Report On

Credit Card Fraud Detection Using Supervised Learning

Submitted in partial fulfillment of the requirements of the Course project in Semester VII of Fourth Year Computer Engineering

by
Polomi Adak (Roll No. 20)
Riya Khot (Roll No. 28)
Nayan Biramane (Roll No. 21)

Supervisor
Dr. Megha Trivedi

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

(2023-24)

Vidyavardhini's College of Engineering & Technology
Department of Computer Engineering

CERTIFICATE

This is to certify that the project entitled “Credit Card Fraud Detection Using Supervised Learning” is a bonafide work of "Polomi Adak (20), Riya Khot (28), Nayan Biramane (21)" submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester VII of Fourth Year Computer Engineering.

Dr. Megha Trivedi
Supervisor

Dr Megha Trivedi
Head of Department

Dr. H.V. Vankudre
Principal

Abstract

Credit card fraud poses a significant threat to financial institutions and cardholders worldwide. Detecting fraudulent transactions in real-time is of paramount importance to mitigate the associated financial losses and maintain customer trust. This paper explores the application of supervised learning techniques for credit card fraud detection, leveraging historical transaction data to build predictive models.

The study employs a comprehensive dataset of credit card transactions, comprising both genuine and fraudulent cases. Various supervised learning algorithms, including logistic regression, decision trees, random forests, support vector machines, and neural networks, are evaluated to identify the most effective method for fraud detection. The models are trained on features extracted from transaction details, such as transaction amount, time, and anonymized cardholder information.

Performance metrics such as precision, recall, F1-score, and the receiver operating characteristic (ROC) curve are used to assess the models' effectiveness in identifying fraudulent transactions. Cross-validation techniques are employed to ensure robustness and prevent overfitting.

The results of the study demonstrate that supervised learning algorithms can be highly effective in detecting credit card fraud. The selected model achieves a balance between precision and recall, minimizing both false positives and false negatives. This enables financial institutions to promptly identify and prevent fraudulent transactions while minimizing inconvenience to legitimate cardholders.

Contents

Pg. No

1. Introduction

1.1. Introduction: -

In an increasingly digital and cashless world, the convenience of credit cards is undeniable. However, this widespread use of credit cards has also made them a prime target for fraudulent activities. Credit card fraud, which includes unauthorized transactions and identity theft, poses a significant financial threat to cardholders, banks, and merchants alike. To counteract this, the field of credit card fraud detection has evolved, leveraging advanced technologies and techniques to identify and prevent fraudulent transactions.

Credit card fraud detection is a vital component of the financial industry's efforts to safeguard transactions and protect consumers. This multifaceted field involves the application of data analytics, machine learning, artificial intelligence, and real-time monitoring to scrutinize card transactions for irregularities. By analyzing patterns, identifying anomalies, and comparing transaction data to user behavior, financial institutions can proactively detect and respond to potentially fraudulent activities.

Nowadays Credit card usage has been drastically increased across the world, now people believe in going cashless and are completely dependent on online transactions. The credit card has made the digital transaction easier and more accessible. A huge number of dollars of loss are caused every year by the criminal credit card transactions. Fraud is as old as mankind itself and can take an unlimited variety of different forms. The PwC global economic crime survey of 2017 suggests that approximately 48% of organizations experienced economic crime. Therefore, there’s positively a necessity to unravel the matter of credit card fraud detection. Moreover, the growth of new technologies provides supplementary ways in which criminals may commit a scam. The use of credit cards is predominant in modern day society and credit card fraud has been kept on increasing in recent years. Huge Financial losses have been fraudulent effects on not only merchants and banks but also the individual person who are using the credits. Fraud may also affect the reputation and image of a merchant causing non-financial losses that. For example, if a cardholder is a victim of fraud with a certain company, he may no longer trust their business and choose a competitor

1.2. Problem Statement & Objectives: -

Problem Statement: - - The Credit Card Fraud Detection Problem includes modeling past credit card transactions with the knowledge of the ones that turned out to be fraud. This model is then used to identify whether a new transaction is fraudulent or not. Our aim here is to detect the fraudulent transactions while minimizing the incorrect fraud classifications.

