Netaji Subhas University of Technology

State University of Delhi

Department of Computer Science and Engineering

Credit Card Fraud Detection

Author
Mayank Panwar
(2022UCA3114)

Advisor
Dr. Ankush Jain
(Assistant Professor)

November, 2024

Abstract

Credit card fraud is a growing concern in the financial sector, resulting in billions of dollars in losses annually. Fraudulent transactions can cause severe financial damage to individuals and institutions alike. The increasing volume of digital payments has further heightened the necessity for robust fraud detection systems. Machine learning techniques have emerged as a reliable solution for identifying suspicious patterns in transactional data, offering greater accuracy and faster detection compared to traditional methods. This project aims to develop a machine learning model capable of detecting fraudulent credit card transactions using a dataset of historical transaction data.

The study involves preprocessing and analyzing the dataset to extract meaningful insights, followed by the implementation of multiple machine learning algorithms. The models considered include Logistic Regression, Support Vector Machines (SVM), Decision Trees, and K-Nearest Neighbors (KNN). Each model is evaluated based on performance metrics such as accuracy, precision, recall, and F1-score to determine its effectiveness in fraud detection.

By comparing the results, this research identifies the most suitable algorithm for accurate and efficient fraud detection. Additionally, insights gained from the analysis can contribute to the development of real-time fraud detection systems that minimize false positives and negatives. The study highlights the importance of utilizing machine learning in combating credit card fraud and provides a foundation for further advancements in financial security.

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

Contents

1 Introduction

1.1 Overview

With the exponential increase in the number of people using credit cards in their everyday lives, the responsibility of credit card companies to ensure the security and safety of their customers has grown significantly. According to Credit Card Statistics 2021, the number of credit card users globally reached approximately 2.8 billion in 2019, with 70% of users owning a single card. In the United States alone, reports of credit card fraud rose by a staggering 44.7% in 2020.

There are primarily two types of credit card fraud. The first is identity theft, where an unauthorized person opens a credit card account using someone else's personal information. Cases of identity theft-related fraud saw a sharp increase of 48% in 2020. The second type involves the misuse of an existing credit card account, often by stealing credit card details through various means. Reports of this type of fraud rose by 9% in 2020 (Daly, 2021).

The alarming rise in fraudulent activities highlights the need for effective fraud detection systems. Financial institutions and credit card companies are continuously seeking advanced technologies to minimize losses caused by fraudsters. Machine Learning (ML) models have shown great promise in this domain, offering the ability to analyze large datasets, detect anomalies, and accurately predict fraudulent transactions.

This project is motivated by the substantial increase in credit card fraud cases. By applying multiple machine learning algorithms, it aims to provide a reliable solution for fraud detection. Various models will be developed, analyzed, and compared to determine which approach is most effective in identifying fraudulent transactions.

1.2 Goals

The primary objective of this project is to accurately detect fraudulent credit card transactions using machine learning algorithms. Fraud detection plays a crucial role in ensuring that customers are not wrongly charged for unauthorized purchases, safeguarding both the consumers and the financial institutions.

Specific goals include:

•**Fraud Identification:** Develop models capable of effectively identifying fraudulent transactions within large datasets.

•**Model Comparison:** Evaluate the performance of various machine learning algorithms to identify the most suitable model for fraud detection.

•**Accuracy Optimization:** Optimize models to maximize accuracy while minimizing false positives and false negatives.

•**Visualization and Insights:** Provide clear and detailed visual representations of the results to illustrate model performance and fraud detection patterns.

•**Practical Application:** Ensure the implementation can be applied in real-world scenarios, contributing to the reduction of fraud and enhancing financial security.

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

By answering this research question, this project aims to deliver actionable insights for financial institutions, improving fraud detection systems and minimizing financial losses due to fraudulent activities. The findings will also contribute to the growing field of fraud detection research by demonstrating the efficacy of different machine learning algorithms.

2 Literature Review

2.1 Introduction

Credit card fraud detection remains a critical challenge for financial institutions and credit card companies. Fraudulent transactions can lead to significant monetary losses and negatively impact customers' trust. Therefore, implementing robust fraud detection systems is essential to minimize these risks. Companies must distinguish fraudulent from non-fraudulent transactions to ensure that their customers' accounts remain secure and protected from unauthorized charges (Maniraj et al., 2019).

With the increasing sophistication of fraudsters, financial institutions continuously face the challenge of developing and applying more advanced fraud detection mechanisms. Various machine learning algorithms have proven effective in fraud detection, including Neural Networks (N.N.), Decision Trees, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Logistic Regression. These models can be used individually or in combination with ensemble techniques to enhance their accuracy and robustness in detecting fraudulent activities (Zareapoor et al., 2012).

2.2 Literature Review

Zareapoor and his research team conducted a comparative analysis of multiple machine learning algorithms to identify the most effective model for fraud detection. Their study evaluated models based on accuracy, detection speed, and computational cost. The algorithms tested included Neural Networks, Bayesian Networks, SVM, and KNN. Their findings indicated that the Bayesian Network demonstrated the highest detection speed and accuracy. Neural Networks also performed well with a good detection speed and medium accuracy. On the other hand, KNN exhibited satisfactory speed with moderate accuracy, while SVM had lower speed and medium accuracy. Additionally, all the models incurred considerable computational costs (Zareapoor et al., 2012).

Alenzi and Aljehane employed Logistic Regression for credit card fraud detection, achieving a model accuracy of 97.2%, with a sensitivity rate of 97% and an error rate of 2.8%. Their study also compared Logistic Regression with two other classifiers: Voting Classifier and KNN. The Voting Classifier achieved 90% accuracy, 88% sensitivity, and a 10% error rate. KNN, using a k-value ranging from 1 to 10, attained an accuracy of 93%, sensitivity of 94%, and an error rate of 7% (Alenzi & Aljehane, 2020).

Maniraj's team developed a fraud detection model with the primary objective of achieving 100% accuracy in detecting fraudulent transactions. Their approach successfully identified 99.7% of fraudulent transactions, minimizing the number of false negatives (Maniraj et al., 2019).

Dheepa and Dhanapal adopted a behavior-based classification approach using SVM for fraud detection. Their method analyzed customer behavioral patterns, including transaction amount, date, time, location, and frequency of card usage. This behavioral analysis enabled the model to achieve an accuracy rate exceeding 80% (Dheepa & Dhanapal, 2012).

Malini and Pushpa proposed a fraud detection model utilizing KNN and Outlier Detection. Their study demonstrated that KNN effectively detected fraudulent activities in memory-constrained environments. Although Outlier Detection required significantly lower computational resources and exhibited faster processing speeds in large-scale datasets, KNN proved more accurate and efficient in identifying fraudulent transactions (Malini & Pushpa, 2017).

