Credit Card Fraud Detection Using Machine Learning

KASIREDDY THONTLA , IIIT-SRIKAKULAM

ABSTRACT

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.

LIST OF ABBREVIATIONS

ML	Machine Learning
CCFD	Credit Card Fraud Detection
KNN	K- Nearest Neighbors
SVM	Support Vector Machine
L.R.	Logistic Regression
DT	Decision Tree

LIST OF FIGURES

Figure

TABLE OF CONTENTS

CHAPTER I

Introduction

1.1 Overview

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, 70those users own a single card. Reports of Credit card fraud in the U.S. rose by 44.7in 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 48to 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 9to 2020 (Daly, 2021). Those statistics caught We's attention as the numbers have increased drastically and rapidly throughout the years, which motivated We 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?

CHAPTER II

Literature Review

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 theresearch 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, andfinally, SVM scored one of the lower scores, as the speed was low, and the Accuracy wasmedium. As for the cost, All models built were expensive (Zareapoor et al.,

2012).[1]

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 two other classifiers , which are3 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).[2]

Maniraj's team built a model to recognize if any new transaction is Fraud or non-fraud. Their goal was to get 100% in detecting fraudulent transactions andtry to minimize the incorrectly classified fraud instances. Their model has performed well as they got 99.7% of the fraudulent transactions (Maniraj et al., 2019).[3]

The classification approach used by Dheepa and Dhanapal was the behaviour-based classification approach, using a Support Vector Machine, where the behavioural patternsof 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 wasmore than 80% (Dheepa Dhanapal, 2012).[4]

Mailini and Pushpa proposed using KNN and Outlier detection in identifying credit card fraud. After performing their model oversampled data, the authors found that the most suitable method for detecting and determining target instance anomaly is KNN, which showed that it is most suited to detecting Fraud with memory limitation. As for Outlier detection, the computation and memory required for credit card fraud detectionis much less in addition to working faster and better in large online datasets. However, their work and results showed that KNN was more accurate and efficient (Malini Pushpa,2017).[5]

Maes and his team proposed using Bayesian and Neural Networks to detect credit card fraud. Their results showed that Bayesian performance is 8% more effective in detecting Fraud than ANN, meaning BBN sometimes detects 8% more fraudulent transactions. In addition to the Learning times, ANN can go up to several hours,whereas BBN takes only 20 minutes (Maes et al., 2002).[6]

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% (Jain et al., 2019).[7]

Dighe and his team used KNN, Logistic Regression and Neural Networks, multi-layer perceptron and Decision Tree in their work, then evaluated the results regarding numerous accuracy metrics. Of all the models created, the best performing one is KNN, which scored 99.13%, then in second place performing model at 96.40% and in last place is logistic Regression with 96.27% (Dighe et al., 2018).[8]

Sahin and Duman used four Support Vector Machine methods in detecting credit card fraud. (SVM) Support Vector Machine with RBF, Polynomial, Sigmoid, and Linear Kernel,all models scored 99.87% in the training model and 83.02% in the testing part of the model (Sahin Duman, 2011).[9]

CHAPTER III

Project Description

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 mostsuited data and preparing/preprocessing are the first and second steps; after making surethat the data is ready, the modelling phase starts, where four models are created: K-NearestNeighbor (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. Thedataset 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 variableswhere "1" is a case of fraudulent transaction, and "0" is not as case of fraudulent transaction.

Dataset : https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

CHAPTER IV

Data Analysis and Processing

4.1 Data Description

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 modeland obtain visualizations.The second figure shows the class distribution; the red bar, which contains 284,315variables, represents the non-fraudulent transactions, and the blue bar, with 492 variables, represents the fraudulent transactions.

4.2 Analysis

```
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
 4   V4      284807 non-null  float64
 5   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
 9   V9      284807 non-null  float64
 10  V10     284807 non-null  float64
 11  V11     284807 non-null  float64
 12  V12     284807 non-null  float64
 13  V13     284807 non-null  float64
 14  V14     284807 non-null  float64
 15  V15     284807 non-null  float64
 16  V16     284807 non-null  float64
 17  V17     284807 non-null  float64
 18  V18     284807 non-null  float64
 19  V19     284807 non-null  float64
 20  V20     284807 non-null  float64
 21  V21     284807 non-null  float64
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64
dtypes: float64(30), int64(1)
```

