A MINI PROJECT REPORT

ON

AUTOMATED CREDIT CARD FRAUD DETECTION

A dissertation submitted in partial fulfilment of the

Requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

Submitted by

Puppala Pothan Prathap (19B81A1278)

Minkuri Vrishin Reddy (19B81A1287)

Marlapally Rohith Reddy (19B81A1289)

Under the esteemed guidance of

Dr. Bipin Bihari Jayasingh

HOD, IT Department

DEPARTMENT OF INFORMATION TECHNOLOGY

CVR COLLEGE OF ENGINEERING

ACCREDITED BY NBA, AICTE & Affiliated to JNTU-H

Vastunagar, Mangalpally (V), Ibrahimpatnam (M), R.R. District, PIN-501 510

2022-2023

Cherabuddi Education Society's

CVR COLLEGE OF ENGINEERING

Accredited By NBA, NAAC 'A' Grade and Affiliated to JNTU, Hyderabad

(Approved by AICTE & Government of Telangana) Vastunagar, Mangalpalli (V), Ibrahimpatnam (M), R.R. District, PIN: 501 510

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the Mini Project Report entitled **"Automated Credit Card Fraud Detection"** is a bonafide work done and submitted by **Puppala Pothan Prathap (19B81A1278) , Minkuri Vrishin Reddy (19B81A1287) , Marlapally Rohith Reddy (19B81A1289)** during the academic year 2022-2023, in partial fulfilment of requirement for the award of Bachelor of Technology degree in Information Technology from Jawaharlal Nehru Technological University Hyderabad, is a bonafide record of work carried out by them under my guidance and supervision.

Certified further that to the best of my knowledge, the work in this dissertation has not been submitted to any other institution for the award of any degree or diploma.

INTERNAL GUIDE

Dr. Bipin Bihari Jayasingh

HOD, IT Department

HEAD OF THE DEPARTMENT

Dr. Bipin Bihari Jayasingh

HOD, IT Department

MINI-PROJECT COORDINATOR

G. Sunitha Rekha

Sr. Assistant Professor, IT Department

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction of completing this mini project would be incomplete without mentioning our gratitude towards all the people who have supported us. Constant guidance and encouragement have been instrumental in the completion of this project.

First and foremost, we thank the Chairman, Principal, Vice Principal for availing infrastructural facilities to complete the mini project in time.

We offer our sincere gratitude to our internal guide **Dr.Bipin Bihari Jayasingh ,** HOD, IT Department, CVR College of Engineering for his immense support, timely co-operation and valuable advice throughout the course of our project work.

We would like to thank the Professor In-Charges of Projects, **Dr.R.Seetharamaiah** and **Dr. S.V. Suryanarayana** of Information Technology for their valuable suggestions in implementing the project.

We would like to thank the Head of Department, Professor **Dr. Bipin Bihari Jayasingh**, for his meticulous care and cooperation throughout the project work.

We are thankful to **G. Sunitha Rekha,** Mini-Project Coordinator, Sr. Assistant Professor, IT Department, CVR College of Engineering for his supportive guidelines and for having provided the necessary help for carrying forward this project without any obstacles and hindrances.

We also thank the **Project Review Committee Members** for their valuable suggestions.

Puppala Pothan Prathap (19B81A1278)

Minkuri Vrishin Reddy (19B81A1287)

Marlapally Rohith Reddy (19B81A1289)

DECLARATION

We hereby declare that the project report entitled "**Automated Credit Card Fraud Detection**" is an original work done and submitted to IT Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad in partial fulfilment of the requirement for the award of Bachelor of Technology in **Information Technology** and it is a record of bonafide project work carried out by us under the guidance of **Dr. Bipin Bihari Jayasingh , HOD , Department of Information Technology.**

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other Institute or University.

_____

Signature of the Student

(Puppala Pothan Prathap)

(19B81A1278)

_____

Signature of the Student

(Minkuri Vrishin Reddy)

(19B81A1287)

_____

Signature of the Student

(Marlapally Rohith Reddy)

(19B81A1289)

ABSTRACT

Everyone in today's generation uses the internet banking system for financial transfers, as well as credit and debit cards for shopping and online purchases. Credit card fraud occurs in a relatively tiny number of cases, but the financial losses can be significant. This necessitates the creation of an Automated Fraud Detection System (FDS). Credit card online transactions are becoming the most popular and widely utilized. However, as internet usage grows at an exponential rate, unlawful transactions and assaults in the financial industry are becoming more common. Unauthorized credit and debit card transactions, credit card theft, and other such illegal operations are causing concern among international governments, clients, and the banking sector. Unusual assaults and illegal access can be detected by financial fraud detection systems. Financial institutions keep these fraud detection methods up to date. To avoid these unlawful transactions and assaults, we may utilize technology such as Machine Learning and Deep Learning approaches to forecast fraud transactions to prevent attacks and make transactions safer and more efficient. In our project, we would want to apply a financial fraud detection method based on a machine learning algorithm for a big volume of data collected from various online sources. Because RF is good for big quantities of data while LR leads to overfitting, this technique is said to work particularly well for fraud detection in large data sets.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

Credit card is one of the popular modes of payment for electronic transactions in many developed and developing countries. Invention of credit cards has made online transactions seamless, easier, comfortable, and convenient. However, it has also provided new fraud opportunities for criminals, and in turn, increased fraud rate. The global impact of credit card fraud is alarming, millions of US dollars have been lost by many companies and individuals. Furthermore, cybercriminals are innovating sophisticated techniques on a regular basis, hence, there is an urgent task to develop improved and dynamic techniques capable of adapting to rapidly evolving fraudulent patterns. Achieving this task is very challenging, primarily due to the dynamic nature of fraud and due to lack of dataset for researchers. There are several different factors that make card fraud research worthwhile. The most obvious advantage of having a proper fraud detection system in place is the restriction and control of potential monetary loss due to fraudulent activity. Today, all around the world data is available very easily, from small to big organizations are storing information that has high volume, variety, speed and worth. This information comes from tons of sources like social media followers, likes and comments, user's purchase behaviours.

All this information used for analysis and visualization of the hidden data pattern. Early analysis of big data was centered primarily on data volume, for example, public databases, biometrics, and financial analysis. For frauds, the credit card is an easy and friendly target because without any risk a significant amount of money is obtained within a short period. To commit credit card fraud, fraudsters try to steal sensitive information such as credit card number, bank account and social security number. Fraudsters try to make every fraudulent transaction legitimate which makes fraud detection a challenging problem. Increased credit card transactions show that approximately 70% of the people in the US can fall into the trap of these fraudsters. Credit card dataset is highly imbalanced because it carries more legitimate transactions as compared to the fraudulent one.

Credit card normally refers to a card that is allocated to the consumers (cardholder), generally consumer can acquisition goods and services through credit card within limit or withdraw cash from anywhere. Banks provide many facilities to customers thorough credit card. For instance, it provide consumer to pay later in a given time by carrying it to the subsequent next bill. Fraud is a criminal or illegal cheating that targeted to fetch financial or

private advantage. In circumventing damage from fraud and there are the following two approaches can be implemented: fraud detection and fraud prevention. Fraud detection is desired when a fake transaction is happened by a cheaters and Fraud prevention is an active technique, where I thalts fraud from trendy in the primarily place.

There are giant data model has transfigured the baking systems and by altering the financial establishments run direction. The outcome of the historical financial disaster has been gradually remedying and people are nowadays improved for opportunities and financial system (Subbas and Lahiri, 2017).Credit card fraud is linked with the prohibited usage of credit card material for acquisitions. Credit card transactions can be skillful either physically or digitally, physical transactions of the credit card are complicated while doing the transactions and in digital transactions it can be happen over the internet and phones.

Fraud is very easy if targeted people have less knowledge about transactions and online money transfer. Hacker mostly attack on innocent people those don't have complete knowledge and usage about online banking applications. Normally cardholders typically deliver expiry date and card number, in case of forgets PIN number banks provide verification numbers respective telephone or email, (Randhawa, Kuldeep, et al 2018). Credit card fraud is easy marks with no any risks, hacker withdraw amount without owner knowledge without bank's knowledge, they transfer amount in very short time of period and then escape from specific networks. Identify to fraudsters is not easy because they use very fast tool and masterminds those have all knowledge about owner's credits cards and financial transactions moments. Fraudsters continuously attempt to variety of each fake transaction legitimate, and it makes fraud detection actual stimulating and problematic job to identify attacks. The data analytics permitted banks to method the data ambitious commerce in an improved method to tackle the actual data generated customers. Credit card default prediction is core forecast that banks are a worry with involves credit counting to healthier comprehend why customers are probable to default. Banks want to each minor detail of the customers for tracking of payment data that is added in the credit history.

Credit card fraud recognition is a problematic issue that becomes the attention of Machine Learning researchers and scientists. Nevertheless, the issue is still challenging for credit card data which suffer from class inequity as no fraud transactions over powering succeed fraud transactions making it tough for numerous machine learning algorithms to achieve good accuracy and performance, a upright illustration can be erudite from the dataset that increase the classification performance of the machine learning techniques. Machine

learning is a thinkable resolution to the challenge of credit fraud prediction because of its extraordinary feature learning aptitude in large and unstable datasets.

That means prediction will get a very high accuracy score without detecting a fraud transaction. To handle this kind of problem one better way is to class distribution, i.e., sampling minority classes. In the sampling minority, class training examples can be increased in proportion to the majority class to raise the chance of correct prediction by the algorithm. Annually, card issuers suffer huge financial losses due to card fraud and, consequently, large sums The use of machine learning in fraud detection has been an interesting topic now days. The following are some commonly used online fraud techniques they are Site Cloning: this involves designing the exact duplicate of web sites. Users ignorantly visit cloned websites and enter their card information. Credit card generators: Some fraudsters use credit card generators for fraudulent activities.

These generators are computer programs capable of generating credit card numbers and expiry dates. A list of card numbers is usually generated for the account of one card holder. Account Takeover: this occurs when fraudsters use stolen information (card number, card number, CVV number, etc.) illegally obtained from card holders to control account of users. In some instances, fraudsters (posing as the card owner) can send a message to a card issuer requesting for change of address and change of card. In other instances, fraudsters can use stolen information (such as username and password) to logon to customer account and change customer details making it difficult for the original account owner to recover the account. Fake Card: Fraudsters clones stolen cards and use them for fraudulent transactions. Fraudsters use different cloning methods.

Credit card is a small thin plastic or fiber card that contains information about the person such as picture or signature and person named on it to charge purchases and service to his linked account charges for which will be debited regularly. Now a day's card information is read by ATM's, swiping machines, store readers, bank and online transaction. Each card as a unique card number which is very important, its security is mainly relies on physical security of the card and also privacy of the credit card number. There is rapid increase in the credit card transaction which as led to substantial growth in fraudulent cases. Many data mining and statistical methods are used to detect fraud. Many fraud detection techniques are implemented using artificial intelligence, pattern matching. Detection of fraud using efficient and secure methods are very important. Credit card frauds are increasing heavily because of fraud financial loss is increasing drastically. Now days Internet or online transaction growing as new technology are coming day by day. In these transaction Credit card holds the maximum share.

In 2018 Credit card fraud losses in London estimated US dollar 844.8 million. To reduce these losses prevention or detection of fraud must be done. There are different types of frauds occurring as technology is growing rapidly. So, there are many machine algorithms are used to detect fraud now days hybrid algorithms, artificial neural network is used as it gives better performance.

In some cases, fraudsters use magnet to erase the magnetic strip in cards. Afterwards, the information on the card will be changed to that of a valid user. In some other instance, fraudsters use skimming devices (containing electronic magnetic strip reader) to copy the magnetic stripe information of a valid card into a fake card. Sometimes the transfer of information is done without the knowledge of card owners because fraudster can put the skimming devices in their pocket and move closer to the place where the valid card is kept. Afterwards, the fraudster will then use the cloned card to perform card-not-present transactions Mail Theft: Some fraudsters divert mails containing card number that are newly supplied. A credit card fraud detection algorithm consists in identifying those transactions with a high probability of being fraud, based on historical fraud patterns. Every card holder is characterized by patterns containing information about distinctive purchase category the time since the last buying, money spent and other things. Falsehood from such.

1.1 MOTIVATION

The use of credit and debit cards has increased significantly in the last years, unfortunately so has fraud. Because of that, billions of Euros are lost every year. According to the European Central Bank (European Central Bank, 2014), during 2012 the total level of fraud reached 1.33 billion Euros in the Single Euro Payments Area, which represents an increase of 14.8% compared with 2011. Moreover, payments across non-traditional channels (mobile, internet, etc.) accounted for 60% of the fraud, whereas it was 46% in 2008. This opens new challenges as new fraud patterns emerge, and current fraud detection systems are less successful in preventing these frauds. The use of machine learning in fraud detection has been an interesting topic in recent years. Several detection systems based on machine learning techniques have been successfully used for this problem. Credit card fraud detection is a cost-sensitive problem, in the sense that the cost due to a false positive is different than the cost of a false negative. When predicting a transaction as fraudulent, when in fact it is not a fraud, there is an administrative cost that is incurred by the financial institution. On the other hand, when failing to detect a fraud, the amount of that transaction is lost. Moreover, it is not enough to assume a

constant cost difference between false positives and false negatives, as the amount of the transactions varies quite significantly; therefore, its financial impact is not constant but depends on each transaction. As per this we have used five machine learning models and compared their Accuracy, TPR, FPR, G-mean, Recall, Precision, Specificity and F1-Score. All machine learning algorithms are evaluated using a real-world credit card transaction to identify fraud or non-fraud transaction. The main motive of this project is to apply supervised learning methods on the real-world dataset. When constructing a credit card fraud detection model, it is very important to use those features that allow accurate classification. Typical models only use raw transactional features, such as time, amount, place of the transaction. However, these approaches do not consider the spending behaviour of the customer, which is expected to help discover fraud patterns.

1.2PROBLEM STATEMENT

Credit card fraud is a significant concern for both customers and issuers. Cardholders risk having their identities stolen and their accounts utilized without their knowledge. Fraudulent transactions and chargebacks pose a danger to issuers. There are several methods for detecting credit card fraud. One method is to keep an eye on account activity for unusual transactions. Another approach is to employ data analytics to uncover fraudulent activity patterns. Detecting credit card theft is difficult because thieves are always devising new methods to avoid detection. To protect oneself against fraud, issuers and cardholders must be alert. System for detecting credit card fraud Credit card fraud refers to theft and fraud conducted using or involving a payment card, such as a credit or debit card, as an illegitimate source of funds in a transaction. Credit card fraud detection 1 abstract Due to the widespread usage of credit cards, they have become an appealing target for fraudsters. Credit card number theft has become frequent in today's culture. System for detecting credit card fraud A credit card is a plastic card that is used to buy goods and services on credit. In this article, we will apply data mining approaches to identify credit card fraud. Credit card fraud is a significant issue that affects everyone who uses a credit card. It may be extremely costly for organizations and create significant financial hardship for individuals. There are several methods for committing credit card fraud, and it is critical to be aware of all of them. There are a few things you can take to assist avoid credit card fraud, and it is critical that you are aware of them. Every year, millions of Americans are victimized by credit card theft. Criminals may perpetrate credit card fraud in

a variety of methods, the most frequent of which is identity theft. Identity thieves steal your personal information, such as your name, SSN, and credit card number, and use it to establish new credit card accounts in your name. They then rack up big charges on these accounts that they do not pay, leaving you with the bill. Credit card fraud is expensive and can harm your credit rating. It can also be challenging to identify and avoid. If you believe you are a victim of credit card fraud, you can take steps to safeguard yourself and your assets. There have always been people who will find new ways to access someone's finances illegally since the advent of e-commerce payment systems. This is a major issue in the modern era because all transactions can be easily completed online by simply entering your credit card information. How can we use different techniques to prevent fraud transactions when using an automated system? In 2017, there were 16.7 million victims of unauthorised card operations. Furthermore, the number of credit card fraud claims in 2017 was 40% higher than the previous year, according to the Federal Trade Commission (FTC). There were approximately 13,000 reported cases in California and 8,000 in Florida, the two states with the highest per capita numbers of such cases.