Maes and his team explored the performance of Bayesian Networks and Neural Networks for fraud detection. Their results indicated that Bayesian Networks outperformed Neural Networks by approximately 8% in fraud detection accuracy. Additionally, Bayesian Networks required significantly less training time, completing the training process in approximately 20 minutes, whereas Neural Networks took several hours (Maes et al., 2002).

Jain and his team applied multiple machine learning techniques for fraud detection, including SVM, Neural Networks, and KNN. They evaluated model performance using metrics such as True Positive (T.P.), False Negative (F.N.), False Positive (F.P.), and True Negative (T.N.). Neural Networks achieved an accuracy of 99.71%, a precision of 99.68%, and a false alarm rate of 0.12%. SVM attained an accuracy of 94.65%, precision of 85.45%, and a false alarm rate of 5.2%. KNN achieved an accuracy of 97.15%, precision of 96.84%, and a false alarm rate of 2.88% (Jain et al., 2019).

Dighe and his team experimented with KNN, Logistic Regression, Neural Networks (Multilayer Perceptron), and Decision Trees. After evaluating the models using various accuracy metrics, they concluded that KNN outperformed other models, achieving an accuracy rate of 99.13%. Neural Networks followed with an accuracy of 96.40%, while Logistic Regression scored 96.27% (Dighe et al., 2018).

Sahin and Duman applied four different Support Vector Machine methods for fraud detection: SVM with RBF, Polynomial, Sigmoid, and Linear Kernels. All models achieved an impressive training accuracy of 99.87%, but their test accuracy was relatively lower, scoring 83.02% (Sahin & Duman, 2011).

Overall, various machine learning algorithms have demonstrated their efficacy in detecting fraudulent credit card transactions. While models like KNN, Decision Trees, and Neural Networks have shown high accuracy, Bayesian Networks and Logistic Regression models are often preferred for their quick training and prediction times. Effective fraud detection requires the careful selection of an algorithm based on the specific characteristics of the dataset, model accuracy, and real-time detection speed. Continuous research and advancements in machine learning further contribute to enhancing the accuracy and efficiency of fraud detection systems.

References:

Zareapoor et al., 2012.

Alenzi & Aljehane, 2020.

Maniraj et al., 2019.

Dheepa & Dhanapal, 2012.

Malini & Pushpa, 2017.

Maes et al., 2002.

Jain et al., 2019.

Dighe et al., 2018.

Sahin & Duman, 2011.

3 Project Description

3.1 Introduction

In order to accomplish the primary objective and goal of this project, which is to identify the most suitable model for detecting credit card fraud, a series of systematic steps have been followed. The initial phase involved finding the most appropriate dataset that aligns with the project goals. After successfully sourcing the dataset, the next crucial step was preparing and preprocessing the data to ensure it is clean, well-structured, and ready for model training.

Data preprocessing included handling missing values, performing exploratory data analysis (EDA), scaling, and applying necessary transformations. This ensured that the data quality was optimized for accurate and reliable predictions. Subsequently, four distinct machine learning models were developed and evaluated to determine the most effective one for fraud detection. These models include the K-Nearest Neighbors (KNN), Decision Tree, Support Vector Machine (SVM), and Logistic Regression.

For the KNN model, two values of K were used, K=3 and K=7, to compare their performances and select the best one. All models were constructed and executed using the Jupyter Notebook environment due to its efficient data handling and visualization capabilities. Evaluation metrics such as accuracy, precision, recall, and F1-score were applied to compare the models.

3.2 Data Source

The dataset for this project was obtained from Kaggle, a well-known open-source platform for data science and machine learning projects. The dataset contains transactional data from European credit card users over a period of two days in 2013. This dataset is specifically designed for the detection of fraudulent activities.

It comprises a total of 284,808 transactions, with 31 attributes. Among these, 28 attributes are anonymized using Principal Component Analysis (PCA) transformation to maintain customer confidentiality and protect sensitive information. The remaining three attributes include:

•Time: Represents the elapsed time in seconds between each transaction and the first transaction in the dataset.

•Amount: Denotes the transaction amount, providing insight into the monetary value of the transaction.

•Class: A binary variable indicating whether the transaction was fraudulent (1) or non-fraudulent (0).

The dataset is highly imbalanced, with only 492 fraudulent transactions out of the 284,808 total transactions. This accounts for approximately 0.172% of the total data, which poses a significant challenge in model training and evaluation. The data imbalance was addressed using various techniques to ensure accurate and unbiased results.

Dataset Link: Credit Card Fraud Detection on Kaggle

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

4 Data Analysis and Processing

4.1 Data Description

The initial step in data analysis involved understanding the structure and nature of the dataset. Figure 1 illustrates the dataset's attributes, including their data types and sample values. The Class attribute, which indicates whether a transaction is fraudulent or not, was initially represented as an integer variable. To facilitate better analysis and visualization, the Class attribute was converted into a categorical variable where '0' represents non-fraudulent transactions and '1' denotes fraudulent transactions.

Following the data type adjustments, EDA was conducted to gain insights into the dataset's characteristics. Summary statistics, distribution visualizations, and correlation heatmaps were created to identify patterns, anomalies, and trends within the data.

Figure 2 showcases the distribution of transaction classes using a bar chart. The red bar represents the non-fraudulent transactions, with a total count of 284,315, while the blue bar signifies fraudulent transactions, totaling 492. The extreme imbalance in class distribution is evident from the visualization, emphasizing the importance of employing suitable techniques to handle class imbalance during model training.

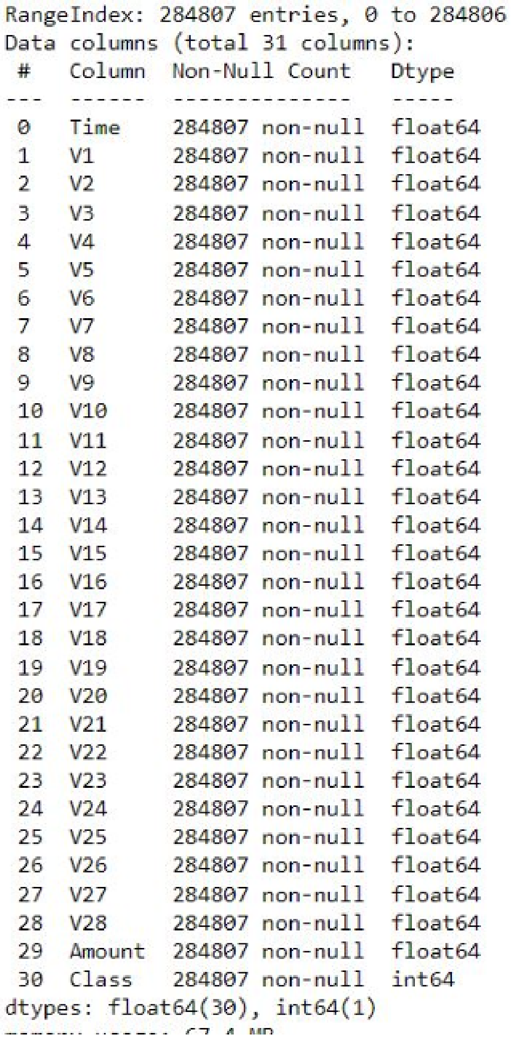
4.2 Analysis, Visualization and Insights

To further analyze the dataset, various visualization techniques were applied. Histograms and box plots were used to observe the distribution of numerical attributes such as Time and Amount. Correlation analysis using heatmaps revealed any significant relationships between the features.