Figure 4.1: Data Structure

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05
mean	94813.859575	1.168375e-15	3.416908e-16	-1.379537e-15	2.074095e-15	9.604066e-16	1.487313e-15	-5.556467e-16	1.213481e-16	-2.406331e-15
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00	1.332271e+00	1.237094e+00	1.194353e+00	1.098632e+00
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02	-2.616051e+01	-4.355724e+01	-7.321672e+01	-1.343407e+01
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01	-5.540759e-01	-2.086297e-01	-6.430976e-01
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02	2.235804e-02	-5.142873e-02
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.704361e-01	3.273459e-01	5.971390e-01
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895e+02	2.000721e+01	1.559499e+01

Figure 4.2: Data Description

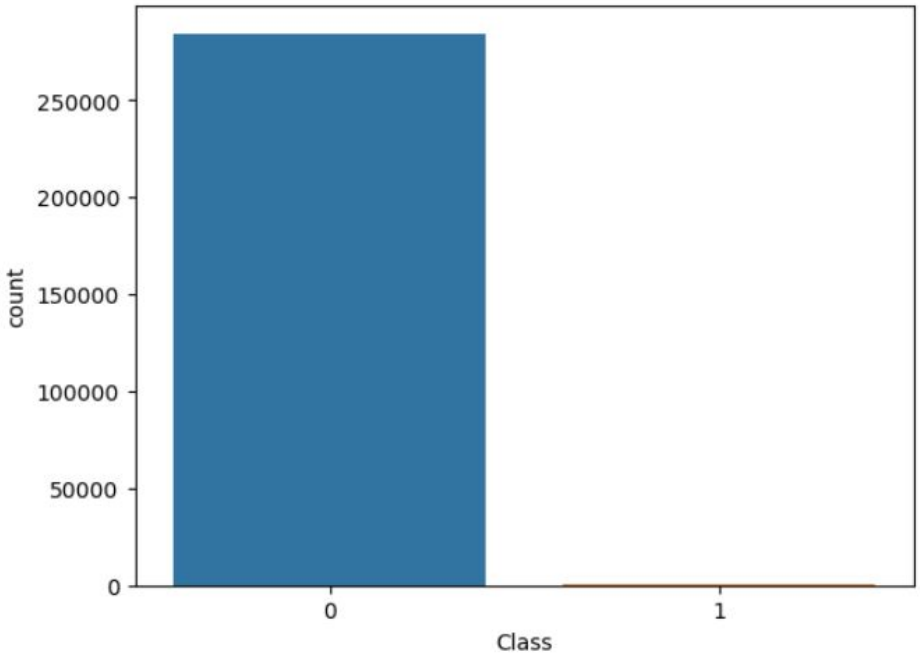

Figure 4.3: Class Count

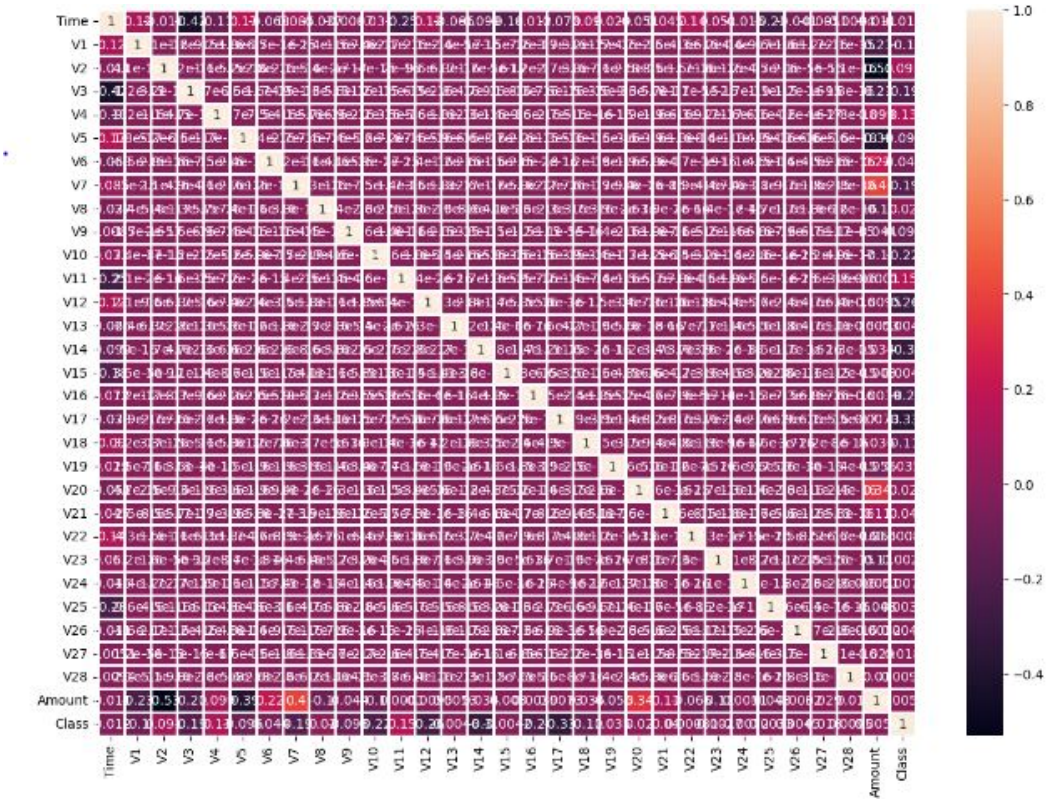

Figure 4.4: Data Correlation

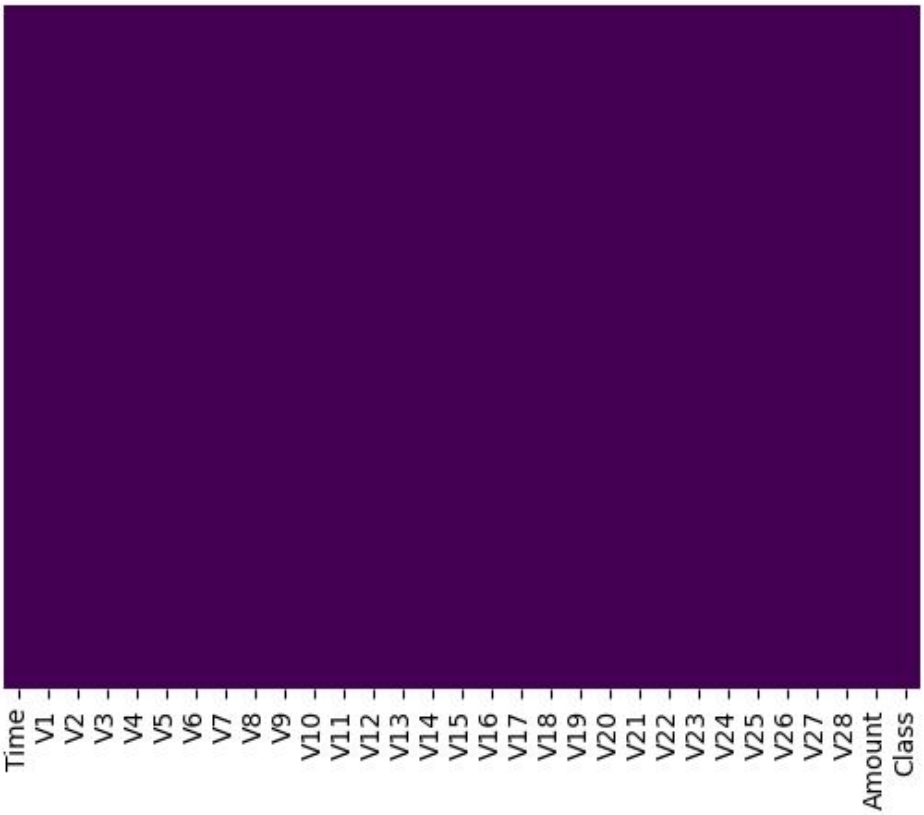

Figure 4.5: Null data

CHAPTER V

Algorithm and Performance Analysis

5.1 K-Nearest Neighbor (KNN):

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.

5.1.1 Algorithm KNN

- Let m be the number of training data samples. Let p be an unknown point that needs to be classified
- Storing the training samples in an array of data points arr[]. Each element of this array denotes a tuple (x, y).
- for i = 0 to m do
- Calculating distance d(arr[i], p)
- end for
- They are making set S of K smallest distances achieved. Each of these distances resembles an already classified data point.