1.3 LITERATURE SURVEY

Fraud acts as unlawful or criminal deception intended to result in financial or personal benefit. It is a deliberate act that is against the law, rule, or policy with an aim to attain unauthorized financial benefit. We went through several types of papers and analysis of website kaggle which provide dataset. In a paper, Suman, Research Scholar, GJUS&T at Hisar HCE presented techniques like Supervised and Unsupervised Learning for credit card fraud detection. Even though these methods and algorithms fetched an unexpected success in some areas, they failed to provide a permanent and consistent solution to fraud detection. In an investigation proposed by John O., Adebayo O. Adetunmbi, and Samuel A. Oluwadaren Awoyemi through which the performances of several algorithms were evaluated when they were applied on credit card fraud data that is highly skewed. The European cardholders‟ 284,807 transactions were used as a source to generate the dataset of credit card transactions. Logistic regression and artificial neural network give flags whenever fraudulent and legitimate transaction happens based upon their transaction score. The performance of all the machine learning models decreases because of the skewness of the training dataset. To make the unbalanced dataset balanced two different methods are used namely, intrinsic features and network-based features. Intrinsic features

compare customer's past transactions looks for any pattern in it. Network-based features work by exploiting the network of credit card holders and merchants and deriving a time-dependent two methods lead to a very high accuracy score in. Random Forest getting a 1% false positive making the perfect model obtaining fraudulent transaction Comparisons are made between different modelling and algorithm techniques on a real dataset. Some of the algorithms underperform because of the unbalanced dataset. To learn from (non-stream credit card and data stream) unbalanced dataset has three different methods used (static, update and DataStream). They also used two methods of under sampling SMOTE and Easy Ensemble to make their dataset balanced from an unbalanced dataset. On the skewed data, a hybrid approach of under-sampling and oversampling is performed. On raw and pre-processed data, there are three different techniques applied in Python. Based on certain limited parameters like precision, sensitivity, accuracy, balanced classification rate and so on, the performances of these techniques are evaluated. They concluded that in comparison to naïve Bayes and logistic regression approaches, the performance of RF is better. Fraud detection is a data mining problem with an aim of segregating transactions into two classes – legitimate and fraudulent (Duman and Ozcelik 2011). Recent fraud detection systems used by merchants and banks are designed to verify transactions by checking spending patterns and behavior of customers (Quah and Sriganesh 2008). To achieve this, fraud detection systems use prediction algorithms to classify pattern observations (Maes et al. 2002). A transaction will be labeled fraudulent if the system observes a deviation in the normal spending pattern of a user. The following are some techniques used in credit card detection (Quah and Sriganesh 2008): Transaction verification through Address Verification System (AVS) using customer zip code. Transaction verification through Card Verification Method (CVM) using a secret number entered by the customer. Transaction verification through Personal Information Number (PIN) using a secret number to be provided by the customer during transactions. Transaction verification through biometrics using Fingerprint. A good fraud detection solution should be capable of reducing high risk fraud to the barest minimum (Duman and Ozcelik 2011). High risk fraud refers to fraud that leads to huge money loss. When a card is stolen, its entire credit limit is habitually the major target. Fraudsters exhaust the credit limit within 4–5 transactions (Duman and Ozcelik 2011). The higher the credit limit, the higher the loss. Misclassification cost of each transaction varies, hence, good credit card fraud detection systems should give priority to transactions with higher misclassification cost. The following are some other characteristics of a good fraud detection system (Maes et al. 2002): Skewed Distribution: A good fraud detection system should be capable of handling skewed distributions. This is because only few fraudulent transactions are

in record. This challenge can be handled by dividing the training dataset into different parts having a reduced skewed distribution. Noise: A dataset is said to be noisy if it contains corrupted or erroneous data. A good fraud detection system should be able to handle noise. Generally, noise affects the performance of a classifier. Overlapping Data: A good fraud detection system should be able to detect fraudulent transactions that look very similar to legitimate transactions. Dynamism: Techniques used by fraudsters change overtime. A good fraud detection system should be dynamic; it should be able to adjust to changes in fraudulent patterns. Good Classification Metrics: The classification metric used in evaluating fraud detection techniques should be chosen carefully. This is because; metrics like classification accuracy is not suitable for skewed distribution. Credit card transactions has two unique peculiarities (Patidar and Sharma 2011). The first peculiarity is centered on the number of credit card transactions. The number of credit card transactions processed at a particular time is numerous (Patidar and Sharma 2011). The second unique attribute is time. Card users have limited time to either reject or accept a given transaction (Patidar and Sharma 2011). On a daily basis, millions of visa card operations are performed by users worldwide, and 98% of these transactions are online based (Patidar and Sharma 2011). Security of credit cards depends on card owners. It depends on how efficient a user can secure his card and card number from theft. We think RF approach is reliable because it is good at handling data as it is memory-based approach in which anyone can use both the classification types (Binary class and Multi class) that too without any extra efforts, also we can use with classification and Regression both. The detection of fraudulent credit card transaction is a challenging task due to the following reasons:

(i) The frequent changes in the patterns of normal and fraudulent activities and

(ii) The high level of skewness related with credit card fraud datasets.

They have used various machine learning techniques to predict credit card fraud in bank system that based on the analysis of the results. They have proposed random forest which has prediction accuracy is more than 80%. According to them banks can use machine learning to measure credit risk of customers before surrendering them credit card. Banks main worry in to offer treasured harvests and facilities to their consumers and in order save up with their contestants they must stay advanced and creative. Randhawa, Kuldeep, et al (2018)they have presented credit card fraud detection by using machine learning algorithms. Some typical models that are NB, SVM, and DL have used in the empirical study. They have proposed the best MCC score is 82% that is achieved by vote. Additional assess the hybrid mock-ups, noise

from10% - 30% has been additional into the data models. Sarah Alexandria Ebiaredoh-Mienye, et al (2020) machine learning algorithms are ineffectual for large datasets performing classification such as large credit card data set. They have proposed stacked sparse auto encoder network to gain optimal features learning. They have introduced batch normalization methods to increase the outcomes and speed of the model and further prevent over fitting. They model was optimized by using Adamax algorithm.

Somayeh Moradi, et al (2019) they have proposed a dynamic model for credit card fraud risk to valuation that outperformance the model used. There model has a self-motivated appliance that evaluates the behavior of corrupt clients in a once-a-month basis, credit risk that include the fuzzy factors, particularly in the financial crises. Their approach can utilize changing indeterminate issues. Vaishnavi Nath Dornadula, et al (2019) they have used novel method to identify credit card fraud detection. Various classifiers are used on three altered collections advanced assessment scores are produced for each type of classifier. These self-motivated variations in strictures lead the organization to familiarize system. They have proposed that decision tree, random forest and logistic regression provided the best results and accuracy. In previous studies, many methods have been implemented to detect fraud using supervised, unsupervised algorithms and hybrid ones. Fraud types and patterns are evolving day by day. It is important to have clear understanding of technologies behind fraud detection. Here discuss machine learning models, algorithms and fraud detection models used in earlier studies, data mining techniques are discussed, and these methods take time dealing with huge data. Overlapping is another problem with credit card transaction data preparation. Imbalanced data distribution is overcome using sampling methods.

Fraud transaction are quite a less compared to normal transaction. When normal transaction looks like fraudulent, or fraud transaction appear as legitimate. Also discuss about difficulties in dealing categorial data. Many machine learning algorithms will not support categorial data. Discuss about the detection cost and adaptability as a challenge. Prevention cost and cost of fraudulent behaviour are taken into consideration. How to handle it and discuss how to work on large dataset. The implemented work was overcome these challenges. Many models are implemented for fraud detection. In every model different algorithm are used. Detection of credit card fraud for new frauds will be problematic if new data has drastic changes in fraud patterns. Replacing the model is risky as machine learning algorithm take much time for training rather than predicting. Logistic Regression algorithm (LR) is implemented to sort the classification problem. Using Gaussian Mixture Models fraudulent cases are discretized. To balance data synthetic minority oversampling is used. Sensitivity analysis is used

calculate economic value this Risk Based Ensemble model is used this model can give good results for data with issues and to remove implicit noise in transaction Naive Bayes algorithm is used. They discuss about huge real time date is a main issue. Real life data contains privacy and sensitive, so it is difficult to analyse and implement algorithms. They were evaluated using both benchmark and real-world data. A summary of the strengths and limitations of the methods were evaluated. The Matthews Correlation Coefficient metric (MCC) has been taken as the performance measure. To evaluate the robustness of the algorithms noise was added to the data. Also, they have proved that the majority voting method was not affected by the added noise. They made comparison study of all the algorithms, and they showed algorithms behave differently for different situation of problems.

1.4 LIMITATIONS OF EXISTING WORK

- Transactional data is not reliable to work with high dimensionality.
- Scaled data is required for detecting the credit card fraud.
- To build a viable machine learning model with long-term functionality, it requires a great deal of good data – which isn't easy to obtain.

CHAPTER 2

SOFTWARE AND HARDWARE SPECIFICATIONS

2.1 SYSTEM REQUIREMENTS

A requirement is a feature that the system must have or a constraint that it must be accepted by the client. Requirement Engineering aims at defining the wants of the system under construction. Requirement Engineering include two main activities requirement elicitation which results in the specification of the system that the client understands and analysis which in analysis model that the developer can unambiguously interpret. A requirement may be a statement about what the proposed system will do. Requirements can be divided into two major categories:

- Functional Requirements.
- Non-Functional Requirements.

2.1.1 FUNCTIONAL REQUIREMENTS

A Functional Requirement may be a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software, its behaviour, and outputs. It is often a calculation, data manipulation, business process, user interaction, or the other specific functionality which defines what function a system is probably going to perform. Functional Requirements describe the interactions between the system and its environment independent of its application.

- Applying the algorithms on the test data.

2.1.2 NON-FUNCTIONAL REQUIREMENTS

Non-Functional Requirements specifies the standard attribute of a software. They judge the software supported Responsiveness, Usability, Security, Portability, and other 34 non-functional standards that are critical to the success of the software. An example of a non-functional requirement, “how fast does the website load?” Failing to satisfy non-functional requirements may result in systems that fail to satisfy user needs. Non-

functional Requirements allow you to impose constraints or restrictions on the planning of the system across the varied agile backlogs.

- Accuracy
- Reliability
- Flexibility
- Portability
- Maintainability
- Availability
- Scalability

➢ **Accuracy:**

The model here uses five algorithms for the prediction as mentioned above. The accuracy of each algorithm is as follows:

Table 1: Algorithms vs Accuracies

Algorithm	Accuracy
Random Forest Classifier	99.9614%
Support Vector Machine	99.9427%
Decision Tree	99.9228%
Naïve Bayes	99.3048%
Logistic Regression	99.8982%

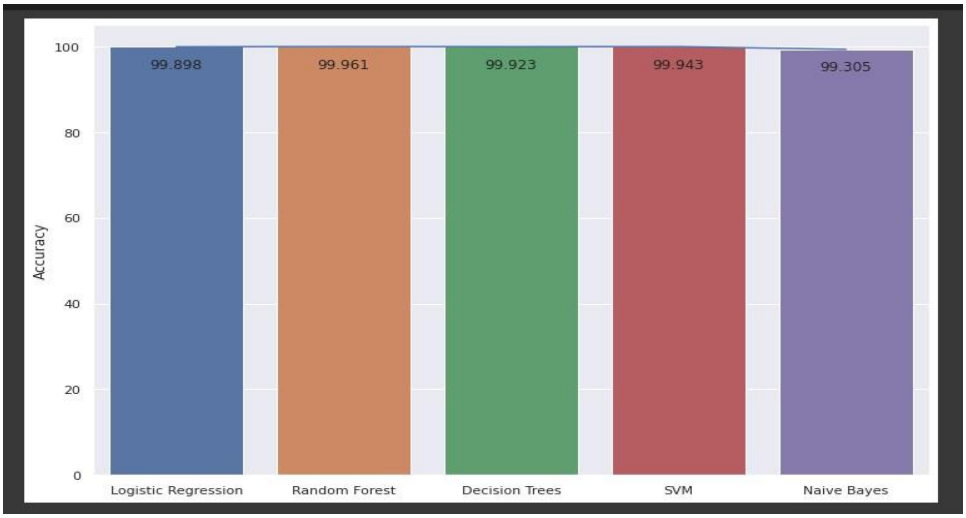


Figure 2.1: Accuracies of different algorithms

- **Reliability:**

  The product should not fail in mid of any operations carrying out.

- **Portability:**

  The project should be executable on any Windows OS.

- **Availability:**

  The software can be used anytime and the capacity of a user to obtain information or resources in a certain place and format.

- **Maintainability:**

  In software engineering, maintainability is the ease with which a software product can be modified to include new functionalities in the project based on the user requirements just by adding the appropriate files to existing project. Since python programming is very simple, it is easier to find and correct the defects and to make the changes in the project.

- **Flexibility:**

  The capacity to enhance the model's degrees of freedom for "fitting" to the training data.

- **Scalability:**

  System can work normally under situations such as low bandwidth.

2.2 HARDWARE REQUIREMENTS

Table 2: Hardware specifications

Peripheral Devices: Monitor, Mouse, and Keyboard.
Hard Disk Drive: 20 GB (free).
Pentium: i3 or higher
RAM: 4 GB or higher

2.3 SOFTWARE REQUIREMENTS

- **Operating System: Windows 10 or above**
- **Google Colab**
- **Python Libraries: NumPy,Pandas,Matplotlib,Sklearn…etc**

➢ GOOGLE COLAB

Colaboratory, sometimes known as "Colab," is a Google Research product. Colab is particularly well suited to machine learning, data analysis, and teaching. It enables anybody to create and execute arbitrary Python code through the browser. Technically speaking, Colab is a hosted Jupyter notebook service that offers free access to computer resources, including GPUs, and requires no setup to use. Jupyter is the open source project on which Colab is based. Colab allows you to use and share Jupyter notebooks with others without having to download, install, or run anything. If you've used Jupyter notebook before, you'll pick up Google Colab quickly. Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, no setup is required, and the notebooks you create can be edited concurrently by your team members, just like documents in Google Docs. Many popular machine learning libraries are supported by Colab and can be easily loaded in your notebook. You may use Colab's free edition to access virtual machines with a typical system memory profile. You can access machines with a high memory system profile in Colab's paid editions, subject to availability and your compute unit balance. Note that the term "memory" refers to system memory. The memory profile is the same across all GPU chips. Colab can offer resources without charging a fee in part because it has dynamic use caps that occasionally change and does not guarantee or give infinite resources. This implies that total utilisation caps, idle timeout durations, maximum VM lifetimes, available GPU kinds, and other parameters change over time. These restrictions are not made public by Colab in part because they can—and occasionally do—change fast.

As a programmer, you can perform the following using Google Colab.

- Write and execute code in Python
- Document your code that supports mathematical equations
- Create/Upload/Share notebooks
- Import/Save notebooks from/to Google Drive
- Import/Publish notebooks from GitHub
- Import external datasets e.g., from Kaggle
- Integrate PyTorch, TensorFlow, Keras, OpenCV
- Free Cloud service with free GPU

Support for multiple programming languages

- Python
- Julia
- Haskell
- Matlab
- Java
- Scala
- Ruby

➢ PYTHON LIBRARIES

o NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the elemental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code

Useful linear algebra, Fourier transform, and random number capabilities Besides its obvious scientific uses, NumPy also can be used as an efficient multidimensional container of generic data.

o Pandas

Pandas is an open-source library that's built on top of NumPy library. It is a Python package that gives various data structures and operations for manipulating numerical data and time series. It is fast and it has high-performance & productivity for users. It provides high-performance and is easy-to-use data structures and data analysis tools for the Python language. Pandas is employed during a wide range of fields including academic and commercial domains including economics, Statistics, analytics, etc.

- **Sklearn:**

  Scikit-learn (Sklearn) is that the most useful and robust library for machine learning in Python. It is an open-source Python library that implements a variety of machine learning, pre-processing, cross-validation and visualization algorithms employing a unified interface. Sklearn provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is essentially written in Python, is made upon NumPy, SciPy and Matplotlib. Pickle: Python pickle module is employed for serializing and de-serializing a Python object structure. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the thing in another python script. Pickling is benefitial for applications where you would like a point of persistency in your data. Your program's state data are often saved to disk, so you‟ll continue working on it later.

