I would like to express my utmost gratitude to the **NATIONAL INSTITUTE OF TECHNOLOGY, SILCHAR** for providing me an opportunity in doing S.N. Bose internship which was required to fulfil the curriculum criteria.

We feel pleased to have the opportunity of expressing our heartfelt thanks and gratitude to those who all rendered their cooperation in making this report. This thesis is performed under the supervision of **Dr. Vipin Chandra Pal Sir**, **NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR (NIT SILCHAR),** **Cachar, Assam**. During the work, he has supplied us a number of books, journals, and materials related to the present investigation. Without his help, kind support and generous time spans he has given, we could not perform the project work successfully in due time. First and foremost, we wish to acknowledge our profound and sincere gratitude to him for his guidance, valuable suggestions, encouragement and cordial cooperation. We express our gratitude to all other sources from where we have found help. We are indebted to those who have helped us directly or indirectly in completing this work. Finally, we would like to thank **Dr. Vipin Sir** and our friend **Raushan Kumar** who have helped us by giving their encouragement and cooperation throughout the work.

I perceive this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way and I will continue to work on improvement in order to attain my desired career objectives. Hope to continue cooperation with you all in the future.

**Rahul Kumar**

Credit card fraud is a significant problem, with billions of dollars lost each year. Machine learning can be used to detect credit card fraud by identifying patterns that are indicative of fraudulent transactions. Credit card fraud refers to the physical loss of a credit card or the loss of sensitive credit card information. Many machine-learning algorithms can be used for detection. This project proposes to develop a machine-learning model to detect credit card fraud. The model will be trained on a dataset of historical credit card transactions and evaluated on a holdout dataset of unseen transactions.

**Keywords:** Credit Card Fraud Detection, Fraud Detection, Fraudulent Transactions, **K- Nearest Neighbors, Support Vector Machine, Logistic Regression, Decision Tree.**

1.1 Introduction ------------------------------------------------------------------------------- 04

1.2 Project Goals ------------------------------------------------------------------------------ 04

**Literature Review:**

2.1 Introduction --------------------------------------------------------------------------- 04

2.2 Literature Review -------------------------------------------------------------------- 05-06

**Project Description:**

3.1 Introduction --------------------------------------------------------------------------- 07

3.2 Data source --------------------------------------------------------------------------- 07

**Data Analysis:**

**4.1** Data Preparation -------------------------------------------------------------------- 08-10

**4.2** Data Preprocessing ----------------------------------------------------------------- 10

**4.3** Data Modeling **----------------------------------------------------------------------** 10-12

**Algorithm and Performance Analysis**

**5.1** Logistic Regression (L.R) -------------------------------------------------------- 13-14

**5.2** Decision Tree (D.T) --------------------------------------------------------------- 14-16

**5.3** K-Nearest Neighbor (KNN) ------------------------------------------------------ 16-20

**5.4** Support vector Machine (SVM) ------------------------------------------------- 21-23

**Evaluation and Deployment -----------------------------------------------------** 23-24

**Conclusion ---------------------------------------------------------------------------**  25

**1.1 Introduction:**

With the increase of people using credit cards in their daily lives, credit card companies should take special care of the security and safety of the customers. According to (Credit card statistics 2021), the number of people using credit cards worldwide was 2.8 billion in 2019; also, 70% of those users own a single card.

Reports of Credit card fraud in the U.S. rose by 44.7% from 271,927 in 2019 to 393,207 words in 2020. There are two kinds of credit card fraud, and the first is having a credit card account opened under your name by an identity thief. Reports of this fraudulent behaviour increased 48% from 2019 to 2020. The second type is when an identity thief uses an existing account you created, usually by stealing the information on the credit card. Reports on this type of Fraud increased 9% from 2019 to 2020 (Daly, 2021). Those statistics caught We's attention as the numbers have increased drastically and rapidly throughout the years, which motivated w to resolve the issue analytically by using different machine learning methods to detect fraudulent credit card transactions within numerous transactions.

**1.2 Project goals:**

The main aim of this project is the detection of fraudulent credit card transactions, as it is essential to figure out the fraudulent transactions so that customers do not get charged for the purchase of products that they did not buy. Fraudulent Credit card transactions will be detected with multiple ML techniques. Then, a comparison will be made between the outcomes and results of each method to find the best and most suited model for detecting fraudulent credit card transactions; graphs and numbers will also be provided. In addition, it explores previous literature and different techniques used to distinguish Fraud within a dataset.

**Research question:** What machine learning model is most suited for detecting fraudulent credit card transactions?

**2.1 Introduction:**

Credit card companies must distinguish fraudulent from non-fraudulent transactions so that their customers' accounts will not get affected and charged for products they did not buy (Maniraj et al., 2019). Many financial Companies and institutions lose massive amounts of money because of Fraud and fraudsters that are seeking different approaches continuously to violate the rules and commit illegal actions; therefore, systems of fraud detection are essential for all banks that issue credit cards to decrease their losses (Zareapoor et al., 2012). Multiple methods are used to detect fraudulent behaviours, such as Neural Networks (N.N.), Decision Trees, K-Nearest Neighbour algorithms, and Support Vector Machines (SVM). Those ML methods can be applied independently or collectively with ensemble or meta-learning techniques to develop classifiers (Zareapoor et al., 2012).