- Returning the majority label among S

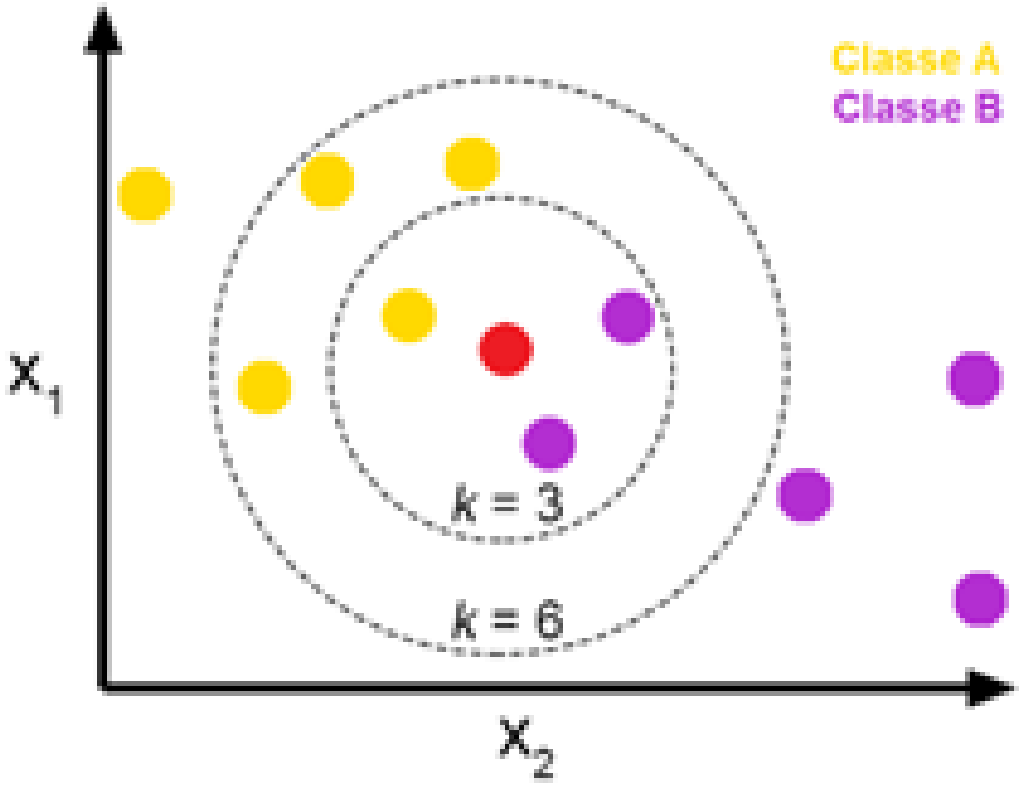

Figure 5.1: KNN

Two Ks were used to determine the best KNN model, K=3 and K =7.

• K = 3 While making the KNN model, We created two models: K =3 and K =7. Figure 5 shows the model created in Jupiter Notebook; the model scored an accuracy of 100% and identified 85,443 transactions correctly and missed 131.

• K = 7

There was a slight decrease in the Accuracy of the model created in Jupiter Notebook (Figure 6) as it scored 100% when K is 7, and the model miss classified 131 fraudulent transactions as no fraudulent. As for the Accuracy is the same as K=3 100% with 52 misclassified transactions; the only difference is within the classifications.

5.2 Logistic Regression (L.R.):

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

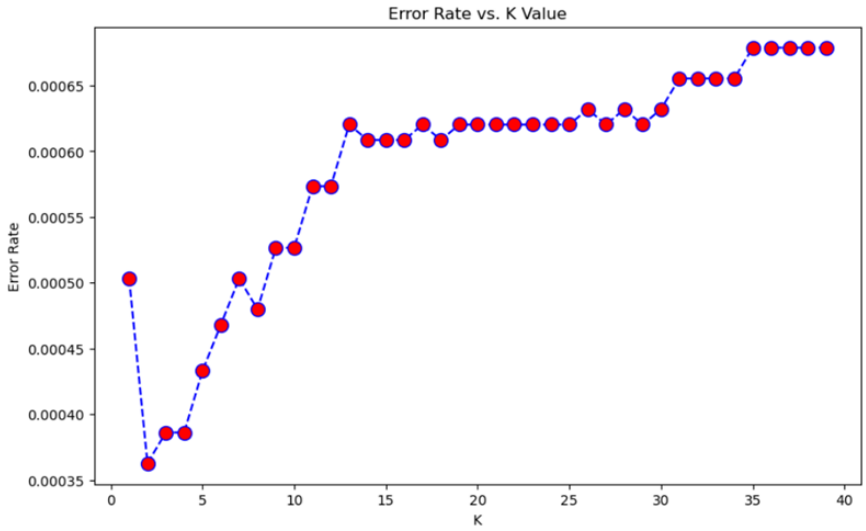

Figure 5.2: Error Rate

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.