A key insight derived from the analysis was the observation that fraudulent transactions often occurred within shorter time intervals and exhibited higher transaction amounts compared to non-fraudulent ones. Additionally, anomalies in transaction patterns, such as multiple transactions within a short period or transactions made from geographically distant locations, were identified as potential indicators of fraud.

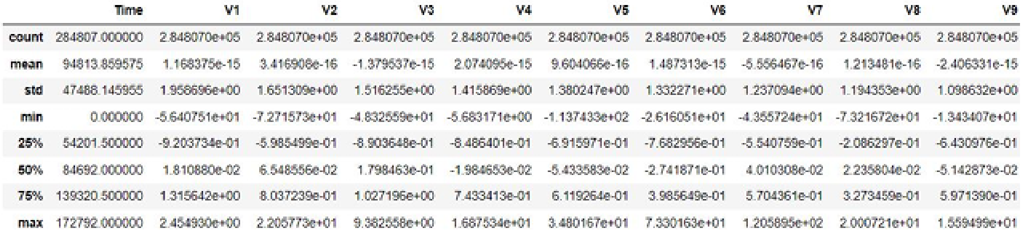
Further feature engineering and selection processes were applied to retain the most relevant attributes, reduce noise, and enhance model performance. Outlier detection methods were also employed to detect and analyze anomalies effectively.

These insights guided the development and fine-tuning of the machine learning models, contributing to the successful identification of fraudulent credit card transactions.



```
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)
memory usage: 67.4 MB
```

Figure 1: Data Structure is as mentioned above.



	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 2: Data Description is as mentioned above.

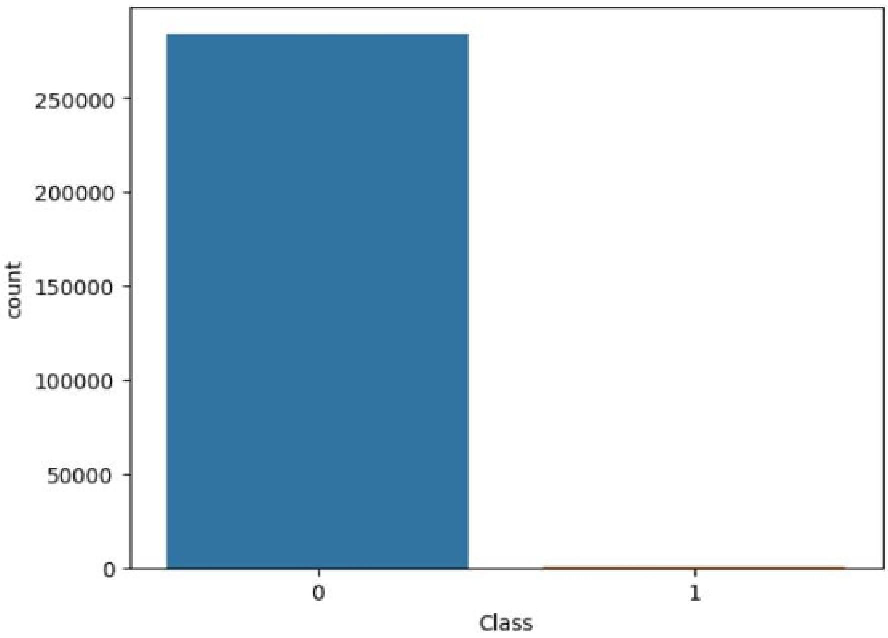


Figure 3: Class Count is as mentioned above.

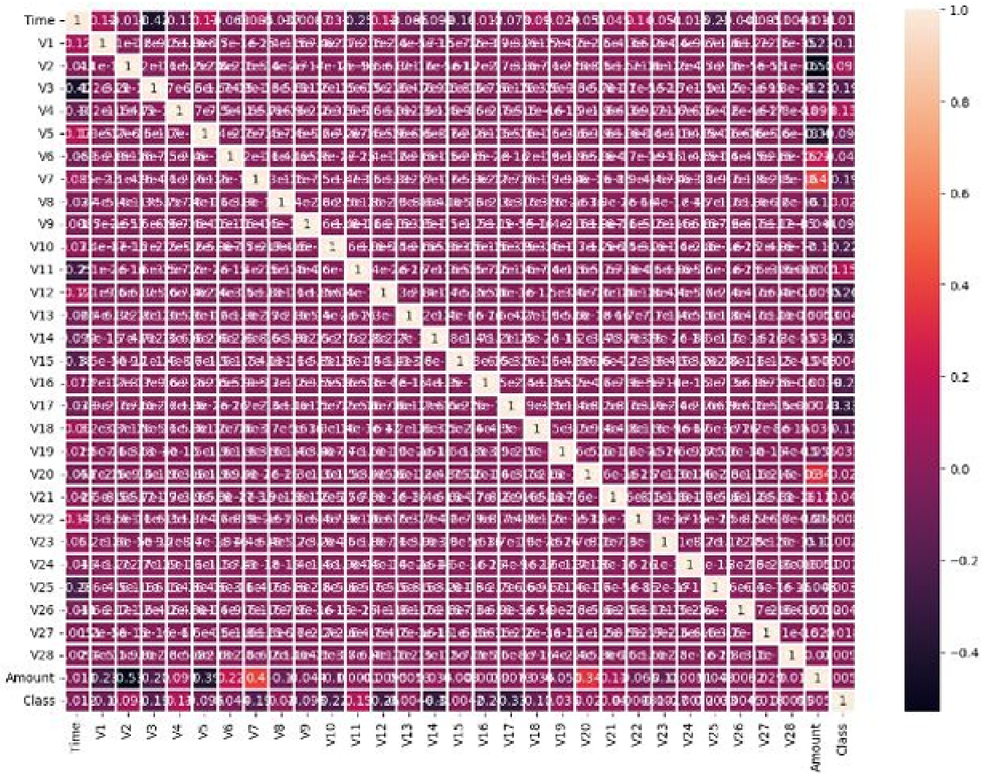


Figure 4: Data Correlation is as mentioned above.