- **Matplotlib**

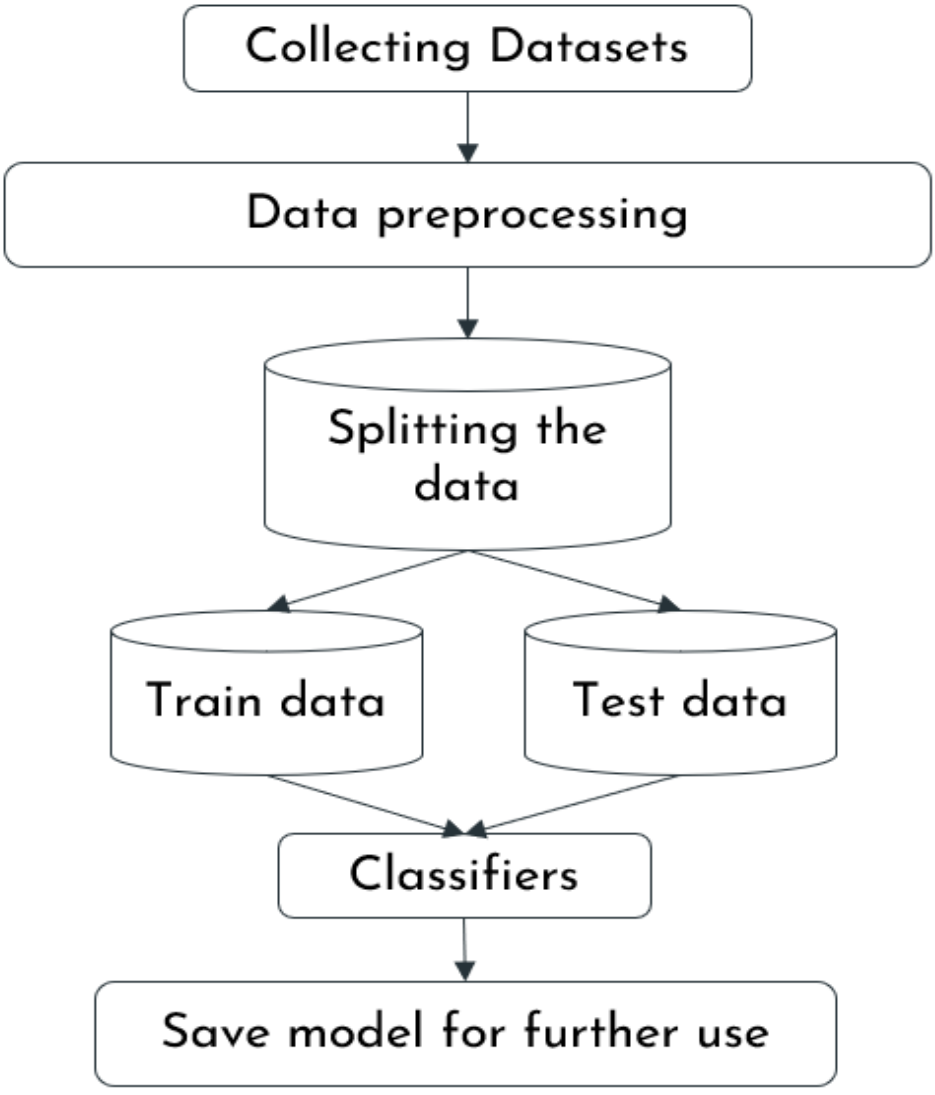
  It is a very powerful plotting library useful for those working with Python and NumPy. And for creating statistical interference, it becomes very necessary to visualize our data and Matplotlib is that the tool which will be very helpful for this purpose. It provides MATLAB like interface only difference is that it uses Python and is open source. Seaborn: Seaborn may be a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is that the central part of Seaborn which helps in exploration and understanding of data. It offers the following functionalities:

  - Dataset oriented API to determine the relationship between variables.
  - Automatic estimation and plotting of linear regression plots.
  - It supports high-level abstractions for multi-plot grids.

CHAPTER 3

DESIGN

The proposed system focuses on predicting the nature of a transaction. On the output side whether the transaction is fraudulent or not is decided. Using RF approach, credit card fraud detection severity can be forecasted. In this process it is required to train data using RF algorithm to predict nature of the transaction. To extract patterns from a credit card dataset, and then build a model based on these extracted patterns. The training data set is now supplied to machine learning model; Based on this data set the model is trained. Every transaction act as a test data set. After the operation of testing, predict whether the transaction is fraudulent or not. To extract essential information and predict if a credit card transaction is non-fraudulent.



3.1 USECASE DIAGRAM

The project fraud detection using machine learning Use Case diagram includes all of the key parts that a standard use case diagram requires. This use case diagram depicts how the model progresses from one stage to the next; here, the use case diagrams of all the entities are linked to each other, and the user is introduced to the system.

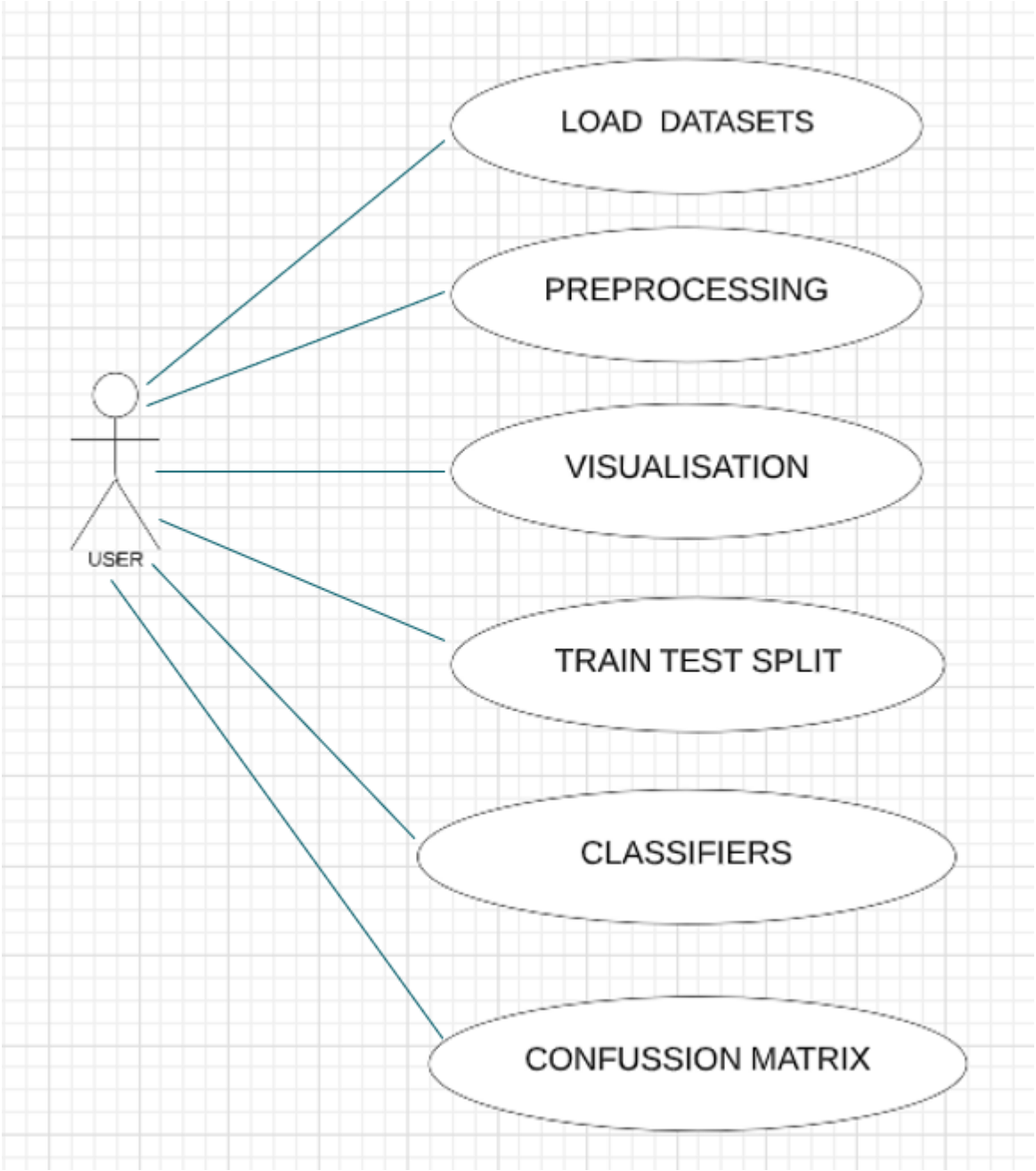


Figure 3.1: Use Case Diagram

3.2 CLASS DIAGRAM

Fraud detection using machine learning consists of a class diagram that all the other applications that consist of the basic class diagram, here the class diagram is the basic entity that is required to carry on with the project. Class diagrams consist of information about all the classes that is used and all the related datasets, and all the other necessary attributes and their relationships with other entities, all these information is necessary in order to use the concept of the prediction.

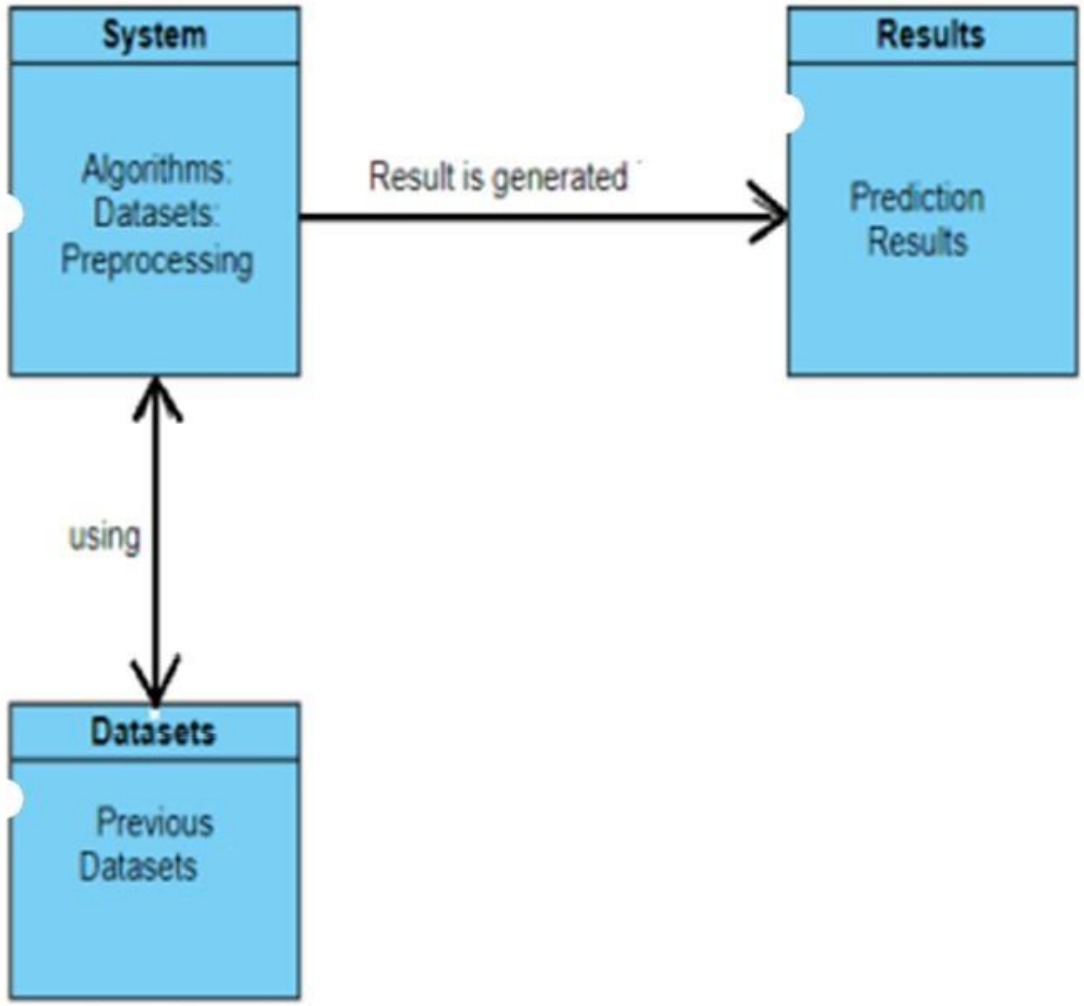


Figure 3.2: Class Diagram

3.3 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another.

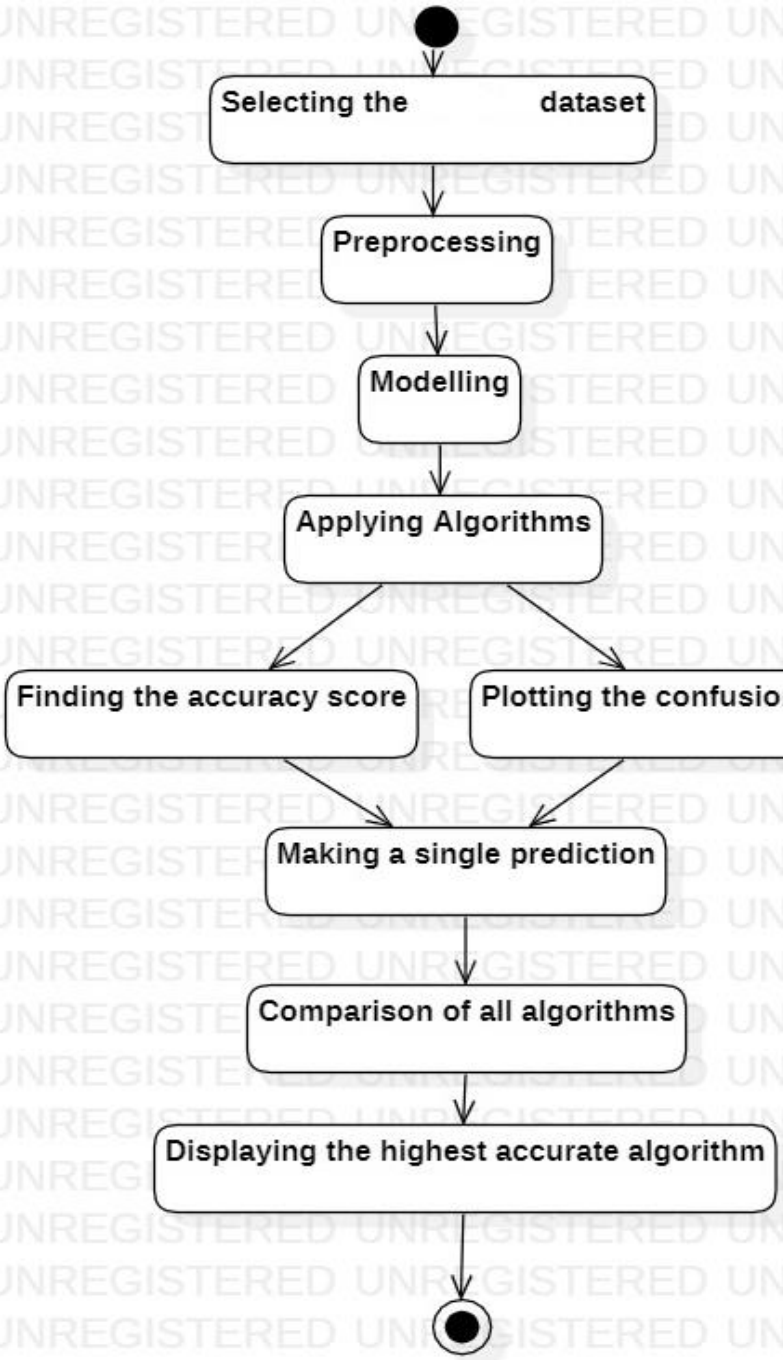


Figure 3.3: Activity Diagram

3.4 SEQUENCE DIAGRAM

The Sequence diagram of the project fraud detection using machine learning consists of all the various aspects a normal sequence diagram requires. This sequence diagram shows how from starting the model flows from one step to other and it compares the data set with the detection model and if true is predicts the appropriate results.