The last model created using Jupiter Notebook is Logistic Regression; the model managed to score an Accuracy on Training data of 93.51% , while it scored an Accuracy score on Test Data of 91.88%, as presented in blew Figure.

5.3 Support Vector Machine (SVM):

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.

Finally, the model Support Vector Machine, as shown in Figure 12, scored 97.59% for the Accuracy.

```
WITH k=3


[[85307     5]
 [   28   103]]


              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85312
           1       0.95      0.79      0.86       131

    accuracy                           1.00     85443
   macro avg       0.98      0.89      0.93     85443
weighted avg       1.00      1.00      1.00     85443
```

Figure 5.3: K=3

5.4 Decision Tree (D.T.):

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.

Algorithm: of Decision Tree

- Create (T)
- Calculate frequencies (Ci, T)
- If all instances belong to the same class, the returning leaf
- for every Attribute, a test is set for splitting criteria. An attribute that satisfies the test is test node K
- Repeating Create (Ti) on each partition Ti.Adding those nodes as children of node K

```
WITH k=7


[[85300    12]
 [   31   100]]


              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85312
           1       0.89      0.76      0.82       131

    accuracy                           1.00     85443
   macro avg       0.95      0.88      0.91     85443
weighted avg       1.00      1.00      1.00     85443
```

Figure 5.4: K=7

5.5 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.

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

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 and Decision Tree as its Accuracy is 100%, the third place is the Support Vector

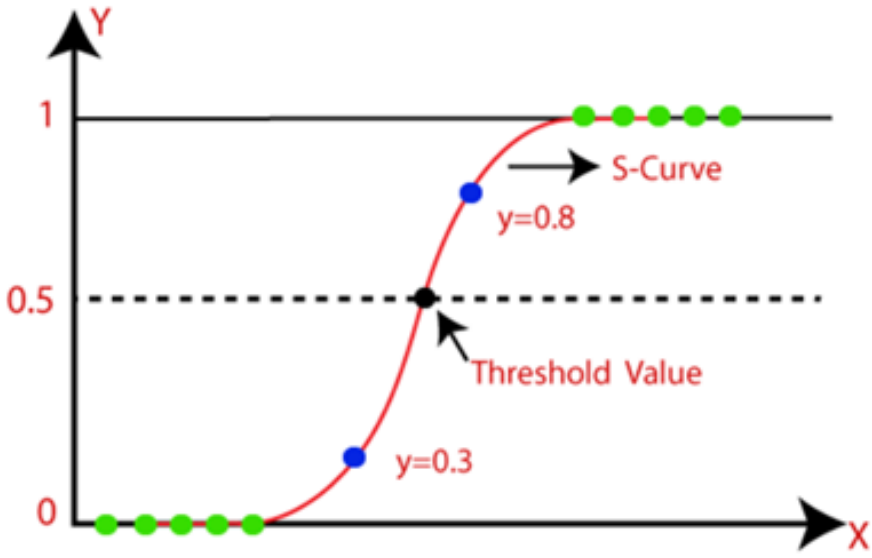

Figure 5.5: Logistic Regression Model

```
In [29]: print('Accuracy on Training data : ', training_data_accuracy)

         Accuracy on Training data :  0.9351969504447268
```

Figure 5.6: Accuracy on Train data

Machine, and the model that scored the lowest Accuracy out of all models is Logistic Regression with a score of 93.51%.

```
In [31]: print('Accuracy score on Test Data : ', test_data_accuracy)

         Accuracy score on Test Data :  0.9187817258883249
```