**2.2 Literature Review:**

Zareapoor and his research team used multiple techniques to determine the best-performing model for detecting fraudulent transactions, which was established using the Accuracy of the model, the speed of detection and the cost. The models used were Neural Network, Bayesian Network, SVM, KNN and. The comparison table in the research paper showed that the Bayesian Network was high-speed in finding fraudulent transactions with high Accuracy. The N.N. performed well, as the detection was fast, with a medium accuracy. KNN’s speed was good with medium Accuracy, and finally, SVM scored one of the lower scores, as the speed was low, and the Accuracy was medium. As for the cost, all models built were expensive (Zareapoor et al., 2012).

The model used by Alenzi and Aljehane to detect Fraud in credit cards was Logistic Regression. Their model scored 97.2% in Accuracy, 97% sensitivity and 2.8% Error Rate. A comparison was performed between their model and

They were voting Classifier and KNN. V.C. scored 90% in Accuracy, 88% sensitivity and 10% error rate, as for KNN where k = 1:10, the Accuracy of the model was 93%, the sensitivity 94% and 7% for the error rate (Alenzi & Aljehane, 2020).

Rahul's team built a model to recognize if any new transaction is Fraud or non-fraud. Their goal was to get 97% in detecting fraudulent transactions and try to minimize the incorrectly classified fraud instances. Their model has performed well as they got 96.7% of the fraudulent transactions.

The classification approach used by Rahul and Raushan was the behaviour-based classification approach, using a Support Vector Machine, where the behavioral patterns of the customers were analyzed to distinguish credit card fraud, such as the amount, date, time, place, and frequency of card usage. The Accuracy achieved by their approach was more than 80%.

Jain’s team used several ML techniques to distinguish credit card fraud; three of them are SVM, ANN and KNN. Then, to compare the outcome of each model, they calculated the true positive (T.P.), false Negative (F.N.), false positive (F.P.), and true negative (T.N.) generated. ANN scored 99.71% accuracy, 99.68% precision, and 0.12% false alarm rate. SVM accuracy is 94.65%, 85.45% for the precision, and 5.2% false alarm rate. Moreover, finally, the Accuracy of KNN is 97.15%, the precision is 96.84%, and the false alarm rate is 2.88%.

**3.1 Introduction:**

In order to accomplish the objective and goal of the project, which is to find the most suited model to detect credit card fraud, several steps need to be taken. Finding the most suited data and preparing/preprocessing are the first and second steps; after making sure that the data is ready, the modelling phase starts, where four models are created: K-Nearest Neighbor (KNN), Decision Tree, SVM and the last one is Logistic Regression. In the KNN model, two Ks were chosen: K=3 and K =7. All models were created in Jupiter Notebook programs.

**3.2 Data Source:**

The dataset was retrieved from an open-source website, Kaggle.com. It contains data on transactions made in 2013 by European credit card users in two days only. The dataset consists of 31 attributes and 284,808 rows. Twenty-eight attributes are numeric variables that, due to the confidentiality and privacy of the customers, have been transformed using PCA transformation; the three remaining attributes are "Time", which contains the elapsed seconds between the first and other transactions of each Attribute, "Amount" is the amount of each transaction, and the final attribute “Class” which contains binary variables where **“1”** is a case of **fraudulent transaction**, and **“0”** is **not as case of fraudulent transaction.**

**Dataset Link:** <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>

**4.1 Data Preparation:**

The first figure below shows the structure of the dataset where all attributes are shown, with their type, in addition to a glimpse of the variables within each Attribute; as shown at the end of the figure, the Class type is integer, which I needed to change to factor and identify the 0 as Not Fraud and the one as Fraud to ease the process of creating the model and obtain visualizations.