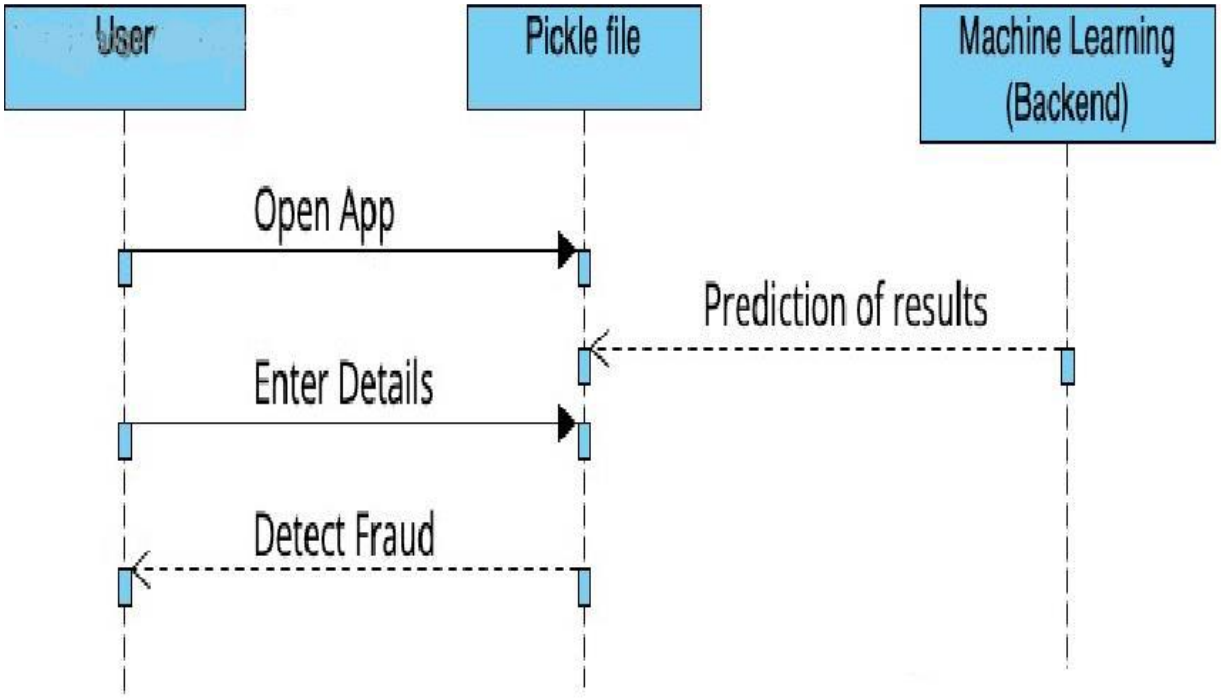


Fig 3.4: Sequence Diagram

3.5 DEPLOYMENT DIADRAM

A deployment diagram shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of structure diagram used in modelling the physical aspects of an object-oriented system. Here the deployment diagram shows the final stage of the project, and it also shows how the model looks after doing all the processes and deploying in the machine. Then the training and testing data uses the algorithms such as RF, Naïve Bayes, Decision Tree, LR, SVM. Then finally processing all that data and information, the system gives the desired result in the interface.

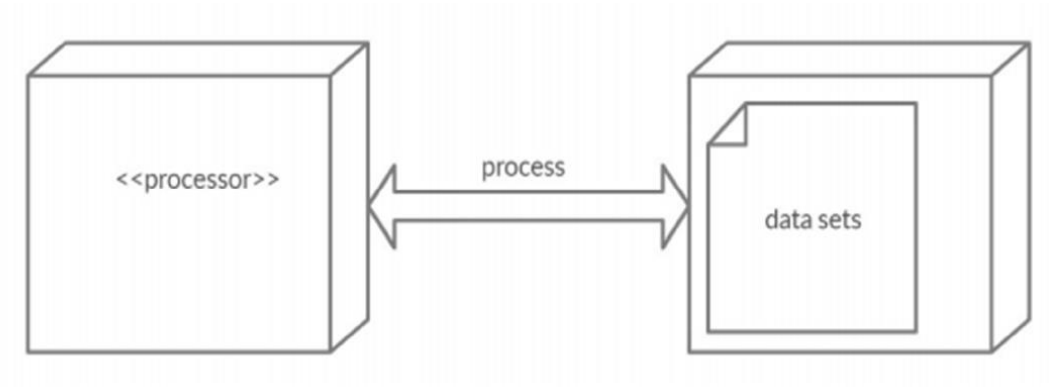


Fig 3.5: Deployment Diagram

3.6 TECHNOLOGY DESCRIPTION

3.6.1 Machine Learning:

Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for *building mathematical models and making predictions using historical data or information.* Machine learning (ML) is defined as a discipline of artificial intelligence (AI) that provides machines the ability to automatically learn from data and past experiences to identify patterns and make predictions with minimal human intervention. Machine learning is the process of teaching computers to learn from data without explicitly programming them. A subset of artificial intelligence is machine learning (AI). Machine learning algorithms create data-driven models that can be used to make predictions or recommendations. Machine learning can be found in a wide range of applications, including email filtering, fraud detection, and computer vision. Machine learning algorithms are classified into three types: supervised, unsupervised, and reinforcement learning. Unsupervised learning algorithms work with unlabelled data, whereas supervised learning algorithms require labelled data. Reinforcement learning algorithms learn through interaction with their surroundings. Machine learning is a rapidly evolving field in which new algorithms and applications are constantly being developed. Machine Learning may be a sub-area of AI, whereby the term refers to the power of IT systems to independently find solutions to problems by recognizing patterns in databases. In other words: Machine Learning enables IT systems to acknowledge patterns in the idea of existing algorithms and data sets and to develop adequate solution concepts. Therefore, in Machine Learning, artificial knowledge is

generated on the idea of experience. In order to enable the software to independently generate solutions, the prior action of 5 people is important. For example, the required algorithms and data must be fed into the systems in advance and the respective analysis rules for the recognition of patterns in the data stock must be defined. Once these two steps have been completed, the system can perform the following tasks by Machine Learning: Finding, extracting and summarizing relevant data . Making predictions based on the analysis data . Calculating probabilities for specific results Basically, algorithms play a crucial role in Machine Learning: On the one hand, they're liable for recognizing patterns and on the opposite hand, they will generate solutions. Algorithms can be divided into different categories:

Supervised learning: During monitored learning, example models are defined beforehand. To make sure an adequate allocation of the knowledge to the respective model groups of the algorithms, these then need to be specified. In other words, the system learns on the idea of given input and output pairs. within the course of monitored learning, a programmer, who acts as a sort of teacher, provides the acceptable values for specific input. The aim is to coach the system within the context of successive calculations with different inputs and outputs to determine connections. Supervised learning is where you've got input variables (X) and an output variable (Y) and you employ an algorithm to find out the mapping function from the input to the output. Y = f(X) The goal is to approximate the mapping function so well that once you have a new input file (X) that you simply can predict the output variables (Y) for that data. It's called supervised learning because the method of an algorithm learning from the training dataset is often thought of as an educator supervising the training process. We all know the correct answers, the algorithm iteratively makes predictions on the training data and is corrected. Learning stops when the algorithm achieves a suitable level of performance. Techniques of Supervised Machine Learning algorithms include linear and logistic regression, multi-class classification, Decision Tree, and Support Vector Machine. Supervised Learning problems are a kind of machine learning technique often further grouped into Regression and Classification problems. The difference between these two is that the dependent attribute is numerical for regression and categorical for classification: Regression: Linear regression could also be a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and thus the only output variable (y). More specifically, that y is usually calculated from a linear combination of the input variables (x). When there's one input variable (x), the tactic is

mentioned as simple linear regression. When there are multiple input variables, literature from statistics often refers to the tactic as multiple linear regression. Classification: Classification could also be a process of categorizing a given set of data into classes, It is often performed on both structured or unstructured data. the tactic starts with predicting the category of given data points. The classes are often mentioned as target, label, or categories. In short, classification either predicts categorical class labels or classification data supported the training set and thus the values (class labels) in classifying attributes and uses it in classifying new data. There is a variety of classification models. Classification models include Logistic Regression, Decision Tree, Random Forest, and Naïve Bayes.

In unsupervised learning, AI learns without predefined target values and without rewards. It's mainly used for learning segmentation (clustering). The machine tries to structure and type the info entered consistent with certain characteristics. For instance, a machine could (very simply) learn that coins of various colors are often sorted consistent with the characteristic "colour" so as to structure them. Unsupervised Machine Learning algorithms are used when the knowledge used to train is neither classified nor labelled. The system doesn't determine the right output but it explores the data and should draw inferences from datasets to elucidate hidden structures from unlabelled data. Unsupervised Learning is that the training of Machines using information that's neither classified nor labelled and allowing the algorithm to act thereon information without guidance. Unsupervised Learning is accessed into two categories of algorithms: Clustering: A clustering problem is where you would like to get the inherent grouping in the data such as grouping customers by purchasing behaviour. Association: An Association rule learning problem is where you would wish to get rules that describe large portions of your data such as folks that buy X also tend to shop for Y.

3.6.2 Machine Learning in Credit Card Fraud Detection:

The digital payments market is soaring as the world shifts towards online and card-based payment methods at a faster rate. With such a shift comes the growing issue of cybersecurity and fraud, which is more common than ever. Machine learning (ML), credit card fraud detection is becoming easier and more efficient.ML-based fraud detection solutions can track patterns and prevent abnormal transactions. Machine

learning models can recognize unusual credit card transactions and fraud. The first and foremost step involves collecting and sorting raw data, which is then used to train the model to predict the probability of fraud.

A machine learning model can quickly identify any drifts from regular transactions and user behaviours in real time. By recognizing anomalies, such as a sudden increase in transactional amount or location change, ML algorithms can minimize the risk of fraud and ensure more secure transactions. One approach would be to employ machine learning algorithms to spot strange trends in credit card usage data that might signal fraud. Another approach is to utilize machine learning to create models that estimate the possibility of fraud based on criteria such as the kind of transaction, the amount of money involved, and the transaction's location. Machine learning may also be used to discover tendencies in credit card theft and build prevention tactics. Machine learning is increasingly being used by credit card firms to detect and prevent fraud, and the technology is growing more complex all the time. Once an algorithm picks up different transactional patterns and behaviours, it can efficiently work with large datasets to separate authentic payments from fraudulent ones. The models can analyse huge amounts of data in seconds while offering real-time insights for improved decision-making capabilities. Using this machine learning in credit card fraud detection we have used 5 classifiers they are:

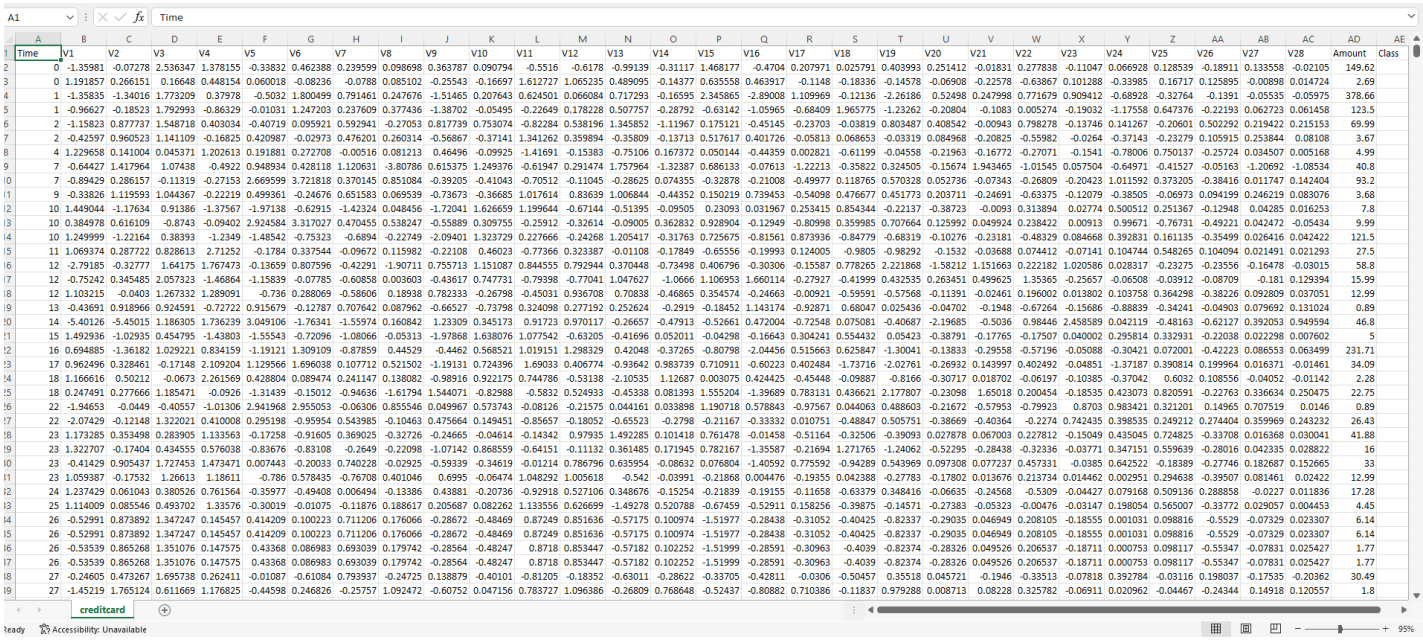
- Logistic Regression
- Decision Tree
- Naïve Bayes
- Support Vector Machine
- Random Forest

CHAPTER 4

IMPLEMENTATION

4.1 DATA SET:

The data set in Fig. 4.1 is a sample dataset, and it is taken from Kaggle. It includes 31 characteristics since it is a pre-processed dataset, and the variables include time, quantity, V-1, 2…..28, as well as class. This dataset is in the form of principal component analysis (PCA), it is a mathematical process. It transforms a set of data into new set of variables, these new set of variables are also called ass principal components non corelated with each other.



Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	-1.35981	-0.07278	2.536347	1.378155	-0.33832	0.462388	0.239599	0.098698	0.363787	0.090794	-0.5516	-0.6178	-0.99139	-0.31117	1.468177	-0.4704	0.207971	0.025791	0.403993	0.251412	-0.01831	0.277838	-0.11047	0.066928	0.128539	-0.18911	0.133558	-0.02105	149.62	
0	1.191857	0.266151	0.16648	0.448154	0.060018	-0.08236	-0.0788	0.085102	-0.25543	-0.16697	1.612727	1.065235	0.489095	-0.14377	0.635558	0.463917	-0.1148	-0.18336	-0.14578	-0.06908	-0.22578	-0.63867	0.101288	-0.33985	0.16717	0.125895	-0.00898	0.014724	2.69	
1	-1.35835	-1.34016	1.773209	0.37978	-0.5032	1.800499	0.791461	0.247676	-1.51465	0.207643	0.624501	0.066084	0.717293	-0.16595	2.345865	-2.89008	1.109969	-0.12136	-2.26186	0.52498	0.247998	0.771679	0.909412	-0.68928	-0.32764	-0.1391	-0.05535	-0.05975	378.66	
1	-0.96627	-0.18523	1.792993	-0.86329	-0.01031	1.247203	0.237609	0.377436	-1.38702	-0.05495	-0.22649	0.178228	0.507757	-0.28792	-0.63142	-1.05965	-0.68409	1.965775	-1.23262	-0.20804	-0.1083	0.005274	-0.19032	-1.17558	0.647376	-0.22193	0.062723	0.061458	123.5	
2	-1.15823	0.877737	1.548718	0.403034	-0.40719	0.095921	0.592941	-0.27053	0.817739	0.753074	-0.82284	0.538196	1.345852	-1.11967	0.175121	-0.45145	-0.23703	-0.03819	0.803487	0.408542	-0.00943	0.798278	-0.13746	0.141267	-0.20601	0.502292	0.219422	0.215153	69.99	
2	-0.42597	0.960523	1.141109	-0.16825	0.420987	-0.02973	0.476201	0.260314	-0.56867	-0.37141	1.341262	0.359894	-0.35809	-0.13713	0.517617	0.401726	-0.05813	0.068653	-0.03319	0.084968	-0.20825	-0.55982	-0.0264	-0.37143	-0.23279	0.105915	0.253844	0.08108	3.67	
4	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.00516	0.081213	0.46496	-0.09925	-1.41691	-0.15383	-0.75106	0.167372	0.050144	-0.44359	0.002821	-0.61199	-0.04558	-0.21963	-0.16772	-0.27071	-0.1541	-0.78006	0.750137	-0.25724	0.034507	0.005168	4.99	
7	-0.64427	1.417964	1.07438	-0.4922	0.948934	0.428118	1.120631	-3.80786	0.615375	1.249376	-0.61947	0.291474	1.757964	-1.32387	0.686133	-0.07613	-1.22213	-0.35822	0.324505	-0.15674	1.943465	-1.01545	0.057504	-0.64971	-0.41527	-0.05163	-1.20692	-1.08534	40.8	
7	-0.89429	0.286157	-0.11319	-0.27153	2.669599	3.721818	0.370145	0.851084	-0.39205	-0.41043	-0.70512	-0.11045	-0.28625	0.074355	-0.32878	-0.21008	-0.49977	0.118765	0.570328	0.052736	-0.07343	-0.26809	-0.20423	1.011592	0.373205	-0.38416	0.011747	0.142404	93.2	
9	-0.33826	1.119593	1.044367	-0.22219	0.499361	-0.24676	0.651583	0.069539	-0.73673	-0.36685	1.017614	0.83639	1.006844	-0.44352	0.150219	0.739453	-0.54098	0.476677	0.451773	0.203711	-0.24691	-0.63375	-0.12079	-0.38505	-0.06973	0.094199	0.246219	0.083076	3.68	
10	1.449044	-1.17634	0.91386	-1.37567	-1.97138	-0.62915	-1.42324	0.048456	-1.72041	1.626659	1.199644	-0.67144	-0.51395	-0.09505	0.23093	0.031967	0.253415	0.854344	-0.22137	-0.38723	-0.0093	0.313894	0.02774	0.500512	0.251367	-0.12948	0.04285	0.016253	7.8	
10	0.384978	0.616109	-0.8743	-0.09402	2.924584	3.317027	0.470455	0.538247	-0.55889	0.309755	-0.25912	-0.32614	-0.09005	0.362832	0.928904	-0.12949	-0.80998	0.359985	0.707664	0.125992	0.049924	0.238422	0.00913	0.99671	-0.76731	-0.49221	0.042472	-0.05434	9.99	
10	1.249999	-1.22164	0.38393	-1.2349	-1.48542	-0.75323	-0.6894	-0.22749	-2.09401	1.323729	0.227666	-0.24268	1.205417	-0.31763	0.725675	-0.81561	0.873936	-0.84779	-0.68319	-0.10276	-0.23181	-0.48329	0.084668	0.392831	0.161135	-0.35499	0.026416	0.042422	121.5	
11	1.069374	0.287722	0.828613	2.71252	-0.1784	0.337544	-0.09672	0.115982	-0.22108	0.46023	-0.77366	0.323387	-0.01108	-0.17849	-0.65556	-0.19993	0.124005	-0.9805	-0.98292	-0.1532	-0.03688	0.074412	-0.07141	0.104744	0.548265	0.104094	0.021491	0.021293	27.5	
12	-2.79185	-0.32777	1.64175	1.767473	-0.13659	0.807596	-0.42291	-1.90711	0.755713	1.151087	0.844555	0.792944	0.370448	-0.73498	0.406796	-0.30306	-0.15587	0.778265	2.221868	-1.58212	1.151663	0.222182	1.020586	0.028317	-0.23275	-0.23556	-0.16478	-0.03015	58.8	
12	-0.75242	0.345485	2.057323	-1.46864	-1.15839	-0.07785	-0.60858	0.003603	-0.43617	0.747731	-0.79398	-0.77041	1.047627	-1.0666	1.106953	1.660114	-0.27927	-0.41999	0.432535	0.263451	0.499625	1.35365	-0.25657	-0.06508	-0.03912	-0.08709	-0.181	0.129394	15.99	
12	1.103215	-0.0403	1.267332	1.289091	-0.736	0.288069	-0.58606	0.18938	0.782333	-0.26798	-0.45031	0.936708	0.70838	-0.46865	0.354574	-0.24663	-0.00921	-0.59591	-0.57568	-0.11391	-0.02461	0.196002	0.013802	0.103758	0.364298	-0.38226	0.092809	0.037051	12.99	
13	-0.43691	0.918966	0.924591	-0.72722	0.915679	-0.12787	0.707642	0.087962	-0.66527	-0.73798	0.324098	0.277192	0.252624	-0.2919	-0.18452	1.143174	-0.92871	0.68047	0.025436	-0.04702	-0.1948	-0.67264	-0.15686	-0.88839	-0.34241	-0.04903	0.079692	0.131024	0.89	
14	-5.40126	-5.45015	1.186305	1.736239	3.049106	-1.76341	-1.55974	0.160842	1.23309	0.345173	0.91723	0.970117	-0.26657	-0.47913	-0.52661	0.472004	-0.72548	0.075081	-0.40687	-2.19685	-0.5036	0.98446	2.458589	0.042119	-0.48163	-0.62127	0.392053	0.949594	46.8	
15	1.492936	-1.02935	0.454795	-1.43803	-1.55543	-0.72096	-1.08066	-0.05313	-1.97868	1.638076	1.077542	-0.63205	-0.41696	0.052011	-0.04298	-0.16643	0.304241	0.554432	0.05423	-0.38791	-0.17765	-0.17507	0.040002	0.295814	0.332931	-0.22038	0.022298	0.007602	5	
16	0.694885	-1.36182	1.029221	0.834159	-1.19121	1.309109	-0.87859	0.44529	-0.4462	0.568521	1.019151	1.298329	0.42048	-0.37265	-0.80798	-2.04456	0.515663	0.625847	-1.30041	-0.13833	-0.29558	-0.57196	-0.05088	-0.30421	0.072001	-0.42223	0.086553	0.063499	231.71	
17	0.962496	0.328461	-0.17148	2.109204	1.129566	1.696038	0.107712	0.521502	-1.19131	0.724396	1.69033	0.406774	-0.93642	0.983739	0.710911	-0.60223	0.402484	-1.73716	-2.02761	-0.26932	0.143997	0.402492	-0.04851	-1.37187	0.390814	0.199964	0.016371	-0.01461	34.09	
18	1.166616	0.50212	-0.0673	2.261569	0.428804	0.089474	0.241147	0.138082	-0.98916	0.922175	0.744786	-0.53138	-2.10535	1.12687	0.003075	0.424425	-0.45448	-0.09887	-0.8166	-0.30717	0.018702	-0.06197	-0.10385	-0.37042	6.0032	0.108556	-0.04052	-0.01142	2.28	
18	0.247491	0.277666	1.185471	-0.0926	-1.31439	-0.15012	-0.94636	-1.61794	1.544071	-0.82988	-0.5832	0.524933	-0.45338	0.081393	1.555204	-1.39689	0.783131	0.436621	2.177807	-0.23098	1.65018	0.200454	-0.18535	0.423073	0.820591	-0.22763	0.336634	0.250475	22.75	
22	-1.94653	-0.0449	-0.40557	-1.01306	2.941968	2.955053	-0.06306	0.855546	0.049967	0.573743	-0.08126	-0.21575	0.044161	0.033898	1.190718	0.578843	-0.97567	0.044063	0.488603	-0.21672	-0.57953	-0.79923	0.8703	0.983421	0.321201	0.14965	0.707519	0.0146	0.89	
22	-2.07429	-0.12148	1.322021	0.410008	0.295198	-0.95954	0.543985	-0.10463	0.475664	0.149451	-0.85657	-0.18052	-0.65523	-0.2798	-0.21167	-0.33332	0.010751	-0.48847	0.505751	-0.38669	-0.40364	-0.2274	0.742435	0.398535	0.249212	0.274404	0.359969	0.243232	26.43	
23	1.173285	0.353498	0.283905	1.133563	-0.17258	-0.91605	0.369025	-0.32726	-0.24665	-0.04614	-0.14342	0.97935	1.492285	0.101418	0.761478	-0.01458	-0.51164	-0.32506	-0.39093	0.027878	0.067003	0.227812	-0.15049	0.435045	0.724825	-0.33708	0.016368	0.030041	41.88	
23	1.322707	-0.17404	0.434555	0.576038	-0.83676	-0.83108	-0.2649	-0.22098	-1.07142	0.868559	-0.64151	-0.11132	0.361485	0.171945	0.782167	-1.35587	-0.21694	1.271765	-1.24062	-0.52295	-0.28438	-0.32336	-0.03771	0.347151	0.559639	-0.28016	0.042335	0.028822	16	
23	-0.41429	0.905437	1.727453	1.473471	0.007443	-0.20033	0.740228	-0.02925	-0.59339	-0.34619	-0.01214	0.786796	0.635954	-0.08632	0.076804	-1.40592	0.775592	-0.94289	0.543969	0.097308	0.077237	0.457331	-0.0385	0.642522	-0.18389	-0.27746	0.182687	0.152665	33	
23	1.059387	-0.17532	1.26613	1.18611	-0.786	0.578435	-0.76708	0.401046	0.6995	-0.06474	1.048292	1.005618	-0.542	-0.03991	-0.21868	0.004476	-0.19355	0.042388	-0.27783	-0.17802	0.013676	0.213734	0.014462	0.002951	0.294638	-0.39507	0.081461	0.02422	12.99	
24	1.237429	0.061043	0.380526	0.761564	-0.35977	-0.49408	0.006494	-0.13386	0.43881	-0.20736	-0.92918	0.527106	0.348676	-0.15254	-0.21839	-0.19155	-0.11658	-0.63379	0.348416	-0.06635	-0.24568	-0.5309	-0.04427	0.079168	0.509136	0.288858	-0.0227	0.011836	17.28	
25	1.114009	0.085546	0.493702	1.33576	-0.30019	-0.01075	-0.11876	0.188617	0.205687	0.082262	1.133556	0.626699	-1.49278	0.520788	-0.67459	-0.52911	0.158256	-0.39875	-0.14571	-0.27383	-0.05323	-0.00476	-0.03147	0.198054	0.565007	-0.33772	0.029057	0.004453	4.45	
26	-0.52991	0.873892	1.347247	0.145457	0.414209	0.100223	0.711206	0.176066	-0.28672	-0.48469	0.87249	0.851636	-0.57175	0.100974	-1.51977	-0.28438	-0.31052	-0.40425	-0.82337	-0.29035	0.046949	0.208105	-0.18555	0.001031	0.098816	-0.5529	-0.07329	0.023307	6.14	
26	-0.52991	0.873892	1.347247	0.145457	0.414209	0.100223	0.711206	0.176066	-0.28672	-0.48469	0.87249	0.851636	-0.57175	0.100974	-1.51977	-0.28438	-0.31052	-0.40425	-0.82337	-0.29035	0.046949	0.208105	-0.18555	0.001031	0.098816	-0.5529	-0.07329	0.023307	6.14	
26	-0.53539	0.865268	1.351076	0.147575	0.43368	0.086983	0.693039	0.179742	-0.28564	-0.48247	0.8718	0.853447	-0.57182	0.102252	-1.51999	-0.28591	-0.30963	-0.4039	-0.82374	-0.28326	0.049526	0.206537	-0.18711	0.000753	0.098117	-0.55347	-0.07831	0.025427	1.77	
26	-0.53539	0.865268	1.351076	0.147575	0.43368	0.086983	0.693039	0.179742	-0.28564	-0.48247	0.8718	0.853447	-0.57182	0.102252	-1.51999	-0.28591	-0.30963	-0.4039	-0.82374	-0.28326	0.049526	0.206537	-0.18711	0.000753	0.098117	-0.55347	-0.07831	0.025427	1.77	
27	-0.24605	0.473267	1.695738	0.262411	-0.01087	-0.61084	0.793937	-0.24725	0.138879	-0.40101	-0.81205	-0.18352	-0.63011	-0.28622	-0.33705	-0.42811	-0.0306	-0.50457	0.35518	0.045721	-0.1946	-0.33513	-0.07818	0.392784	-0.03116	0.198037	-0.17535	-0.20362	30.49	
27	-1.45219	1.765124	0.611669	1.176825	-0.44598	0.246826	-0.25757	1.092472	-0.60752	0.047156	0.783727	1.096386	-0.26809	0.768648	-0.52437	-0.80882	0.710386	-0.11837	0.979288	0.008713	0.08228	0.325782	-0.06911	0.020962	-0.04467	-0.24344	0.14918	0.120557	1.8	

Figure 4.1: Sample data set from Kaggle

4.1.1 TRAINING DATA:

Dataset division into training and test sets: When pre-processing machine learning data, we must separate our dataset into training and test sets. This is frequently one of the most important knowledge pre-processing phases since by executing it, we will improve the functionality of our machine learning model. Consider the case when we train our machine learning model on one dataset and then test it on a completely other dataset. The knowledge about the relationships between the models will then be tough for our model to understand. If we successfully train our model and it has excellent training accuracy as well, but we also provide a replacement

dataset, the performance will suffer. Therefore, we always strive to create a machine learning model that does well with both the training set and the test dataset. The training data set is now supplied to machine learning model, based on this data set the model is trained. Every new applicant detail filled at the time of application form acts as a test data set. After the operation of testing, model predict whether the new applicant is a fit case for detecting credit card fraud or not based upon the inference it concludes based on the training data sets. Usually, we split the dataset into train and test in the ratio of 7:3 i.e., 70 percent of data is used for training and 30 percent of data is used for testing the model. We have done it in the same way. Now, we've both the train and test data. The subsequent step is to spot the possible training methods and train our models. As this is often a classification problem, we've used three different classification methods: Naive Bayes, Logistic Regression, Decision Tree , Support Vector Machine and Random Forest. Each algorithm has been run over the Training dataset and their performance in terms of accuracy is evaluated alongside the prediction wiped out the testing data set. We used the publicly available Lending Club dataset from Kaggle and prepare it accordingly to meet our goals. The data covers loans funded by the platform between 2007 and 2015.

```
In [18]: from sklearn.model_selection import train_test_split
         X = df.drop('status',axis=1)
         y = df['status']
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

4.2 CLASSIFIERS

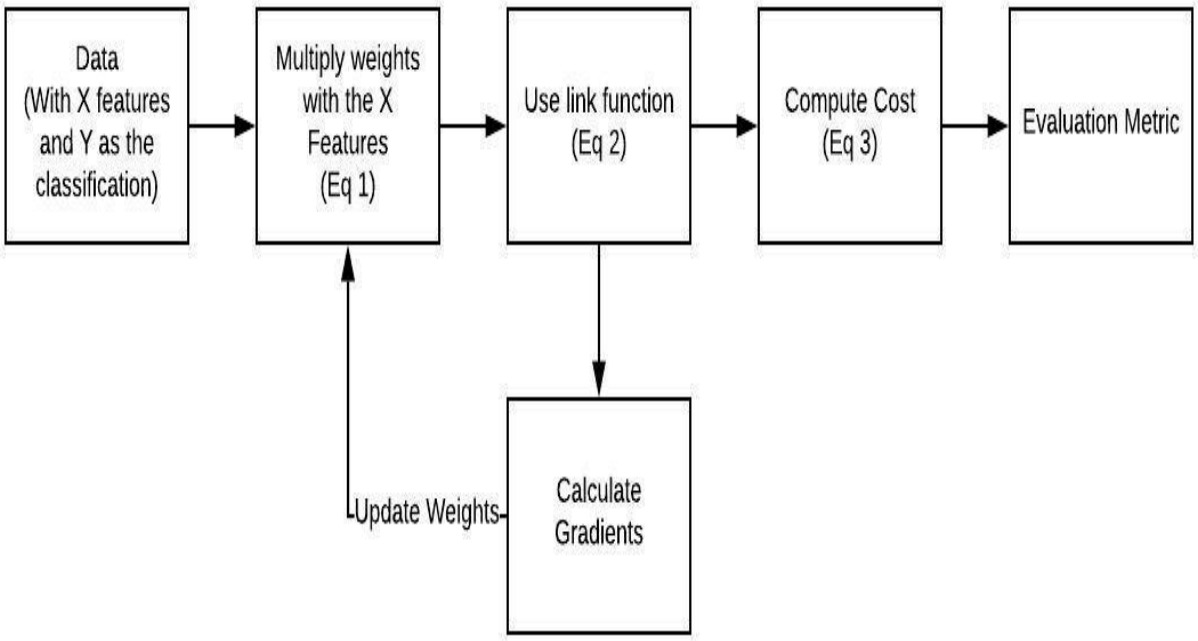
4.2.1 LOGISTIC REGRESSION CLASSIFIER:

Logistic regression is a type of statistical classification model that is used to predict the probability of a categorical dependent variable. The dependent variable in logistic regression is binary, meaning it can take on only two values, 0 and 1. Logistic regression is a supervised learning algorithm for classification problems. It is a linear model that is used to predict the probability of a binary outcome. The outcome is either 0 or 1, which represents the two classes of a binary classification problem. Logistic

regression is used to model the relationship between a dependent variable and one or more independent variables. The independent variables can be categorical or continuous. The dependent variable is always binary. Logistic regression is a special case of a generalized linear model. The logistic regression model is estimated using maximum likelihood estimation. Logistic regression is a widely used machine learning algorithm and is available in most machine learning software packages.

There are a few key points to remember when working with logistic regression in machine learning:

- Logistic regression is a linear model, which means that it will make predictions based on a linear combination of the features.
- The coefficients in a logistic regression model represent the importance of each feature in predicting the outcome.
- The intercept in a logistic regression model represents the value of the predicted outcome when all feature values are 0.
- Logistic regression is a probabilistic model, which means that it will output a probability for each possible outcome.



Algorithm for Logistic Regression

Begin
For i = 1 to k
For each training data instance d_i.
Set the target value for the regression to $z_i = \frac{y_i - P(1|d_j)}{[P(1|d_j)(1-P(1|d_j))]}$
Initialize the weight of instance d_j to $[P(1|d_j)(1 - P(1|d_j))]$
Finalize a $f(j)$ to the data with class value (Z_j) and weight (w_j)
Classical label decision
Assign (class label: 1) if $P_{id} > 0.5$, otherwise (class label: 2)
End

Fig 4.2: Logistic Regression Algorithm

Code snippet for Logistic Regression