Figure 5.7: Accuracy on Test data

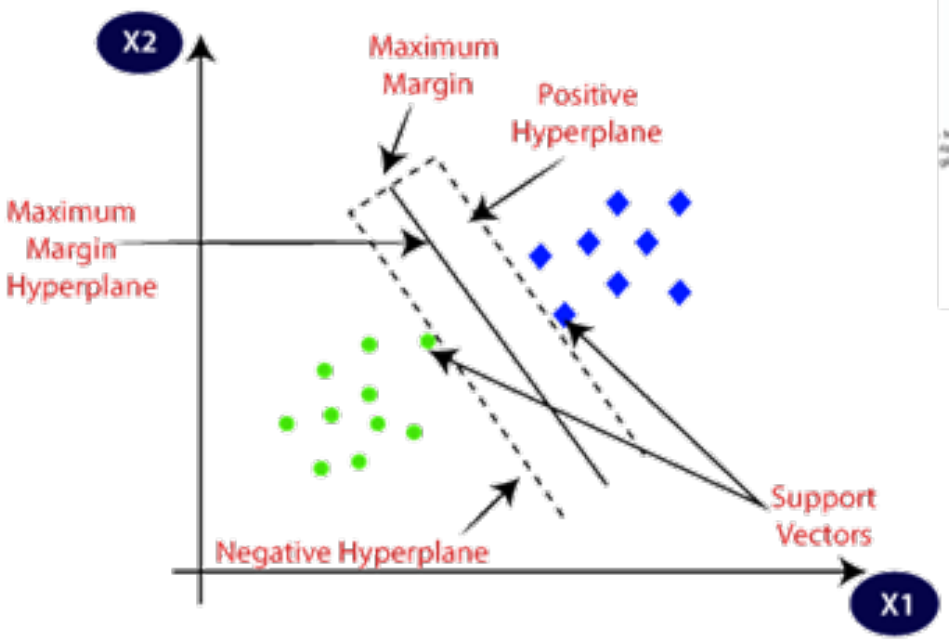

Figure 5.8: Support Vector Machine Algorithm

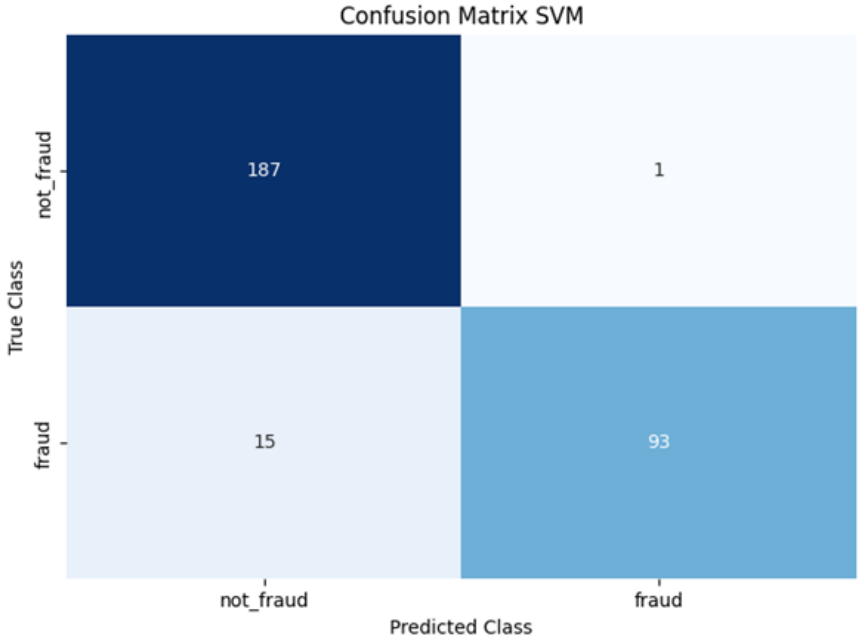

Figure 5.9: SVM Confusion Matrix

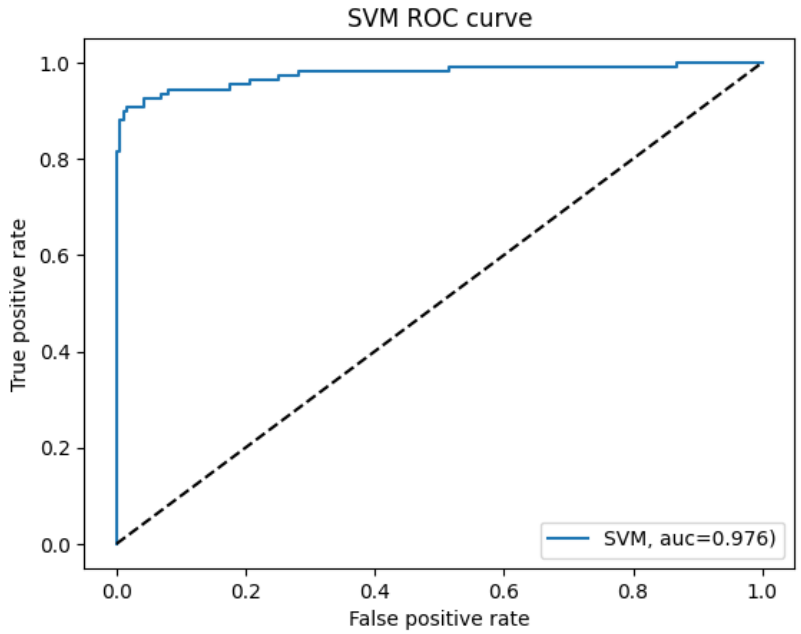

Figure 5.10: SVM ROC Curve