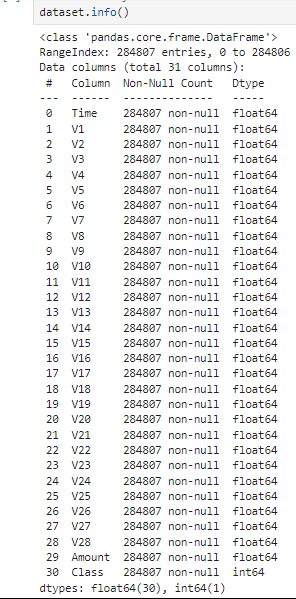
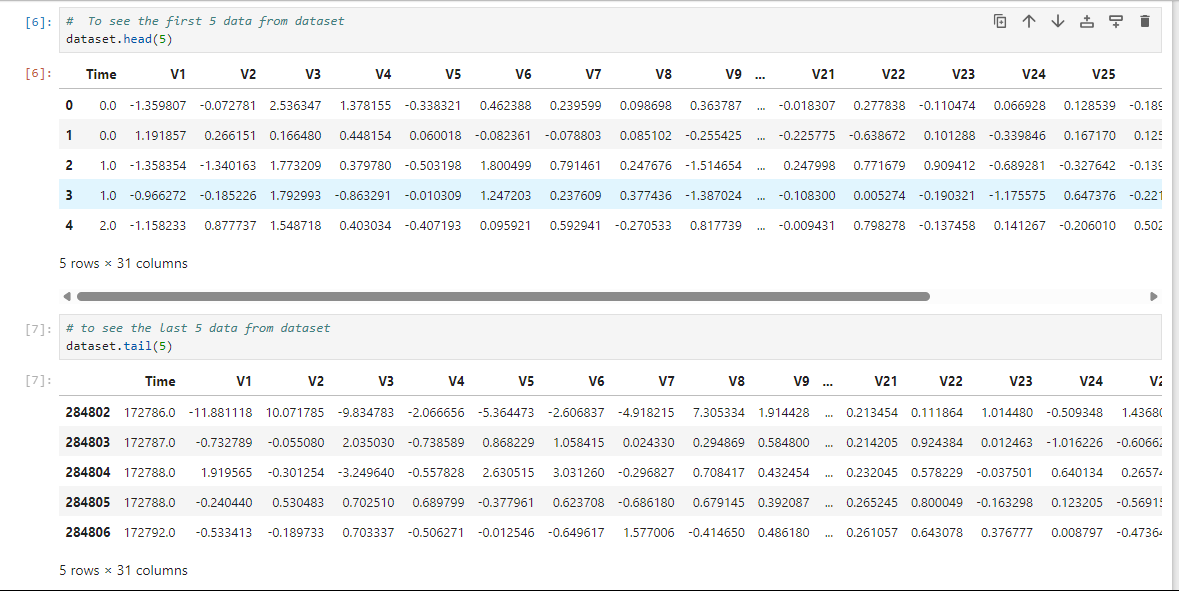
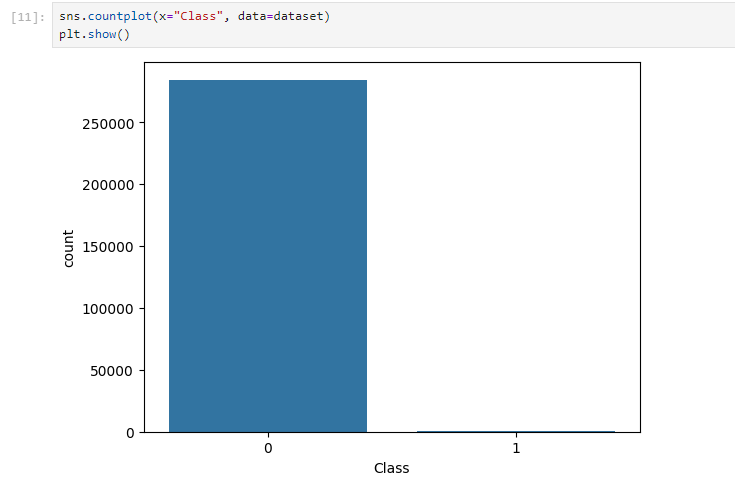
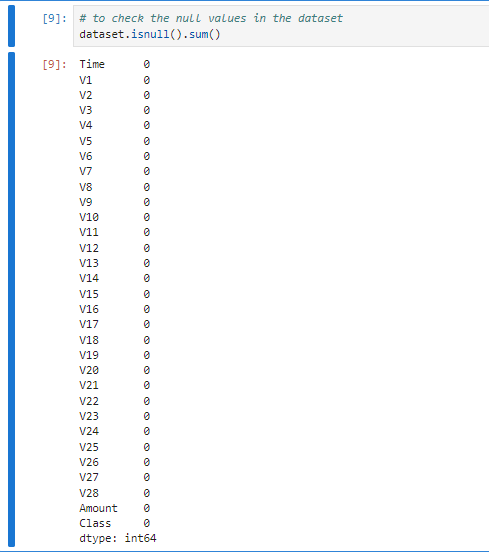


Fig 1: *Dataset Structure*



*Fig 2: Data Description*

*Fig 3: Class Count*



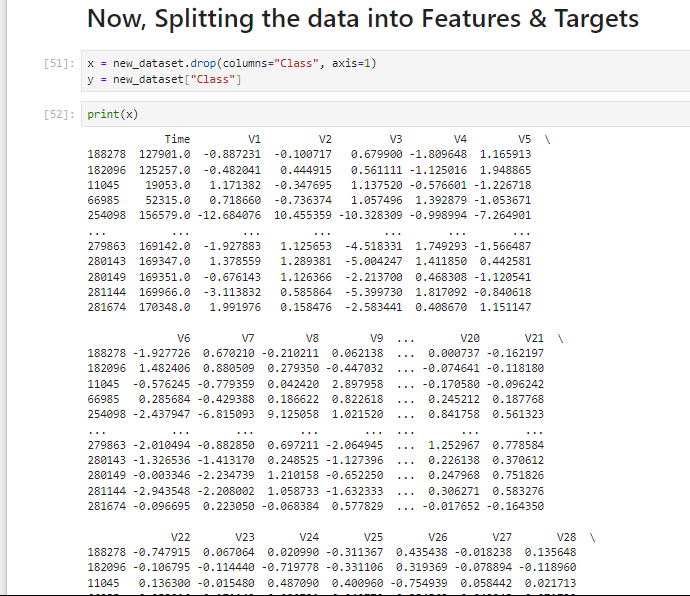
*Fig 4: Null Data*

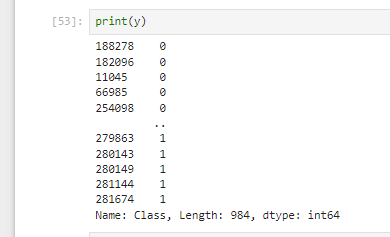
**4.2 Data Preprocessing:**

As there are no N.A.s nor duplicated variables, preparing the dataset was simple. The first alteration made to open the dataset on the Jupiter Notebook program is changing the type of the class attribute from Numeric to Class and identifying the class as {1, 0} using the program Sublime Text. Another alteration was made to the type and the Jupiter Notebook program to create the model and the visualization.

