CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING

A Main Project Report submitted in the partial fulfillment of

the requirements for the award of the degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

M.KoteswaraRao (19471A05F7)
J.Suryanarayana (19471A05E9)
M.V.Aditya Kumar. (19471A05G8)

Under the esteemed guidance of

Y. Chandana,M.Tech.
Asst.Prof

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE

AUTONOMOUS

Accredited by NAAC with A+ Grade and NBA under (Tier-1)

NIRF rank in the band of 251-320 and an ISO 9001:2015 Certified

Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada

KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET-522601

2022-2023

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPETA (AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the main project entitled **"CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING"** is a bonafide work done by "M.KoteswaraRao(19471A05F7), J.Suryanarayana(19471A05E9), M.V.Aditya Kumar (19471A05G8)" in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in the department of **COMPUTER SCIENCE AND ENGINEERING during 2022-2023.**

PROJECT GUIDE

Y.Chandana,M.Tech.

PROJECT CO-ORDINATOR

Dr. M.Sireesha ,MTech.,Ph.D

HEAD OF THE DEPARTMENT

Dr.S.N.TirumalaRao,M.Tech.,Ph.D

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We wish to express our thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman sir **M.V. Koteswara Rao**, **B.sc** who took keen interest in us in every effort throughout this course. We owe out gratitude to our principal **Dr.M.Sreenivasa Kumar,** **M.Tech,Ph.D(UK),MISTE,FIE(1)** for his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude to **Dr.S.N.Tirumala Rao,M.Tech, Ph.D** H.O.D. CSE department and our guide **Y.Chandana,M.Tech.** of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks to **Dr.M.Sireesha,M.Tech,PhD.** professor for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B.Tech degree. We have no words to acknowledge the warm affection, constant inspiration and encouragement that we receive from our parents. We affectionately acknowledge the encouragement received from our friends those who involved in giving valuable suggestions had clarifying out all doubts which had really helped us in successfully completing our project.

By

M.KoteswaraRao (19471A05F7)

J.Suryanarayana (19471A05E9)

M.V.AdityaKumar(19471A05G8)

ABSTRACT

Nowadays credit card became one of the essential parts of the people. Sudden increase in E-commerce, customer started using credit card for online purchasing therefore risk of fraud also increases. Instead of carrying a huge amount in hand it is easier to keep credit cards. But nowadays that too becomes unsafe. Now a days we are facing a big problem on credit card fraud which is increasing in a good percentage. The main purpose is the survey on the various methods applied to detect credit card frauds. From the abnormalities, in the transaction, the fraudulent one is identified. We address this issue in order to implement some machine learning algorithm like Isolation Random Forest Algorithm in order to detect this kind of fraud. In this paper we increase the efficiency in finding the fraud. However, we discussed and evaluated employee criteria. Currently, the issues of credit card fraud detection have become a big problem for new researchers. We implement an intelligent algorithm which will detect all kind of fraud in a credit card transaction. We handled the problem by finding a pattern of each customer in between fraud and legal transaction. Various Machine Learning Algorithms are used to predict the pattern of transaction for each customer and a decision is made according to them. In order to prevent data from mismatching, all attribute are marked equally.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community,

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching,imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a centre of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Outcomes

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, research literature, and analyzecomplex engineering problems reaching substantiated conclusions using firstprinciples of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complexengineering problems and design system components or processes that meetthe specified needs with appropriate consideration for the public health andsafety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-basedknowledge and research methods including design of experiments, analysis andinterpretation of data, and synthesis of the information to provide validconclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques,resources, and modern engineering and IT tools including prediction andmodeling to complex engineering activities with an understanding of thelimitations.

6. The engineer and society: Apply reasoning informed by the contextualknowledge to assess societal, health, safety, legal and cultural issues and theconsequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of theprofessional engineering solutions in societal and environmental contexts,and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Project Course Outcomes (CO'S):

CO425.1: Analyse the System of Examinations and identify the problem.

CO425.2: Identify and classify the requirements.

CO425.3: Review the Related Literature

CO425.4: Design and Modularize the project

CO425.5: Construct, Integrate, Test and Implement the Project.

CO425.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C425.1		✓											✓		
C425.2	✓		✓		✓								✓		
C425.3				✓		✓	✓	✓					✓		
C425.4			✓			✓	✓	✓					✓	✓	
C425.5					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C425.6									✓	✓	✓		✓	✓	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C425.1	2	3											2		
C425.2			2		3								2		
C425.3				2		2	3	3					2		
C425.4			2			1	1	2					3	2	
C425.5					3	3	3	2	3	2	2	1	3	2	1
C425.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's:

1. **Low level**

2. **Medium level**

3. **High level**

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	Description of the device	Attained PO
C3.2.4, C3.2.5	Gathering the requirements and defining the problem, plan to develop a **credit card fraud detection using machine learning.**	PO1, PO3
CC4.2.5	Each and every requirement is critically analyzed, the process model is identified and divided into **five modules**	PO2, PO3
CC4.2.5	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC4.2.5	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC4.2.5	Documentation is done by all our four members in the form of a group	PO10
CC4.2.5	Each and every phase of the work in group is presented periodically	PO10, PO11
CC4.2.5	Implementation is done and the project will be handled by the **credit card fraud detection using machine learning.**	PO4, PO7
CC4.2.8 CC4.2.	**The physical design includes software components like Jupyter, Anaconda, Tensorsflow, PyTorch.**	PO5, PO6

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Credit card fraud is a widespread problem in the financial industry, and to prevent it, banks and credit card companies use machine learning algorithms to identify and stop fraudulent transactions. This document discusses the use of machine learning algorithms in credit card fraud detection, which has become increasingly important as more payments are made online with credit cards. The goal is to create an efficient model to reduce losses from fraudulent activity, which can take many forms, such as false tax returns or fraudulent loans. The project proposes machine learning algorithms to detect suspicious credit card transactions and involves experimenting with different classifiers and features, such as True Positive and False Alarm.

1.2 OBJECTIVES

The objective of any business is to make a profit, which is calculated by subtracting the cost of doing business from the total sell price. However, online payments have increased the risk of fraud for both vendors and financing companies. Fraudulent activities pose a financial risk to both financial companies and cardholder's banks. Fraud detection techniques are necessary to overcome these risks. The main objective is to prevent customers from becoming victims of fraud because frequent incidents can discourage people from using credit cards and other financial services. Hence, it has become essential to prevent fraudulent activities. People should also take precautions to safeguard their personal information since fraudulent activities often begin with the leaking of personal information such as credit card numbers, one-time passwords, and registered mobile numbers. Therefore, it is necessary to reduce the sharing of personal information to prevent fraudulent activities.

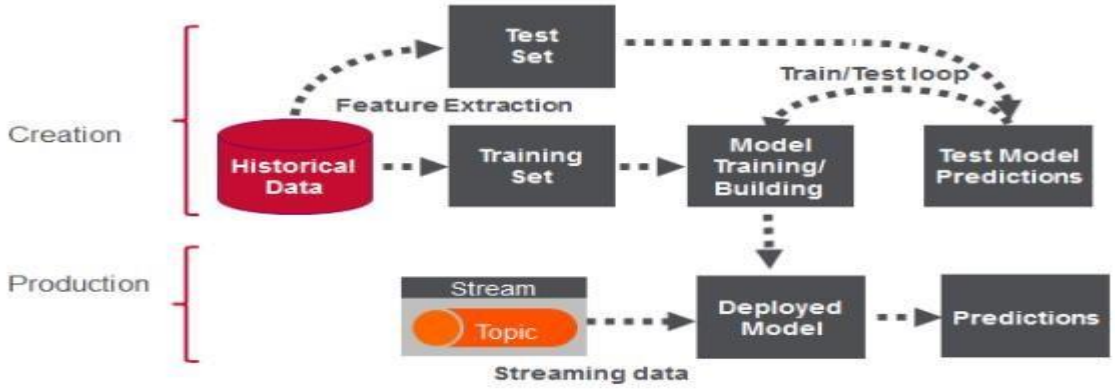


Figure 1.1: System Mechanism

1.3 EXISTING SYSTEM

The previous credit card fraud detection system was slow and not very accurate. There was also a class imbalance problem. An algorithm based on analysis and credit rating was used to detect fraud, but there were limitations to this approach. A new approach using the Isolation Random Forest (iForest) algorithm has been proposed. iForest is a machine learning algorithm used for anomaly detection and can identify patterns in the data that deviate from normal transaction behavior. The iForest algorithm constructs a random forest of decision trees on subsets of the data and can be trained on a dataset of credit card transactions to identify fraudulent behavior. The iForest model calculates a score based on the transaction's deviation from normal behavior and flags potentially fraudulent transactions for further review. iForest is a promising approach to credit card fraud detection, but requires careful tuning and validation to ensure effectiveness and reliability.

1.3.1 DISADVANTAGES OF EXISTING SYSTEM

- In case of fraud there is a high amount losses and thus because of this loss, card limit should be reduced.

- The fraudulent should be detected in real time and omission in false transactions is mandatory.

- Reasons of fraudulent should be identified from data available.

- System should be capable in identifying the trend of fraud transaction.

- Credit card fraudulent transaction should be based on web service scheme.

1.4 THE SYSTEM PROPOSED

- In this model we overcome with the issues in a significant way. Using Random forest, Decision Tree algorithm, Logistic Regression we can detect the fraud in actual time and find out the way to minimize the fraud to produces an optimized result so that it will perform a better prediction. On the basis of customer's behavior, we can detect fraudulent. Here the local outlier factor is used.

- We have used logistic regression and random forest. We can get more accuracy like 0.99 etc…

- We are taking the dataset with help of simple GUI from our local directory where we downloaded the dataset.

- With the help of random forest algorithm and local outlier factor we are finding the data point which is different from its neighbour and can be a fraudulent transaction with its outlier behavior.

- We have two classification class which is named as class 0 and class 1.

- If there is legal transaction then the result will store in class 0 and if there is a fraudulent transaction then the result will store in class 1.

1.5 SOFTWARE REQUIREMENTS

- **Operating system** - Windows7,8,10 and 11 **(32 and 64 bit)**
- **Dataset** - csv
- **Language** - Python
- **Platform** - Google Colab, Anaconda.
- **Processor** – I3 processor.(minimum)

1.5.1 SOFTWARE USED

- Python – 3.x
- Numpy – 1.19.2
- Scikit-learn – 0.24.1
- Matplotlib – 3.3.4
- Imblearn – 0.8.0
- Collections, Itertools

1.6 HARDWARE REQUIREMENTS

- System - Core i3(minimum)
- Mobile - Android
- Monitor - RGB Colour
- Hard Disk - 50 GB or more
- Ram - 8GB

CHAPTER 2

LITERATURE SURVEY

2.1 GENERAL

In our paper, we conducted a literature survey and analyzed various techniques to improve routing performance, reduce information delay, packet loss rate, link failure, packet delivery rate, and energy consumption. We also reviewed numerous algorithms for detecting credit card fraud. Our paper proposes a machine learning algorithm with two features: True Positive and False Alarm, which help in quickly detecting fraudulent behavior. As network security is crucial, our model needs to be updated regularly to detect new fraudulent activities in real-time. Additionally, our paper explores the use of Neural Networks for Fraud Detection Systems. To prevent personal information theft, our model provides charts that show abnormal behavior based on different columns such as time and amount.

2.2 FRAUDULENT DETECTION IN CREDIT CARD SYSTEM

With growing advancement in the electronic commerce field, fraud is spreading all over the world, causing major financial losses. In the current scenario, Major cause of financial losses is credit card fraud; it not only affects tradesperson but also individual clients. Decision tree, Random Forest Algorithm, Metalearning strategy, Logistic Regression are the presented methods used to detect credit card frauds. In contemplating system for fraudulent detection, artificial intelligence concept of Support Vector Machine (SVM) & decision tree is being used to solve the problem. Thus by the implementation of this hybrid approach, financial losses can be reduced to greater extent.

2.3 MACHINE LEARNING BASED APPROACH TO FINANCIAL FRAUD DETECTION PROCESS IN MOBILE PAYMENT SYSTEM

Mobile payment fraud is a rising problem that occurs when someone uses stolen credit card information or personal identity to make unauthorized transactions through mobile payment services. Since financial fraud results in financial loss, it is important to have a highly accurate process for detecting mobile payment fraud. Our proposed approach utilizes machine learning, specifically supervised and unsupervised methods, to detect fraud and handle large volumes of financial data. To achieve high accuracy in mobile payment fraud detection, we perform a sampling process and feature selection process to quickly process large amounts of transaction data. Our model's effectiveness is validated through F-measure and ROC curve analysis.

2.4 CREDIT CARD FRAUDULENT DETECTION

We have developed a Credit Card Fraudulent Detection model that performs well on anonymized datasets and can adapt to changing consumer behavior. The model can detect outlier values and abnormal behaviors that indicate fraudulent transactions. To improve its real-time analysis capabilities, we have reduced dataset redundancy. The model can predict fraudulent activity by running after a fixed amount of transactions or at regular intervals. Our goal is to achieve real-time analysis, and reducing the dataset helps speed up the algorithm's performance.

2.5 DATA SAMPLING

To perform the Random Forest algorithm for Credit Card Fraudulent Detection, a trained dataset is needed to be loaded into the system's main memory. Since the dataset has almost 300,000 values, it is difficult to load it all at once. To address this, redundant datasets were removed to reduce the size of the dataset. The model was trained on previous data, enabling it to detect fraudulent transactions in real-time.

2.6 CREDIT CARD FRAUD DETECTION DEFINITION

Credit card fraud detection systems are complex because the definition of fraud is not clear and the problem is not dichotomous but multi-classification, as different types of fraud exist. Fraud is constantly evolving, and there is no single type of fraud. Banks, insurance companies, and customers are perennial victims of fraud, so they must update their prediction systems regularly. Rather than relying on the same model, fraud detection faces the challenge of constantly improving and evolving to keep up with the changing nature of fraud.

2.7 CREDIT CARD FRAUD DETECTION

Credit card fraud detection is designed to prevent any unauthorized credit card transactions from fraudsters and to recover losses and credibility for customers and businesses. Although there are better financial mechanisms, the fraudster is continually updating his techniques. Also, it makes the anticredit card fraud techniques very challenging; the standard anti-credit card fraud methods available in the market today are listed below.

● Validation method through merchant trade

The merchants often require a complete list of receipts to identify the user and have added tokenisation techniques to protect credit card information by using the referenced card number instead of the current card number. It can make sure that they offer additional information like a PIN, zip code or card security code. Also, they may be requested to show them during the merchant transaction, and they are currently used by merchants to combat fraud(Contributors 2020).

● Geolocation of transactions by IP address

Geolocation technology provides an absolute geographic location through the IP address of the computer where the order placed in a real-time e-commerce transaction which can identify areas with a high potential for fraud. It might allow merchants to attach authentication acne to transaction applications that vary widely in realistic examples to protect them from credit card fraud(FTC.gov 2012).

● Detect IP address countries and whether they are high-risk areas

Detection system makes sure that the IP address country is the same as the billing address country. By using a fraud prevention service, the service can detect the IP address country for the customer placing the order. If the customer's billing and shipping address are in the UK (Duman et al. 2013) but the person placing the order logged in from a Russian IP address, a more rigorous review is required, and anti-fraud precautions are often triggered. It is also always needed that orders shipped to international addresses scrutinised if the card or shipping address is in an area prone to credit card fraud.

● Detecting the use of anonymous mailboxes and proxies

Many legitimate customers use free email addresses because they are convenient and economical. 11 Indeed, most fraudsters use free email addresses to remain anonymous. Detecting new domain registrations for email addresses is one of the most important ways to do a better job of fighting fraud(Bhatla et al. 2003). Secondly, anonymous proxy servers allow Internet users to hide their actual IP addresses. The primary purpose of using a proxy server is to remain anonymous or to avoid detection so people need to save the list of proxies as a web service to prevent credit card fraud.

● Using Neural Networks to Detect Credit Card Payment Fraud

Most of the existing techniques based on deep learning and oversampling algorithms for credit card fraud detection. The Long Short Term Memory Networks (LSTM) fraud detection model for serial classification of transaction data and integration of synthetic minority class oversampling. The Smote and the k-Nearest Neighbor (kNN) classification algorithm design and build a kNN-Smote-LSTM based fraud detection network model which can Improve fraud detection performance by continuously filtering out security-generating samples through kNN discriminant classifiers(Maes et al. 2002).

● Machine learning detection

They are using Machine Learning Classification Algorithms to Detect Credit Card Fraud. Machine learning is a very effective way to detect fraudulent transactions if his performance is good enough because he determined by choice of features, the training of the data drink testing, and the classification methods of machine learning. All of these factors contribute to different generation rates. Many studies have shown that using machine learning classification algorithms to detect credit card fraud has resulted in better accuracy. They have also compared the results of different algorithms and other studies and agreed that machine learning detection is the right choice.

2.8 CREDIT CARD FRAUD IDENTIFICATION

The identification of credit card fraud detection is currently facing challenging because of most people not familiar with credit card fraud. After all, most of the scam comes out through the valid pathway following the banks as well as financial companies, and the only difference is that they are unauthorised third party pathways. The recent credit fraud, as well as becomes more challenging to identify. Because if there has anyone who knows them credit card number, as well as expiration date, he can make a transaction on the website without them permission. Fraudsters will get more information about people's finances, and they will also have more opportunities to make fraudulent transactions by swiping credit cards, rather than just the ones we see.

2.9 CONSEQUENCES OF CREDIT CARD FRAUD

Credit card fraud and process directly concern the user and the financial company; it is a reason we keep focus credit card fraud this year. The following are examples of fraud transaction outcomes.

● Economic losses to users and businesses

● Customer Personal Information Breach and Corporate Disclosure Enterprise trust crisis in information security

Over 45.6 million credit cards were exposed due to TJX's systems from July 2005 to mid-January 2007, with Albert Gonzalez accused of leading the organization responsible for the theft.

Gonzalez was also indicted in August 2009 for stealing information from over 130 million credit and debit cards from Heartland Payment Systems, retailers 7-11 and Hannaford Brothers, and two unidentified companies.

A group of about 100 people used data from 1,600 South African credit cards to steal $12.7 million from 1,400 convenience stores in Tokyo over three hours on May 15, 2016.

2.10 FRAUD COUNTERMEASURES

2.10.1 GENERAL COUNTERMEASURES

The general countermeasures are increasing protection of customer transactions. For instance, they are adding signs to direct cardholders to designated areas. Every cardholder in the self-service program should be protected accordingly, protecting ATMs and bank assets from unauthorised use. Protected areas for any transaction can be monitored through the bank's CCTV system. Also, Cards use CHIP identification to reduce the likelihood of card theft. (Little 2009)

2.10.2 USER TRAINING AND EDUCATION

Credit cardholders should receive training and education on how to use credit cards safely, including reporting card loss or theft to avoid fraud. They should regularly check billing charges and report any unauthorized transactions immediately to their financial center. It is also recommended to install virus protection software on their computers, securely store account information, and avoid using credit cards on untrusted websites. Additionally, customers should not send credit card information via unencrypted emails and avoid retaining PINs when using credit cards.

2.10.3 GOVERNMENT LEGISLATION

The identification of credit card fraud also requires the help of our government regulators. For example, the enactment laws of consumer protection related to card fraud transaction. It will help to optimise the market environment and ensure the safety of the credit card transaction market. Also based on the EU GDPR principles, any company and card issuers should publish standards, guidelines and codes to protect cardholder information and monitor fraudulent activities, or be fined(Foulsham 2019).