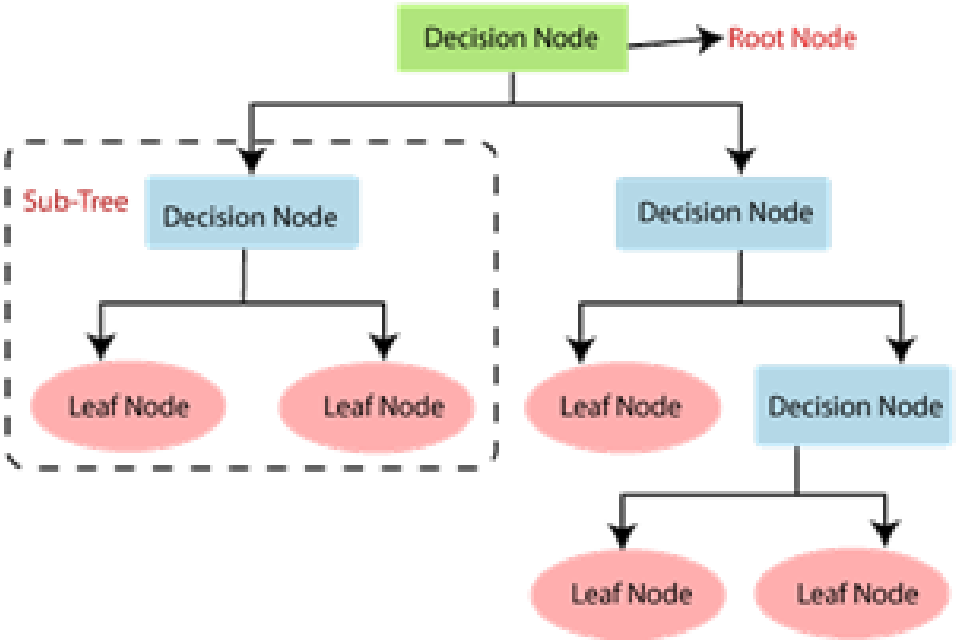

Figure 5.11: Decision Tree Algorithm

	precision	recall	f1-score	support
0	1.00	1.00	1.00	85285
1	0.83	0.81	0.82	158
accuracy			1.00	85443
macro avg	0.92	0.90	0.91	85443
weighted avg	1.00	1.00	1.00	85443