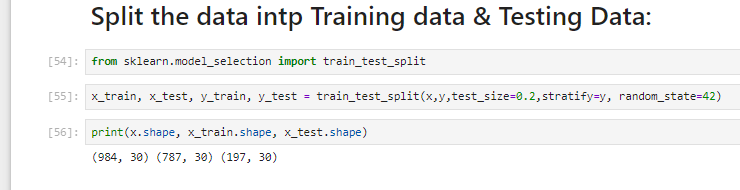
**4.3 Data Modeling:**

After ensuring that the data is ready to get modelled, the four models were created for KNN, Logistic Regression and Decision Tree using Jupiter Notebook.

Fig 5: *Splitting the data's*



*Fig 6: Print the data*

*Fig 7: Modelling*

**5.1 Logistic Regression (L.R.) Algorithm:**

This statistical classiﬁcation model based on probabilities detects Fraud using a logistic curve. Since the value of this logistic curve varies from 0 to 1, it can be used to interpret class membership probabilities. The dataset fed as input to the model is classiﬁed for training and testing the model. Post-model activity is tested for some minimum threshold cut-off value for prediction. Based on some threshold probabilities, the logistic Regression can divide the plane using a single line and divide dataset points into exactly two regions.

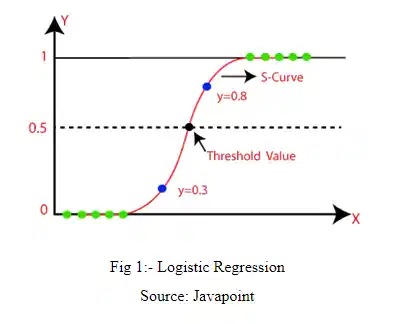


Fig 5: Logistic Regression

The last model created using Jupiter Notebook is Logistic Regression; the model managed to score an Accuracy on **Training data of 94.0279%** (Fig 6), while it scored an Accuracy score on **Testing Data of 95.4314%**, as presented in Figure 6.

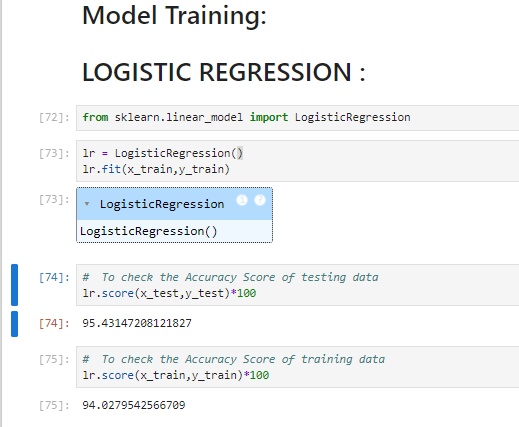


Fig 6: *Logistic Regression*

**5.2 Decision Tree (DT) Algorithm:**

A supervised learning algorithm is a decision tree in the form of a tree structure consisting of the root node and other nodes split in a binary or multi-split manner further into child nodes, with each tree using its algorithm to perform the splitting process. With the tree growing, there may be possibilities of overﬁtting the training data with possible anomalies in branches, some errors or noise. Hence, pruning is used for improving classiﬁcation performance of the tree by removing specific nodes. Ease in use and the ﬂexibility that the decision trees provide to handle different data types of attributes make them quite popular.