Objectives: -

The objectives of credit card fraud detection using machine learning typically include:

1) Detection of Fraudulent Transactions: The primary goal is to identify and flag transactions that are likely to be fraudulent, preventing unauthorized charges.

2) Reducing False Positives: While detecting fraud is essential, minimizing false alarms (legitimate transactions flagged as fraud) is equally important to avoid inconveniencing customers. Real-time Detection: Implementing a system that can detect fraudulent transactions in real-time, allowing for immediate action to prevent potential losses.

3) Scalability: Ensuring the system can handle a large volume of transactions as the number of credit card transactions can be massive.

4) Accuracy: Maximizing the accuracy of fraud detection models to reduce the chances of false negatives (genuine fraud cases going undetected).

5) Model Robustness: Developing models that can adapt to changing fraud patterns and techniques used by fraudsters.

6) Cost Reduction: Reducing financial losses for both the cardholder and the issuing bank by preventing fraudulent transactions.

7) Compliance: Ensuring that the fraud detection system complies with relevant regulations and data privacy laws.

8) User Experience: Maintaining a seamless user experience by minimizing the impact on genuine transactions while preventing fraud.

9) Data Security: Protecting the sensitive data used for fraud detection to prevent any breaches or unauthorized access.

10) Monitoring and Reporting: Implementing a system that provides real-time monitoring and generates reports on detected fraud cases for analysis and improvement.

1.3 Scope

Our project provides better data insights and more accuracy. Scope of any system depends on the following things:

1. It satisfies the user requirements.
2. Be easy to understand by the user and operator.
3. Be easy to operate.
4. Provide valuable data insights.
5. Be expandable.

We have tried to make such type of model, which satisfy the above given requirements. Overall, the scope of Credit Card Fraud Detection Using Supervised Learning is to provide a comprehensive solution for credit card fraud cases, providing valuable data insights and a much-improved model with a user-friendly interface and automated processes.

2. Proposed System

2.1. Introduction: -

Credit card fraud has emerged as a formidable challenge in the modern era of electronic commerce. With the increasing reliance on credit and debit cards for transactions, financial institutions and cardholders face an ever-growing threat from fraudulent activities. Detecting and preventing credit card fraud is not only crucial for limiting financial losses but also for upholding the integrity of financial systems and maintaining the trust of consumers. In response to this evolving threat, the application of sophisticated data analytics and machine learning techniques has gained prominence, and in this study, we delve into the use of supervised learning to address the issue of credit card fraud detection.