```
from sklearn.linear_model import LogisticRegression
lr_model = LogisticRegression()
```

```
# Training the algorithm
lr_model.fit(xtrain, ytrain)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='warn',
          tol=0.0001, verbose=0, warm_start=False)
```

```
# Predictions on training and testing data
lr_pred_train = lr_model.predict(xtrain)
lr_pred_test = lr_model.predict(xtest)
```

4.2.2 DECISION TREE CLASSIFIER:

A decision tree is a type of machine learning algorithm that can categorise and predict data. It functions by making a sequence of judgments that divide the data into two groups. The first group trains the decision tree, while the second group tests it. The decision tree is then used to new data to produce predictions. Decision trees are a form of machine learning method that uses a group of predictor variables to predict the value of a target variable. Decision trees are a non-parametric approach, which means they

make no assumptions about the data's distribution. The goal of utilising a decision tree is to learn from the training data and develop a model that can be used to generate predictions about the target variable. A decision tree's Tree structure may be used to depict the decision process, making it simple to grasp and interpret. Because of their simplicity and interpretability, decision trees are a popular machine learning method. They are, however, known to be prone to overfitting, especially if the tree is allowed to grow too deep. A decision tree is a type of machine learning algorithm that can do both regression and classification tasks. The algorithm divides the data into smaller groups based on a set of criteria. The algorithm then generates a forecast for each group based on the data in that group. The forecasts are then pooled to get an overall prediction for the dataset.

- Decision trees are a type of supervised learning algorithm used for classification and regression tasks.
- Decision trees learn from data to create a model of the target variable.
- Decision trees are used to make predictions about the target variable, by using the model created from the data.
- Decision trees can be used for both categorical and numerical data.
- Decision trees are easy to interpret and explain, and they can be used to create decision rules that can be applied to new data.

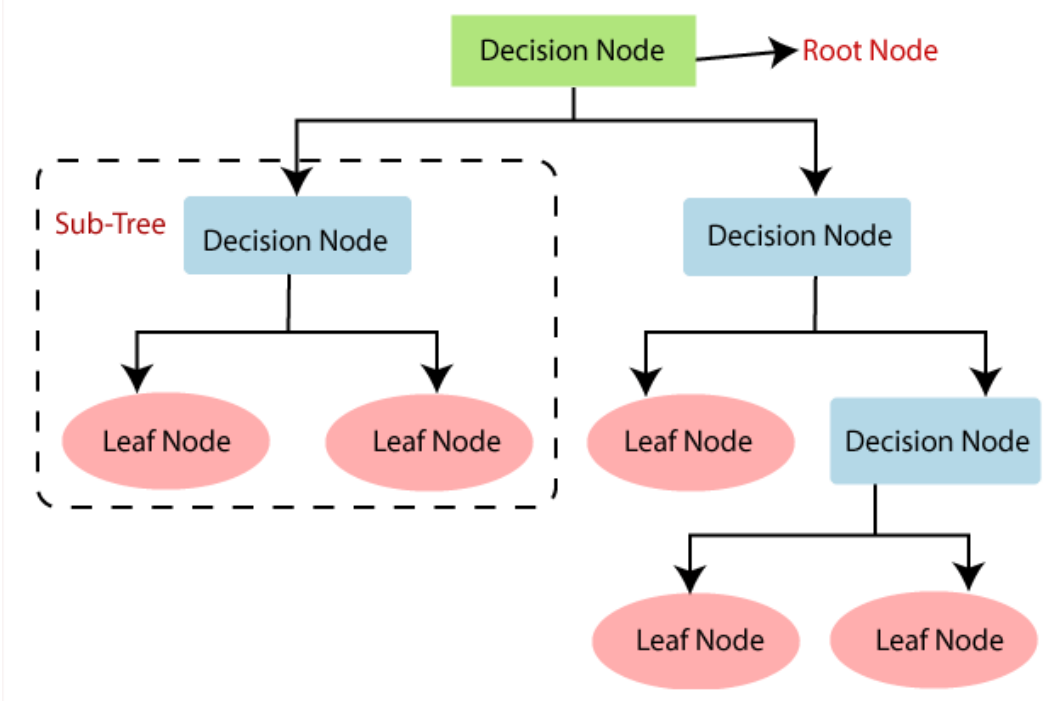


Figure 4.3: Decision Tree model

Algorithm for Decision Tree

GenDecTree(Sample S, Features F)

Steps:

If** stopping_condition(S, F) = true **then

a. *Leaf = createNode()*

b. *leafLabel = classify(s)*

c. ***return** leaf*

root = createNode()

root.test_condition =findBestSpilt(S,F)

V = {v | v a possible outcomecfroot.test_condition}

***For each** value v ∈ V:*

a. S_v = {s | root.test_condition(s) = v and s ∈ S };

b. *Child = TreeGrowth* (S_v, F);

c. *Add child as descent of root and label the edge {root → child} as v*

***return** root*

Figure 4.4: Decision tree algorithm

Code snippet for Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
decision_tree = DecisionTreeClassifier()
```

```
decision_tree.fit(X_train,y_train.values.ravel())
```

```
DecisionTreeClassifier()
```

```
y_pred = decision_tree.predict(X_test)
```

```
decision_tree.score(X_test,y_test)
```

4.2.3 NAIVE BAYES CLASSIFIER:

Naive Bayes is a straightforward machine learning technique used for categorization tasks. It is a probabilistic algorithm that generates predictions based on previous data. The algorithm is called for the fact that it simplifies data assumptions. Even though the naïve assumption is frequently incorrect, the method performs well in practice. In fact, it is frequently employed as a baseline algorithm against which more complicated algorithms are compared. A probabilistic algorithm, the Naive Bayes method in other words, it makes predictions based on the data's probabilities. The algorithm computes each class's probability before predicting the class with the highest probability. The Naive Bayes method is simple to use and efficient in terms of computing. It also offers certain advantages over other machine learning algorithms, such as being resistant to noisy data and capable of handling missing data. Despite its benefits, the Naive Bayes algorithm has a few drawbacks. One of the most significant drawbacks is that the algorithm assumes that all the characteristics in the data are independent of one another. This assumption is frequently inaccurate in practice, which can lead to the algorithm making erroneous decisions.

- A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions between the features.
- Naive Bayes classifiers are extremely fast compared to more sophisticated methods and are often used in very large-scale applications such as spam filtering and document classification.
- Despite their naive design and apparently over-simplified assumptions, naive Bayes classifiers have been shown to work well in many complex real-world situations.

Algorithm for Naïve Bayes

1. for q = 1… w // loop for each mining models element
2. μ[q] = 0; // initialization of mining models elements
3. end for;
4. for j = 1… m // loop for each row
5. μ[d[j,p]]++; // increment number of row for value $x_{j.p}$ of object x_j;
6. for k = 1 …p-1 // loop for each column
7. $\mu[\varphi(k-1)+(d[j, k]-1)\cdot\varphi(0)+ d[j, p]]$++; // increment number of rows with value $x_{j.k}$ // and value $x_{j.p}$, where $\varphi(k)=s+\sum_{q=1}^{k}(|T_q|\cdot s)$
8. end for;
9. end for;

Figure 4.5: Naïve Bayes algorithm

Code snippet for Naïve Bayes

```
from sklearn.naive_bayes import GaussianNB
```

```
nb = GaussianNB()
```

Let's first train the algorithm on the default settings.

```
nb.fit(xtrain, ytrain)
```

```
GaussianNB(priors=None, var_smoothing=1e-09)
```

```
nb_pred = nb.predict(xtest)
nb_pred_proba = nb.predict_proba(xtest)[:, 1]
```

4.2.4 SUPPORT VECTOR MACHINE CLASSIFIER:

A support vector machine (SVM) is a supervised machine learning technique that may be used for classification as well as regression. The primary goal of an SVM is to create a hyperplane that best divides a given dataset into two groups. The SVM algorithm initially converts the data into a higher-dimensional space using a kernel function in order to locate the hyperplane that maximizes the margin between the two classes. The SVM algorithm can then locate the hyperplane that maximizes the margin once the data is in this higher-dimensional space. An SVM has the benefit of frequently finding a hyperplane that provides acceptable generalization performance even when the data is not linearly separable in the original space. This is because the SVM method may use a kernel function to determine a non-linear decision boundary. There are several kinds of kernel functions that may be employed with an SVM. The linear, polynomial, and radial basis function (RBF) kernels are the most frequent. When the data is linearly separable, the linear kernel is the simplest and is utilized. When the data is not linearly separable and there are many data points, the polynomial kernel can be employed, and the RBF kernel is frequently utilized.

- A function is a set of ordered pairs (x, y) such that each x corresponds to a unique y.
- A function can be represented using a graph on a coordinate plane. The domain of a function is the set of all x-values for which the function produces a valid y-value.

- The range of a function is the set of all y-values that the function produces. A function is continuous if given any two points within the function's domain, there exists a smooth curve that connects those points.
- A function is discontinuous if there is a point within the domain at which the function produces two different y-values.
- A function is linear if the graph of the function is a straight line.
- A function is nonlinear if the graph of the function is not a straight line.
- The degree of a polynomial function is the highest exponent of the variable in the function. For example, the function f(x) = x2 + 3x + 5 is a polynomial function of degree 2.
- The order of a differential equation is the highest derivative that appears in the equation. For example, the equation y" + y = 0 is a second-order differential equation.

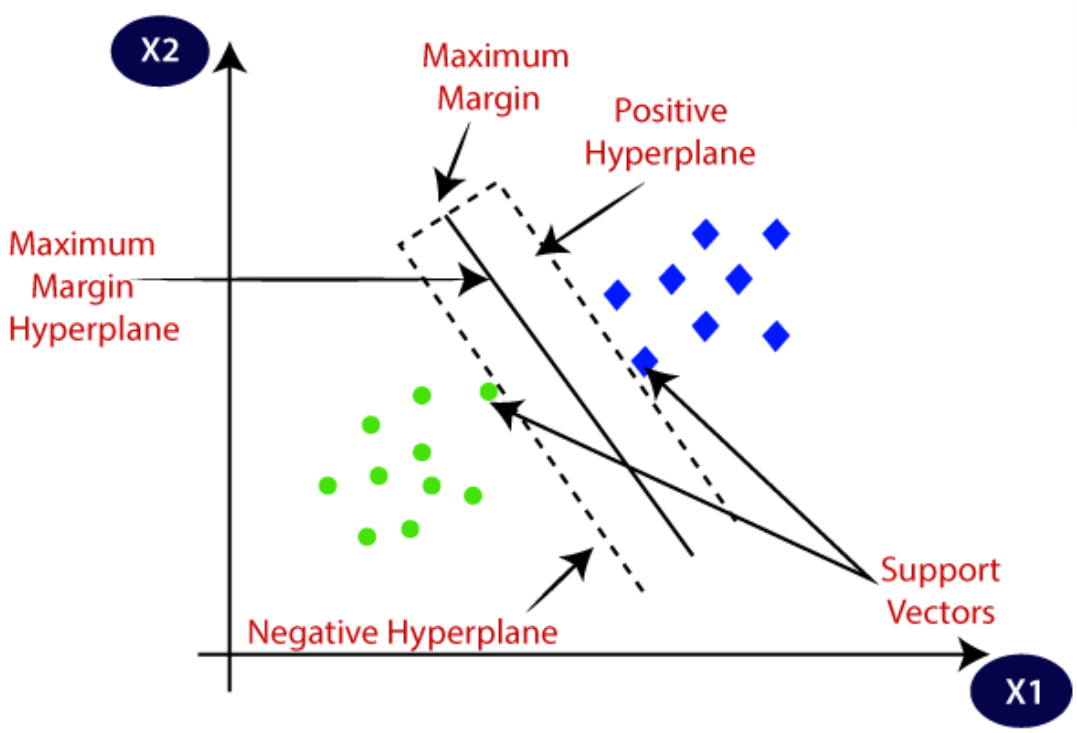


Figure 4.6: Support Vector Machine model

Code snippet for Support Vector Machine

```
from sklearn.svm import SVC
```

```
svc_model = SVC(kernel='linear', probability=True)
```

```
svc_model.fit(xtrainS, ytrainS)
```

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='linear', max_iter=-1, probability=True, random_state=None,
  shrinking=True, tol=0.001, verbose=False)
```

```
svc_pred = svc_model.predict(xtestS)
```

4.2.5 RANDOM FOREST CLASSIFIER:

A random forest is a classification and regression machine learning technique. The random forest technique is an ensemble learning approach that builds many decision trees and then averages the results to generate a final forecast. The random forest technique is a versatile tool that may be used for classification as well as regression. The algorithm is a basic and straightforward machine learning approach that produces accurate results. Random forests are supervised machine learning algorithms that may be used for classification as well as regression. At training time, the method constructs many decision trees and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random forests are an example of an ensemble learning approach, which combines the results of numerous learning algorithms to produce a more accurate final model.

The individual decision trees required to build the random forest are developed using a process known as bootstrap aggregation, often known as bagging. Bagging is a way of averaging the outcomes of numerous models to produce a more robust and less prone to overfitting final model. Random forests offer several advantages and disadvantages over other machine learning algorithms: They are generally simple to train and adjust and may be utilized for both classification and regression problems. They are also less prone than other approaches, such as decision trees, to overfit the data. Random forests may be used to evaluate the significance of each feature in the data, which can then be used to choose a subset of features for inclusion in a final model. Because it is built on

a mixture of many decision trees, the model might be difficult to grasp. It is possible that the model will be slower to train and forecast than alternative approaches, such as linear models. Random forests may not be suitable for data sets with a high number of features or features with several levels.

Functions of random forest classifier are:

- A random forest is a machine learning algorithm that is used for classification and regression.
- It is an ensemble learning method that is used to create a forest of random Decision Trees.
- The random forest algorithm is a supervised learning algorithm, non-parametric algorithm, an efficient and a robust algorithm.

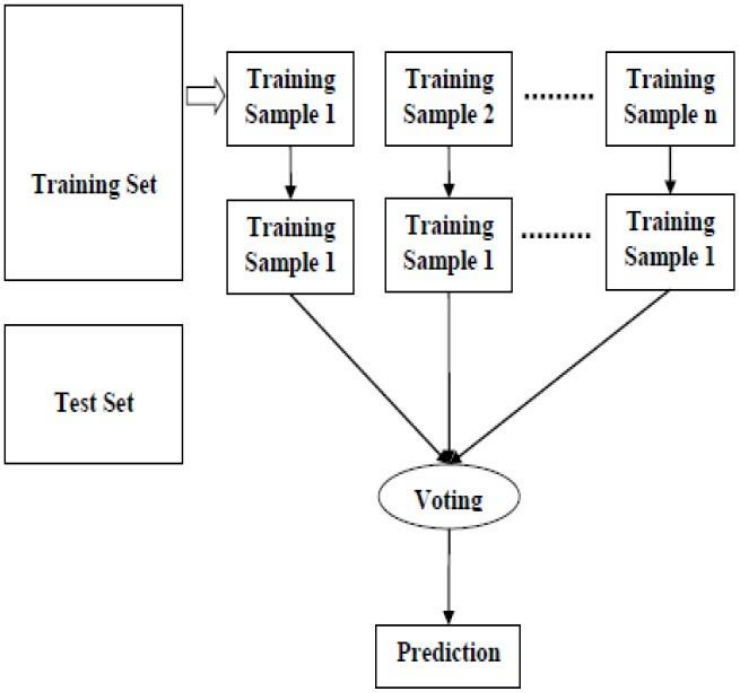


Figure 4.7: Random Forest model

- **Random Forest Algorithm**

Algorithm Random Forest:
To generate c classifiers:
For i=1 to c do
Randomly select the training data D with
replacement to produce Di
Create a root node N containing Di and cell
Build Tree(N)
End for
Majority Vote

Build Tree(N)
Randomly select x% of all the possible splitting features in N
Select the features F that has the highest Information
A gain for further splitting
Gain (T,X)=Entropy (T)-Entropy(T,X)
Now to calculate the entropy we use,
$E(S) = \Sigma^{c}_{i=1} (-Pi \log Pi)$
Create f child nodes
For i=1 to f do
Set contents f N to Di
Call Build Tree(Ni)
End for
End

Code snippet for Random Forest