CHAPTER 3

CREDIT CARD FRAUDULENT DETECTION SYSTEM

3.1 GENERAL

Anomaly detection is the identification of unusual patterns, known as outliers, that do not conform to expected behaviors. It is used in many business applications, including network monitoring, credit card fraud detection, and operating system fraud detection. In our case study, we use the Isolated Random Forest and Local Outlier Factor models to detect fraudulent activity with a focus on minimizing false alarms. By plotting charts for each feature from V1 to V28, we can easily recognize fraudulent transactions. This system has the advantage of working efficiently with limited amounts of data and maintaining the privacy of personal information. The accuracy of our model is satisfactory.

3.2 PROBLEM DEFINITION OF CREDIT CARD FRAUDULENT

As online payment and credit card usage continue to increase, so does credit card fraud. This type of fraud often involves the use of stolen credit card information during online transactions. However, relying solely on credit card billing to detect and prevent fraudulent activities is not sufficient. Therefore, an efficient system that can quickly detect and alert customers of fraudulent transactions is necessary. In addition, limiting the amount of transactions that can be made in a day or at a time can also help reduce losses.

To determine whether a transaction is legal or fraudulent, two analyzers, namely the random forest algorithm and local factor outlier, are used. These analyzers help calculate a score prediction, which represents a more balanced result between legal and fraudulent transactions. The accuracy of the prediction is based on a balanced dataset, which plays an essential role in building the problem model. It is also necessary to split the dataset into a training dataset and a test dataset to train and test the class classifier for the problem model. Overall, the goal is to develop a system that can efficiently detect and prevent credit card fraud while minimizing losses for customers.

3.3 BLOCK DIAGRAM

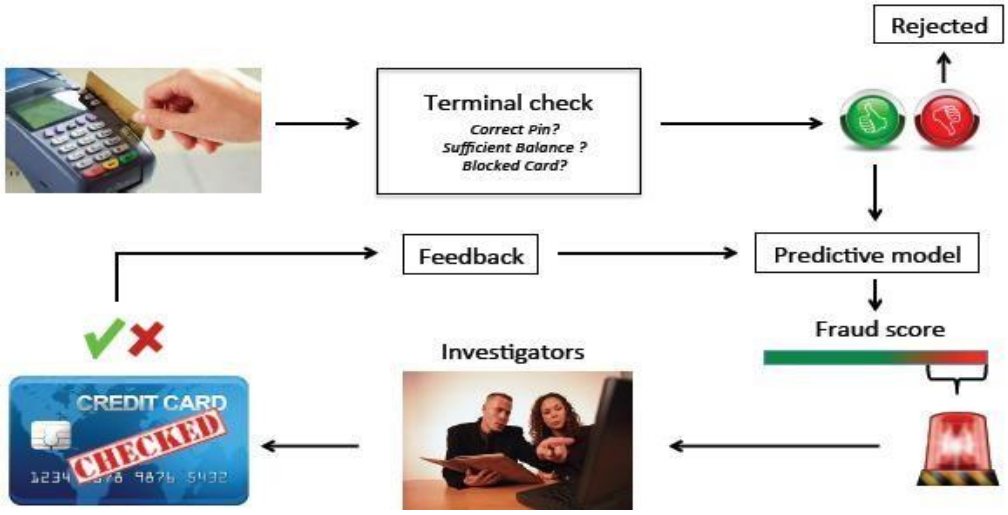


Figure 3.1: Block Diagram

3.4 METHODOLOGY

The task which is performed for the prediction of transaction and labelled as fraud is detected on the basis of binary classification. We make two class for the prediction of fraud: class 0 and class 1.

Class 0 if there is no fraud and class 1 to catch the fraud. This can be done with the help of binary classification.

3.4.1 WHAT ARE ANOMALIES?

Anomalies can be categorized as following:

- Point Anomalies: Point anomaly is a single instance of data. The credit card fraudulent detection technique is based on “amount spend”.

- Contextual Anomalies: The best example of contextual anomaly is time-series data.

- Collective Anomalies: Here, Detection of anomaly is based on a set of data instances collectively. Therefore, a set of data will help in detecting fraudulent anomaly. If someone try to theft personal data from server it will come under collective anomaly and named as cyber-attack.

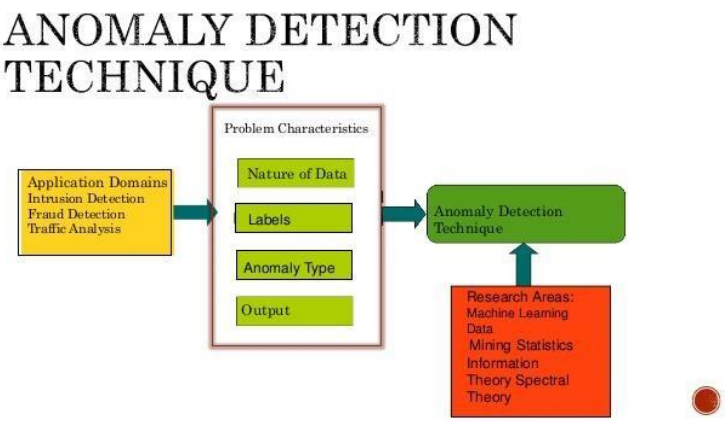


Figure 3.2: Anomaly Detection Technique

3.4.2 ANOMALY DETECTION

Identifying an unobserved pattern in new observation is the main area of concern. It's include training of dataset.

3.4.3 NOISE REMOVAL

Noise removal is the process of removing noise from meaningful data, noise is unnecessary data along with the meaningful data.

3.5 ANOMALY DETECTION TECHNIQUES

The various Anomaly Detection Techniques are as follows

3.5.1 SIMPLE STATISTICAL METHODS

The simple way by which we can determine the irregularities in dataset by determining the deviation of data point from common statistical distribution, for example mean, mode and median.

Anomaly data point is that deviates by a certain standard deviation from mean. To compute average data point we need a rolling window across data points which is known as moving average which is used to find low pass filter.

3.5.2 CHALLENGES WITH SIMPLE STATISTICAL METHODS

The low pass filter allows us to identify anomalies in simple use cases, but there are some framework where this method fails to determine anomaly data point. Data which contain noise data which can be named as abnormal data, as the boundary between normal and abnormal are not accurate. Therefore, it's a big problem to identify threshold value because the moving average can't apply in that framework.

3.6 CREDIT CARD FRAUDULENT DETECTION SYSTEMS

All the credit card fraudulent detecting models are evaluated and compared using this model.

Accuracy - It is characterized as a bit of all the quantity of exchanges which are distinguished effectively.

Methodology - This indicates the instrument pursued by the credit card FDS.

True Positive or TP - Legal and fraud transaction are detected on this basis. Genuine transaction only counted here.

False Positive or FP - Legal and fraud transaction are detected on this basis. Fraud transaction only counted here.

Supervised Learning - In this supervised data is fed in the machine.

3.7 FUNCTIONALITIES

Many organization and banks will take the benefit from this model. Because this will be a significant model for the prediction of credit card fraudulent. This will detect the consumer behaviors and his last transaction and predict whether the consumer is fraud or not. We use random forest and local outlier factor for the fraudulent. We need to have controls over the algorithm in order to fit with the data set. It will help our application to improve and to be more efficient in order to detect the fraudulent transactions and help us in solving problems.

3.8 ACCURACY

The Fraudulent Detection is done on basis of previous transaction history of consumer. We will detect out of whole transaction how much result in fraud. Then we will identify whether a new transaction made by customer is fraudulent or not. With the help of this model we achieve 99.97% accuracy in finding fraudulent transactions.

3.9 OBSERVATION

The data set contains 492 frauds out of almost 300,000. This results a probability of 17.2% fraudulent cases. This identified that there is much more fraud customer. The data sets consists of column which start from v1 and end as V28. There are much features present from V1 to V28. Furthermore, there is no missing value present in datasets. The datasets has column name as Time & Amount. The analysis is done on the basis of ranges present in this two columns.

The datasets contains the numerical value which can be called as PCA transformation. Due to security issue, unfortunately we cannot take the original features and information about data. Column V1 to V28 are taken as principal components. The features which is not transformed with PCA are “Time” and “Amount”.

“Time” plays an important role here as it is used to determine the time between each transaction and it is calculated in seconds.

“Amount” is another feature which is used to determine the transactional Amount.

“Class” is the most important feature here in our model which is response variable and it takes the value as 1 and 0. It gives value 1 in case of fraud and value 0 in case of legal transaction. The main goal of this model is to predict the credit card fraudulent, for all transaction which is received as online payment to check whether the transaction is legal or not. If the transaction is genuine then it is consider as legal transaction and the transaction which has fraudulent should be recognize as fraud transaction. All this is performed with the help of random forest algorithm and local outlier factor to make an assumption of true probability and false probability. The result obtain after this algorithm performed successfully is then plotted as graph and heat-map. This model is also tested for different test cases and also compared with the previous all model and the accuracy is also compared.

CHAPTER 4

INTRODUCTION OF MACHINE LEARNING

4.1 GENERAL

Machine learning is a type of artificial intelligence that involves creating algorithms and statistical models to enable computer systems to learn from data and make predictions or decisions based on that data. The aim is to develop intelligent systems that can improve over time without being explicitly programmed. It has applications in many fields and has already transformed industries like advertising, e-commerce, and search engines. AI uses algorithms and calculations based on human intelligence to solve problems and can address a wide range of problems. AI is trained with general problem-solving abilities and creates its own algorithms for solving problems. AI engines perform predictions or analysis using a set of data and a PC framework, and unsupervised learning is used for operation. Modifications are made for business purposes before AI is applied.

4.2 OVERVIEW OF MACHINE LEARNING

Machine learning was first named in 1959 by Arthur Samuel, but it was Tom M. Mitchell who gave a more formal definition of the field. This definition offers an operational definition rather than defining the field in psychological terms, as suggested by Alan Turing's paper "Computing Machinery and Intelligence". Prior to the introduction of machine learning, it was assumed that robots needed to learn everything from human brains to function properly. However, it was found to be difficult to teach robots everything humans know, so the idea of making robots learn on their own was proposed, leading to the birth of the term "machine learning". Machine learning uses different approaches and algorithms to train a model based on the dataset being used. The machine learning process is iterative, and modifications are made to achieve the desired output.

4.3 MACHINE LEARNING CLASSIFIERS

In this project, we used a total of five classifications methods（Random Forest, Logistic regression, KNN, Support vector machine (SVM), Decision tree(DT), Category&boosting(Cat boots). These classification algorithm methods are widely used for problems such as differential training dataset. Also, it commonly used in classification learning. That is the reason I compare them in the same training dataset. Also, it can be a cross-sectional comparison with other current studies in the final results.

4.3.1 Random Forest algorithm

Random Forest is a popular machine learning algorithm that belongs to the ensemble methods family. It combines multiple decision trees, each trained on a random subset of input data and features, to produce a final prediction by aggregating the predictions of these individual trees. The model can capture a wider range of relationships between input features and the target variable, and mitigate the risk of poor performance on new data by preventing models from becoming overly complex and overfitting. Random Forest outperforms other machine learning algorithms in handling a large number of input features, resilience to noisy data, and generating feature importance rankings. It has a high accuracy rate and requires little tuning.

4.3.2 Logistic regression algorithm

Logistic regression. It is a classical and effective bicategorical algorithm for classification problems, especially those with two possible outcomes. It is based on the principle of simplicity before complexity and is a recognized statistical method for predicting the outcome of a binomial or polynomial. It can also be regenerated as a multinomial logistic regression algorithm for fields with more than two possible values.

Logistic regression is advantageous because it is faster to process and suitable for bicategorical problems. It is also easier for beginners to understand and update with new data. However, it has a limitation in adaptability to large datasets compared to the decision tree algorithm. This issue can be determined based on the project's specific situation, such as credit card transactions.

The main methods of logistic regression method:

Objective: It is to look for some risk factor, then in this project, They want to find a particular transaction factor or reasons that are suspected of being fraudulent.

Prediction: Predicting the probability of fraud under other independent variables, based on different algorithmic models.

Judgment: It is somewhat similar to prediction. It is also based on different models to see how likely it is that a transaction is a risk factor in a situation where fraud falls into a specific category.

Regression General Steps

● Finding the h-function (i.e., the prediction function)

Constructing the predictive function h(x), the logistic function, or also known as the sigmoid function, we generally the first step is to build the predictive process, where the training data for the vector, as well as the best parameters. The basic form of the function shown in figure 1

$$g(z) = \frac{1}{1+e^{-z}}$$

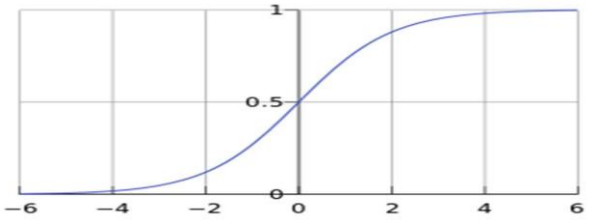


Figure 1: Logical function expressions

- Constructing the J-function (loss function)

The second step is that we need to construct the loss function-j. In general, there will be m samples, each with n characteristics. The Cost and J functions are as follows, and they are derived based on maximum likelihood estimation(Sahin and Duman 2011).

$$Cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & if\ y = 1 \\ -\log(1 - h_\theta(x)) & if\ y = 0 \end{cases}$$

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x_i), y_i) = -\frac{1}{m}\left[\sum_{i=1}^{m}(y_i log h_\theta(x_i) + (1 - y_i)\log(1 - h_\theta(x_i)))\right]$$

Figure 2: The Cost and J functions

- Figure out how to make the J-function minimal and find the regression parameter (θ)

The final step is that we, using gradient descent, solve for the minimum value of θ. The process of updating θ can then be summarised as follows

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i) x_i^j$$

Figure 3: The process function of updating θ

4.3.3 Decision tree (DT)

Decision tree is a method that is often used to evaluate the risk of a training project based on the known probability of various scenarios. By forming a decision tree, we can find the possibility that the expected net present value is greater than or equal to zero, which helps to judge the feasibility of the decision analysis method. Decision tree is named as such because its decision branch is drawn as a graph much like the trunk of a tree. In machine learning, decision trees are used for classification and regression and typically involve three steps: feature selection, decision tree generation, and decision tree pruning. Decision tree is a popular classification method used to analyze data and make predictions, including the detection of credit card fraud. That is why it was chosen for the training of the fraud detection system in the mentioned paper.

That is a simple decision tree classification model: the red boxes are features.

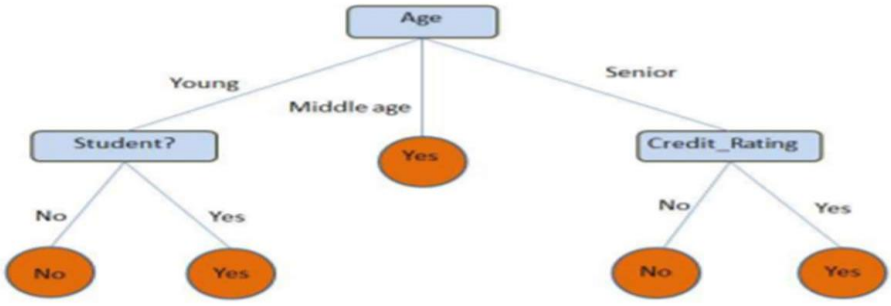


Figure 4: A simple decision tree

People may be wondering why we chose a decision tree? There are two universal reasons: Decision trees usually mimic human horizontal thinking, so it is easy to understand the data we provide and make some excellent interpretations. Decision trees allow you to see the logic of how the data is interpreted, unlike SVM, NN. and other similar black-box algorithms where you do not see any internal information(Gaikwad et al. 2014).For example, as the figure above, we can see how the logic makes decisions. Plain and simple.

Then, what is a decision tree now? A decision tree is kind like a tree which each node represents an element (attribute), each link (branch) means a decision (rule), and each leaf represents a result (categorical or continuous value). The core of the entire decision tree is to create a tree-like this for the whole of the data. And the decision tree process individual results (or minimise errors in each leaf) on each plate.

4.3.4 k-nearest neighbour (KNN)

Knn is a classification technique proposed by Cover and Hart in 1968, which uses the K nearest neighbours to represent each sample. The algorithm calculates the distance between the unknown samples and all available known examples to determine the category of the unknown samples. The K nearest known examples are selected based on the majority rule and the unknown sample is classified based on the category with the most votes.

$$d((x_1,\ldots,x_n),(y_1,\ldots,y_n)) = \left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{1/p}$$

Figure 5: The formula for calculating the distance between two points

The K value of the KNN algorithm in 'scikit-learn' is adjusted by the n_neighbors parameter, and the default value is 5.

As shown in the figure below, how do people determine which Category a green circle should belong to, whether it is a red triangle or a blue square? If K=3, the green process will be judged to belong to the red triangle class because the proportion of red triangles is 2/3, and if K =5, the green circle will be considered to belong to the blue square class because the ratio of blue squares is 3/5(Gaikwad et al.2014).

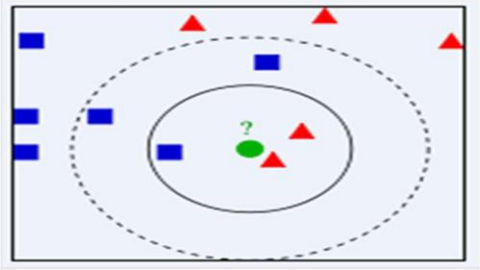


Figure 6: The k-nearest neighbour sample

4.3.5 Support vector machine (SVM)

Support Vector Machine (often abbreviated as SVM) is a supervised learning method, most widely used in statistical classification and regression analysis. It is also the focus of this project. Support vector machines belong to a family of generalised linear classifiers which are characterised by their ability to both minimise empirical errors and maximise geometric edge regions. Hence support vector machines are also known as maximum edge region classifiers.

The core principle of the support vector machine is: mapping the vectors into a higher dimensional space where a maximum spacing hyperplane is established. Two parallel hyperplanes are built on either side of the hyperplane that separates the data. Also, the separated hyperplanes maximise the distance between the two parallel hyperplanes(Singh et al. 2012). It is assumed that the greater the space or gap between the parallel hyperplanes, the smaller the total error of the classifier. In this project, SVM is the supervised learning algorithm used to solve the multi-class classification(Bhattacharyya et al. 2011).

Distance from example to the separator is $r = y\frac{\mathbf{w}^T\mathbf{x}+b}{\|\mathbf{w}\|}$

Examples closest to the hyperplane are support vectors. Margin ρ of the separator is the width of separation between support vectors of classes.

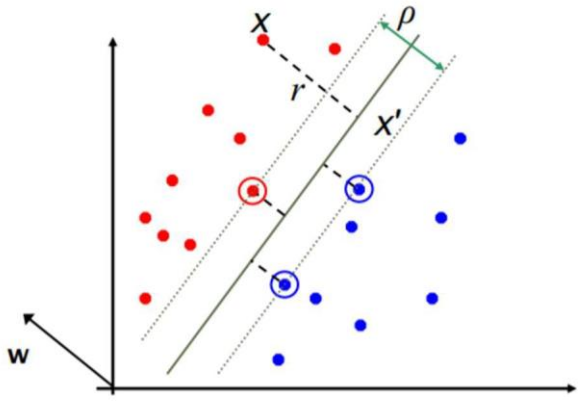


Figure 7: Geometric Margin

4.3.6 Category&boosting(Catboost)

CatBoost is a machine learning library that is based on gradient boosting decision trees and was open-sourced by Yandex in 2017. It is specifically designed to handle categorical features well, and its name comes from the words "Category" and "Boosting". CatBoost contains various tree type algorithms and is a universal library of gradient boosting algorithms. It was chosen for comparison with four individual classification algorithms, including DT, because of its comprehensive performance and ability to handle categorical features.

Boost algorithm is another even beyond "lintbm" and "xgboost" by the author from a developer's point of view. The catboost has some of the following advantages:

- Catboost is a machine learning algorithm that has a unique approach to dealing with categorical features. It uses statistics on the categories to calculate their frequency and generates new numerical features through hyper-parameters. It is also robust, reducing the need for tuning many hyperparameters and the chance of overfitting. Finally, it is practical for real-world use..