2.2. Architecture/Block Diagram: -

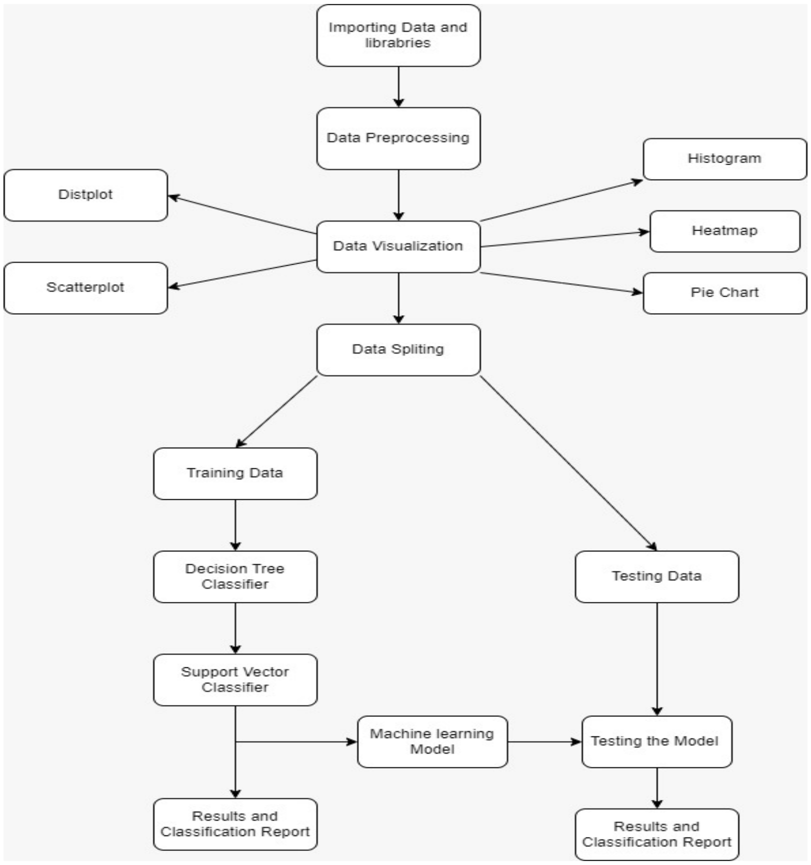


Fig 2.2.1: Block diagram of the system

The architecture block diagram in Fig. 2.2.1 shows where data is first gathered from internet sources such as Kaggle. Next, data is preprocessed to remove missing data and inconsistencies, turning the data into meaningful data. To improve the performance of the model, features are extracted based on the data. To obtain insights from data, data visualization techniques including scatterplots, pie charts, distant plots, and histograms are utilized. In order to apply models, the data is then divided into training and testing sets. To determine the model's accuracy, support vector machines and decision tree classifiers are used to the data.

2.3. Algorithm and Process Design: -

This model uses two machine learning algorithms Decision Tree and Support Vector Classifier for predictive analysis. Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed on the basis of features of the given dataset.

It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

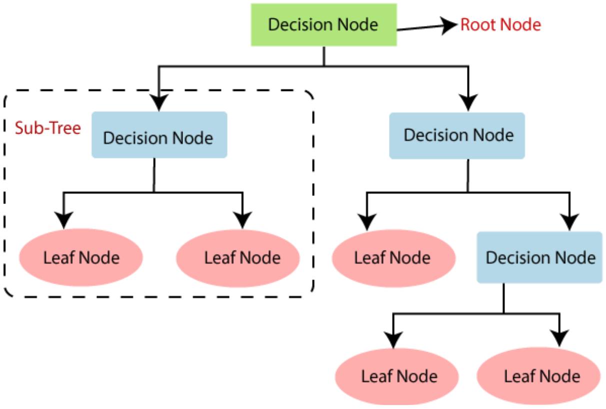


Fig 2.3.1: Decision Tree Classifier

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

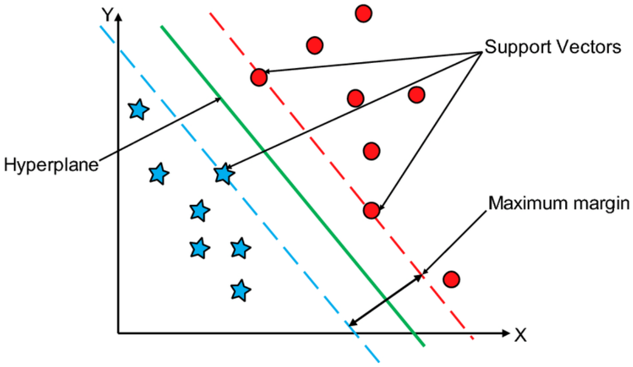


Fig 2.3.2: Support Vector Machine

The optimal model accuracy was determined using these two approaches. We first gathered the dataset utilizing an internet resource, and as it could have a lot of inconsistencies, we preprocessed the data. The data was then transformed to get an improved outcome. Additionally, data visualization techniques were used to demonstrate the relationship between the statistics and data points in terms of frequency and time. The decision tree algorithm yielded 99.92% accuracy, whereas the support vector machine yielded 99.82% accuracy. We may therefore assume from this that the decision tree model has superior accuracy, which is why we chose to utilize it.

Packages which are being used for data exploration, pro processing and for using random forest algorithm are:

1)NumPy: For simple arrays.