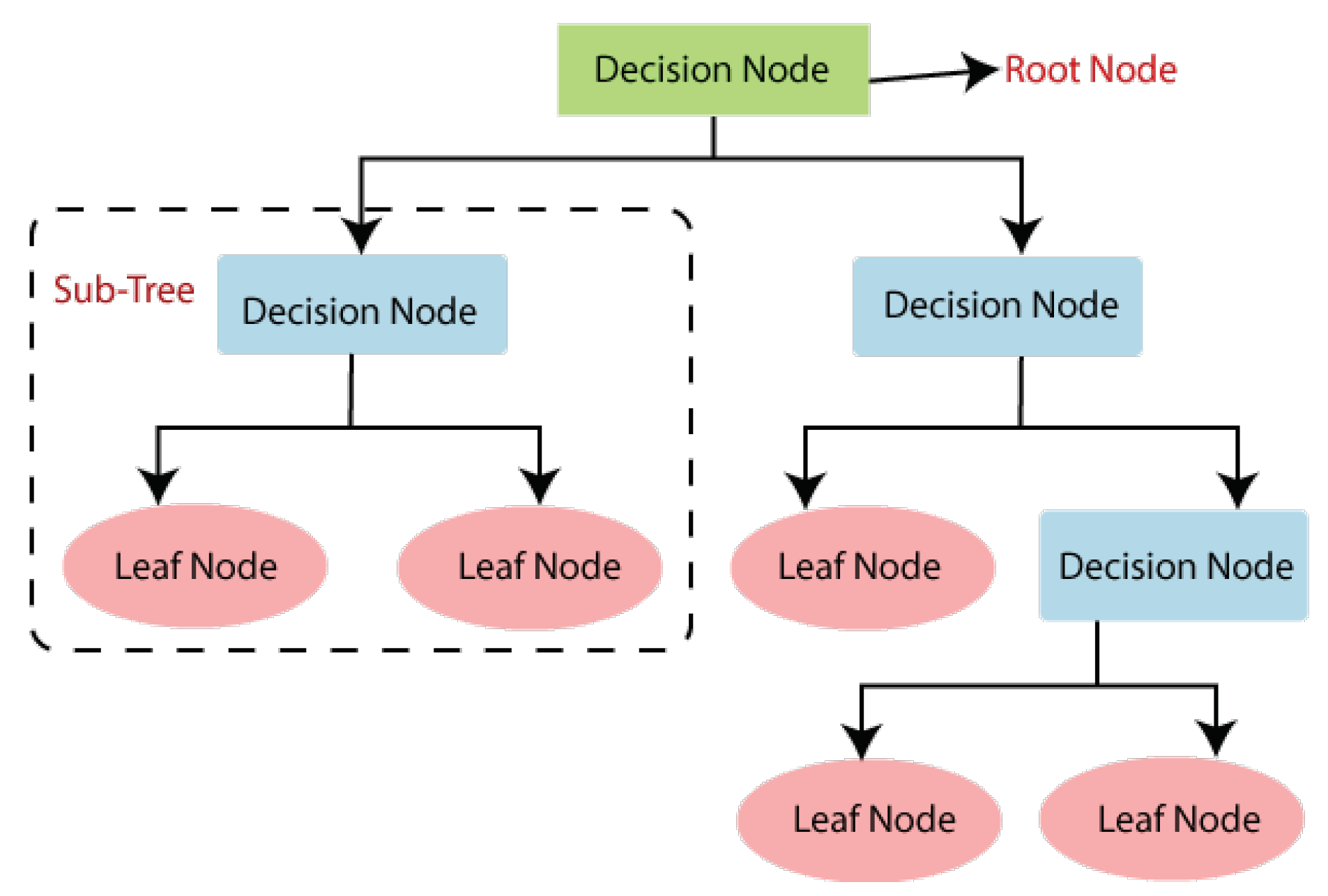
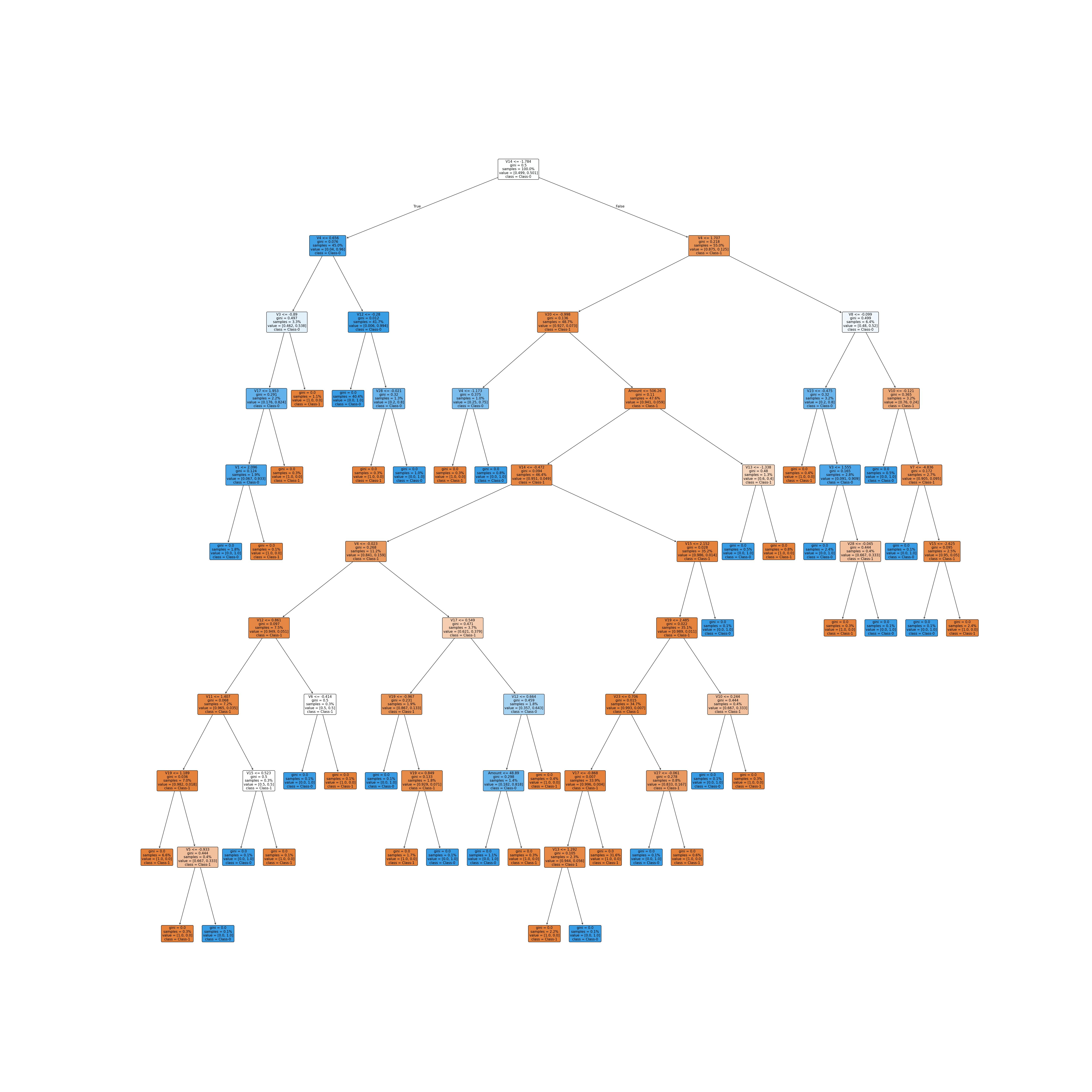