- The catboost is also a more practical method. It can handle both Category and numerical features and uses combined category features that can take advantage of the links between elements which significantly enriches the feature dimension.

- The base model of catboost uses symmetric trees, and the way to calculate the leaf-value is different from the traditional booster algorithm which calculates the average. However, catboost has been optimized to use other algorithms to prevent overfitting of the model. That is why the catboost algorithm can rival any advanced machine learning algorithm in terms of performance.
- Catboost is easy to use: catboost provides a Python interface for integration with scikit, as well as R and command-line interfaces which facilitate quick calls and reduce the number of calls. Also holds a custom loss function which also reflects his extensibility.

4.3.7 Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE is called Synthetic Minority Oversampling Technique which is an improvement of the random oversampling algorithm. The basic idea of the SMOTE algorithm is to analyse a small number of samples data and add new samples to the dataset based on the analysis of a small number of sample (Stolfo et al. 1997)s.

However, the class-imbalance problem that we need to solve next in this project refers to the uneven distribution of classes in the training set used in the training classifier (Pun 2011). For example, for a binary problem with 1000 training samples, ideally, the number of positive and negative models are similar; if there are 995 positive samples and only five negative samples, it means there is class imbalance. There is also the case for the dataset in this project. We can see more details in section 3.3.

For now, there are three main approaches.

- Adjusting the value of θ

Adjust the value of θ according to the proportion of positive and negative samples in the training set. It is done based on the assumptions made about the training set, as described above. However, whether this assumption holds in the given task is open to discussion.

- Over sampling

The classes with a small number of samples inside the training set (few types) are oversampled, and new models are synthesized to mitigate class imbalance.

- Under sampling

Under-sampling of classes with a large number of samples inside the training set (most categories), discarding some examples to mitigate class imbalance(Dal Pozzolo et al. 2015).

In this project, we use oversampling and under sampling to perform comparison operations. At the same time, we can also compare the results to analyse whether the two methods are more suitable for this project's dataset, and what are the advantages and disadvantages of each technique(Alghamdi et al. 2017).

The core idea of SMOTE (synthetic minority oversampling technique) in a nutshell is to interpolate between minority class samples to generate additional models. For example, for a minority sample xi use the k-nearest neighbour method (k values need to be specified in advance) to find the k nearest minority samples to xi(Sahin et al. 2013). The distance is defined as the Euclidean distance in the n dimensional feature space between the models. One of the k nearest neighbours is then randomly selected to generate a new sample using the following formula(Han et al. 2005).

$$\mathbf{x}_{new} = \mathbf{x}_i + (\hat{\mathbf{x}}_i - \mathbf{x}_i) \times \delta$$

Where x^ is the elected k-nearest neighbour point, and δ ∈ [0,1] is a random number. An example of a SMOTE-generated sample, using 3-nearest neighbours, is shown in the following figure which shows that the SMOTE-generated model generally lies on the line connected by xi and x^i.

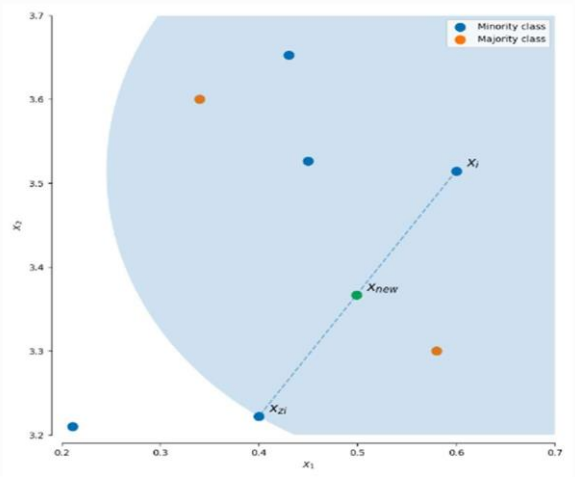


Figure 8: The example of SMOTE formation sample

4.4 DATASET

A dataset is a collection of data that may or may not be related to each other, which consists of multiple columns (parameters) and rows (tuples). It can consist of data related to a particular domain and its purpose defines its size. Data pieces are also called datum, and the relational parameters are often numerical or logical. A dataset is modified while keeping interdependencies in mind. Large datasets are called Big Data, which require new algorithms and tools to cope up with the vast amount of data. Data can also be classified on the basis of its dynamism.

4.4.1 DATASET DETAILS

• Time

• Number of seconds slipped by between this exchange and the primary exchange in the dataset

• V1 up to V28

• It might be consequence of a PCA Dimensionality decrease to secure client personalities and touchy features (v1-v28)

4.4.2 AMOUNT

- Transaction amount
- Class
- 1 for fraudulent transactions, 0 otherwise

CHAPTER 5

DESIGN ENGINEERING

5.1 GENERAL

The UML is used for business and production based works. The task of using UML is to provide a solution or working of a product or model using visual representation. UML involves usage of lock diagrams and flow chart to depict the interrelation and workflow of a model. Sometimes it is also used for planning purposes or analysis as a reference for further development of a project.

- Provides direction with regards to the requests of the group exercise.

- Software ancient rarities create.

- Directs of errand to individual designers and group.

- Offer the criteria to check & estimate the task's item & exercise.

The UML intestinally process autonomous and can be attached with regards to various procedures. All things considered, It is the most reasonable for utilize driven, intuitive and gradual improvement forms. A case for such procedure is Rational Unified Process (RUP).

5.2 ACTIVITY DIAGRAM

It portray the work process conduct of a framework. It ought to be utilized related to other displaying methods, for example, connection and state chart.

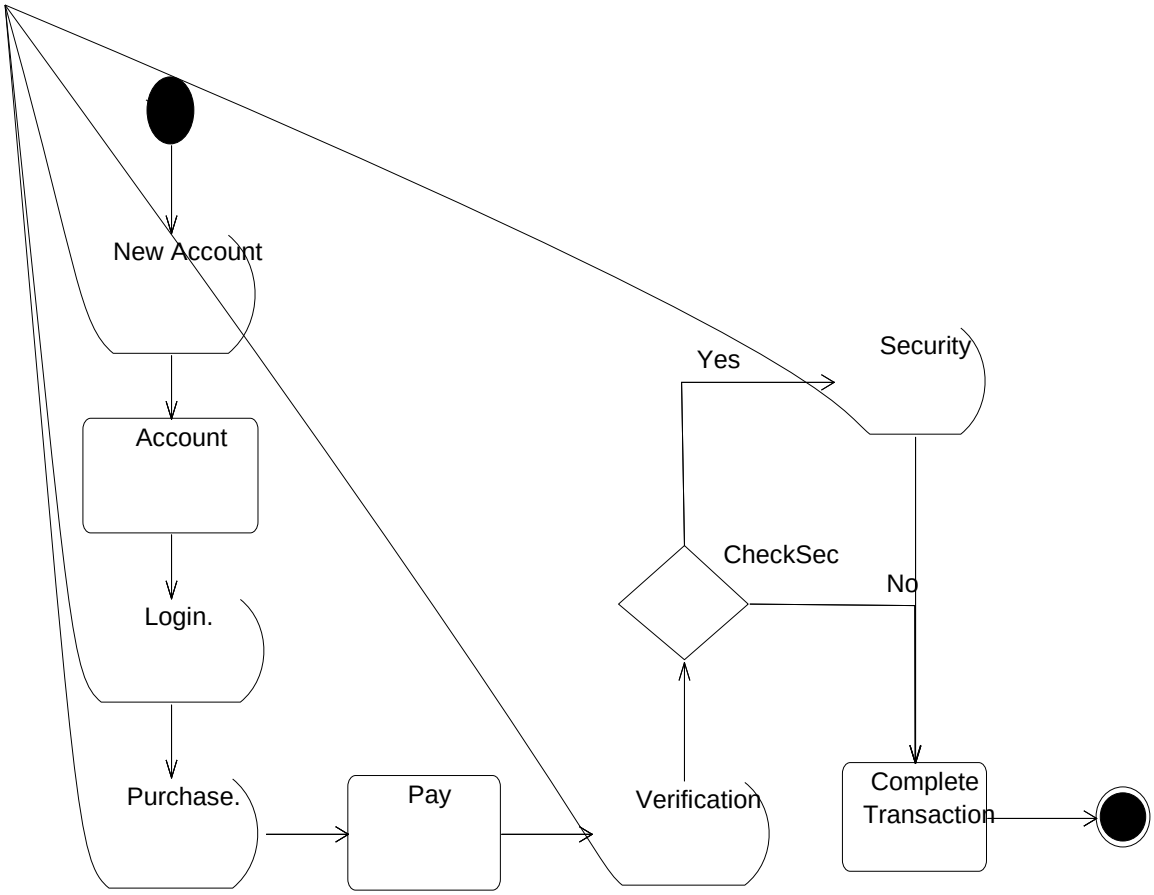


Figure 5.1: Activity Diagram

5.3 USE CASE DIAGRAM

Use case chart show the relationship, among performers and clients. Use cases are utilized in pretty much every task they are useful in uncovering prerequisites and arranging the venture. Amid the underlying phase of a task most use cases ought to be characterized yet as a venture proceeds with more become an obvious.

Use case diagrams are used to describe association of actors along with the working model. It is often used to describe a static state of a model. Use case model consists mainly of two components: The one who interacts with the system (Actor) and the system in consideration.

Use case charts are3 formally incorporate into two displaying dialects they are Unified Modelling Language (UML) and System Modelling Language (SML).

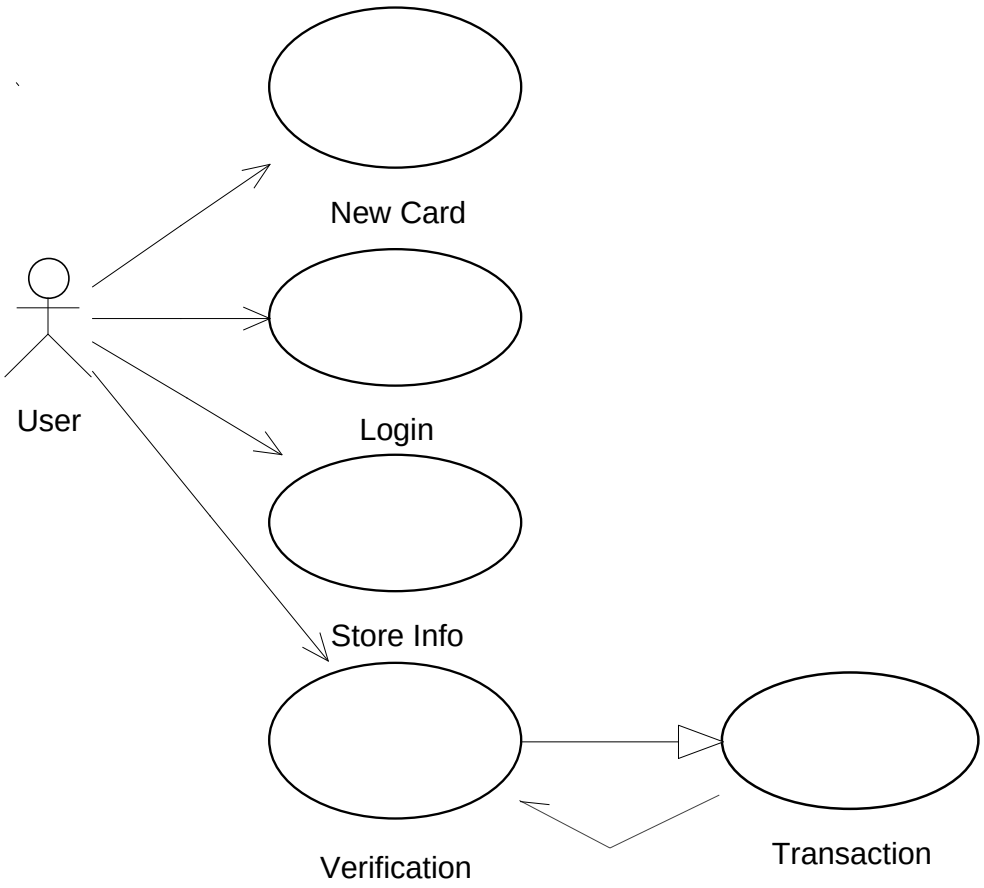


Figure 5.2: Use Case Diagram

5.4 SEQUENCE DIAGRAM

Arrangement graph is a collaboration outline that indicates how work with each other and in what request object. Its build of a message arrangement short. A succession outline indicates object connection organized in time arrangement. It delineates the article and classes included the situation and the arrangement of message trade between the items expected to complete the utilitarian of the situation.

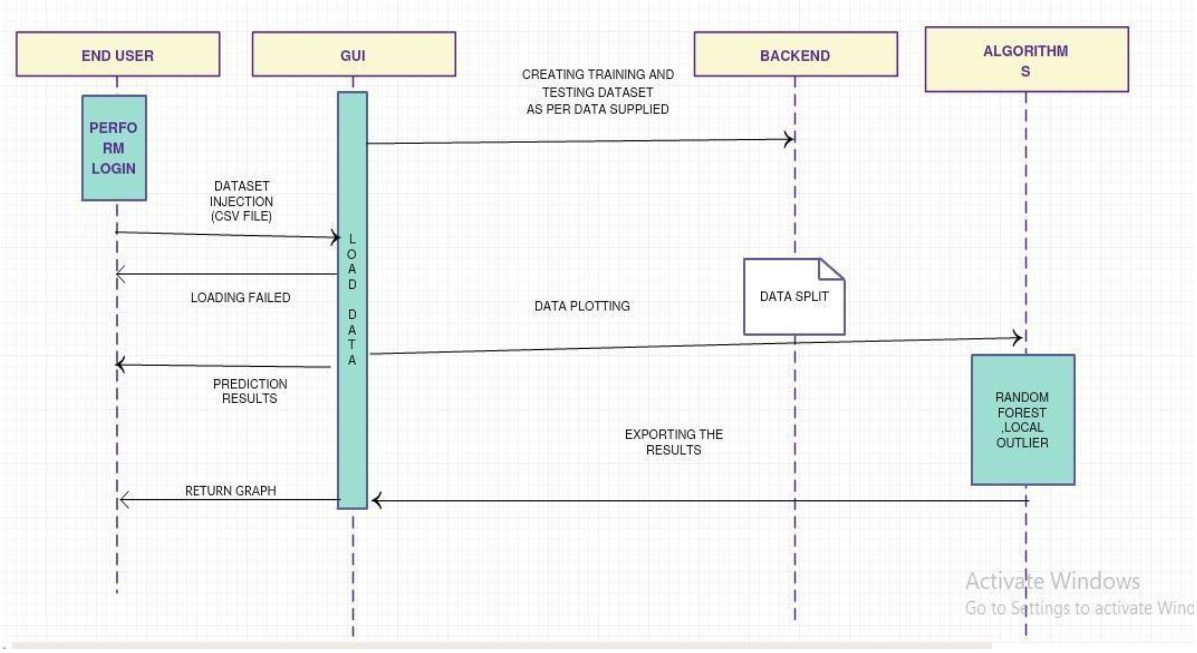


Figure 5.3: Sequence Diagram

5.5 CLASS FEATURE

Class diagram makes us of inter-related structures which consists of package, entities, objects and variables. This depicts relationship between each of the entities through associations, containment and inheritance etc. Using class diagram it becomes easier to understand the holistic working of entities in work along with their inner functionalitites. It is widely used in Object Oriented Software designs.

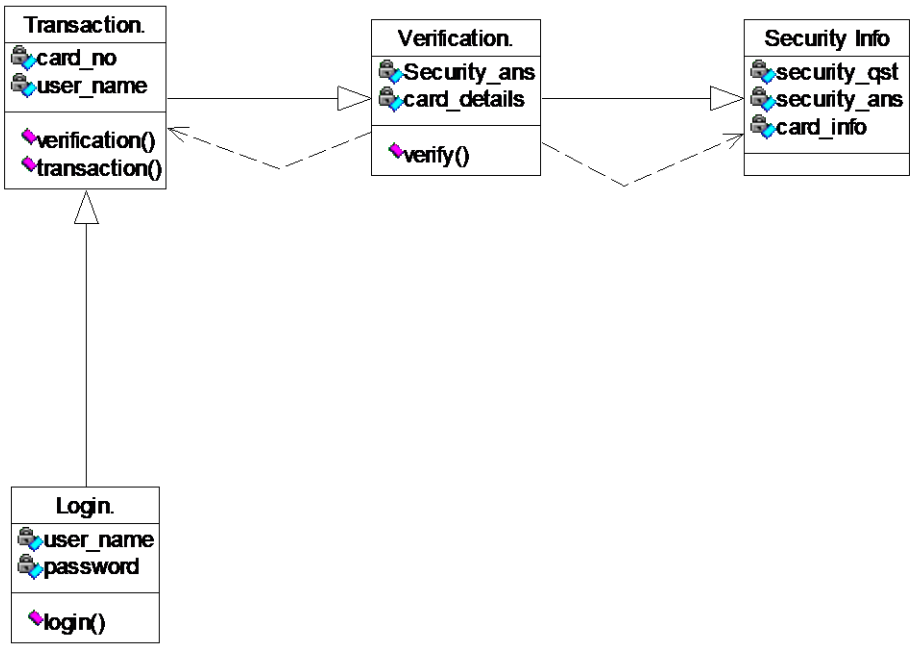


Figure 5.4: Class Diagram

5.6 THE DATA-FLOW-DIAGRAM

Data Flow Diagram is used to represent the requirements of a system in graphical form. It depicts what the data flow is rather than how they are being processed.It is known as bubbler chart. It defines important transaction in a system as a part of requirement of the model. It is used in the starting phase of a design process for reference of further development on the basis of the current workflow.

It is depicted by collection of bubbles and lines. The bubbles represents the transactions and operations whereas the lines demonstrates the connection/flow between each transactions. It is independent of hardware, software and datasets used and is a general outline in simple words.

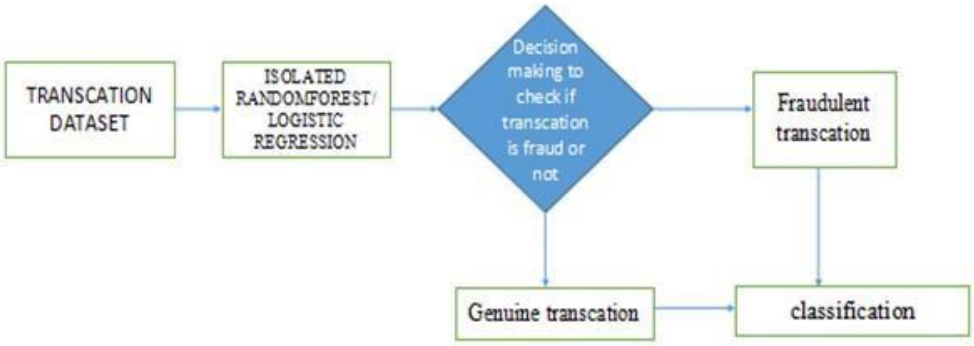


Figure 5.5: Data Flow Diagram

5.7 COMPONENT DIAGRAM

Segment chart shows the abnormal state bundle structure of the code itself. Conditions among parts are demonstrated including source code segments, double code segments, and executable segments. A few parts exist at arrange time, at connection time, at run time well as at more than one time

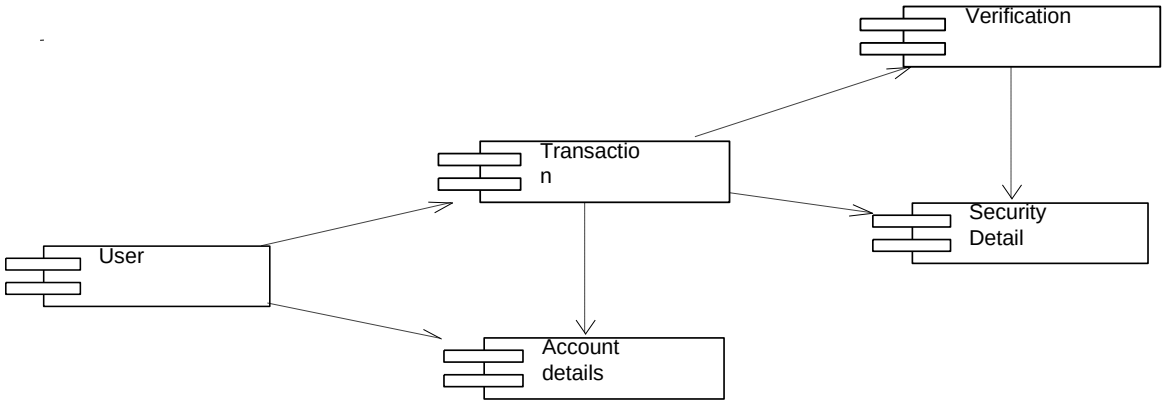


Figure 5.6: Component Diagram

5.8 DEPLOYMENT DIAGRAM

Arrangement graphs shows the design of run time handling components and the product parts, procedures, and items that live on them. Programming part occasion speak to run time appearances of code units.