2)Pandas: For reading the file.

3)SciKit: Learn- for pre-processing.

4)Matplotlib or Seaborn: For plotting and representing confusion matrix color format.

5)Tensor flow: For matrix format

2.4. Details of Hardware & Software: -

Hardware**: -**

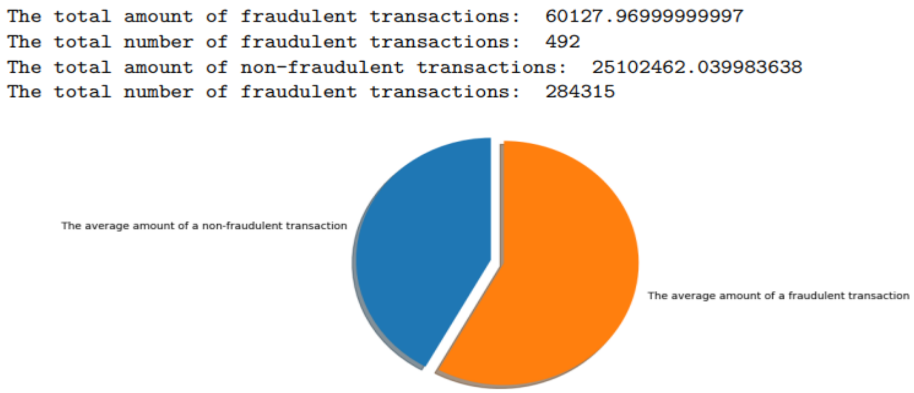
- Windows 10
- Processor Base Frequency of 1.8 GHz or higher
- 4 GB RAM or more
- 75 MB of available disk space or more
- Minimal screen resolution: 1366x768 pixels
- Camera and Mic (USB or Built in)

Software: -

- Windows10 (OS)
- Python Language
- Google Collab
- Jupyter Notebook
- Lucid

2.5. Experiment and Results for Validation and Verification

Data Visualization: In this project we have used various data visualization techniques tohelp the law enforcers derive various valuable insights from the data.



```
The total amount of fraudulent transactions:  60127.96999999997
The total number of fraudulent transactions:  492
The total amount of non-fraudulent transactions:  25102462.039983638
The total number of fraudulent transactions:  284315
```

Fig 2.5.1: Pie chart of fraud and non-fraud transactions

The above figure shows us total amount of transactions and total number of transactions of fraudulent and non-fraudulent data.

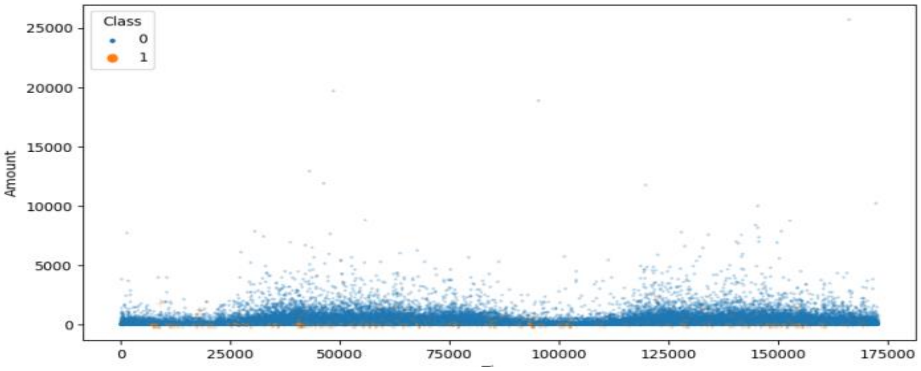


Fig 2.5.2: Scatterplot of frequency of class with respect to time

The above figure shows the scatterplot diagram of class scatter in the time with respect to number of transactions of dataset.