Fig 7: *Decision tree*

The last model created using Jupiter Notebook is Decision Tree; the model managed to score an Accuracy on **Training data of 89.847%** (Fig 7), while it scored an Accuracy score on **Testing Data of 89.847 %**, as presented in Figure 7.

Fig 7: *Accuracy of Decision Tree*

*Fig 8: Decision Tree*

**5.3. K-Nearest Neighbors Algorithm:**

This supervised learning technique achieves consistently high performance compared to other fraud detection techniques of supervised statistical pattern recognition. Three factors majorly affect its performance distance to identify the least distant neighbours. There are some rules to deduce a categorization from the k-nearest neighbour and the count of neighbours to label the new sample. This algorithm classiﬁes transactions by computing the least distant point to this particular transaction. If this least distant neighbour is classiﬁed as fraudulent, the latest marketing is also labelled as fraudulent. Euclidean distance is an excellent choice to calculate the distances in this scenario. This technique is fast and results in fault alerts. Its performance can be improved by distance metric optimization.

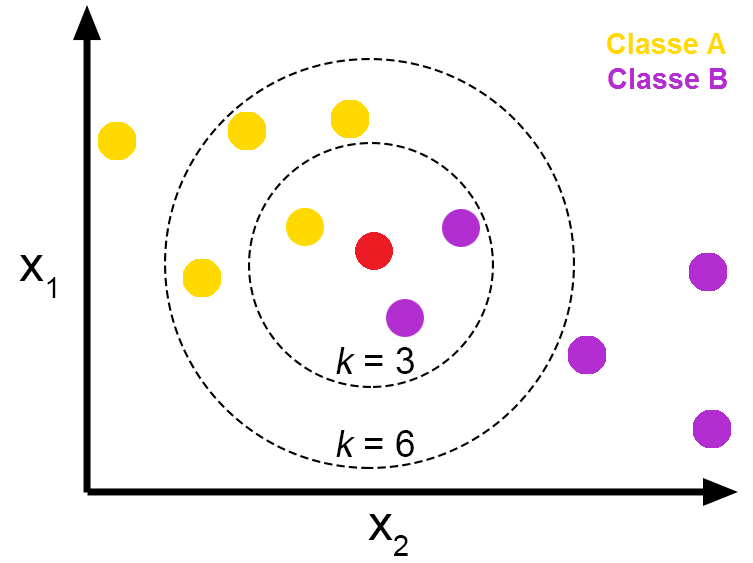
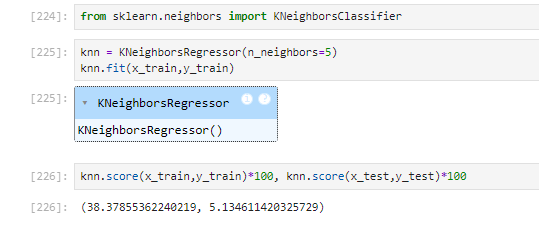
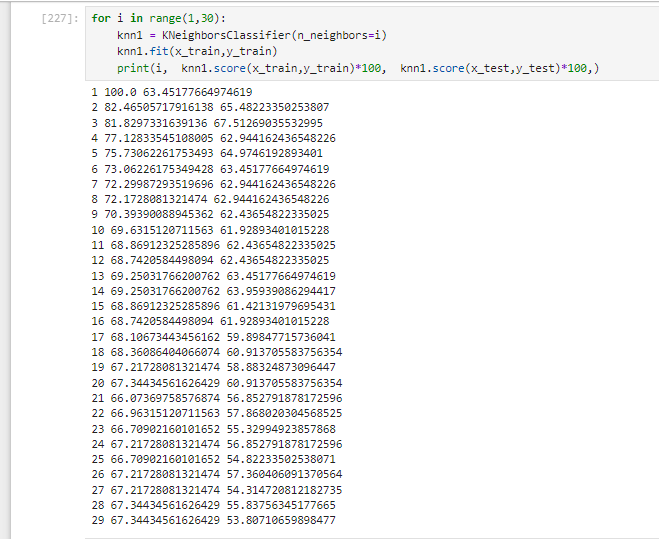


Fig 9: Pros and Cons of K-Nearest Neighbors - From The GENESIS

The last model created using Jupiter Notebook is K-Nearest Neighbors; the model managed to score an Accuracy on **Training data of 38.37 %** (Fig 10), while it scored an Accuracy score on **Testing Data of 5.134 %**, as presented in Figure 10.