```
from sklearn.ensemble import RandomForestClassifier
```

```
random_forest = RandomForestClassifier(n_estimators=100)
```

```
# Pandas Series.ravel() function returns the flattened underlying data as an ndarray.
random_forest.fit(X_train,y_train.values.ravel())    # np.ravel() Return a contiguous flattened array
```

```
RandomForestClassifier()
```

```
y_pred = random_forest.predict(X_test)
```

```
random_forest.score(X_test,y_test)
```

CHAPTER 5

RESULTS AND ANALYSIS

Performance evaluations measures are the factors that aid in the comparative study of various machine learning approaches, i.e., they reveal which algorithm is better among all others that can be used to forecast the detection of credit card fraud. To assess the accuracy of the predictions, we have employed a variety of metrics. In results and analysis, accuracy is the main factor we focus on, but there are also additional factors including precision, recall, and f1-score.

5.1 CONFUSION MATRIX

The error matrix is another name for the confusion matrix. It is a table that is frequently used to summarize how well a classification algorithm performs on a set of test data when the real values are known. Instances belonging to a predicted class are represented in each column of the matrix. The correlation matrix is shown as follows:

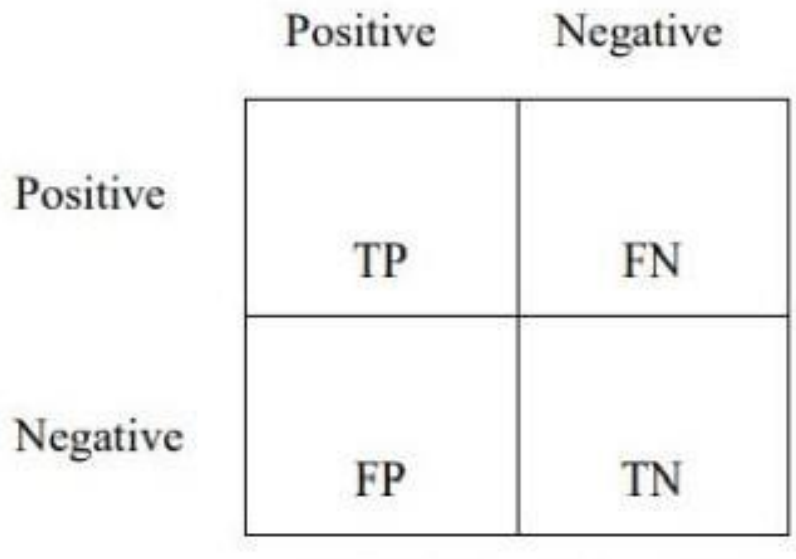


Figure 5.1: Confusion Matrix model

Where TP: True positive
FP: False Positive
FN: False Negative
TN: True Negative

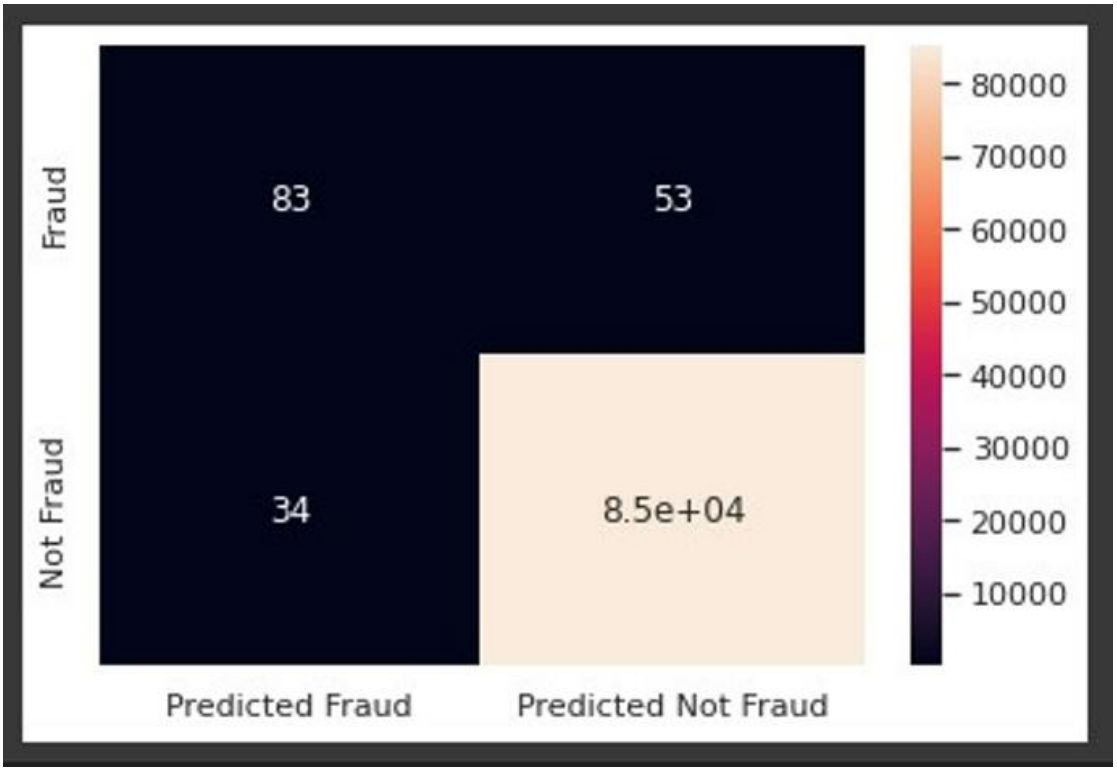


Figure 5.2: Confusion Matrix of Fraud and Not-Fraud

- **Accuracy:**

Accuracy is the proportion of the total number of predictions that were correct. It can be obtained by the sum of true positive and true negative instances divided by the total number of Samples. It is expressed as: Accuracy =(TP+TN)/(TP+FP+FN+TN)

- **Precision:**

Precision is fraction of true positive and predicted yes instances. It is also known as the ratio of correct positive results to the total number of positive results predicted by the system. It is expressed as: Precision(P) =TP/ (TP + FP).

- **Recall:**

Recall is defined as the fraction between True Positive instances and Actual yes instances, or it is the ratio of correct positive results to the number of all relevant samples. It is expressed as: Recall(R) =TP/ (TP +FN).

- **F1-Score:**

F1-score is the fraction between product of the recall and precision to the summation of recall and precision parameter of classification. It is the harmonic mean of Precision and Recall. It measures the test accuracy. The range of this metric is 0 to F1. It is expressed as: F1 score = 2 ∗1/((1/Precision) +(1/Recall))=2PR/(P+R).

- **Comparative Analysis:**

The model that best fits our system was discovered by comparing all the accuracy, precision, recall, and F1-score graphs and tables. This graph depicts a comparison of the three models based on some of the assessment metrics listed above.

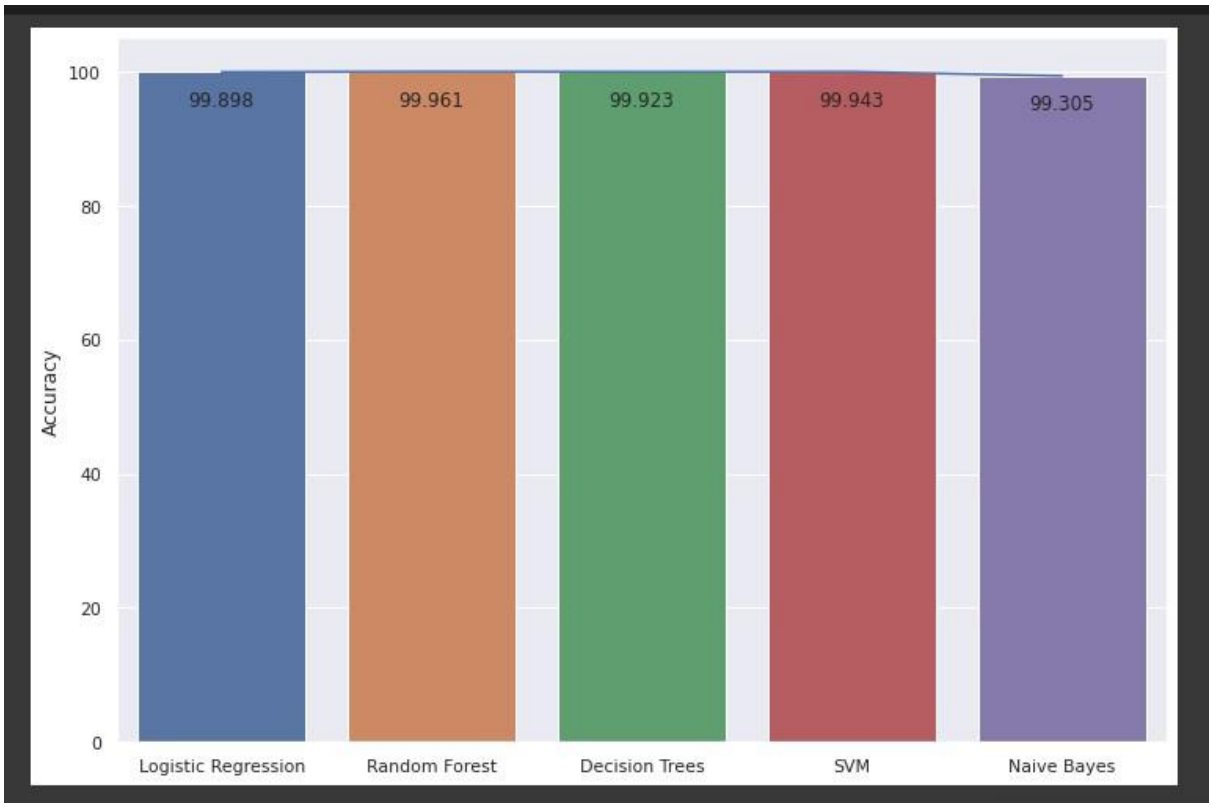


Figure 5.3: Comparative Analysis of five classifiers

In Fig 5.3, we have observed that results of all five techniques after applying LR, SVM, Decision Tress, Naïve Bayes and RF , it shows that all the accuracies of all the classifiers is 99% so we are not able to pick a accurate classifier. Furthermore, in the Fig 5.4 we are

comparing all the other variables such as precision, recall and F1-score. To decide which classifier suits the project.

```
model_results = pd.DataFrame([['Logistic Regression', acc_lr, rec_lr, prec_lr, f1_lr],
                              ['Random Forest', acc_rf, rec_rf, prec_rf, f1_rf],
                              ['Decision Trees', acc_dt, rec_dt, prec_dt, f1_dt],
                              ['Support Vector Machine', acc_svc, rec_svc, prec_svc, f1_svc],
                              ['Naive Bayes', acc_nb, rec_nb, prec_nb, f1_nb]],
                columns = ['Model', 'Accuracy', 'Recall', 'Precision', 'F1 Score'])
results_fullset = results_fullset.append(model_results, ignore_index = True)
results_fullset
```

	Model	Accuracy	Recall	Precision	F1 Score
0	random forest	0.999614	0.801471	0.947826	0.868526
1	Logistic Regression	0.998982	0.610294	0.709402	0.656126
2	Random Forest	0.999614	0.801471	0.947826	0.868526
3	Decision Trees	0.999228	0.816176	0.730263	0.770833
4	Support Vector Machine	0.999427	0.801471	0.832061	0.816479
5	Naive Bayes	0.993048	0.661765	0.141066	0.232558

Figure 5.4: Results Obtained from five models

5.2 RESULTS

To demonstrate the results of our project, we take the remaining test data, and it is tested using five algorithms. After that our trained model to ready to predict the fraud is present or not. The test accuracy is done in the colab which is our python notebook. Below we described how the five algorithms are processed. First, LR algorithm is trained with the training dataset and later it was tested with the remaining test data. In Fig 5.5, a screenshot of our notebook is showing that how the process of RF algorithms is done and the accuracy the model returns and it is of 99.89%, recall is 61%, precision is 70% and at last F1-score is 65%.

```
from sklearn.metrics import roc_curve, roc_auc_score
fpr, tpr, threshold = roc_curve(ytest, lr_pred_test_prob)
acc = roc_auc_score(ytest, lr_pred_test_prob)
acc
acc_lr = accuracy_score(ytest, lr_pred_test)
prec_lr = precision_score(ytest, lr_pred_test)
rec_lr = recall_score(ytest, lr_pred_test)
f1_lr = f1_score(ytest, lr_pred_test)
results_fullset = pd.DataFrame([['Logistic Regression', acc_lr, rec_lr, prec_lr, f1_lr]],
            columns = ['Model', 'Accuracy', 'Recall', 'Precision', 'F1 Score'])
results_fullset
def plot_roc_curve(fpr, tpr, label=None):
  plt.figure(figsize=(8, 6))
  plt.title('ROC Curve', fontsize=15)
  plt.plot([0, 1], [0, 1], 'k--')
  plt.plot(fpr, tpr, linewidth=2, label=label)
  plt.xticks(np.arange(0, 1, 0.05), rotation=90)
  plt.xlabel('False Positive Rates', fontsize=15)
  plt.ylabel('True Positive Rates', fontsize=15)
  plt.legend(loc='best')

  plt.show()
  plot_roc_curve(fpr=fpr, tpr=tpr, label="AUC = %.3f" % acc_lr)
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	85307
1	0.71	0.61	0.66	136
accuracy			1.00	85443
macro avg	0.85	0.80	0.83	85443
weighted avg	1.00	1.00	1.00	85443

```
results_fullset = pd.DataFrame([['Logistic Regression', acc_lr, rec_lr, prec_lr, f1_lr]],
            columns = ['Model', 'Accuracy', 'Recall', 'Precision', 'F1 Score'])
results_fullset
```

	Model	Accuracy	Recall	Precision	F1 Score
0	Logistic Regression	0.998982	0.610294	0.709402	0.656126

Figure 5.5: Results of LR

Secondly, Decision Tree algorithm is trained with the training dataset and later it was tested with the remaining test data. In Fig 5.6, a screenshot of our notebook is showing that how the process of Decision Tree algorithms is done and the accuracy the model returns and it is of 99.92%, precision is 73%, recall is 81% and F1-score is 77%.

```
[[85266    41]
 [   25   111]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85307
           1       0.73      0.82      0.77       136

    accuracy                           1.00     85443
   macro avg       0.86      0.91      0.89     85443
weighted avg       1.00      1.00      1.00     85443
```

	Model	Accuracy	Recall	Precision	F1 Score
0	Decision Trees	0.999228	0.816176	0.730263	0.770833

Figure 5.6: Results of Decision Tree

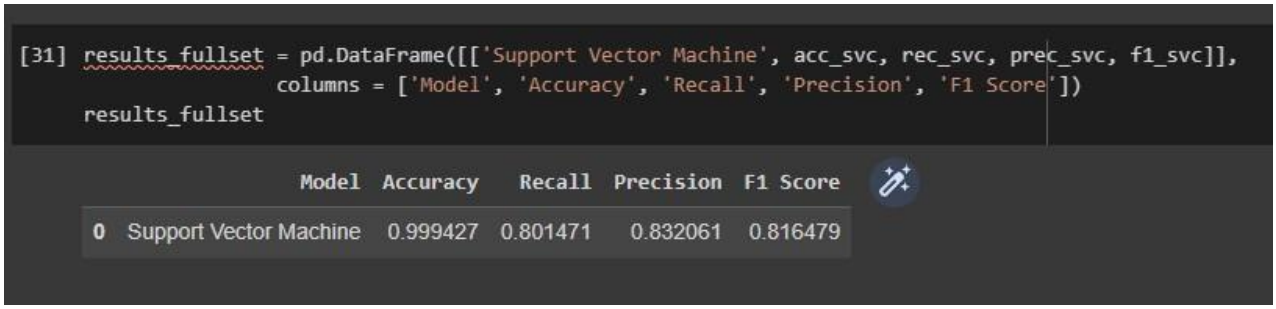
Thirdly, Random Forest algorithm is trained with the training dataset and later it was tested with the remaining test data. In Fig 5.7, a screenshot of our notebook is showing that how the process of Random Forest algorithms is done and the accuracy the model returns, and it is of 99.96%, recall is 80%, precision is 94.78% and F1-score is 86%.

```
[[85301     6]
 [   27   109]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85307
           1       0.95      0.80      0.87       136

    accuracy                           1.00     85443
   macro avg       0.97      0.90      0.93     85443
weighted avg       1.00      1.00      1.00     85443
```

	Model	Accuracy	Recall	Precision	F1 Score
0	random forest	0.999614	0.801471	0.947826	0.868526

Figure 5.7: Results of RF

Fourthly, Support Vector Machine algorithm is trained with the training dataset and later it was tested with the remaining test data. In Fig 5.8, a screenshot of our notebook is showing that how the process of Support Vector Machine algorithms is done and the accuracy the model returns, and it is of 99.94%, recall is 80%, precision is 83%, F1-score is 81%.