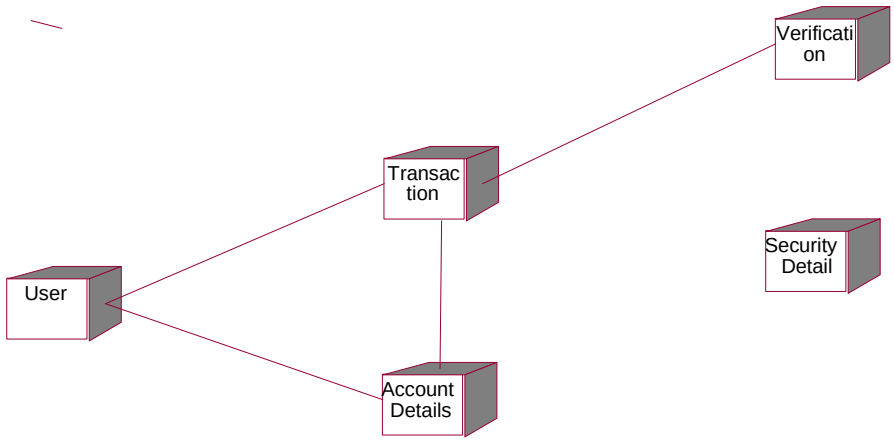


Figure 5.7: Deployment Diagram

CHAPTER 6

IMPLEMENTAION

6.1 GENERAL

Implementation phase brings out the design tweaked out into a operational system. Hence this can be deliberated to be most precarious juncture in accomplishing the efficacious system and in convincing the user faith that system will operate and be effective. This phase encompasses vigilant planning & design, examination of prevailing system and constraints on execution, design & scheming of methods to change over.

6.2 PROCEDURE FOLLOWED DURING IMPLEMENTATION

The application – Credit Card Fraud Detection which is in itself the complete & full-fledged GUI enabled application to envisage/foresee the authenticity & legitimacy of a transaction has been implemented, as per the following steps:

- Install Anaconda from an reliable source.
- Import packages: pandas, Scipy, Matplotlib, Seaborn
- Load the dataset, a dataset is the pool of data for analytical/critical purpose, a (.CSV) file.
- Reconnoiter and get through the dataset through data. shape, data. describe.
- Split the dataset into training dataset and testing dataset.
- Plot histogram of the dataset to epitomize/depict numerical data.
- Determine the count of fraud cases by checking if class is 0 or 1.
- In the similar procedure, get the correlation matrix.

- Next, there is a need to determine the local outlier factor.
- This is followed by use of random forest algorithm to find accurate results.
- The GUI is developed using PyQt library.
- The PyQt library, provides tools to achieve a complete GUI enabled application, similar to swings in java environment.
- Define the constructor in the file.
- Write down the entire implementation inside, thus encapsulating everything inside a GUI-enabled python file.

6.3 Steps to Develop Credit Card Fraud Classifier in Machine Learning

Our approach to building the classifier is discussed in the steps:

- Perform Exploratory Data Analysis (EDA) on our dataset
- Apply different Machine Learning algorithms to our dataset
- Train and Evaluate our models on the dataset and pick the best one.

Step 1. Perform Exploratory Data Analysis (EDA)

There are a total of 284,807 transactions with only 492 of them being fraud. Let's import the necessary modules, load our dataset, and perform EDA on our dataset. Here is a peek at our dataset:

 # Import the necessary modules

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from collections import Counter
import itertools
import os
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, confusion_matrix, recall_score,
 f1_score
from google.colab import drive
drive.mount('gdrive')
%cd gdrive
```

 # Load the csv file

```
dataframe = pd.read_csv("My Drive/1.MKR/creditcard.csv")
dataframe.head()
```

Out[2]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267

5 rows × 31 columns

Now, check for null values in the credit card dataset. Luckily, there aren't any null or NaN values in our dataset.

```
dataframe.isnull().values.any()
```

OutPut:

False

The feature we are most interested in is the "Amount". Here is the summary of the feature.

```
dataframe["Amount"].describe()
```

Output:

```
dataframe["Amount"].describe()

count    284807.000000
mean         88.349619
std         250.120109
min           0.000000
25%           5.600000
50%          22.000000
75%          77.165000
max       25691.160000
Name: Amount, dtype: float64
```

Now, let's check the number of occurrences of each class label and plot the information using matplotlib.

```
non_fraud = len(dataframe[dataframe.Class == 0])
fraud = len(dataframe[dataframe.Class == 1])
fraud_percent = (fraud / (fraud + non_fraud)) * 100
print("Number of Genuine transactions: ", non_fraud)
print("Number of Fraud transactions: ", fraud)
print("Percentage of Fraud transactions: {:.4f}".format(fraud_percent))
```

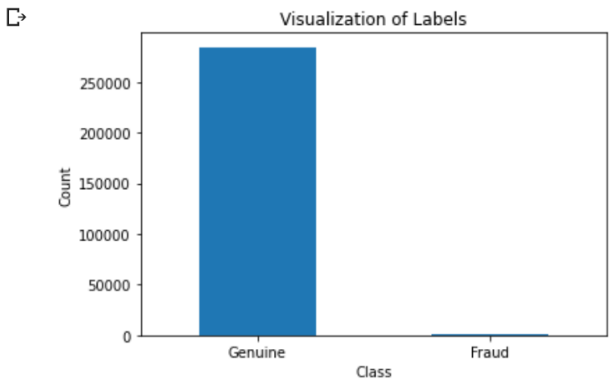
Output:

```
Number of Genuine transactions:  284315
Number of Fraud transactions:  492
Percentage of Fraud transactions: 0.1727
```

Let's plot the above information using matplotlib.

```
import matplotlib.pyplot as plt
labels = ["Genuine", "Fraud"]
count_classes = dataframe.value_counts(dataframe['Class'], sort= True)
count_classes.plot(kind = "bar", rot = 0)
plt.title("Visualization of Labels")
plt.ylabel("Count")
plt.xticks(range(2), labels)
plt.show()
```

Output:



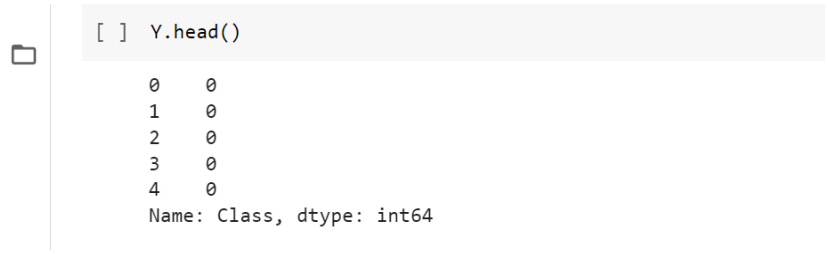
We can observe that the genuine transactions are over 99%! This is not good.

Let’s apply scaling techniques on the “Amount” feature to transform the range of values. We drop the original “Amount” column and add a new column with the scaled values. We also drop the “Time” column as it is irrelevant.

 # Perform Scaling

```
scaler = StandardScaler()
import numpy as np
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
dataframe["NormalizedAmount"] =
scaler.fit_transform(dataframe["Amount"].values.reshape(-1, 1))
dataframe.drop(["Amount", "Time"], inplace= True, axis= 1)
Y = dataframe["Class"]
X = dataframe.drop(["Class"], axis= 1)
```

OutPut:

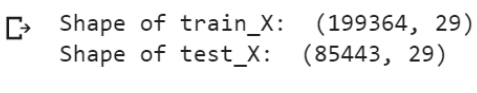


Now, it’s time to split credit card data with a split of 70-30 using train_test_split().

#Split the data

```
from sklearn.model_selection import train_test_split
(train_X, test_X, train_Y, test_Y) = train_test_split(X, Y, test_size= 0.3, random_state= 42)
print("Shape of train_X: ", train_X.shape)
print("Shape of test_X: ", test_X.shape)
```

Output:



Step 2: Apply Machine Learning Algorithms to Credit Card Dataset

Let's train different models on our dataset and observe which algorithm works better for our problem. This is actually a binary classification problem as we have to predict only 1 of the 2 class labels. We can apply a variety of algorithms for this problem like Random Forest, Decision Tree, Support Vector Machine algorithms, etc.

In this machine learning project, we build Random Forest and Decision Tree classifiers and see which one works best. We address the "class imbalance" problem by picking the best-performed model.

But before we go into the code, let's understand what random forests and decision trees are.

The Decision Tree algorithm is a supervised machine learning algorithm used for classification and regression tasks. The algorithm's aim is to build a training model that predicts the value of a target class variable by learning simple if-then-else decision rules inferred from the training data.

Random forest (one of the most popular algorithms) is a supervised machine learning algorithm. It creates a "forest" out of an ensemble of "decision trees", which are normally trained using the "bagging" technique. The bagging method's basic principle is that combining different learning models improves the outcome.

To get a more precise and reliable forecast, random forest creates several decision trees and merges them.

Let's build the Random Forest and Decision Tree Classifiers. They are present in the sklearn package in the form of RandomForestClassifier() and DecisionTreeClassifier() respectively.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
#Decision Tree
decision_tree = DecisionTreeClassifier()
```

 # Random Forest

random_forest = RandomForestClassifier(n_estimators= 100)

Step 3: Train and Evaluate our Models on the Dataset

Now, Let's train and evaluate the newly created models on the dataset and pick the best one.

Train the decision tree and random forest models on the dataset using the fit() function. Record the predictions made by the models using the predict() function and evaluate.

Let's visualize the scores of each of our credit card fraud classifiers.

 # Decision Tree Classifier

decision_tree = DecisionTreeClassifier()

decision_tree.fit(train_X, train_Y)

predictions_dt = decision_tree.predict(test_X)

decision_tree_score = decision_tree.score(test_X, test_Y) * 100

 # Random Forest

random_forest = RandomForestClassifier(n_estimators= 100)

random_forest.fit(train_X, train_Y)

predictions_rf = random_forest.predict(test_X)

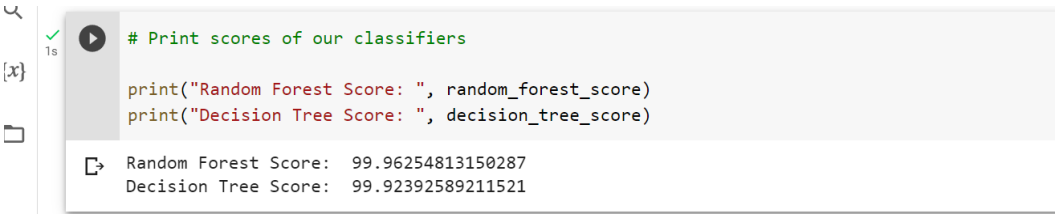
random_forest_score = random_forest.score(test_X, test_Y) * 100

 # Print scores of our classifiers

print("Random Forest Score: ", random_forest_score)

print("Decision Tree Score: ", decision_tree_score)

OutPut:



The Random Forest classifier has slightly an edge over the Decision Tree classifier.

Let's create a function to print the metrics: accuracy, precision, recall, and f1-score.

```
from sklearn.metrics import accuracy_score, precision_score, confusion_matrix, recall_score, f1_score
def metrics(actuals, predictions):
    print("Accuracy: {:.5f}".format(accuracy_score(actuals, predictions)))
    print("Precision: {:.5f}".format(precision_score(actuals, predictions)))
    print("Recall: {:.5f}".format(recall_score(actuals, predictions)))
    print("F1-score: {:.5f}".format(f1_score(actuals, predictions)))
```

Let's visualize the confusion matrix and the evaluation metrics of our Decision Tree model.

 # Plot confusion matrix for Decision Trees

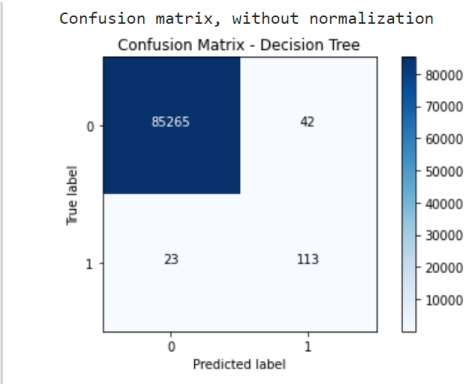
```
confusion_matrix_dt = confusion_matrix(test_Y, predictions_dt.round())
print("Confusion Matrix - Decision Tree")
print(confusion_matrix_dt)
```

OutPut:

```
Confusion Matrix - Decision Tree
[[85265    42]
 [   23   113]]
```

```
plot_confusion_matrix(confusion_matrix_dt, classes=[0, 1], title= "Confusion Matrix - Decision Tree")
```

OutPut:



Let's visualize the confusion matrix and the evaluation metrics of our Random Forest model.

 # Plot confusion matrix for Random Forests

confusion_matrix_rf = confusion_matrix(test_Y, predictions_rf.round())

print("Confusion Matrix - Random Forest")

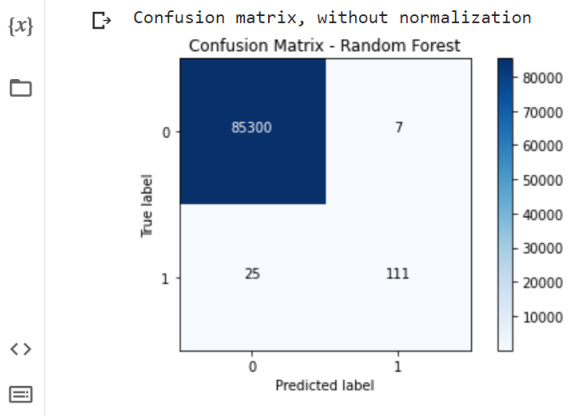
print(confusion_matrix_rf)

OutPut:

```
Confusion Matrix - Random Forest
[[85300     7]
 [   25   111]]
```

plot_confusion_matrix(confusion_matrix_rf, classes=[0, 1], title= "Confusion Matrix - Random Forest")

OutPut:



 # The below function prints the following necesary metrics

```
def metrics(actuals, predictions):
    print("Accuracy: {:.5f}".format(accuracy_score(actuals, predictions)))
    print("Precision: {:.5f}".format(precision_score(actuals, predictions)))
    print("Recall: {:.5f}".format(recall_score(actuals, predictions)))
    print("F1-score: {:.5f}".format(f1_score(actuals, predictions)))
print("Evaluation of Decision Tree Model")
print()
metrics(test_Y, predictions_dt.round())
```

OutPut:

```
Evaluation of Decision Tree Model

Accuracy: 0.99923
Precision: 0.72436
Recall: 0.83088
F1-score: 0.77397
```

```
print("Evaluation of Random Forest Model")
print()
metrics(test_Y, predictions_rf.round())
```

OutPut:

```
Evaluation of Random Forest Model

Accuracy: 0.99963
Precision: 0.93333
Recall: 0.82353
F1-score: 0.87500
```

Address the Class-Imbalance issue

The Random Forest model works better than Decision Trees. But, if we observe our dataset suffers a serious problem of class imbalance. The genuine (not fraud) transactions are more than 99% with the credit card fraud transactions constituting 0.17%.

With such a distribution, if we train our model without taking care of the imbalance issues, it predicts the label with higher importance given to genuine transactions (as there is more data about them) and hence obtains more accuracy.

The class imbalance problem can be solved by various techniques. Oversampling is one of them.

Oversample the minority class is one of the approaches to address the imbalanced datasets. The easiest solution entails doubling examples in the minority class, even though these examples contribute no new data to the model.

Instead, new examples may be generated by replicating existing ones. The Synthetic Minority Oversampling Technique, or SMOTE for short, is a method of data augmentation for the minority class.

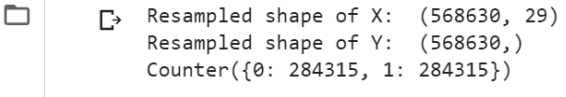
The above SMOTE is present in the imblearn package. Let's import that and resample our data.

In the following code below, we resampled our data and we split it using train_test_split() with a split of 70-30.

 # Performing oversampling on RF and DT

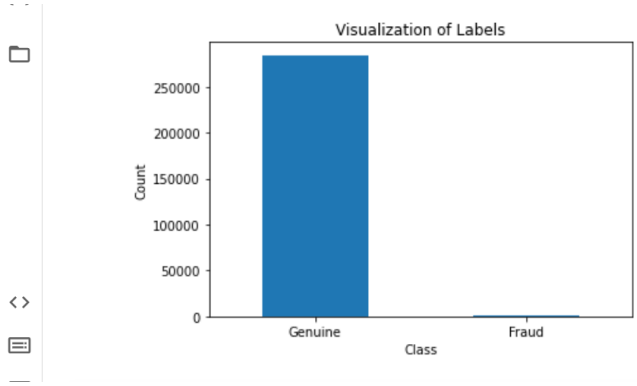
```
from imblearn.over_sampling import SMOTE
X_resampled, Y_resampled = SMOTE().fit_resample(X, Y)
print("Resampled shape of X: ", X_resampled.shape)
print("Resampled shape of Y: ", Y_resampled.shape)
value_counts = Counter(Y_resampled)
print(value_counts)
(train_X, test_X, train_Y, test_Y) = train_test_split(X_resampled, Y_resampled, test_size= 0.3, random_state= 42)
```

OutPut:



```
labels = ["Genuine", "Fraud"]
count_classes = dataframe.value_counts(Y_resampled, sort= True)
count_classes.plot(kind = "bar", rot = 0)
plt.title("Visualization of Labels")
plt.ylabel("Count")
plt.xticks(range(2), labels)
plt.show()
```

OutPut:



As the Random Forest algorithm performed better than the Decision Tree algorithm, we will apply the Random Forest algorithm to our resampled data.