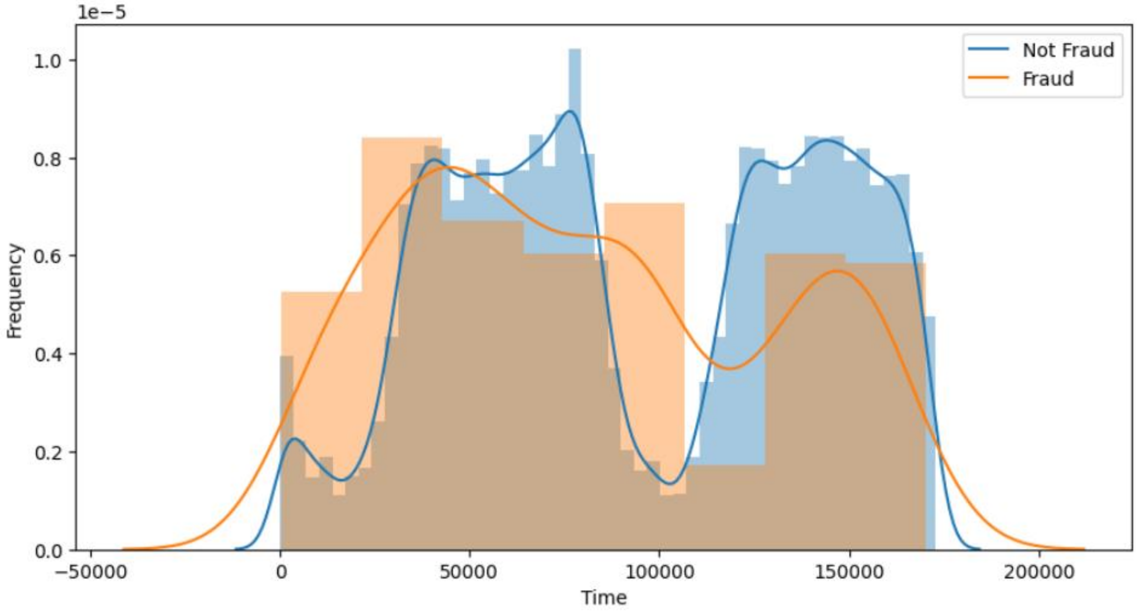


Fig 2.5.3: Distant Plot between time and frequency

The above figure shows the distant plot between fraud and non fraud data with time with respect to frequency.