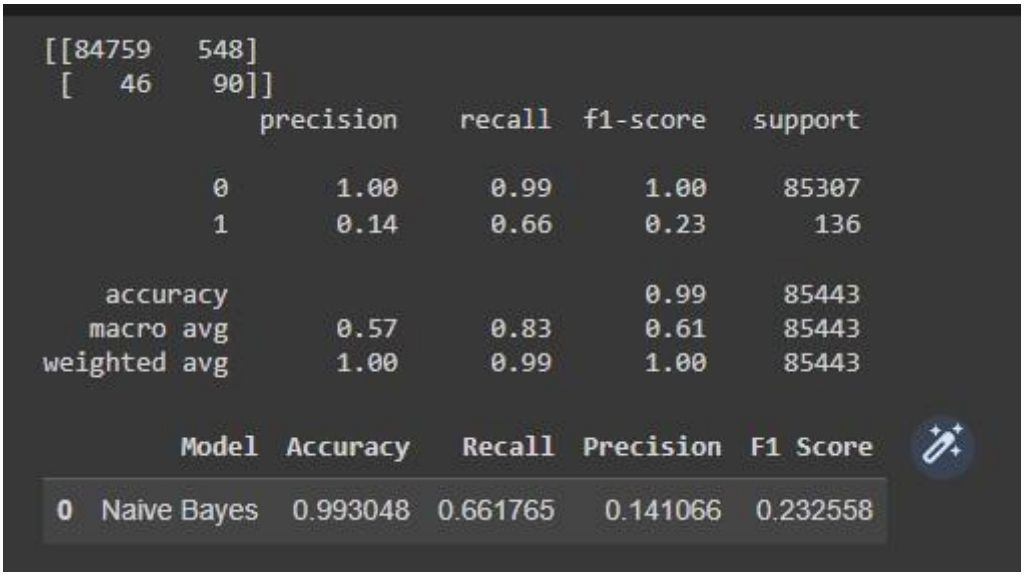


```
[31] results_fullset = pd.DataFrame([['Support Vector Machine', acc_svc, rec_svc, prec_svc, f1_svc]],
                                    columns = ['Model', 'Accuracy', 'Recall', 'Precision', 'F1 Score'])
     results_fullset
```

	Model	Accuracy	Recall	Precision	F1 Score
0	Support Vector Machine	0.999427	0.801471	0.832061	0.816479

Figure 5.8: Results of SVM

Fifthly, Naïve Bayes algorithm is trained with the training dataset and later it was tested with the remaining test data. In Fig 5.9, a screenshot of our notebook is showing that how the process of Support Vector Machine algorithms is done and the accuracy the model returns, and it is of 99.30%, recall is 66%, precision is 14%, F1-score is 23.25%.



```
[[84759   548]
 [   46    90]]
              precision    recall  f1-score   support

           0       1.00      0.99      1.00     85307
           1       0.14      0.66      0.23       136

    accuracy                           0.99     85443
   macro avg       0.57      0.83      0.61     85443
weighted avg       1.00      0.99      1.00     85443
```

	Model	Accuracy	Recall	Precision	F1 Score
0	Naive Bayes	0.993048	0.661765	0.141066	0.232558

Figure 5.9: Results of Naïve Bayes

Table 3: Model Evaluation Parameters

Model	Accuracy	Recall	Precision	F1-Score
Naïve Bayes	0.993048	0.661765	0.141066	0.232558
Decision Tree	0.999064	0.727941	0.697183	0.712230
Logistic Regression	0.998982	0.610294	0.709402	0.656126
Support Vector Machine	0.999427	0.801471	0.832061	0.816479
Random Forest	0.999614	0.808824	0.940171	0.869565

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

- Algorithms operate in accordance with balanced datasets, which offer the best improvements in efficiency**.**
- The data was pre-processed before being utilized in the model.
- In our project, we employed five different types of algorithms, and the results from the random forest when combined with F1-scores are superior.
- Although Random Forest and Ensemble models use many algorithms (in the case of Random Forest, multiple Decision Trees) to address the issue of overfitting, they have done quite well. Our objective is to make predictions using fewer characteristics and tests.

6.2 FUTURE ENHANCEMENTS

- We'd want to include a function that detects live-based fraud detection.
- Enhancing the data sets with information from real-time data to improve our model's real-time capability.
- Provide a UI to the model to deal with live datasets.

REFERENCES

[1] S. Bhattacharyya, S. Jha, K. Tharakunnel,and J. C. Westland, “Data mining for credit card fraud: A comparative study,” *Decis. Support Syst.*, vol. 50, no. 3, pp. 602–613, 2011.

[2] K. Chaudhary, J. Yadav, and B. Mallick, “A review of Fraud Detection Techniques: Credit Card,” *Int. J. Comput. Appl.*, vol. 45, no. 1, pp. 975–8887, 2012.

[3] F. N. Ogwueleka, “Data Mining Application in Credit Card Fraud Detection System,” vol. 6, no. 3, pp. 311–322, 2011.

[4] H. Nordberg, K. Bhatia, K. Wang, and Z. Wang, “BioPig: a Hadoop-based analytic toolkit for large-scale sequence data,” *Bioinformatics*, vol. 29, no. 23, pp. 3014– 3019, Dec. 2013.

[5] M. Hegazy, A. Madian, and M. Ragaie, “Enhanced Fraud Miner: Credit Card Fraud Detection using Clustering Data Mining Techniques,” *Egypt. Comput. Sci.*, no. 03, pp. 72–81, 2016.

[6] M. Zareapoor and P. Shamsolmoali, “Application of credit card fraud detection: Based on bagging ensemble classifier,” *Procedia Comput. Sci.*, vol. 48, no. C, pp. 679–686, 2015.

[7] O. S. Yee, S. Sagadevan, N. Hashimah, and A. Hassain, “Credit Card Fraud Detection Using Machine Learning As Data Mining Technique,” vol. 10, no. 1, pp. 23–27.

[8] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, “Credit Card Fraud Detection Using AdaBoost and Majority Voting,” *IEEE Access*, vol. 6, pp. 14277– 14284, 2018.

[9] Bahnsen, A. C., Stojanovic, A., Aouada, D., & Ottersten, B. (2014). Improving credit card fraud detection with calibrated probabilities. In *Proceedings of the 2014 SIAM International Conference on Data Mining* (pp. 677-685). Society for Industrial and Applied Mathematics.

[10] Ng, A. Y., and Jordan, M. I., (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, *2*, 841-848.

[11] Maes, S., Tuyls, K., Vanschoenwinkel, B., & Manderick, B. (2002). Credit card fraud detection using Bayesian and neural networks. In *Proceedings of the 1st international naiso congress on neuro fuzzy technologies* (pp. 261-270).

[12] Shen, A., Tong, R., & Deng, Y. (2007). Application of classification models on credit card fraud detection. In *Service Systems and Service Management, 2007 International Conference on* (pp. 1-4). IEEE.

[13] Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*, *50*(3), 602-613.

[14] Sahin, Y. and Duman, E., (2011). Detecting credit card fraud by ANN and logistic regression. In *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on* (pp. 315-319). IEEE.

[15] Chaudhary, K. and Mallick, B., (2012). Credit Card Fraud: The study of its impact and detection techniques, *International Journal of Computer Science and Network (IJCSN)*, Volume 1, Issue 4, pp. 31 – 35, ISSN: 2277-5420.

[16] Bhatla, T.P.; Prabhu, V.; and Dua, A. (2003). *Understanding credit card frauds*. Crads Business Review#

2003-1, Tata Consultancy Services.

[17] The Nilson Report. (2015). U.S. Credit & Debit Cards 2015. David Robertson.

[18] Stolfo, S., Fan, D. W., Lee, W., Prodromidis, A., & Chan, P. (1997). Credit card fraud detection using meta-learning: Issues and initial results. In *AAAI-97 Workshop on Fraud Detection and Risk Management*.

APPENDIX A – ABBREVIATIONS

Abbreviations	Full Form
ML	Machine Learning
AI	Artificial Intelligence
SVM	Support Vector Machine
LR	Logistic Regression
FPR	False Positive Rate
TPR	True Positive Rate
RF	Random Forest

APPENDIX B – PROCEDURE TO USE GOOGLE COLAB

1. Creating your first. ipynb notebook in colab

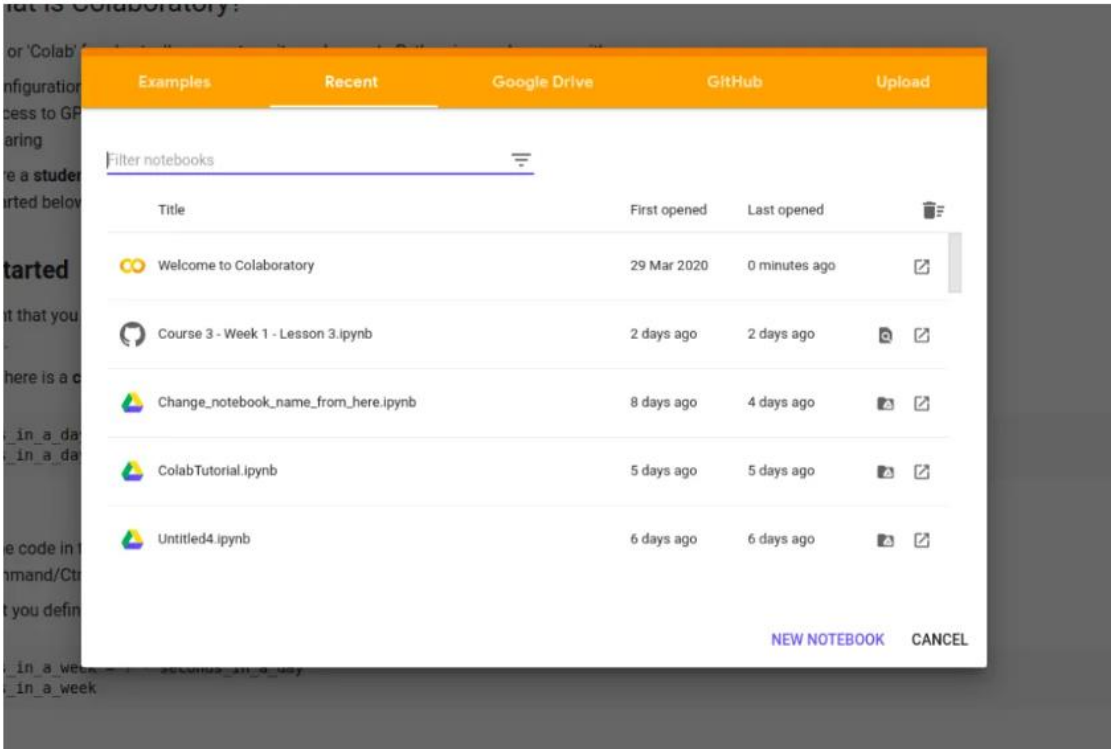


Figure B.1: Creating a notebook

2. Installing packages in colab

Fig B.2: Installing packages

3. Initiating a run time with GPU/TPU enabled

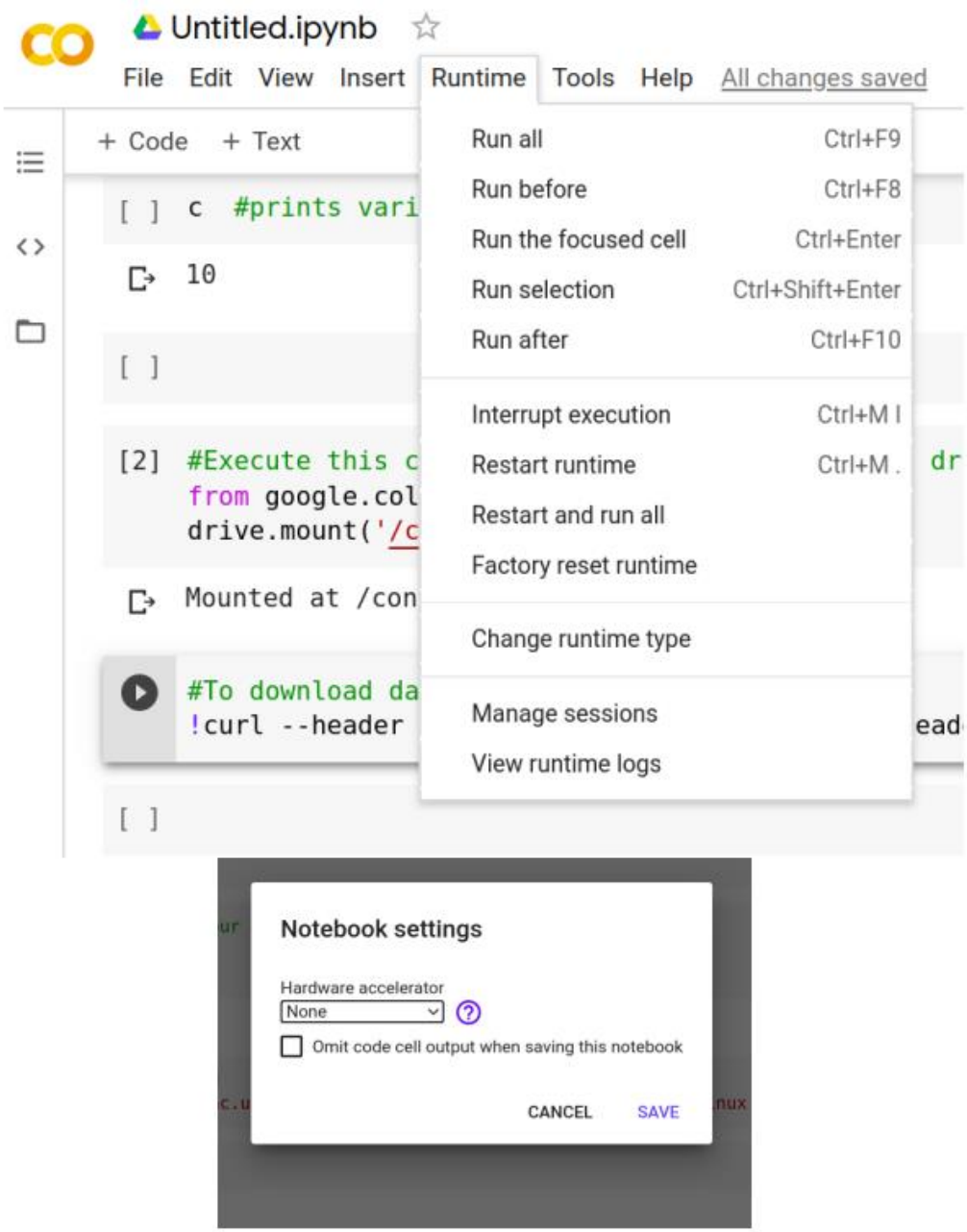


Figure B.3: Run time initialization

4. Mounting a drive

- **Using GUI**
- **Using Code**

➢ Using GUI

Figure B.4: Using GUI

➢ Using Code

```
from google.colab import drive
drive.mount('/content/drive')
```

Figure B.5: Using code snippet