 # Build the Random Forest classifier on the new dataset

```
rf_resampled = RandomForestClassifier(n_estimators = 100)
rf_resampled.fit(train_X, train_Y)

predictions_resampled = rf_resampled.predict(test_X)
random_forest_score_resampled = rf_resampled.score(test_X, test_Y) * 100
```

 # Visualize the confusion matrix

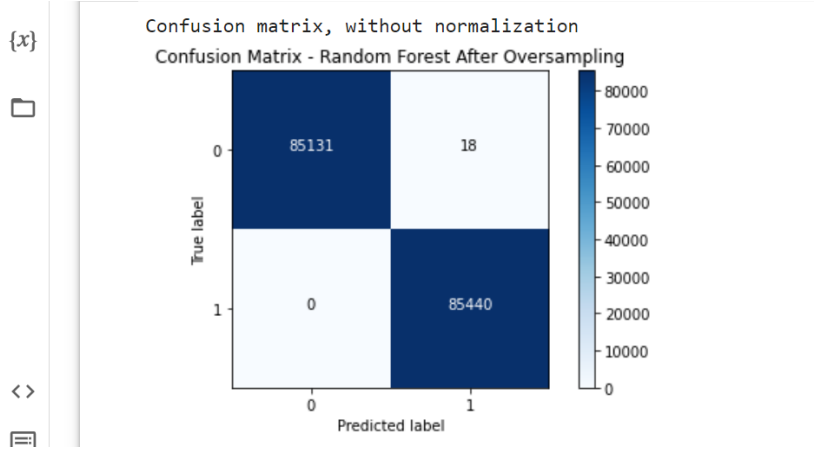
```
cm_resampled = confusion_matrix(test_Y, predictions_resampled.round())
print("Confusion Matrix - Random Forest")
print(cm_resampled)
```

OutPut:

```
Confusion Matrix - Random Forest
[[85131    18]
 [    0 85440]]
```

plot_confusion_matrix(cm_resampled, classes=[0, 1], title= "Confusion Matrix - Random Forest After Oversampling")

OutPut:




```
print("Evaluation of Random Forest Model")
print()
metrics(test_Y, predictions_resampled.round())
```

OutPut:

```
Evaluation of Random Forest Model

Accuracy: 0.99989
Precision: 0.99979
Recall: 1.00000
F1-score: 0.99989
```

Now, it is clearly evident that our model performed much better than our previous Random Forest classifier without oversampling**.**

Summary

In this python machine learning project, we built a binary classifier using the Random Forest algorithm to detect credit card fraud transactions. Through this project, we understood and applied techniques to address the class imbalance issues and achieved an accuracy of more than 99%.

6.4 SOURCE CODE:

indexstyle.css

```
input,
button {
  position: fixed;
  top: 50%;
  left: 50%;
  -webkit-transform: translate(-50%, -300%);
        transform: translate(-50%, -300%);
  display: block;
  width: 70vw;
  opacity: 0;
  pointer-events: none;
  -webkit-transition: all 0.5s cubic-bezier(0.4, 0.25, 0.8, 0.3);
  transition: all 0.5s cubic-bezier(0.4, 0.25, 0.8, 0.3);
}

input {
  padding: .25rem 0;
  border: 0;
  border-bottom: 1px solid #bb1515;
  outline: 0;
  background: transparent;
  color: #fff;
  font-size: 3rem;
  line-height: 4rem;
  letter-spacing: .125rem;
  -webkit-transition: all 0.5s cubic-bezier(0.4, 0.25, 0.8, 0.3);
  transition: all 0.5s cubic-bezier(0.4, 0.25, 0.8, 0.3);
}

input::-moz-selection {
```

```
  background: rgba(187, 21, 21, 0.25);
}

input::selection {
  background: rgba(187, 21, 21, 0.25);
}

button,
.signup-button {
  padding: .25em 0;
  border: 0;
  outline: 0;
  background: #bb1515;
  color: rgba(255, 255, 255, 0.85);
  font-size: 2rem;
  line-height: 3.6rem;
  letter-spacing: .0625rem;
  box-shadow: 0 3px 5px 1px rgba(0, 0, 0, 0.25);
  text-shadow: 0 -2px 0 rgba(0, 0, 0, 0.25), 0 1px 0 rgba(255, 255, 255, 0.2);
}

input:focus,
button:focus {
  opacity: 1;
  -webkit-transform: translate(-50%, -100%);
          transform: translate(-50%, -100%);
  pointer-events: auto;
  -webkit-transition: all 0.4s cubic-bezier(0.1, 0.45, 0.1, 0.85) 0.5s;
  transition: all 0.4s cubic-bezier(0.1, 0.45, 0.1, 0.85) 0.5s;
  z-index: 10;
}

input:focus ~ input,
```

```
input:focus ~ button {
  -webkit-transform: translate(-50%, 500%);
          transform: translate(-50%, 500%);
  -webkit-transition: all .5s ease-in;
  transition: all .5s ease-in;
}

input:focus ~ label .label-text {
  -webkit-transform: translate(-50%, 300%);
          transform: translate(-50%, 300%);
  -webkit-transition: all .5s ease-in;
  transition: all .5s ease-in;
}

input:focus ~ .tip {
  opacity: 1;
}

input:focus ~ .signup-button,
button:focus ~ .signup-button {
  opacity: 0;
}

input:focus + label .label-text {
  opacity: 1;
  -webkit-transform: translate(-50%, -100%);
          transform: translate(-50%, -100%);
  -webkit-transition: all 0.3s cubic-bezier(0.1, 0.45, 0.1, 0.85) 0.4s;
  transition: all 0.3s cubic-bezier(0.1, 0.45, 0.1, 0.85) 0.4s;
}

input:focus + label .nav-dot:before {
  background: #a41212;
```

```
  box-shadow: 0 0 0 0.15rem #111, 0 0 0.05rem 0.26rem #bb1515;
}

.tip {
  position: fixed;
  top: 57%;
  left: 50%;
  -webkit-transform: translate(-50%, -50%);
          transform: translate(-50%, -50%);
  width: 70%;
  opacity: 0;
  color: #fff;
  font-size: .875rem;
  font-weight: 300;
  letter-spacing: .125rem;
  text-transform: uppercase;
  text-align: right;
  -webkit-transition: opacity .25s .5s;
  transition: opacity .25s .5s;
}
.signup-button,
.signup-button-trigger {
  position: fixed;
  top: 50%;
  left: 50%;
  -webkit-transform: translate(-50%, -100%);
          transform: translate(-50%, -100%);
  width: 70vw;
  padding: .25rem 0;
  line-height: 3.6rem;
  text-align: center;
  pointer-events: none;
  cursor: pointer;
```

```
  -webkit-transition: opacity .4s .3s;
  transition: opacity .4s .3s;
}

.signup-button-trigger {
  opacity: 0;
  pointer-events: auto;
}

.label-text {
  position: fixed;
  top: calc(50% - 4rem);
  left: 50%;
  -webkit-transform: translate(-50%, -300%);
          transform: translate(-50%, -300%);
  width: 70vw;
  padding: 3.125rem 0 1.5rem;
  text-transform: uppercase;
  color: #fff;
  opacity: 0;
  font-size: 1.125rem;
  font-weight: 300;
  letter-spacing: .125rem;
  pointer-events: none;
  -webkit-transition: all 0.4s cubic-bezier(0.4, 0.25, 0.8, 0.3) 0.05s;
  transition: all 0.4s cubic-bezier(0.4, 0.25, 0.8, 0.3) 0.05s;
}

.nav-dot {
  cursor: pointer;
  position: fixed;
  padding: .625rem 1.25rem .625rem .625rem;
  top: 52%;
```

```
  right: 1.25rem;
}
.nav-dot:before {
  content: '';
  display: inline-block;
  border-radius: 50%;
  width: .375rem;
  height: .375rem;
  margin-right: .625rem;
  position: fixed;
  background-color: #16272f;
  border: 0;
  -webkit-transition: all 0.25s;
  transition: all 0.25s;
}
.nav-dot:hover:before {
  width: .625rem;
  height: .625rem;
  margin-top: -.125rem;
  margin-left: -.125rem;
  background-color: #a41212;
}

label[for="input-1"] .nav-dot {
  margin-top: -125px;
}

label[for="input-2"] .nav-dot {
  margin-top: -100px;
}

label[for="input-3"] .nav-dot {
  margin-top: -75px;
```

```
}

label[for="input-4"] .nav-dot {
  margin-top: -50px;
}

label[for="input-5"] .nav-dot {
  margin-top: -25px;
}
label[for="input-6"] .nav-dot {
  margin-top: 0px;
}
label[for="input-7"] .nav-dot {
  margin-top: 25px;
}

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

html, body {
  width: 100%;
  height: 100%;
  background-image: -webkit-gradient(linear, left top, right bottom, from(#111E25), to(#111));
  background-image: linear-gradient(to bottom right, #111E25 0%, #111 100%);
  font-family: 'Lato', sans-serif;
}
form {
  width: 100%;
  height: 100%;
  overflow: hidden;
```

```
}
```

indexstyle.scss

```
$background: #111E25;
$dark: #111;
$primary: #bb1515;

input,
button {
  position: fixed;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -300%);
  display: block;
  width: 70vw;
  opacity: 0;
  pointer-events: none;
  transition: all .5s cubic-bezier(.4, .25, .8, .3);
}

input {
  padding: .25rem 0;
  border: 0;
  border-bottom: 1px solid $primary;
  outline: 0;
  background: transparent;
  color: #fff;
  font-size: 3rem;
  line-height: 4rem;
  letter-spacing: .125rem;
  transition: all .5s cubic-bezier(.4, .25, .8, .3);
}
input::selection {
  background: rgba($primary, 0.25);
```

```
}

button,
.signup-button {
  padding: .25em 0;
  border: 0;
  outline: 0;
  background: $primary;
  color: rgba(#fff, 0.85);
  font-size: 2rem;
  line-height: 3.6rem;
  letter-spacing: .0625rem;
  box-shadow: 0 3px 5px 1px rgba(#000, 0.25);
  text-shadow: 0 -2px 0 rgba(#000, 0.25), 0 1px 0 rgba(#fff, 0.2);
}

input:focus,
button:focus {
  opacity: 1;
  transform: translate(-50%, -100%);
  pointer-events: auto;
  transition: all .4s cubic-bezier(.1, .45, .1, .85) .5s;
  z-index: 10;
}

input:focus ~ input,
input:focus ~ button {
  transform: translate(-50%, 500%);
  transition: all .5s ease-in;
}
input:focus ~ label .label-text {
  transform: translate(-50%, 300%);
  transition: all .5s ease-in;
```

```
}
input:focus ~ .tip {
  opacity: 1;
}
input:focus ~ .signup-button,
button:focus ~ .signup-button {
  opacity: 0;
}

input:focus + label .label-text {
  opacity: 1;
  transform: translate(-50%, -100%);
  transition: all .3s cubic-bezier(.1, .45, .1, .85) .4s;
}
input:focus + label .nav-dot:before {
  background: darken($primary, 5%);
  box-shadow: 0 0 0 .15rem $dark, 0 0 .05rem .26rem $primary;
}

.tip {
  position: fixed;
  top: 57%;
  left: 50%;
  transform: translate(-50%, -50%);
  width: 70%;
  opacity: 0;
  color: #fff;
  font-size: .875rem;
  font-weight: 300;
  letter-spacing: .125rem;
  text-transform: uppercase;
  text-align: right;
  transition: opacity .25s .5s;
```

```
}

.signup-button,
.signup-button-trigger {
  position: fixed;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -100%);
  width: 70vw;
  padding: .25rem 0;
  line-height: 3.6rem;
  text-align: center;
  pointer-events: none;
  cursor: pointer;
  transition: opacity .4s .3s;
}

.signup-button-trigger {
  opacity: 0;
  pointer-events: auto;
}

.label-text {
  position: fixed;
  top: calc(50% - 4rem);
  left: 50%;
  transform: translate(-50%, -300%);
  width: 70vw;
  padding: 3.125rem 0 1.5rem;
  text-transform: uppercase;
  color: #fff;
  opacity: 0;
  font-size: 1.125rem;
```

```
  font-weight: 300;
  letter-spacing: .125rem;
  pointer-events: none;
  transition: all .4s cubic-bezier(.4, .25, .8, .3) .05s;
}

.nav-dot {
  cursor: pointer;
  position: fixed;
  padding: .625rem 1.25rem .625rem .625rem;
  top: 52%;
  right: 1.25rem;
  &:before {
    content: '';
    display: inline-block;
    border-radius: 50%;
    width: .375rem;
    height: .375rem;
    margin-right: .625rem;
    position: fixed;
    background-color: lighten($background, 3%);
    border: 0;
    transition: all 0.25s;
  }
  &:hover:before {
    width: .625rem;
    height: .625rem;
    margin-top: -.125rem;
    margin-left: -.125rem;
    background-color: darken($primary, 5%);
  }
}
```

```
@for $i from 1 through 5 {
  label[for="input-#{$i}"] .nav-dot {
      margin-top: -150px + (25 * $i);
  }
}

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

html, body {
  width: 100%;
  height: 100%;
  background-image: linear-gradient(to bottom right, $background 0%, $dark 100%);
  font-family: 'Lato', sans-serif;
}

form {
  width: 100%;
  height: 100%;
  overflow: hidden;
}
```

resultstyle.css

```
@import url(//fonts.googleapis.com/css?family=Lato:300:400);

body {
  margin:0;
}

h1 {
  font-family: 'Lato', sans-serif;
```

```
  font-weight:300;
  letter-spacing: 2px;
  font-size:48px;
}
p {
  font-family: 'Lato', sans-serif;
  letter-spacing: 1px;
  font-size:14px;
  color: #333333;
}

.header {
  position:relative;
  text-align:center;
  background: linear-gradient(60deg, rgba(84,58,183,1) 0%, rgba(0,172,193,1) 100%);
  color:white;
  background-image: -webkit-gradient(linear, left top, right bottom, from(#111E25), to(#111));
  background-image: linear-gradient(to bottom right, #111E25 0%, #111 100%);
}
.logo {
  width:50px;
  fill:white;
  padding-right:15px;
  display:inline-block;
  vertical-align: middle;
}

.inner-header {
  height:65vh;
  width:100%;
  margin: 0;
  padding: 0;
```

```
}

.flex { /*Flexbox for containers*/
  display: flex;
  justify-content: center;
  align-items: center;
  text-align: center;
}

.waves {
  position:relative;
  width: 100%;
  height:15vh;
  margin-bottom:-7px; /*Fix for safari gap*/
  min-height:100px;
  max-height:150px;
}

.content {
  position:relative;
  height:20vh;
  text-align:center;
  background-color: #bb1515;

}
/* Animation */

.parallax > use {
  animation: move-forever 25s cubic-bezier(.55,.5,.45,.5)     infinite;
}
.parallax > use:nth-child(1) {
  animation-delay: -2s;
  animation-duration: 7s;
```

```
}
.parallax > use:nth-child(2) {
  animation-delay: -3s;
  animation-duration: 10s;
}
.parallax > use:nth-child(3) {
  animation-delay: -4s;
  animation-duration: 13s;
}
.parallax > use:nth-child(4) {
  animation-delay: -5s;
  animation-duration: 20s;
}
@keyframes move-forever {
  0% {
   transform: translate3d(-90px,0,0);
  }
  100% {
    transform: translate3d(85px,0,0);
  }
}
/*Shrinking for mobile*/
@media (max-width: 768px) {
  .waves {
    height:40px;
    min-height:40px;
  }
  .content {
    height:30vh;
  }
  h1 {
    font-size:24px;
  }
```

}

index.html

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="{{ url_for('static', filename='css/indexstyle.css') }}">
<title>ML API</title>
</head>
<body>
<form action="{{ url_for('predict')}}" method="POST">
  <input id="input-1" type="text" placeholder="Enter time" name ="time" required autofocus />
  <label for="input-1">
    <span class="label-text">TIME</span>
    <span class="nav-dot"></span>
    <div class="signup-button-trigger">Credit Card Fraud Prediction</div>
  </label>
  <input id="input-2" type="text" placeholder="Enter Amount" name="amount" required />
  <label for="input-2">
    <span class="label-text">AMOUNT</span>
    <span class="nav-dot"></span>
  </label>
  <input id="input-3" type="text" placeholder="Enter Transaction Method" name="tm" required />
  <label for="input-3">
    <span class="label-text">Transaction Method</span>
    <span class="nav-dot"></span>
  </label>
  <input id="input-4" type="text" placeholder="Transaction id" name="ti" required />
  <label for="input-4">
    <span class="label-text">Transaction id</span>
    <span class="nav-dot"></span>
  </label>
  <input id="input-5" type="text" placeholder="Enter Type Of card" name="ct" required />
  <label for="input-5">
```

```
  <span class="label-text">Card Type</span>
  <span class="nav-dot"></span>
 </label>
 <input id="input-6" type="text" placeholder="Enter Location" name="location" required />
 <label for="input-6">
  <span class="label-text">Enter Location</span>
  <span class="nav-dot"></span>
 </label>
 <input id="input-7" type="text" placeholder="Enter Bank" name="em" required />
 <label for="input-7">
  <span class="label-text">Enter Bank</span>
  <span class="nav-dot"></span>
 </label>
 <button type="submit">Predict</button>
 <p class="tip">Press Tab</p>
 <div class="signup-button">Credit card Fraud Detection</div>
</form>
</body>
</html>
```

Result.html

```
<!--Hey! This is the original version
of Simple CSS Waves-->
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
<link rel="stylesheet" href="{{ url_for('static', filename='css/resultstyle.css') }}">

</head>

<!--Hey! This is the original version
of Simple CSS Waves-->
```

```
<body>
<div class="header">

<!--Content before waves-->
<div class="inner-header flex">
<!--Just the logo.. Don't mind this-->
<h1>{{ prediction }}</h1>
</div>

<!--Waves Container-->
<div>
<svg class="waves" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink"
viewBox="0 24 150 28" preserveAspectRatio="none" shape-rendering="auto">
<defs>
<path id="gentle-wave" d="M-160 44c30 0 58-18 88-18s 58 18 88 18 58-18 88-18 58 18 88 18 v44h-352z" />
</defs>
<g class="parallax">
<use xlink:href="#gentle-wave" x="48" y="0" fill="#bb1515" />
<use xlink:href="#gentle-wave" x="48" y="3" fill="#bb1515" />
<use xlink:href="#gentle-wave" x="48" y="5" fill="rgba(255,255,255,0.3)" />
<use xlink:href="#gentle-wave" x="48" y="7" fill="#bb1515" />
</g>
</svg>
</div>
<!--Waves end-->