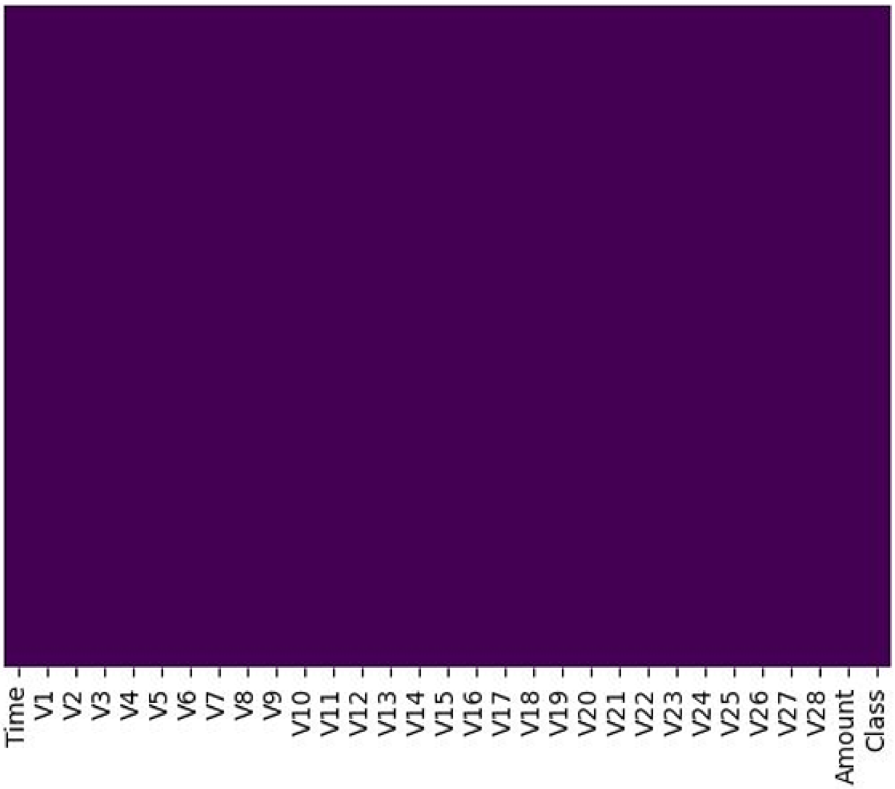


Figure 5: Null Data Visualization is as mentioned above.

5 Algorithm and Performance Analysis

5.1 K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) is a supervised learning algorithm known for its high accuracy in fraud detection compared to other supervised statistical pattern recognition methods. Its performance primarily depends on three factors: the distance metric used to identify the nearest neighbors, the method for deriving a classification from these neighbors, and the number of neighbors (K) considered.

KNN classifies transactions by calculating the distance between a given transaction and its nearest neighbors. If the nearest neighbor is labeled as fraudulent, the current transaction is also flagged as fraudulent. Euclidean distance is commonly applied for this purpose, offering reliable results with fast computation. Although KNN is efficient, its performance can be further improved by optimizing the distance metric.

5.1.1 KNN Algorithm

- Let m represent the number of training data samples, and p be the unknown data point to be classified.

- Store the training samples in an array arr[] where each element contains a tuple (x, y).

- For each i from 0 to m, Calculate the distance d(arr[i], p) using a suitable distance metric.

- Identify the set S of K smallest distances, representing the nearest neighbors.

- Return the majority label among the neighbors in S.

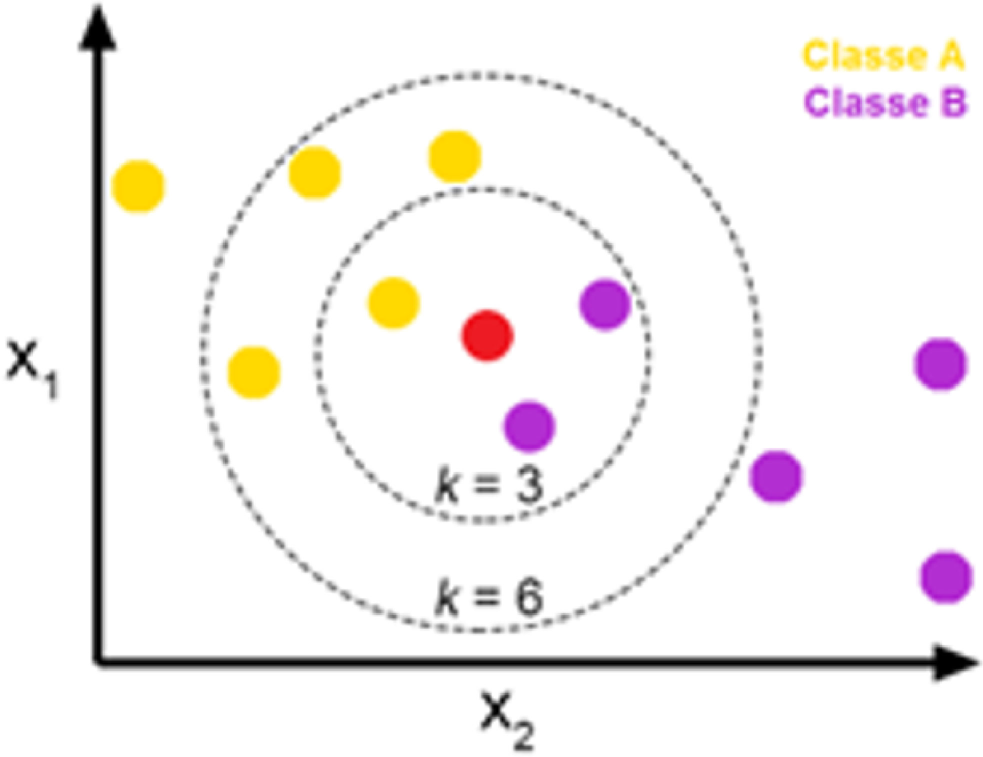


Figure 5.1: KNN

To identify the most effective model, two values of K were tested: K=3 and K=7.

• K=3: The KNN model with K=3 was created using Jupyter Notebook (Figure 5). It achieved a 100% accuracy, correctly identifying 85,443 transactions while misclassifying 131 transactions.

• K=7: A slight decrease in classification performance was observed when K was increased to 7 (Figure 6). Although the model still achieved 100% accuracy, it misclassified 131 fraudulent transactions as non-fraudulent. Additionally, 52 transactions were inaccurately labeled, indicating a difference in classification behavior compared to K=3.

Both models demonstrated strong fraud detection capabilities, with K=3 exhibiting slightly better classification consistency.