Fig 11: *Accuracy of K-Nearest Neighbors Algorithm*

*To improve the accuracy of training and testing data of the datasets by using some parameters inside k-nearest neighbors' algorithm.*

Fig 12: *To improve* *Accuracy of K-Nearest Neighbors*

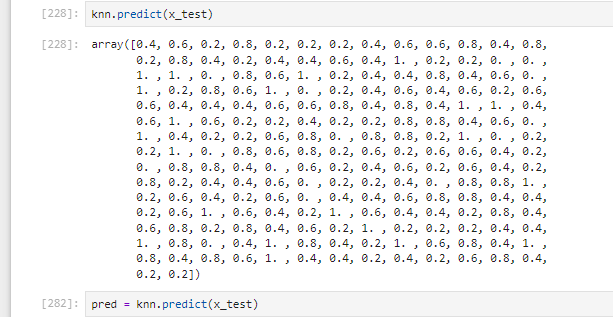
Fig 13: *Predict value*

Fig 14: *Error Rate of training and testing data*

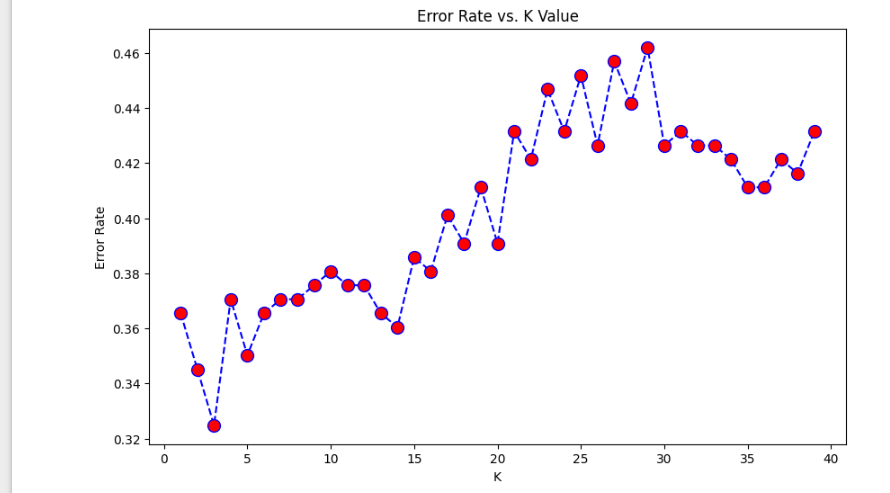
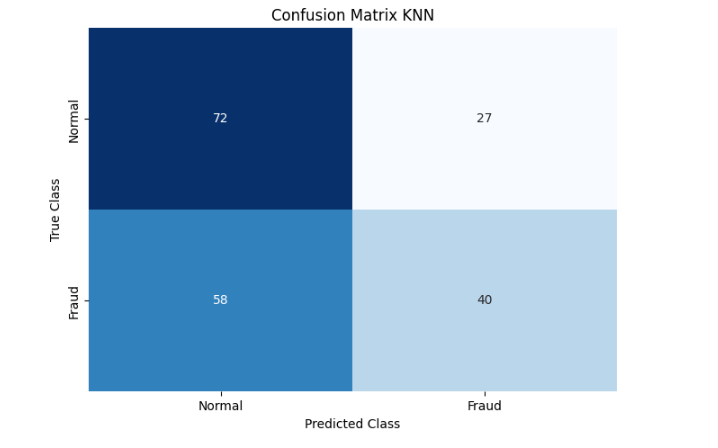
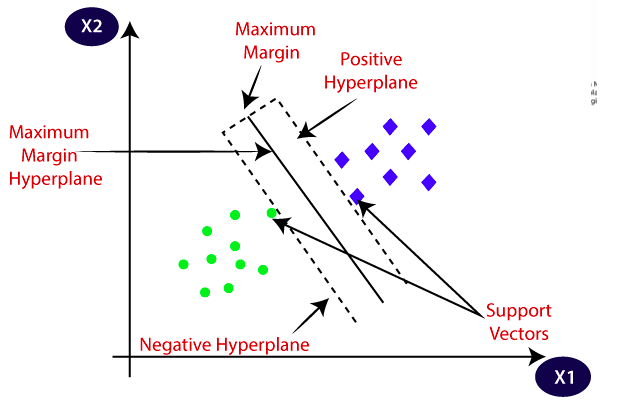
*Fig 15: Error rate*

Fig 16: *Accuracy and Confusion Matrix*

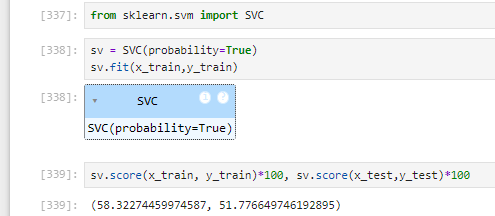
*Fig 17: Confusion Matrix*

**5.4Support Vector Machine (SVM) Algorithm:**

Support vector machines or SVMs are linear classiﬁers, as stated in that work in high dimensionality because, in high dimensions, a non-linear task in input becomes linear. Hence, this makes SVMs highly useful for detecting Fraud. Its two most important features that is a kernel function to represent the classiﬁcation function in the dot product of input data point projection and the fact that it tries ﬁnding a hyperplane to maximize separation between classes while minimizing overﬁtting of training data; it provides a very high generalization capability.