</div>
<!--Header ends-->

<!--Content starts-->
<div class="content flex">
```

```
</div>
<!--Content ends-->
</body>
```

app.py

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__)
# prediction function
def ValuePredictor(to_predict_list):
    to_predict = np.array(to_predict_list).reshape(1, 7)
    loaded_model = pickle.load(open("model.pkl", "rb"))
    result = loaded_model.predict(to_predict)
    return result[0]

@app.route('/')
def home():
    return render_template("index.html")

@app.route('/predict',methods=['POST','GET'])
def predict():
    if request.method == 'POST':
        to_predict_list = request.form.to_dict()
        to_predict_list = list(to_predict_list.values())
        to_predict_list = list(map(float, to_predict_list))
        result = ValuePredictor(to_predict_list)
    if int(result)== 1:
        prediction ='Given transaction is fradulent'
    else:
        prediction ='Given transaction is NOT fradulent'
    return render_template("result.html", prediction = prediction)
```

```
if __name__ == "__main__":
    app.run(debug=True)
```

project_all_model.py

```
# -*- coding: utf-8 -*-
"""Project_Final.ipynb

Automatically generated by Colaboratory.

# Data Preprocessing and Visualisation
"""

# Commented out IPython magic to ensure Python compatibility.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# %matplotlib inline

# Commented out IPython magic to ensure Python compatibility.
import sklearn
import random

from sklearn.utils import shuffle
# %matplotlib inline

from zipfile import ZipFile
with ZipFile('creditcardfraud.zip','r') as zip:
```

```
  zip.printdir()
  zip.extractall()

d=pd.read_csv('creditcard.csv')

sns.distplot(data['Amount'])

sns.distplot(data['Time'])

data.hist(figsize=(20,20))
plt.show()

sns.jointplot(x= 'Time', y= 'Amount', data= d)

class0 = d[d['Class']==0]

len(class0)

class1 = d[d['Class']==1]

len(class1)

class0
temp = shuffle(class0)

d1 = temp.iloc[:2000,:]

d1

frames = [d1, class1]
df_temp = pd.concat(frames)

df_temp.info()
```

```
df= shuffle(df_temp)

df.to_csv('creditcardsampling.csv')

sns.countplot('Class', data=df)

"""# SMOTE"""

!pip install --user imblearn

import imblearn

from imblearn.over_sampling import  SMOTE
oversample=SMOTE()
X=df.iloc[ : ,:-1]
Y=df.iloc[: , -1]
X,Y=oversample.fit_resample(X,Y)

X=pd.DataFrame(X)
X.shape

Y=pd.DataFrame(Y)
Y.head()

names=['Time','V1','V2','V3','V4','V5','V6','V7','V8','V9','V10','V11','V12','V13','V14','V15','V16','V1
7','V18','V19','V20','V21','V22','V23','V24','V25','V26','V27','V28','Amount','Class']

data=pd.concat([X,Y],axis=1)

d=data.values

data=pd.DataFrame(d,columns=names)
```

```
sns.countplot('Class', data=data)

data.describe()

data.info()

plt.figure(figsize=(12,10))
sns.heatmap(data.corr())

!pip install --user lightgbm

!pip install --user utils

import math
import sklearn.preprocessing

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score , classification_report, confusion_matrix, precision_recall_curve, f1_score, auc

X_train, X_test, y_train, y_test = train_test_split(data.drop('Class', axis=1), data['Class'], test_size=0.3, random_state=42)

"""# Feature Scaling"""

cols= ['V22', 'V24', 'V25', 'V26', 'V27', 'V28']

scaler = StandardScaler()

frames= ['Time', 'Amount']
```

```
x= data[frames]

d_temp = data.drop(frames, axis=1)

temp_col=scaler.fit_transform(x)

scaled_col = pd.DataFrame(temp_col, columns=frames)

scaled_col.head()

d_scaled = pd.concat([scaled_col, d_temp], axis =1)

d_scaled.head()

y = data['Class']

d_scaled.head()

"""# Dimensionality Reduction"""

from sklearn.decomposition import PCA

pca = PCA(n_components=7)

X_temp_reduced = pca.fit_transform(d_scaled)

pca.explained_variance_ratio_

pca.explained_variance_

names=['Time','Amount','Transaction Method','Transaction Id','Location','Type of Card','Bank']

X_reduced= pd.DataFrame(X_temp_reduced,columns=names)
```

```
X_reduced.head()

Y=d_scaled['Class']

new_data=pd.concat([X_reduced,Y],axis=1)
new_data.head()
new_data.shape

new_data.to_csv('finaldata.csv')

X_train, X_test, y_train, y_test= train_test_split(X_reduced, d_scaled['Class'], test_size = 0.30, random_state = 42)

X_train.shape, X_test.shape

"""# Logistic Regression"""

from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(X_train,y_train)
y_pred_lr=lr.predict(X_test)
y_pred_lr

from sklearn.metrics import classification_report,confusion_matrix
print(confusion_matrix(y_test,y_pred_lr))

#Hyperparamter tuning
from sklearn.model_selection import GridSearchCV
lr_model = LogisticRegression()
lr_params = {'penalty': ['l1', 'l2'],'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000]}
grid_lr= GridSearchCV(lr_model, param_grid = lr_params)
grid_lr.fit(X_train, y_train)
```

```
grid_lr.best_params_

y_pred_lr3=grid_lr.predict(X_test)
print(classification_report(y_test,y_pred_lr3))



"""# Support Vector Machine"""

from sklearn.svm import SVC
svc=SVC(kernel='rbf')
svc.fit(X_train,y_train)
y_pred_svc=svc.predict(X_test)
y_pred_svc

print(classification_report(y_test,y_pred_svc))

print(confusion_matrix(y_test,y_pred_svc))

from sklearn.model_selection import GridSearchCV
parameters = [ {'C': [1, 10, 100, 1000], 'kernel': ['rbf'], 'gamma': [0.1, 1, 0.01, 0.0001 ,0.001]}]
grid_search = GridSearchCV(estimator = svc,
                           param_grid = parameters,
                           scoring = 'accuracy',
                           n_jobs = -1)
grid_search = grid_search.fit(X_train, y_train)
best_accuracy = grid_search.best_score_
best_parameters = grid_search.best_params_
print("Best Accuracy: {:.2f} %".format(best_accuracy*100))
print("Best Parameters:", best_parameters)

svc_param=SVC(kernel='rbf',gamma=0.01,C=100)
svc_param.fit(X_train,y_train)
```

```
y_pred_svc2=svc_param.predict(X_test)
print(classification_report(y_test,y_pred_svc2))



"""# Decision Tree"""

from sklearn.tree import DecisionTreeClassifier
dtree=DecisionTreeClassifier()
dtree.fit(X_train,y_train)
y_pred_dtree=dtree.predict(X_test)
print(classification_report(y_test,y_pred_dtree))

print(confusion_matrix(y_test,y_pred_dtree))

d_tree_param=DecisionTreeClassifier()
tree_parameters={'criterion':['gini','entropy'],'max_depth':list(range(2,4,1)),
                 'min_samples_leaf':list(range(5,7,1))}
grid_tree=GridSearchCV(d_tree_param,tree_parameters)
grid_tree.fit(X_train,y_train)

y_pred_dtree2=grid_tree.predict(X_test)

print(classification_report(y_test,y_pred_dtree2))



"""# Random Forest"""

from sklearn.ensemble import RandomForestClassifier
randomforest=RandomForestClassifier(n_estimators=5)
randomforest.fit(X_train,y_train)
y_pred_rf=randomforest.predict(X_test)
```

```
print(confusion_matrix(y_test,y_pred_rf))

print(classification_report(y_test,y_pred_rf))



"""# K Nearest Neighbors"""

from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train,y_train)
y_pred_knn=knn.predict(X_test)
y_pred_knn

print(classification_report(y_test,y_pred_knn))

print(confusion_matrix(y_test,y_pred_knn))

knn_param=KNeighborsClassifier()
knn_params={"n_neighbors": list(range(2,5,1)), 'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute']}
grid_knn=GridSearchCV(knn_param,param_grid=knn_params)
grid_knn.fit(X_train,y_train)
grid_knn.best_params_

knn = KNeighborsClassifier(n_neighbors=2)

knn.fit(X_train,y_train)
pred_knn2 = knn.predict(X_test)

print('WITH K=3')
print('\n')
print(confusion_matrix(y_test,pred_knn2))
print('\n')
```

```
print(classification_report(y_test,pred_knn2))

"""# XGBoost"""

from xgboost import XGBClassifier
xgb=XGBClassifier()
xgb.fit(X_train,y_train)
y_pred_xg=xgb.predict(X_test)
print(classification_report(y_test,y_pred_xg))

"""# LGB"""

import lightgbm as lgb

lgb_train = lgb.Dataset(X_train, y_train, free_raw_data= False)

lgb_test = lgb.Dataset(X_test, y_test, reference=lgb_train, free_raw_data= False)

parameters = {'num_leaves': 2**8,
              'learning_rate': 0.1,
              'is_unbalance': True,
              'min_split_gain': 0.1,
              'min_child_weight': 1,
              'reg_lambda': 1,
              'subsample': 1,
              'objective':'binary',
              #'device': 'gpu', # comment this line if you are not using GPU
              'task': 'train'
              }
num_rounds = 300

lgb_train = lgb.Dataset(X_train, y_train)
```

```
lgb_test = lgb.Dataset(X_test, y_test)

clf = lgb.train(parameters, lgb_train, num_boost_round=num_rounds)

y_prob = clf.predict(X_test)
y_pred = sklearn.preprocessing.binarize(np.reshape(y_prob, (-1,1)), threshold= 0.5)

accuracy_score(y_test, y_pred)

print(classification_report(y_test,y_pred))



"""# ROC"""

from sklearn.metrics import roc_curve,roc_auc_score
lg_fpr,lg_tpr,lg_threshold=roc_curve(y_test,y_pred_lr3)
svc_fpr,svc_tpr,svc_threshold=roc_curve(y_test,y_pred_svc2)
dtree_fpr,dtree_tpr,dtree_threshold=roc_curve(y_test,y_pred_dtree2)
rf_fpr,rf_tpr,rf_threshold=roc_curve(y_test,y_pred_rf)
knn_fpr,knn_tpr,rf_threshold=roc_curve(y_test,pred_knn2)
xg_fpr,xg_tpr,xg_threshold=roc_curve(y_test,y_pred_xg)
lgb_fpr,lgb_tpr,lgb_threshold=roc_curve(y_test,y_pred)

plt.figure(figsize=(15,10))
plt.title("Roc Curve")
plt.plot(lg_fpr,lg_tpr, label='Logistic Regression Classifier Score: {:.4f}'.format(roc_auc_score(y_test, y_pred_lr3)))
plt.plot(knn_fpr,knn_tpr, label='KNears Neighbors Classifier Score: {:.4f}'.format(roc_auc_score(y_test, pred_knn2)))
plt.plot(svc_fpr, svc_tpr, label='Support Vector Classifier Score: {:.4f}'.format(roc_auc_score(y_test, y_pred_svc2)))
```

```
plt.plot(dtree_fpr, dtree_tpr, label='Decision Tree Classifier Score: {:.4f}'.format(roc_auc_score(y_test, y_pred_dtree2)))
plt.plot(rf_fpr,rf_tpr, label='Random Forest Classifier Score: {:.4f}'.format(roc_auc_score(y_test, y_pred_rf)))
plt.plot(xg_fpr,xg_tpr, label='XGBoost Classifier Score: {:.4f}'.format(roc_auc_score(y_test, y_pred_xg)))
plt.plot(lgb_fpr,lgb_tpr, label='Light Gradient Boosting Classifier Score: {:.4f}'.format(roc_auc_score(y_test, y_pred)))
plt.xlabel('False Positive Rate', fontsize=16)
plt.ylabel('True Positive Rate', fontsize=16)
plt.legend()
plt.show()
```

project_final.py

```
# Commented out IPython magic to ensure Python compatibility.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# %matplotlib inline

# Commented out IPython magic to ensure Python compatibility.
import sklearn
import random
from sklearn.utils import shuffle
# %matplotlib inline
data=pd.read_csv('creditcard.csv')
sns.distplot(data['Amount'])
sns.distplot(data['Time'])
data.hist(figsize=(20,20))
plt.show()
sns.jointplot(x= 'Time', y= 'Amount', data= d)
d=data
```

```
class0 = d[d['Class']==0]
len(class0)
class1 = d[d['Class']==1]
len(class1)
class0
temp = shuffle(class0)
d1 = temp.iloc[:2000,:]
d1
frames = [d1, class1]
df_temp = pd.concat(frames)
df_temp.info()
df= shuffle(df_temp)
df.to_csv('creditcardsampling.csv')
sns.countplot('Class', data=df)

"""# SMOTE"""
#!pip install --user imblearn
import imblearn
from imblearn.over_sampling import  SMOTE
oversample=SMOTE()
X=df.iloc[ : ,:-1]
Y=df.iloc[: , -1]
X,Y=oversample.fit_resample(X,Y)
X=pd.DataFrame(X)
X.shape
Y=pd.DataFrame(Y)
Y.head()
names=['Time','V1','V2','V3','V4','V5','V6','V7','V8','V9','V10','V11','V12','V13','V14','V15','V16','V1
7','V18','V19','V20','V21','V22','V23','V24','V25','V26','V27','V28','Amount','Class']
data=pd.concat([X,Y],axis=1)
d=data.values
data=pd.DataFrame(d,columns=names)
sns.countplot('Class', data=data)
```

```
data.describe()
data.info()
plt.figure(figsize=(12,10))
sns.heatmap(data.corr())
import math
import sklearn.preprocessing
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score , classification_report, confusion_matrix, precision_recall_curve, f1_score, auc
X_train, X_test, y_train, y_test = train_test_split(data.drop('Class', axis=1), data['Class'], test_size=0.3, random_state=42)

"""# Feature Scaling"""
cols= ['V22', 'V24', 'V25', 'V26', 'V27', 'V28']
scaler = StandardScaler()
frames= ['Time', 'Amount']
x= data[frames]
d_temp = data.drop(frames, axis=1)
temp_col=scaler.fit_transform(x)
scaled_col = pd.DataFrame(temp_col, columns=frames)
scaled_col.head()
d_scaled = pd.concat([scaled_col, d_temp], axis =1)
d_scaled.head()
y = data['Class']
d_scaled.head()

"""# Dimensionality Reduction"""

from sklearn.decomposition import PCA
pca = PCA(n_components=7)
X_temp_reduced = pca.fit_transform(d_scaled)
pca.explained_variance_ratio_
```

```
pca.explained_variance_
names=['Time','Amount','Transaction Method','Transaction Id','Location','Type of Card','Bank']
X_reduced= pd.DataFrame(X_temp_reduced,columns=names)
X_reduced.head()
Y=d_scaled['Class']
new_data=pd.concat([X_reduced,Y],axis=1)
new_data.head()
new_data.shape
new_data.to_csv('finaldata.csv')
X_train, X_test, y_train, y_test= train_test_split(X_reduced, d_scaled['Class'], test_size = 0.30, random_state = 42)
X_train.shape, X_test.shape
from sklearn.metrics import classification_report,confusion_matrix

"""# Support Vector Machine"""

from sklearn.svm import SVC
svc=SVC(kernel='rbf',probability=True)
svc.fit(X_train,y_train)
y_pred_svc=svc.predict(X_test)
y_pred_svc
type(X_test)
X_test.to_csv('testing.csv')
from sklearn.model_selection import GridSearchCV
parameters = [ {'C': [1, 10, 100, 1000], 'kernel': ['rbf'], 'gamma': [0.1, 1, 0.01, 0.0001 ,0.001]}]
grid_search = GridSearchCV(estimator = svc,
                           param_grid = parameters,
                           scoring = 'accuracy',
                           n_jobs = -1)
grid_search = grid_search.fit(X_train, y_train)
best_accuracy = grid_search.best_score_
best_parameters = grid_search.best_params_
print("Best Accuracy: {:.2f} %".format(best_accuracy*100))
```

```
print("Best Parameters:", best_parameters)

svc_param=SVC(kernel='rbf',gamma=0.01,C=100,probability=True)
svc_param.fit(X_train,y_train)

import pickle
# Saving model to disk
pickle.dump(svc_param, open('model.pkl','wb'))
model=pickle.load(open('model.pkl','rb'))
```