5.2 Logistic Regression (LR)

Logistic Regression is a statistical classification model that uses probabilities to detect fraud by applying a logistic curve. Since the logistic curve ranges from 0 to 1, it is ideal for interpreting class membership probabilities. The dataset is divided into training and testing sets for model development. After building the model, it is evaluated using a threshold cut-off value to predict class membership. By applying a suitable threshold, Logistic Regression effectively separates the dataset into two distinct regions using a single decision boundary.

The Logistic Regression model was developed using Jupyter Notebook. It achieved a training data accuracy of 93.51% and a test data accuracy of 91.88%, as depicted in the figure below.

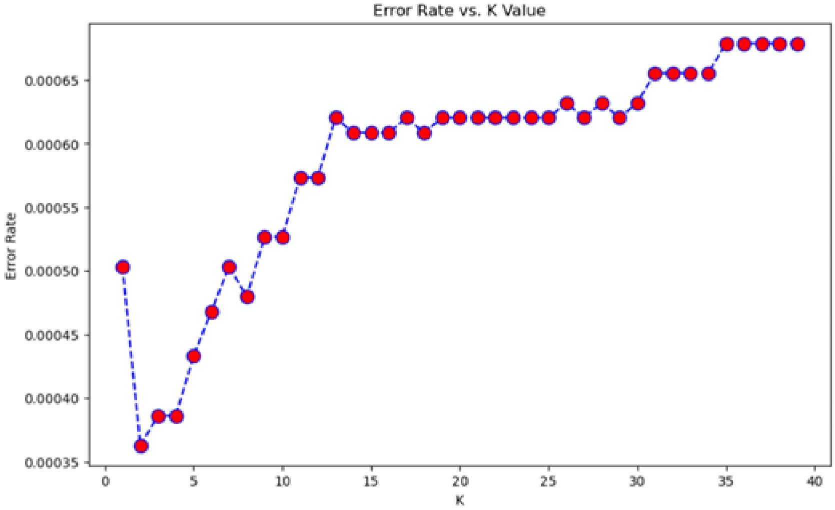
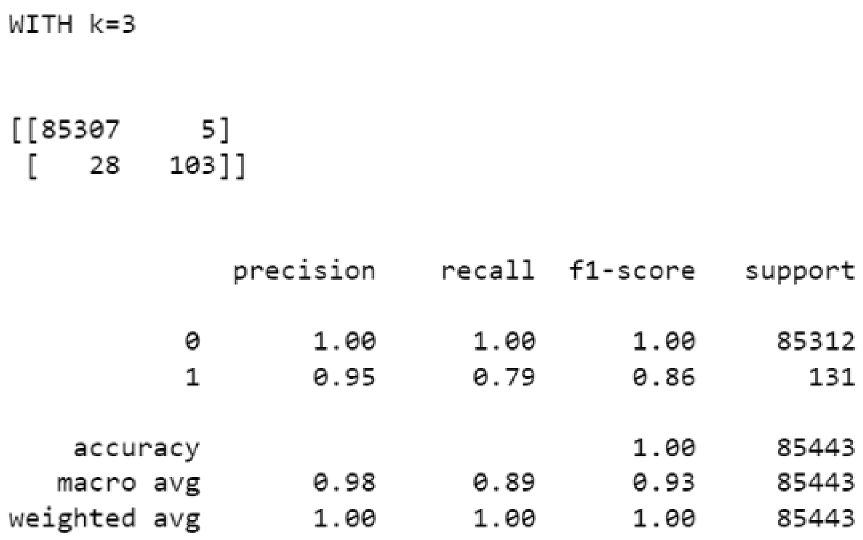


Figure 5.2: Error Rate

5.3 Support Vector Machine (SVM)

Support Vector Machines (SVMs) are linear classifiers that perform effectively in high-dimensional spaces. In such spaces, tasks that are non-linear in the input space often become linear, making SVMs particularly advantageous for fraud detection. SVMs employ a kernel function to represent the classification function through the dot product of input data projections. The algorithm aims to find a hyperplane that maximizes the separation between classes while minimizing overfitting on the training data, resulting in excellent generalization capability.