Figure 5.12: D.T. Accuracy

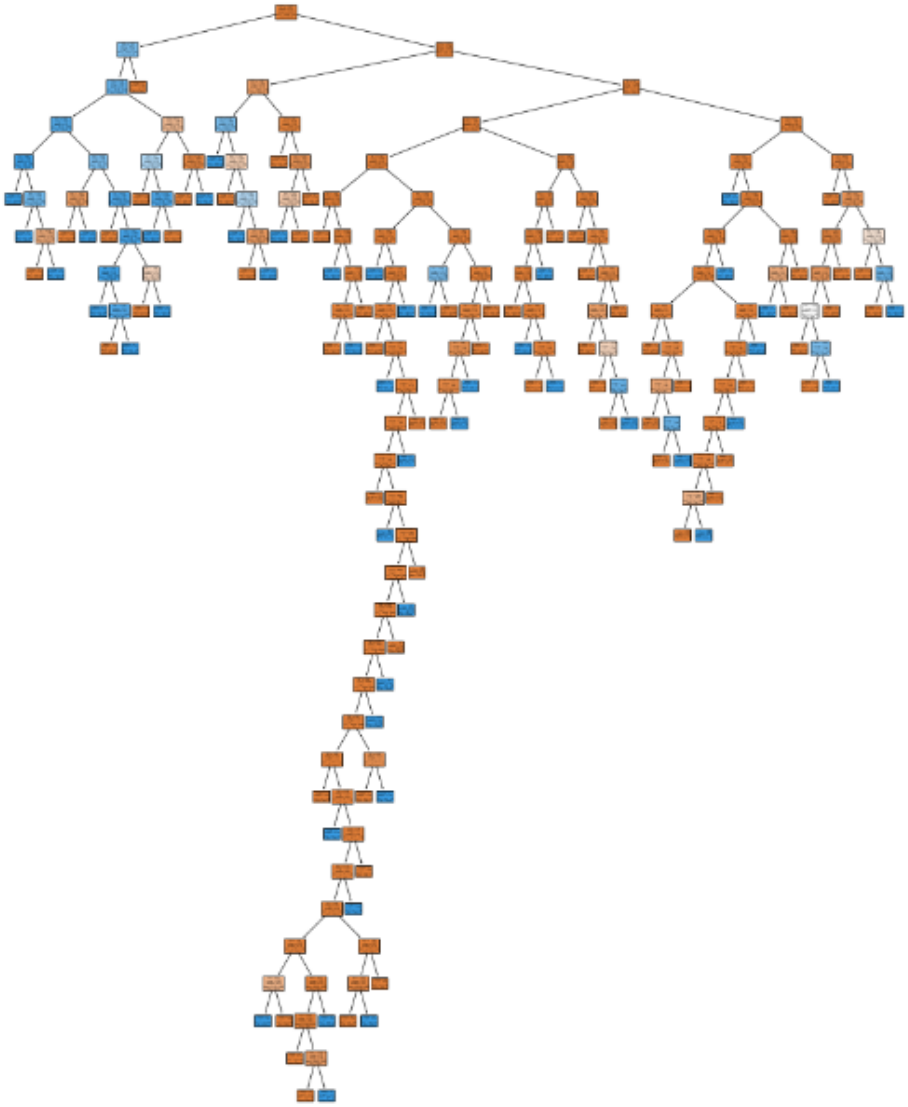

Figure 5.13: Decision Tree

Actual/Predicted	Positive	Negative
Positive	TP	FN
Negative	F.P.	TN

Figure 5.14: Confusion Matrix

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Model		Accuracy
KNN	K = 3	100%
	K = 7	100%
Logistic Regression	Training Data	93.51%
	Test Data	91.88%
Support Vector Machine	SVM	97.59%
Decision Tree	DT	100%

Figure 5.15: Table of Accuracy

CHAPTER VI

Future Work & Conclusion

6.1 Future Work

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. An example can be merging telecom datato calculate the location of people to have better knowledge of the location of the card owner while his/her credit card is being used; this will ease the detection because if the card owner is in Dubai and a transaction of his card was made in Abu Dhabi, it will easilybe detected as Fraud..

6.2 Conclusion

In conclusion, the main objective of this project was to find the most suited model for creditcard 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 bestmodel in terms of accuracies is KNN and Decision Tree, which scored 100the amount of credit card fraud and increase the customer's satisfaction as it will provide themwith a better experience and feeling secure.

References

[1] Z. et al, "Analysis on credit card fraud detection techniques: Based on certain design criteria." https://research.ijcaonline.org/volume52/number3/pxc3881538.pdf, 2012. Accessed: 26-oct-2023.

[2] . A. N. O. Alenzi, H. Z., "Fraud detection in credit cards using logistic regression." https://thesai.org/Publications/ViewPaper?Volume=11&Issue=12&Code=IJACSA&SerialNo=65, 2020. Accessed: 26-oct-2023.

[3] S. A. A. S. . S. S. D. Maniraj, S. P., "Credit card fraud detection using machine learning and data science." https://doi.org/10.17577/ijertv8is090031, 2019. Accessed: 25-oct-2023.

[4] . D. R. Dheepa, V., "Analysis on credit card fraud detection techniques: Based on certain design behaviour-based credit card fraud detection using support vector machinescriteria." https://doi.org/10.21917/ijsc.2012.0061, 2012. Accessed: 26-oct-2023.

[5] . P. M. Malini, N., "Analysis of credit card fraud identification techniques based on knn and outlier detection." https://doi.org/10.1109/aeeicb.2017.7972424, 2017. Accessed: 26-oct-2023.

[6] T. K. V. B. . M. B. Maes, S., "Credit card fraud detection using bayesian and neural networks." https://www.ijert.org/research/credit-card-fraud-detection-using-machine-learning-and-data-science-IJERTV8IS0
pdf, 2002. Accessed: 23-oct-2023.

[7] N. S. . J. S. Jain, Y., "A comparative analysis of various credit card fraud detection techniques," 2019. Accessed: 25-oct-2023.

[8] P. S. . K. S. Dighe, D., "Detection of credit card fraud transactions using machine learning algorithms and neural networks." https://doi.org/10.1109/iccubea.2018.8697799, 2018. Accessed: 26-oct-2023.

[9] . D. E. . Sahin, Y., "Detecting credit card fraud by decision trees and support vector machines," 2011. Accessed: 23-oct-2023.