7. SCREEN SHOTS

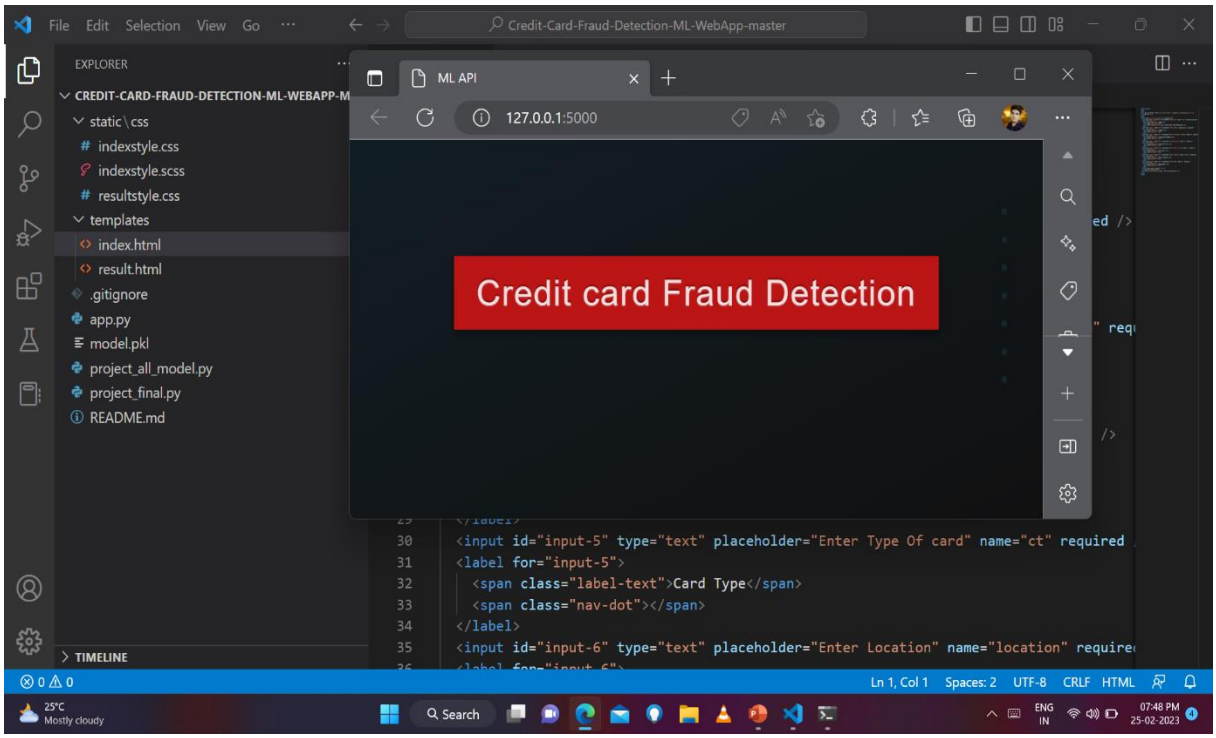


Figure 7.1: Home Page

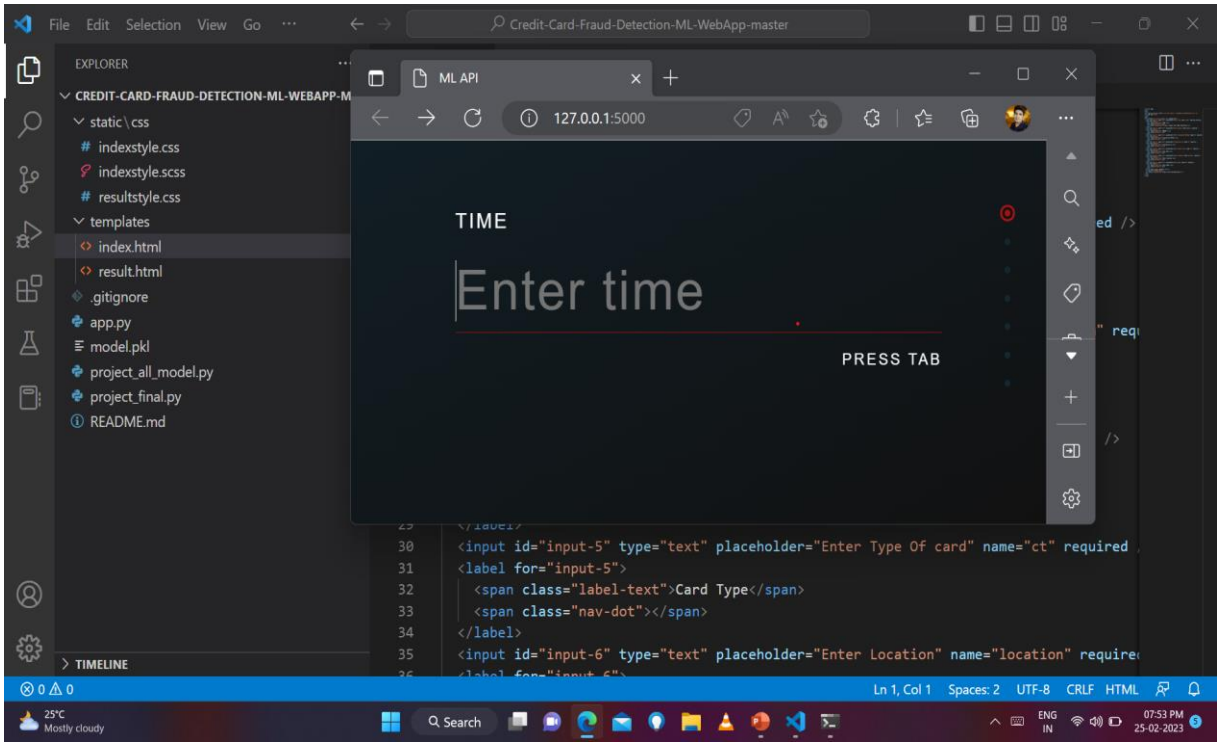


Figure 7.2: Interface to Ready to Enter Time

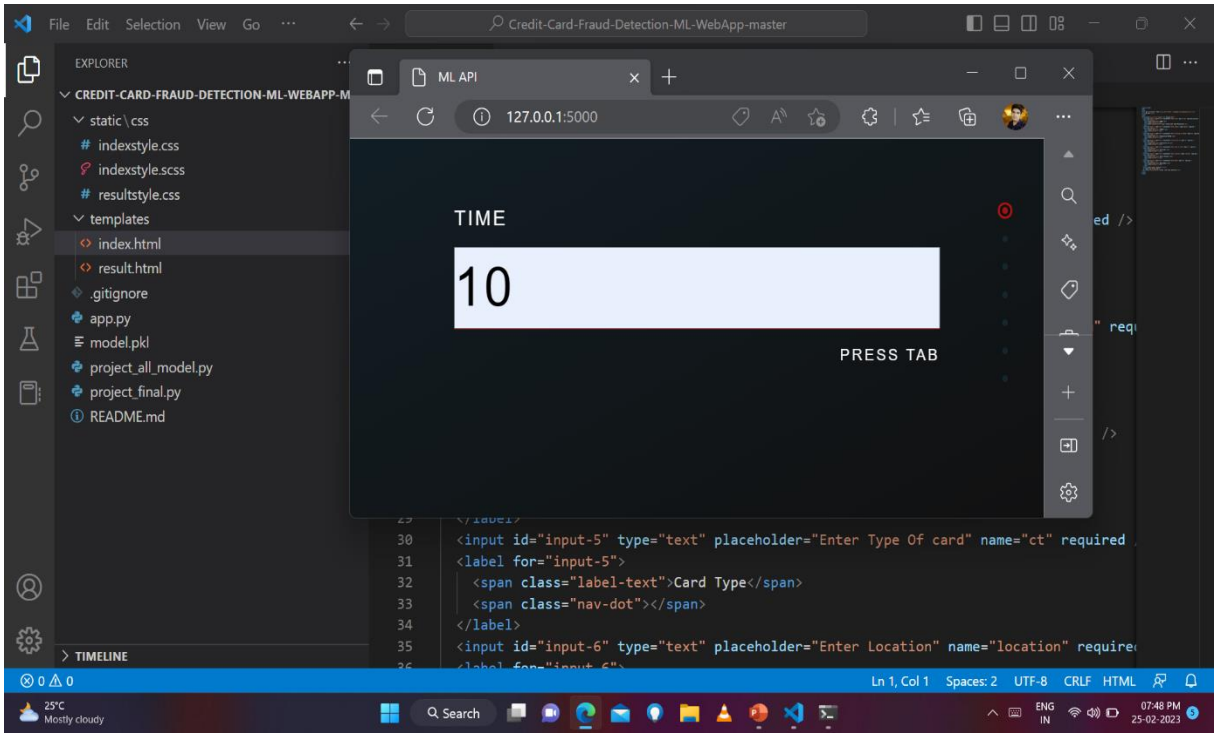


Figure 7.3: Interface to Enter Time

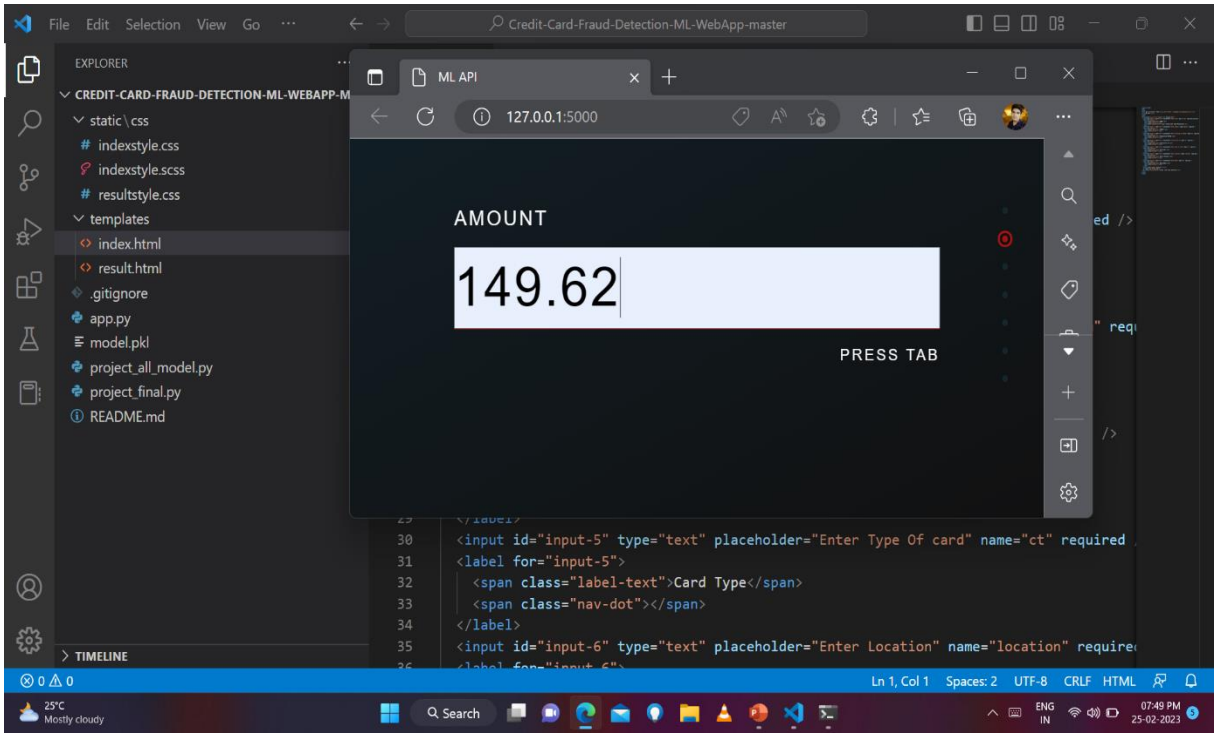


Figure 7.4: Interface to Amount

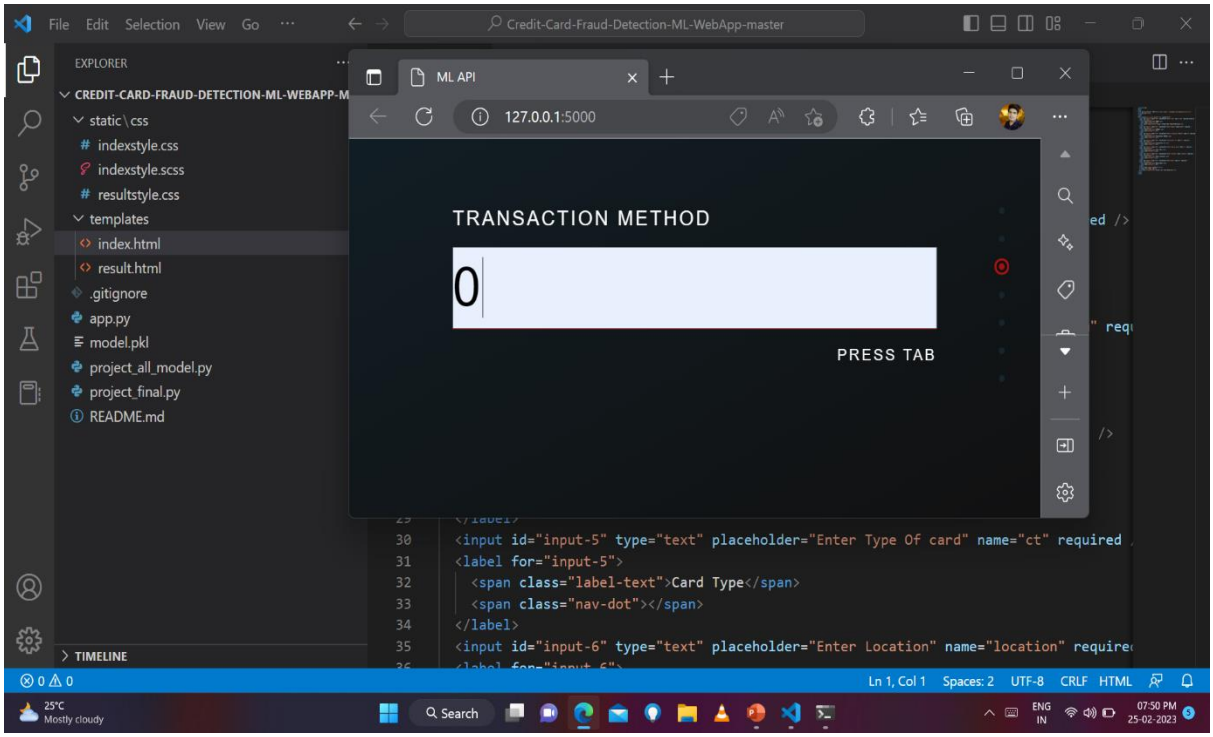


Figure 7.5: Interface to Enter Transaction method

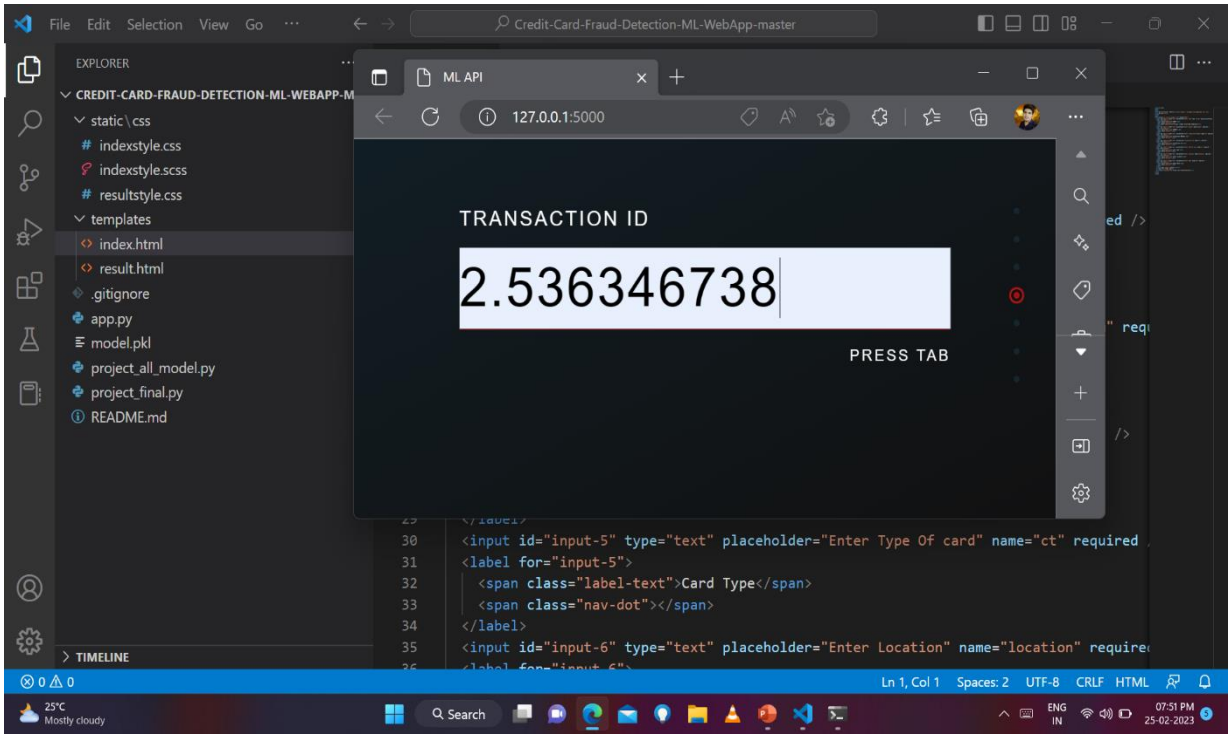


Figure 7.6: Interface to Transaction ID

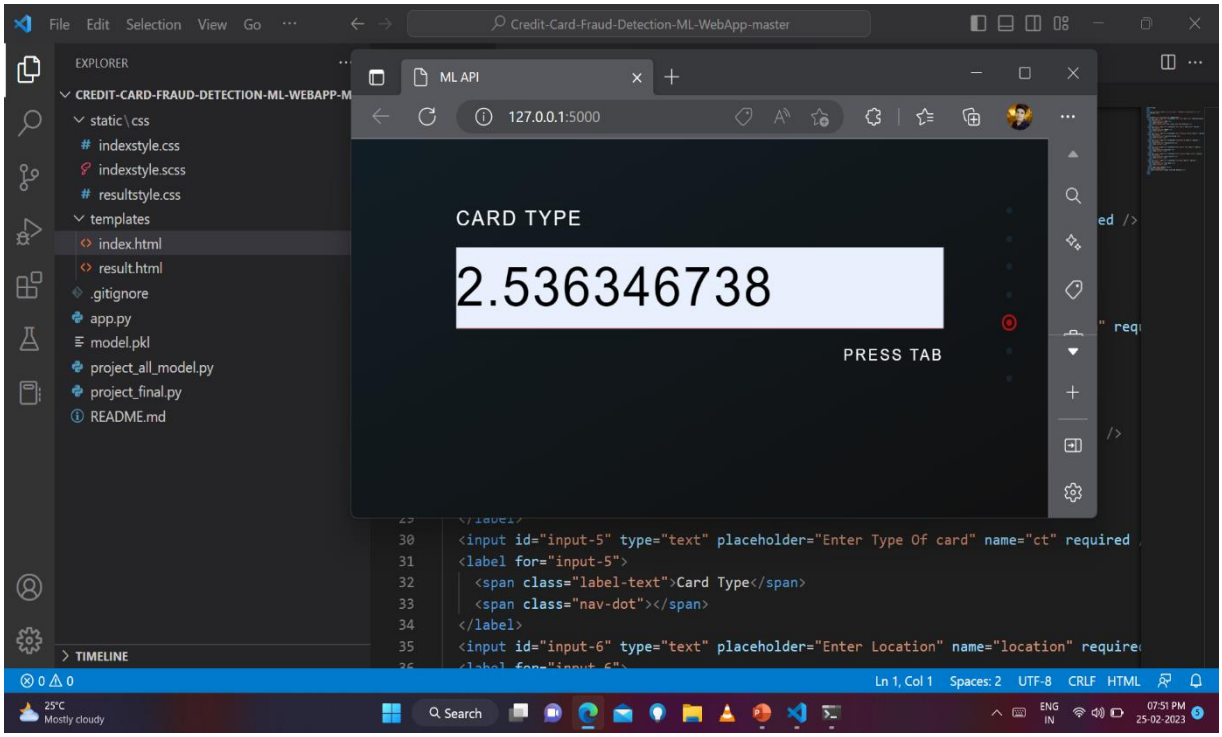


Figure 7.7: Interface to Enter Card Type

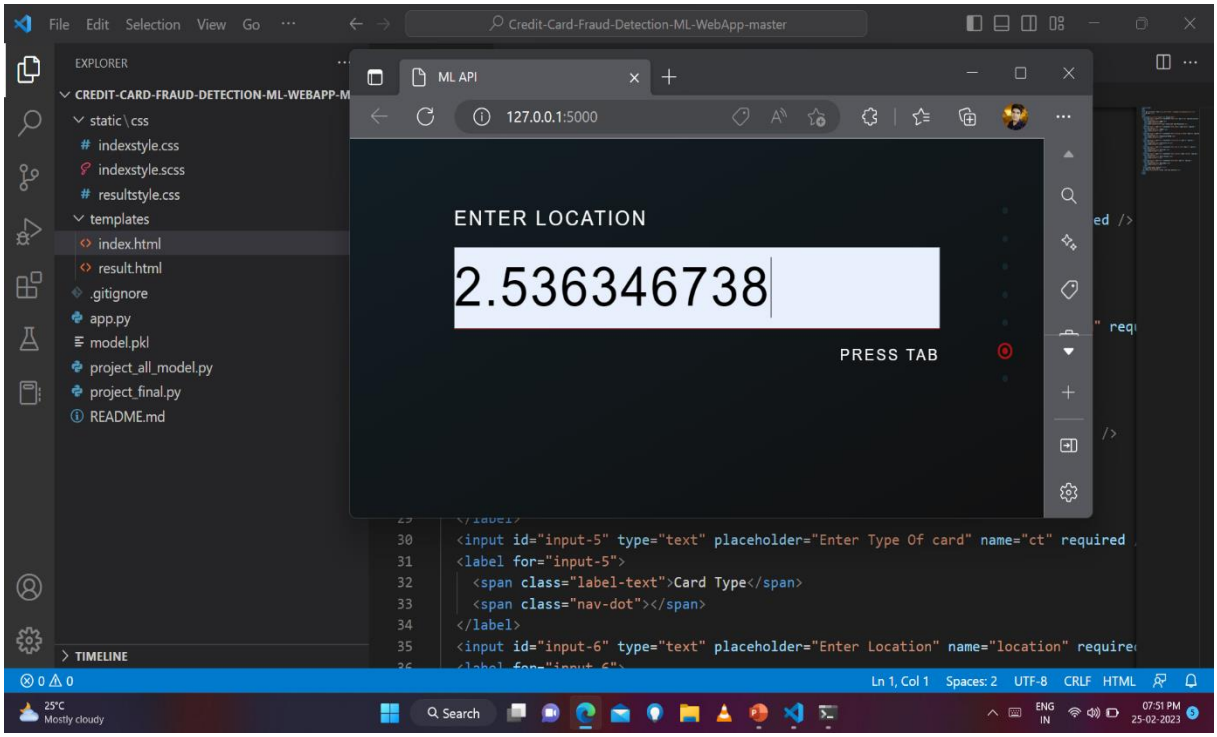


Figure 7.8: Interface to Enter Location

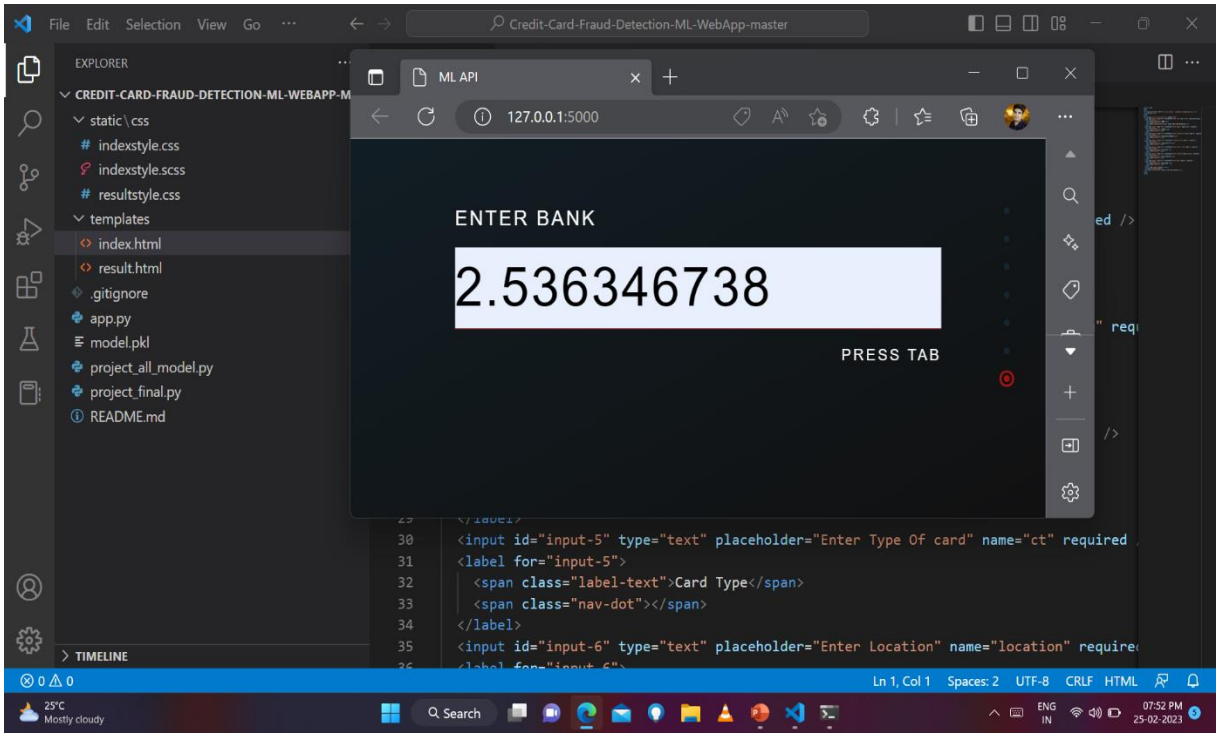


Figure 7.9: Interface to Enter Bank

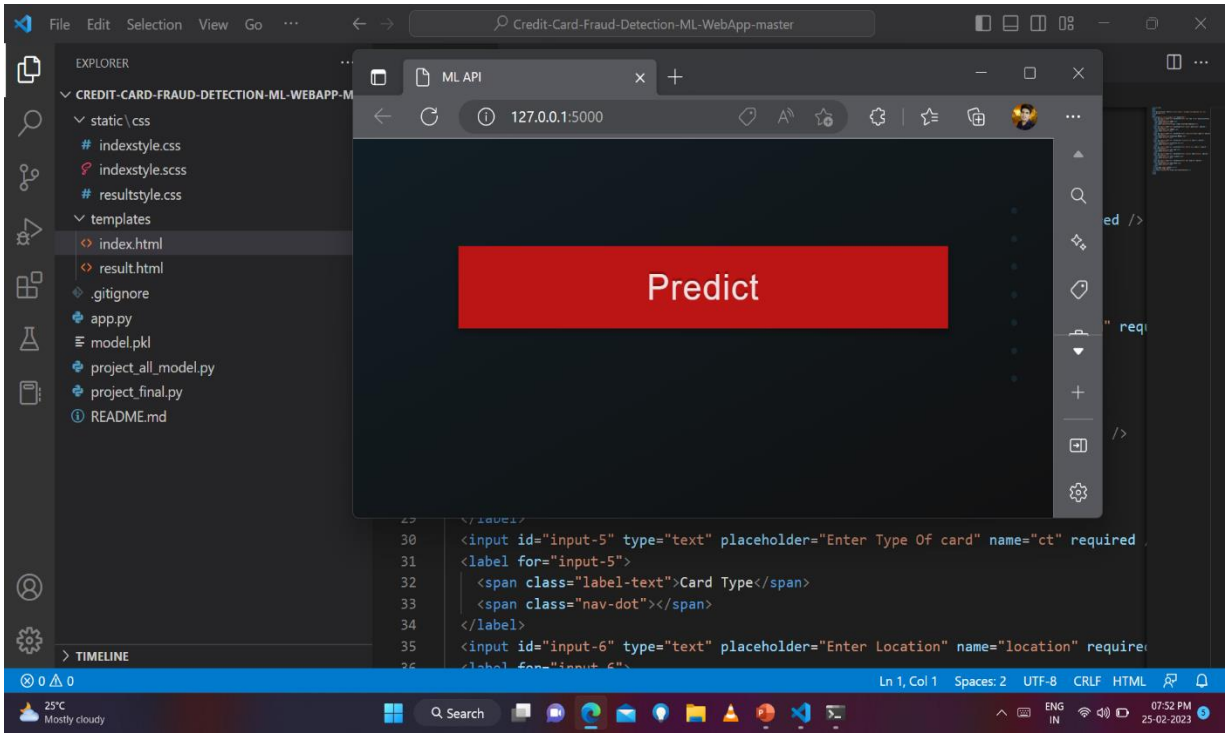


Figure 7.10: Prediction Page

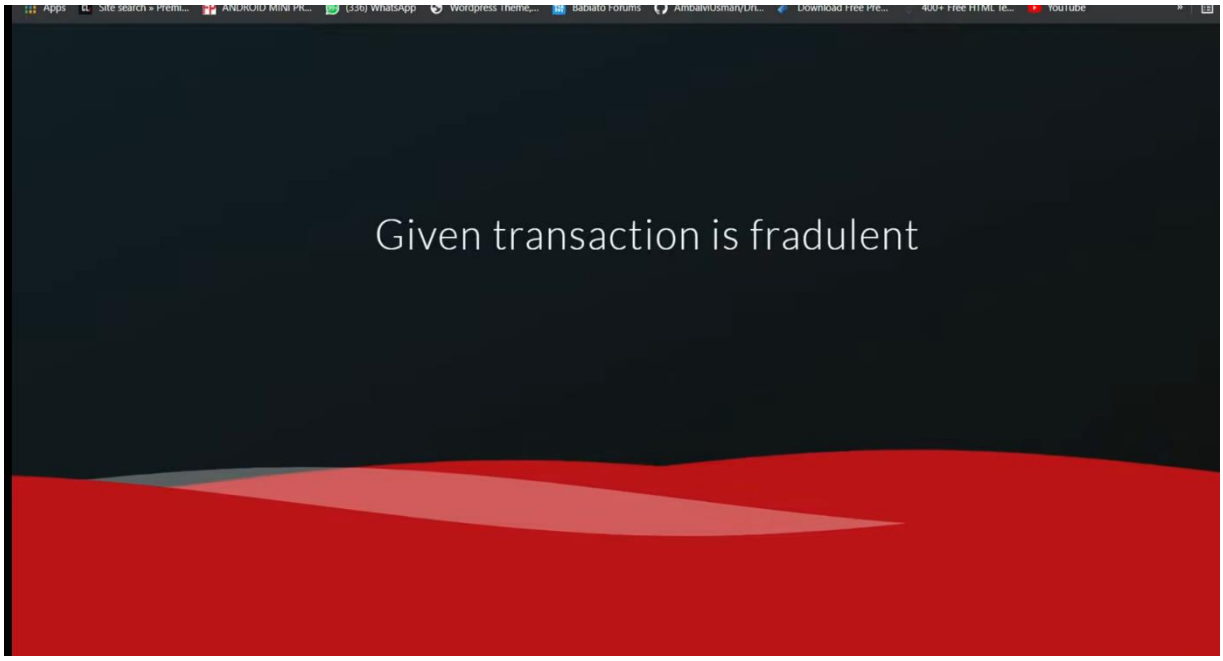


Figure 7.11: Result Page

8. FUTURE SCOPE

There is a very strong possibility of the system being adopted as a norm for the major banking and financial services applications as fraud detection and prevention is the major checkpoint in financial and banking sector. The above system is also likely to be embedded in other applications based, modified as per platformspecific/application specific environment. The banks, financial and retail institutes have faced huge losses owing to cause of a robust and accurate system to predict and prevent the fraudulent transactions going on in an institution. This in-turn affects the business capabilities and consumer trust of the company. Thus, the organizations have moved their focus onto implementing a system which can depict inconsistent transactions, providing banks a privilege to act upon it take necessary measures.

9. CONCLUSION

In this model, we detected the fraudulent transactions and recognized which illustrates the robustness of the proposed system. This proposed model took the trained dataset and performed classification on basis of them, if the transaction was legal then it moved to class 0 and if the transaction was fraud then it moved to class 1, and significantly improve the detection accuracy. The proposed method works efficiently in various platform, vivid environment and is a full_fledged cross platform application. The system has depicted robust, scalable and accurate performance to the degree that efficiency is taken into consideration in the Credit Card Fraud Detection System. The system takes into consideration various factors and has been fulfilling or meeting all the project specifications documented.

10. REFERENCES

[1] R. R. Subramanian, R. Ramar, “Design of Offline and Online Writer Inference Technique”, International Journal of Innovative Technology and Exploring Engineering, vol. 9, no. 2S2, Dec. 2019, ISSN: 2278-3075.

[2]https://www.kaggle.com/mlg-ulb/creditcardfraud database of cards

[3] Delamaire. L. Abdou, HAH and Pointon. J,”Credit card fraud and detection techniques”, Banks and Bank Systems, Volume 4, Issue 2, 2009,2014.

[4] R. R. Subramanian, B. R. Babu, K. Mamta and K. Manogna, "Design and Evaluation of a Hybrid Feature Descriptor based Handwritten Character Inference Technique," 2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Tamilnadu, India, 2019, pp. 1-5.

[5] Şahin, Y. G. and Duman, E. 2011. Detecting credit card fraud by decision trees and support vector machines.

[6] John Richard D. Kho, Larry A. Vea “Credit Card Fraud Detection Based on Transaction Behaviour” published by Proc. of the 2017 IEEE Region 10 Conference (TENCON), Malaysia, November 5-8, 2017.

[7] Yashvi Jain, Namrata Tiwari, ShripriyaDubey, Sarika Jain, “A Comparative Analysis of Various Credit Card Fraud Detection Techniques, Blue Eyes Intelligence Engineering and Sciences Publications 2019” .

[8] Learning Robert A. Sowah, Moses A. Agebure, Godfrey A. Mills, Koudjo M. Kaumudi, “New Cluster Undersampling Technique for Class Imbalance “of 2016 IJMLC.

[9] Baraneetharan, E. "Role of Machine Learning Algorithms Intrusion Detection in WSNs: A Survey." Journal of Information Technology 2, no. 03 (2020): 161-173.

[10] Mitra, Ayushi. "Sentiment Analysis Using Machine Learning Approaches (Lexicon based on movie review dataset)." Journal of Ubiquitous Computing and Communication Technologies (UCCT) 2, no. 03 (2020): 145-152.

[11] Mohamed Jaward Bah, Mohamed Hammad “Progress in Outlier Detection Techniques: A Survey” Hongzhi Wang, of the 2019 IEE. Aibus, S. et al. 2007. Application of Classification Models on Credit Card Fraud Detection. IEEE

[12] Al Daoud, E. J. 1. 1. a. C. and Engineering, 1. 2019. Comparison between XGBoost, LightGBM and CatBoost Using a Home Credit Dataset. 13(1), pp. 6-10. Alghamdi, M. et al.

2017. Predicting diabetes mellitus using SMOTE and ensemble machine learning approach: The Henry Ford Exercise Testing (FIT) project. 12(7), p. 0179805.

[13] Awoyemi, J. O. et al, eds. 2017. Credit card fraud detection using machine learning techniques: A comparative analysis. 2017 International Conference on Computing Networking and Informatics (ICCNI) IEEE.

[14] Bahnsen, A. C. et al, eds. 2014. Improving credit card fraud detection with calibrated probabilities. Proceedings of the 2014 SIAM international conference on data mining SIAM.

[15] Barandela, R. et al, eds. 2004. The imbalanced training sample problem: Under or aver sampling? Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR). Springer.

Credit Card Fraud Detection Using Machine Learning

M.KoteswaraRao
Student
Department of Computer Science and Engineering
Narasaraopeta Engineering College
Narasaraopet, India.
mahankalikoteswararao78@gmail.com

J.SuryaNarayana
Student
Department of Computer Science and Engineering
Narasaraopeta Engineering College
Narasaraopet, India.
suryajakkula999@gmail.com

M.V.Aditya Kumar
Student
Department of Computer Science and Engineering
Narasaraopeta Engineering College
Narasaraopet, India.
adityakumarmudra@gmail.com

Y. Chandana
Asst. Professor
Department of Computer Science and Engineering
Narasaraopeta Engineering College
Narasaraopet, India.
chandana.nrtec@gmail.com

Abstract—Nowadays credit card became one of the essential parts of the people. Sudden increase in E-commerce, customer started using credit card for online purchasing therefore risk of fraud also increases. Instead of carrying a huge amount in hand it is easier to keep credit cards. But nowadays that too becomes unsafe. Now a days we are facing a big problem on credit card fraud which is increasing in a good percentage. The main purpose is the survey on the various methods applied to detect credit card frauds. From the abnormalities, in the transaction, the fraudulent one is identified. We address this issue in order to implement some machine learning algorithm like Isolation Random Forest Algorithm in order to detect this kind of fraud. In this paper we increase the efficiency in finding the fraud. However, we discussed and evaluated employee criteria. Currently, the issues of credit card fraud detection have become a big problem for new researchers. We implement an intelligent algorithm which will detect all kind of fraud in a credit card transaction. We handled the problem by finding a pattern of each customer in between fraud and legal transaction. Random Forest Algorithm and Decision Tree Algorithm are used to predict the pattern of transaction for each customer and a decision is made according to them. In order to prevent data from mismatching, all attribute are marked equally.