Fig 18: *Support Vector Machine (SVM)*

The last model created using Jupiter Notebook is Support Vector Machine (SVM); the model managed to score an Accuracy on **Training data of 58.3227 %** (Fig 10), while it scored an Accuracy score on **Testing Data of 51.776 %**, as presented in Figure 10.

Fig 19: *Accuracy of training and testing data*

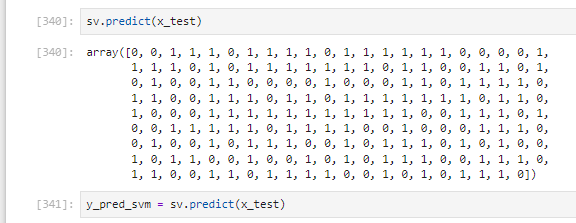
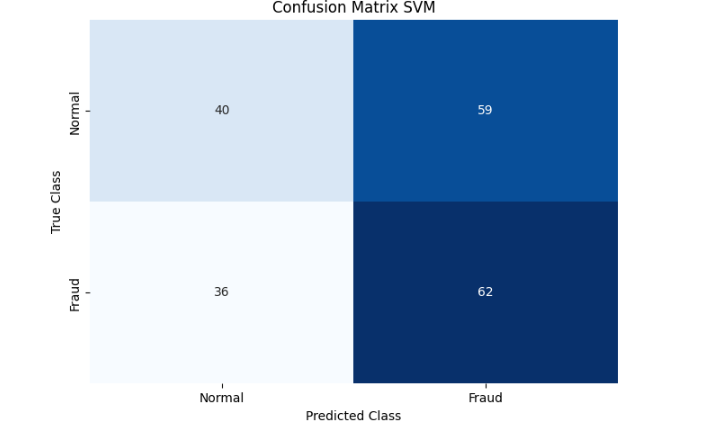


Fig 20: *Predict value*

Fig 21: *Accuracy and Confusion Matrix values*

*Fig 22: Confusion Matrix*

* **Evaluation and Deployment:**

The last stage of the CRISP-DM model is the evaluation and deployment stage, as presented in Table 2 below. All models are being compared to determine the best model for identifying fraudulent credit card transactions.

Accuracy is the overall number of instances that are predicted correctly; accuracies are represented by a confusion matrix where it shows the True Positive (T.P.), True Negative (T.N.), False Positive (F.P.) and False Negative (F.N.). True Positive represents the transactions that are fraudulent and were correctly classified by the model as fraudulent. True Negative represents the not fraudulent transactions that the model correctly predicted as not fraudulent. The third rating is False positive, which represents the fraudulent transaction but was misclassified as not fraudulent. Moreover, finally, False Negative, which are the not fraudulent transactions identified as fraudulent; Table 1 below shows the confusion matrix.

|  |  |  |
| --- | --- | --- |
| Actual/Predicted Values | Positive | Negative |
| Positive | T.P | F.N |
| Negative | F.P | T.P |

Table 1: *Confusion Matrix*

The table above shows all the components to calculate the Accuracy of a model, which is displayed in the below equation.

𝐴𝑐𝑐𝑢𝑟𝑎𝑐𝑦 =

|  |  |  |
| --- | --- | --- |
| Models | | Accuracy |
| Logistic Regression (LR) | Training Data | 94.53% |
| Testing Data | 94.92% |
| Decision tree (DT) | Training Data | 92.38 % |
| Testing Data | 92.38% |
| K-Nearest Neighbors (KNN) | KNN | 100.0% |
| Support Vector Machine (SVM) | SVM | 58.68% |

Table 2: Table of Accuracies

Table 2 shows all of the accuracies of all the models that were created in the project; all models performed well in detecting fraudulent transactions and managed to score high accuracies. Out of all the models, the model that scored the best is KNN is 100% and Decision Tree is 92.38%, the third place is the Support Vector Machine, and the model that scored the lowest Accuracy out of all models is Logistic Regression with a score of 94.53%.

In conclusion, the main objective of this project was to find the most suited model for credit card fraud detection in terms of the machine learning techniques chosen for the project. It was met by building the four models and finding the accuracies of them all; the best model in terms of accuracies is KNN and Decision Tree, which scored 100%. We believe that using the model will help decrease the amount of credit card fraud and increase the customer's satisfaction as it will provide them with a better experience and feeling secure.

There are many ways to improve the model, such as using it on different datasets with various sizes and data types or by changing the data splitting ratio and viewing it from a different algorithm perspective.