Code: -

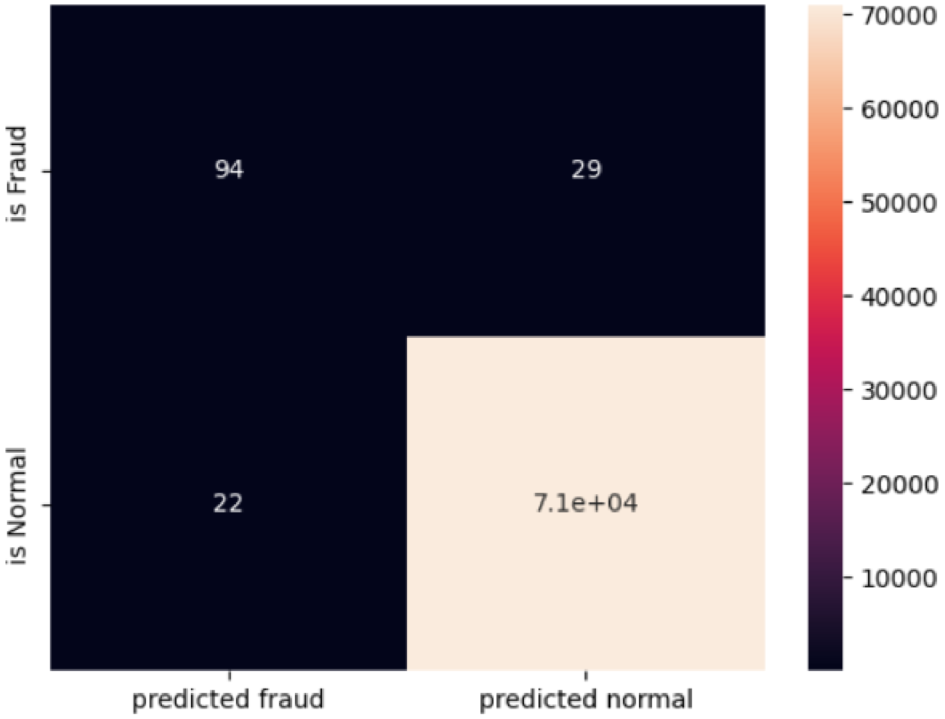
```
  import numpy as np
  import pandas as pd
  import matplotlib.pyplot as plt
  import seaborn as sns
  from sklearn.linear_model import LogisticRegression
  from sklearn.model_selection import train_test_split
  from sklearn.metrics import accuracy_score

 df= pd.read_csv("/content/drive/MyDrive/creditcard.csv")
 df.head(10)


df.info()

df.dtypes

df.describe()

df.isnull().sum()

df['Class'].value_counts()

legit = df[df.Class == 0]
fraud = df[df.Class == 1]
print("Shape of legit : ", legit.shape)
print("Shape of fraud : ", fraud.shape)

df.hist(figsize = (20,15))
plt.show()

import seaborn as sb
plt.figure(figsize = (20,15))
sb.heatmap(df.corr(), annot = True, cmap = 'coolwarm')
plt.title("Heatmap for correlation matrix for credit card data ", fontsize = 22)
plt.show()

fraudulent_amount = 0
number_of_fraudulent_transactions = 0;
for record in df.itertuples():
if record.Class == True:
fraudulent_amount += record.Amount
number_of_fraudulent_transactions += 1
print('The total amount of fraudulent transactions: ', fraudulent_amount)
print('The total number of fraudulent transactions: ',number_of_fraudulent_transactions)
fraudulent_mean = fraudulent_amount / number_of_fraudulent_transactions
nonfraudulent_amount = 0
number_of_nonfraudulent_transactions = 0
for record in df.itertuples():
if record.Class == False:
nonfraudulent_amount += record.Amount
```

```
number_of_nonfraudulent_transactions += 1
print('The total amount of non-fraudulent transactions: ', nonfraudulent_amount)
print('The total number of fraudulent transactions: ',number_of_nonfraudulent_transactions)
nonfraudulent_mean = nonfraudulent_amount / number_of_nonfraudulent_transactions
d = {'Class':[0, 1], 'Mean':[nonfraudulent_mean, fraudulent_mean]}
fig, ax = plt.subplots(figsize=(10,5))
ax.pie(d["Mean"],
explode=[0, 0.1],
labels=['The average amount of a non-fraudulent transaction', 'The average amount of a fraudulent
transaction'],
shadow=True, startangle=90)
ax.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()

plt.figure(figsize=(10,5))
sb.scatterplot(x=df["Time"], y=df["Amount"], hue=df["Class"], size=df["Class"],sizes=(40, 8),
marker="+")

plt.figure(figsize=(10,5))
plot = sns.distplot(a=df["Time"], kde=True, color='purple')
plot.set(xlabel ='Time', ylabel ='Frequency')
plt.show()

fraudulent_transactions = df[df['Class'] == 1]
nonfraudulent_transactions = df[df['Class'] == 0]
plt.figure(figsize=(10,5))
sb.distplot(a=nonfraudulent_transactions["Time"], kde=True)
plot = sb.distplot(a=fraudulent_transactions["Time"], kde=True)
plot.set(xlabel ='Time', ylabel ='Frequency')
plot.legend(['Not Fraud', 'Fraud'])
plt.show()

X = df.drop(columns = 'Class', axis = 1)
Y = df['Class']

X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.25,␣
↪stratify = Y, random_state = 0)

 X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.25,␣
↪stratify = Y, random_state = 0)

from sklearn.preprocessing import StandardScaler
stdsc = StandardScaler()
xtrain = stdsc.fit_transform(X_train)
xtest = stdsc.transform(X_test)

from sklearn.tree import DecisionTreeClassifier
dt_classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
dt_classifier.fit(X_train, Y_train)

y_pred_decision_tree = dt_classifier.predict(X_test)

print("y_pred_decision_tree : \n", y_pred_decision_tree)
```

