loss of time and money and build a better system for consumers for credit card transactions.

References:

1. https://towardsdatascience.com/histograms-and-density-plots-in-python-f6bda88f5ac0
2. https://machinelearningmastery.com/naive-bayes-for-machine-learning/
3. https://towardsdatascience.com/knn-using-scikit-learn-c6bed765be75
4. Classification: ROC Curve and AUC | Machine Learning Crash Course | Google Developers
5. A simple guide to confusion matrix terminology
6. Confusion Matrix in Machine Learning - GeeksforGeeks