The Support Vector Machine model, as illustrated in the figure, achieved an accuracy score of 97.59%, demonstrating its effectiveness in identifying fraudulent transactions.



```
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 (DT)

A Decision Tree is a supervised learning algorithm that operates using a tree structure composed of a root node and subsequent nodes. These nodes are split into child nodes using either binary or multi-split approaches, applying specific algorithms for the splitting process. However, as the tree expands, overfitting may occur due to anomalies, errors, or noise in the data. To mitigate this, pruning techniques are applied to eliminate unnecessary nodes and enhance classification accuracy. Decision Trees are widely appreciated for their simplicity, interpretability, and ability to handle diverse data types.

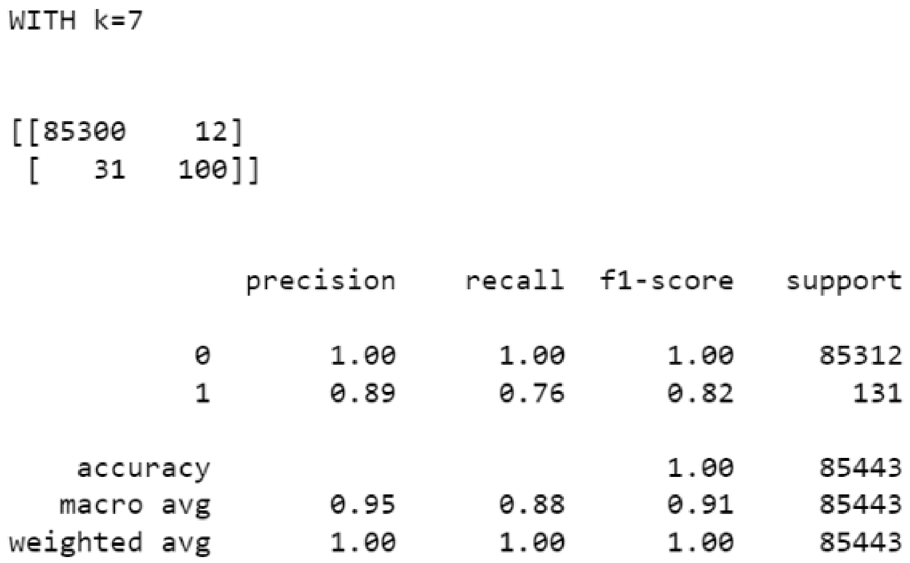
5.4.1 Algorithm

- Create (T) and Calculate frequencies (Ci, T)

- If all instances belong to the same class, return a leaf node

- For every attribute, apply a test to determine the best splitting criterion. The attribute that meets the criteria is assigned as the test node K

- Recursively repeat Create (Ti) for each partition Ti, adding the resulting 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 final stage of the CRISP-DM model is the evaluation and deployment phase, as illustrated in Table 2. This step involves comparing all models to identify the most effective one for detecting fraudulent credit card transactions. Accuracy, representing the percentage of correctly predicted instances, is calculated using a confusion matrix. The matrix consists of four key components: True Positive (T.P.), True Negative (T.N.), False Positive (F.P.), and False Negative (F.N.).

- **True Positive (T.P.)** indicates fraudulent transactions accurately identified by the model.

- **True Negative (T.N.)** refers to non-fraud transactions that were classified right as legit.

- **False Positive (F.P.)** represents non-fraudulent transactions misclassified as fraudulent.

- **False Negative (F.N.)** refers to fraudulent transactions incorrectly predicted as non-fraud.

The confusion matrix components are essential for evaluating the model's accuracy, which is calculated using the following formula:

Table 2 presents the accuracies of all models developed in this project. Each model demonstrated effective performance in detecting fraudulent transactions, achieving high accuracy scores. Among the models, K-Nearest Neighbor (KNN) and Decision Tree achieved the highest accuracy at 100%. Support Vector Machine (SVM) secured third place, while Logistic Regression had the lowest accuracy score of 93.51%.

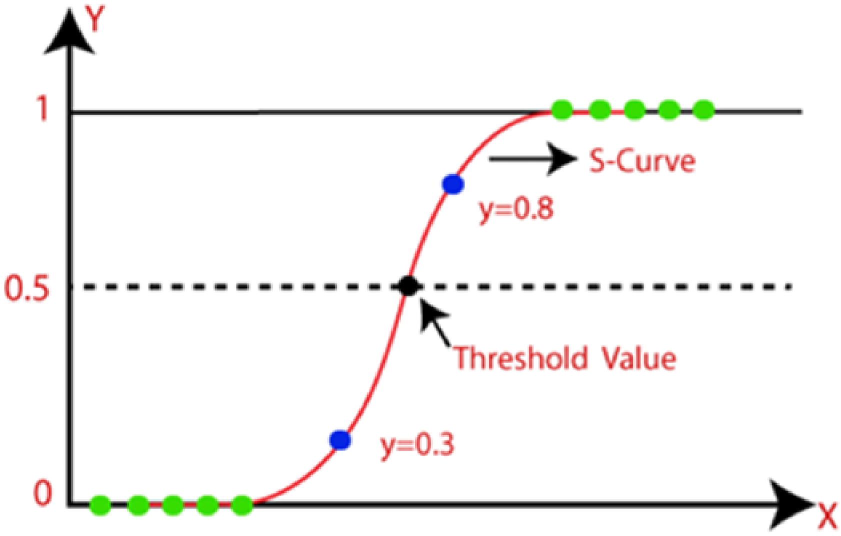
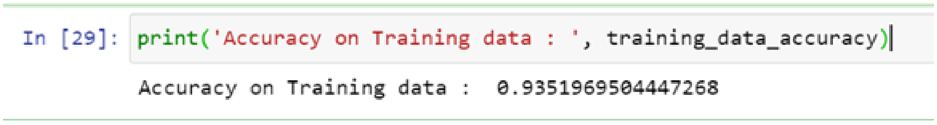


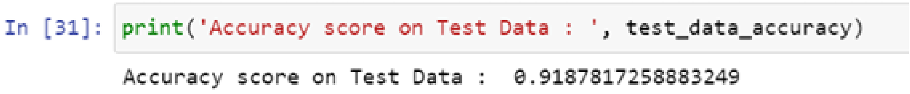
Figure 5.5: Logistic Regression Model



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

         Accuracy on Training data :  0.9351969504447268
```