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
com_decision = confusion_matrix(Y_test, y_pred_decision_tree)
print("Confusion Matrix : \n", com_decision)

Accuracy_Model = ((com_decision[0][0] + com_decision[1][1]) / com_decision.
↪sum()) *100
print("Accuracy_Decison : ", Accuracy_Model)
Error_rate_Model= ((com_decision[0][1] + com_decision[1][0]) / com_decision.
↪sum()) *100
print("Error_rate_Decison : ", Error_rate_Model)
# True Fake Rate
Specificity_Model= (com_decision[1][1] / (com_decision[1][1] +␣
↪com_decision[0][1])) *100
print("Specificity_Decison : ", Specificity_Model)
# True Genuine Rate
Sensitivity_Model = (com_decision[0][0] / (com_decision[0][0] +␣
↪com_decision[1][0])) *100
print("Sensitivity_Decison : ", Sensitivity_Model)

from sklearn.svm import SVC
svc_classifier = SVC(kernel = 'rbf', random_state =0)
svc_classifier.fit(X_train, Y_train)

y_pred2 = svc_classifier.predict(X_test)

print("y_pred_randomforest : \n", y_pred2)

cm2 = confusion_matrix(Y_test, y_pred2)
print("Confusion Matrix : \n\n", cm2)


Accuracy_Model = ((cm2[0][0] + cm2[1][1]) / cm2.sum()) *100
print("Accuracy_svc : ", Accuracy_Model)
Error_rate_Model = ((cm2[0][1] + cm2[1][0]) / cm2.sum()) *100
print("Error_rate_svc : ", Error_rate_Model)
# True Fake Rate
Specificity_Model= (cm2[1][1] / (cm2[1][1] + cm2[0][1])) *100
print("Specificity_svc : ", Specificity_Model)
# True Genuine Rate
Sensitivity_Model= (cm2[0][0] / (cm2[0][0] + cm2[1][0])) *100
print("Sensitivity_svc : ", Sensitivity_Model)
```

2.6. Analysis: -

We have improved the accuracy of our prediction analysis through this project by using Decision Tree classifiers and support vector machines, which will enable the model to function well on untested data. We were able to achieve a 99.92% accuracy rate with our predictive model. The confusion matrix produced by the decision tree classifier is shown below.



2.7. Conclusion and Future scope: -

Here we conclude that we have successfully designed, developed and implemented credit card fraud detection system using supervised techniques to identify the frauds and eliminate it for better performance.

Hence, we have acquired the result of an accurate value of credit card fraud detection i.e., 0.99928372798516 (99.92%) using a decision tree algorithm with new enhancements. In comparison to existing modules, this proposed module is applicable for the larger dataset and provides more accurate results.

Future Scope:

- Credit card fraud detection can be used for developing real-time fraud detection systems is crucial to prevent fraudulent transactions as they occur. The future will likely see advancements in real-time data processing and model deployment, reducing the response time to potential fraud cases.
- Anomaly detection algorithms can be combined with supervised learning models to improve fraud detection. These hybrid systems can identify unusual patterns and deviations from normal behavior, providing an added layer of security.
- As digital currencies and blockchain technology gain prominence, fraud detection will extend to these domains. Innovative approaches will be required to secure these new financial.
- With the increasing use of IoT devices and mobile payment systems, fraud detection will need to encompass these channels, and new models tailored to the unique characteristics of these transactions will be required. systems.

3. References

1. A machine learning based credit card fraud detection using the GA algorithm for feature selection” by E.Ileberi, Y. Sun, Z. Wang.

2. Credit Card Fraud Detection Based on Transaction Behavior -by John Richard D Kho Larry A Vea” published by Proc. of, the,2017 IEEE Region 10 Conference, (TENCON) November 5/8/2017

3. Jain R., Gour B., Dubey S., A hybrid approach for credit card fraud detection using rough set and decision tree technique, International Journal of Computer Applications 139(10) (2016)