Figure 5.6: Accuracy on Train data



```
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

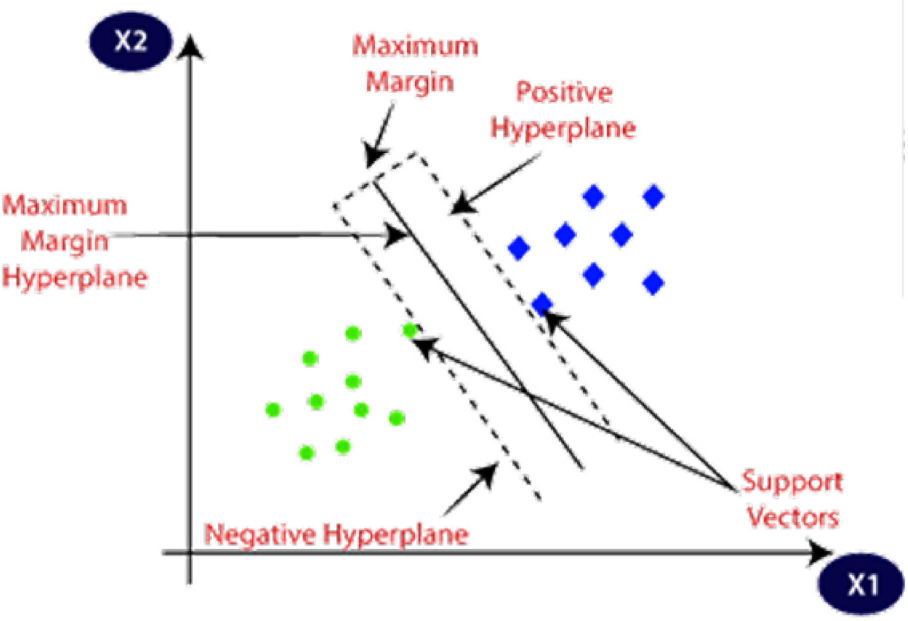


Figure 5.8: Support Vector Machine Algorithm

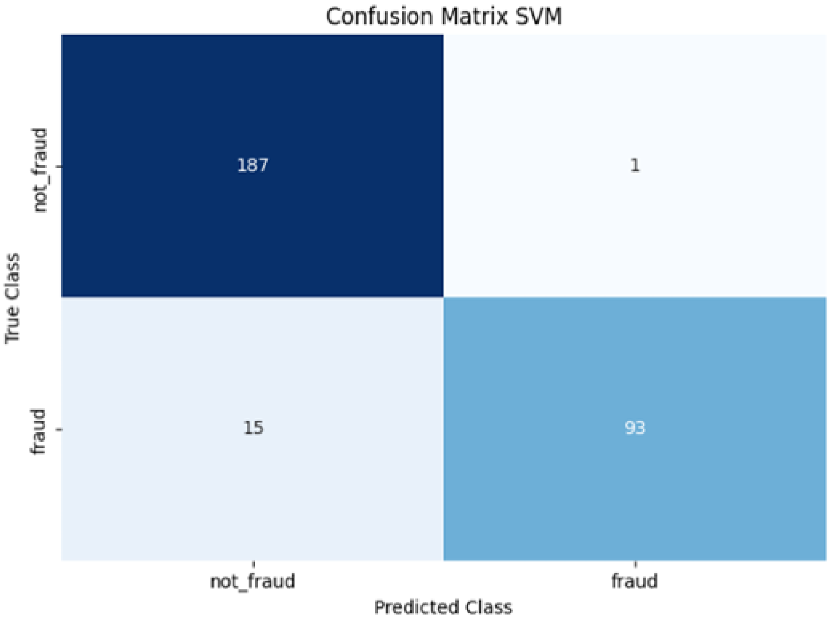


Figure 5.9: SVM Confusion Matrix

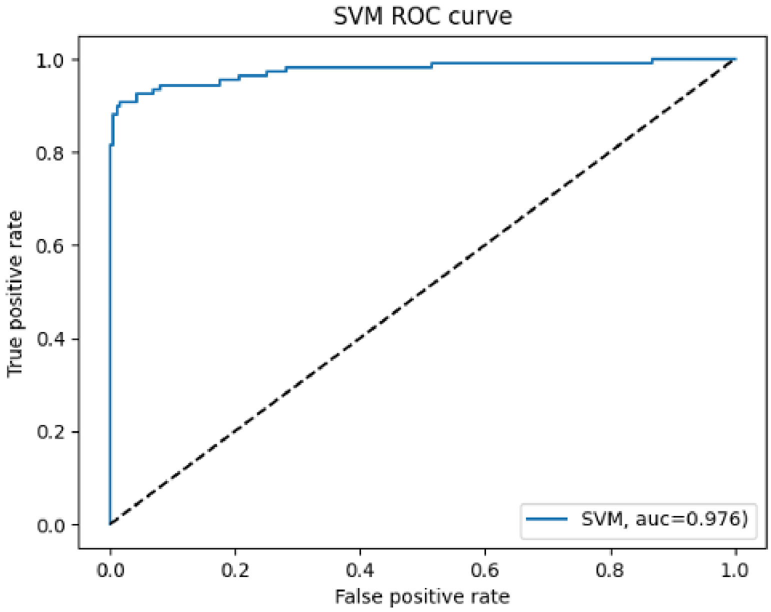


Figure 5.10: SVM ROC Curve

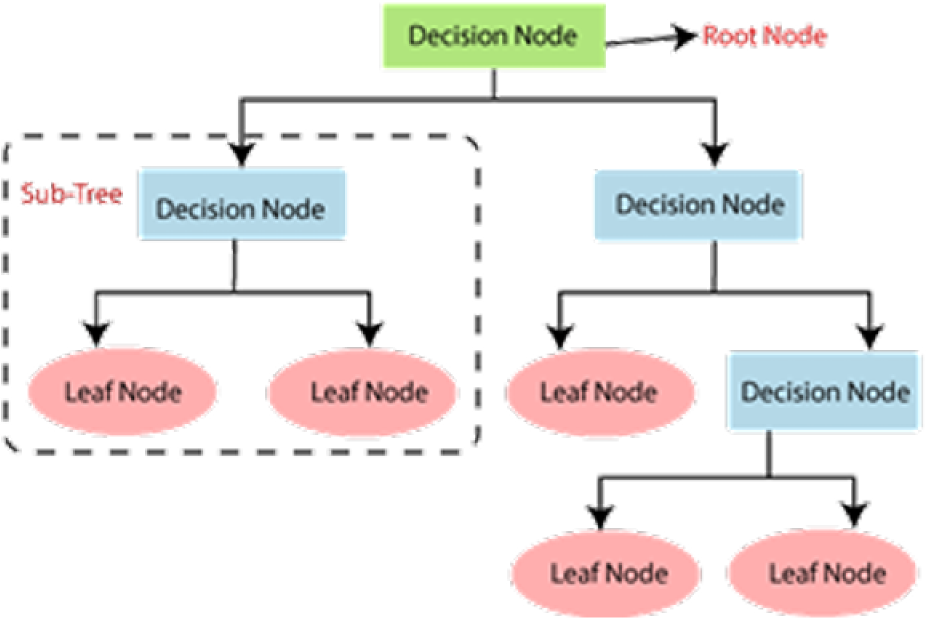
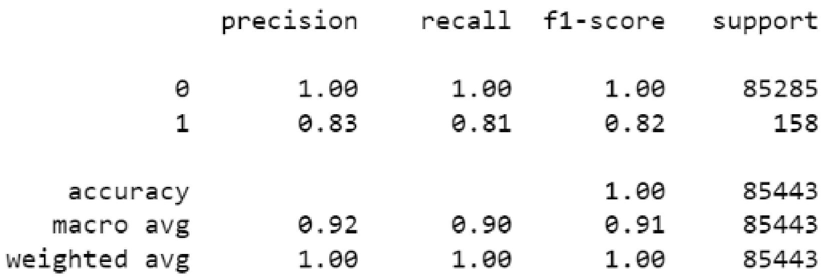


Figure 5.11: Decision Tree Algorithm



```
              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

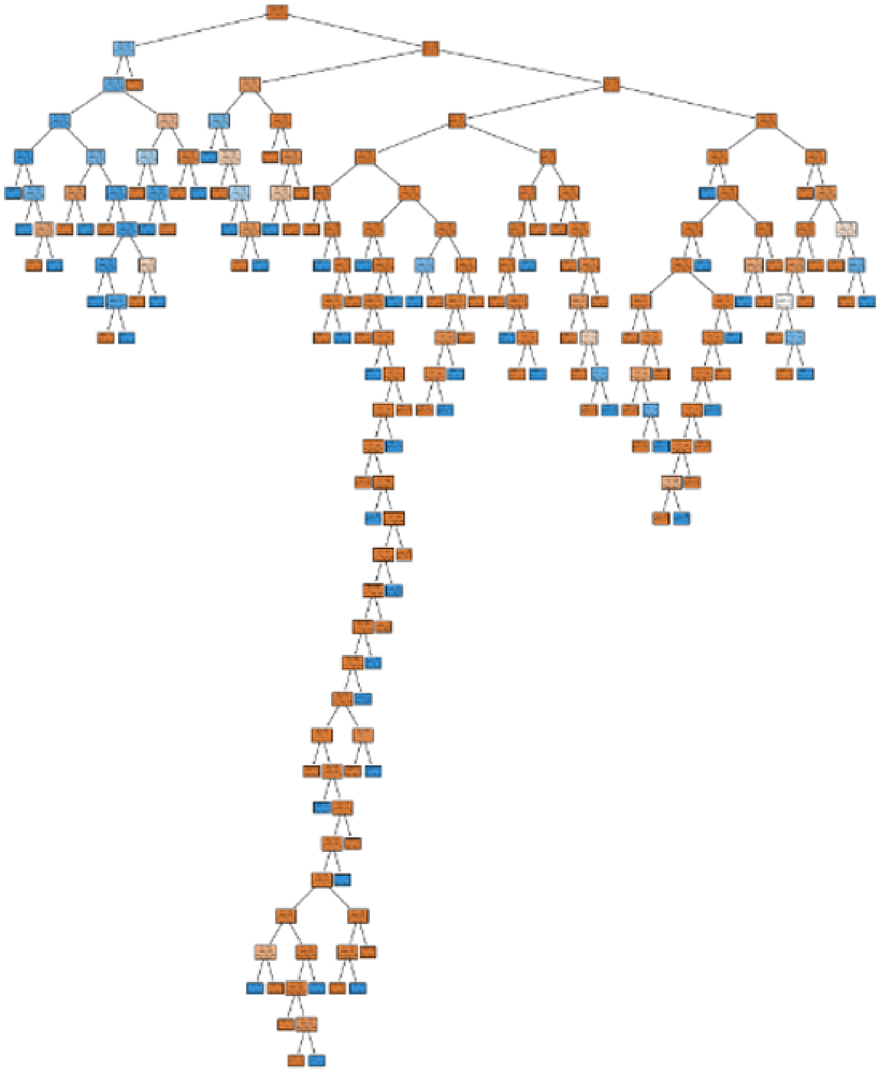
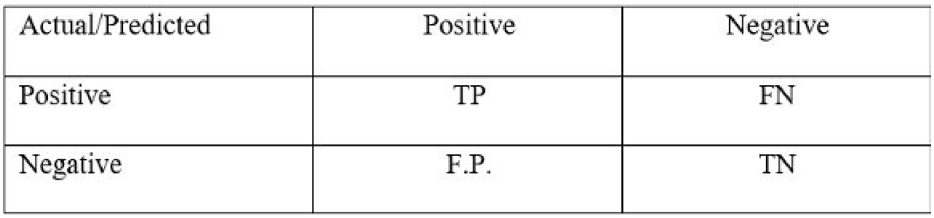


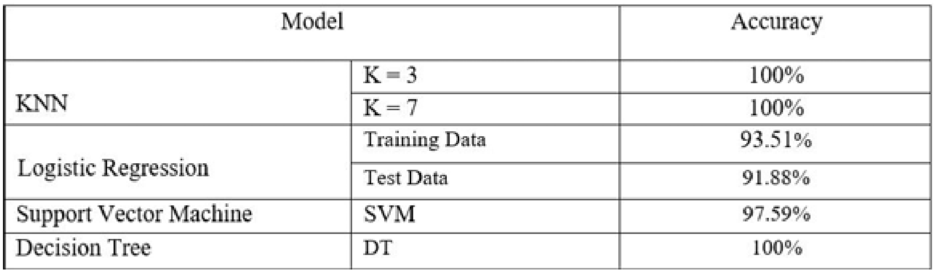
Figure 5.13: Decision Tree



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

6 Future Work & Conclusion

6.1 Future Work

There are numerous avenues for enhancing the model's performance and applicability. Applying the model to different datasets with diverse sizes and data types can provide a more comprehensive evaluation of its effectiveness. Experimenting with alternative data splitting ratios and exploring other machine learning algorithms may further improve accuracy and robustness. Incorporating external data sources, such as telecom data for geolocation tracking, can significantly enhance fraud detection accuracy. For instance, if a cardholder is in Dubai and a transaction occurs in Abu Dhabi, the system can instantly flag it as potentially fraudulent. Additionally, integrating real-time data streams, implementing advanced ensemble models, and applying reinforcement learning could optimize fraud detection capabilities. Developing explainable AI (XAI) models can also increase transparency and trust in the system. Collaborating with financial institutions to apply domain-specific knowledge and validating the model in real-world scenarios would further refine its performance.

6.2 Conclusion

The primary objective of this project was to identify the most suitable model for credit card fraud detection using various machine learning techniques. This objective was successfully achieved by developing and evaluating four models: K-Nearest Neighbors (KNN), Decision Tree, Support Vector Machine (SVM), and Logistic Regression. KNN and Decision Tree models demonstrated exceptional accuracy, achieving a perfect score of 100%, making them the most effective at detecting fraudulent transactions. However, while accuracy is crucial, additional metrics such as precision, recall, and F1-score should be considered to assess the model's true performance in real-world applications. By deploying these models in a financial setting, organizations can significantly reduce credit card fraud, enhance customer satisfaction, and ensure a more secure banking experience. Continuous monitoring, retraining, and model tuning are recommended to maintain high accuracy levels and adapt to evolving fraud patterns.

7 References

[1] Z. et al., ”Analysis on Credit Card Fraud Detection Techniques: Based on Certain Design Criteria.”

Available at: https://research.ijcaonline.org/volume52/number3/pxc3881538.pdf, 2012. Accessed: 26-Oct-2023.

[2] Alenzi, H. Z., ”Fraud Detection in Credit Cards Using Logistic Regression.”

Available at: https://thesai.org/Publications/ViewPaper?Volume=11&Issue=12&Code=IJACSA&SerialNo=
2020. Accessed: 26-Oct-2023.

[3] Maniraj, S. P., ”Credit Card Fraud Detection Using Machine Learning and Data Science.”

Available at: https://doi.org/10.17577/ijertv8is090031, 2019. Accessed: 25-Oct-2023.

[4] Dheepa, V. D. R., ”Behavior-Based Credit Card Fraud Detection Using Support Vector Machines.”

Available at: https://doi.org/10.21917/ijsc.2012.0061, 2012. Accessed: 26-Oct-2023.

[5] Malini, N. P. M., ”Analysis of Credit Card Fraud Identification Techniques Based on KNN and Outlier Detection.”

Available at: https://doi.org/10.1109/aeeicb.2017.7972424, 2017. Accessed: 26-Oct-2023.

[6] Maes, S., ”Credit Card Fraud Detection Using Bayesian and Neural Networks.”

Available at: https://www.ijert.org/research/credit-card-fraud-detection-using-machine-learning-and-data-science-IJERTV8IS090031.pdf, 2002. Accessed: 23-Oct-2023.

[7] Jain, Y. N. S., ”A Comparative Analysis of Various Credit Card Fraud Detection Techniques.” 2019. Accessed: 25-Oct-2023.

[8] Dighe, D. P. S. K. S., ”Detection of Credit Card Fraud Transactions Using Machine Learning Algorithms and Neural Networks.”

Available at: https://doi.org/10.1109/iccubea.2018.8697799, 2018. Accessed: 26-Oct-2023.

[9] Sahin, Y. D. E., ”Detecting Credit Card Fraud by Decision Trees and Support Vector Machines.” 2011. Accessed: 23-Oct-2023.