Keywords— CreditCard, Criminal Transactions.

I. INTRODUCTION

At Present Situations as we can see that there is a huge increase online payment and the payment is mostly done with the help of credit cards. It becomes a big problem for marketing company to overcome with the credit card fraudulent activities. Fraudulent can be done in many ways such as tax return in any other account, taking loans with wrong information etc. Therefore, we need an efficient fraudulent detection model to minimize fraudulent activity and to minimize their losses. There are a huge number of new techniques which provide different algorithms which help in detecting number of credit card fraudulent activity. Basic understanding of these algorithms will help us in making a significant credit card fraudulent detection model. This paper helps us in finding doubtful credit card transaction by proposing a machine learning algorithms. Credit Card Fraudulent detection comes under machine learning, and the objective is to reduce such type of fraudulent activity[6].

This type of fraud is happening from past, and till now not much research has done here in this particular area. The types of credit fraud in transactions are bankruptcy fraud, behavioral fraud, counterfeit fraud, application fraud[3]. There are experiments done before on credit card fraudulent activity on basis of meta-learning. There is certain limit of meta-learning. There are two features which is introduced here in our report is True Positive and False alarm. Both these features play an important role in catching fraudulent because the rate of determining fraudulent behavior is quick[7].

II. DATASET DESCRIPTION

The dataset holds information about credit card transactions which has been made in a span of two days. The number of frauds have been calculated as 492 out of 284,807 transactions[2]. The details have been given in form of positive and non-positive numerical values. The dataset contains 31 features which has been labelled as V1-V28 due to confidential reasons. The feature which has been revealed are Time and Amount of transaction. Here time denotes the number of seconds elapsed from the first transaction of Day 1. Amount of transaction consists of positive value denoting deposit and non-positive value denoting withdrawal[12].

III. DATA PREPROCESSING

Data preprocessing is a way of making raw data more suitable for analysis. It involves cleaning, transforming, and integrating data to make it more complete, consistent, and understandable. Data preprocessing helps to improve the accuracy and quality of data mining or machine learning results.

In this project we have performed various preprocessing techniques which have helped clean the dataset into useful format.

IV. DATA VISUALIZATION

Data visualization is a way of showing data using graphics, such as charts, plots, infographics, and animations[8],[15].

Fig.3:Number of occurrences of each class label.

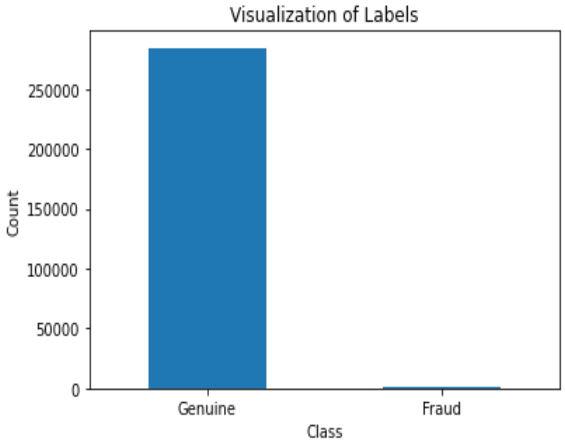


Fig.5:Approximate costing of online orders

V. MODEL BUILDING

The model is built in such way that the dataset we have is split into 70-30 using train_test_split().The model is built using Random Forest algorithm and Decision Tree Algorithm[9].

a)Isolation Random Forest (Existing system)

The previous detecting technique takes a long time to catch fraud which is basically depend on the database, not that much accurate and not give the result in-time. After that algorithm which is used for the detection of credit card fraudulent is generally on basis of analysis, fraudulent detection based on credit card transaction made by cardholder and the credit rate for cardholders.

There are certain limits of meta-learning. There are two features which is introduced here in our report is True Positive and False alarm. Both these features play an important role in catching fraudulent because the rate of determining fraudulent behavior is quick. For the better performance of model, we need a better classifier. Different classifier can be combined together with help of meta-learning.

Previously attempts have been made to work out Credit Card Fraud Detection system using SVM (Select Vector Machine). SVM makes use of hyperplane to classify the data points in a collection. A good hyperplane associates greater number of data points within its margin[5].

Processing a large amount of data sets can be inefficient due to the possibility of redundant data, which increases processing time. Therefore, it usually delayed in calculating the fraud or there might be probability to not calculate in time.

b) Random Forest(Proposed System)

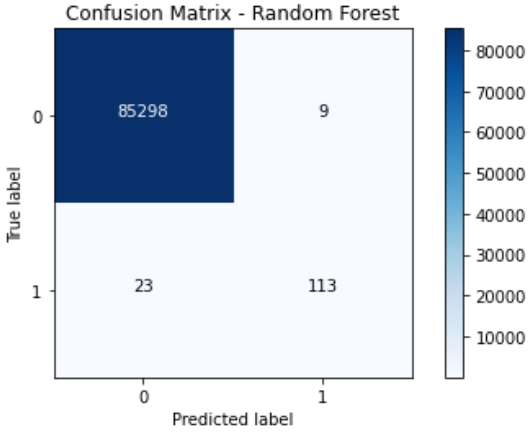
Belonging to the family of ensemble methods, Random Forest is a widely used machine learning algorithm. Ensemble methods combine multiple models to improve the accuracy and robustness of predictions. By utilizing decision trees- simple models that segment the feature space into smaller regions based on input feature values, Random Forest combines multiple models to form a powerful machine learning algorithm.

Multiple decision trees are created by the Random Forest algorithm, where each tree is trained on a random subset of input data and input features. The algorithm then aggregates the predictions of these individual trees to produce a final prediction. In order to aggregate results, two methods can be used depending on the problem type: majority vote for classification problems or averaging for regression problems.

The idea behind Random Forest is that by combining multiple decision trees, the model can capture a wider range of relationships between the input features and the target variable. Additionally, by using random subsets of the input data and features, By preventing models from becoming overly complex and overfitting to the training data, the algorithm mitigates the risk of poor performance on new and unseen data.

Random Forest outperforms other machine learning algorithms in multiple aspects, such as its capability to handle a large number of input features, its resilience to noisy data, and its ability to generate feature importance rankings that help uncover relationships between input features and target variables.

Random Forest is a versatile and powerful algorithm that can produce highly accurate predictions with relatively little tuning. Using this Random Forest We got 99.6% better Accuracy than Decision tree. And we are also performed train and evaluation of Dataset over Confusion Matrix-Random Forest.



c)Decision Tree (Proposed System)

A decision tree is a type of machine learning algorithm that is used for classification and regression analysis. It is a tree-like model where each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label or a numerical value.

Decision trees are commonly employed in problems involving classification, where the objective is to anticipate a categorical output variable. The algorithm constructs a tree by dividing the data into smaller subsets repeatedly, based on the feature values. The splits are chosen to maximize the separation of the classes, usually based on metrics like information gain or Gini impurity.

Decision trees possess several benefits, including their simplicity in interpretation and visualization, which facilitates human comprehension of the model's prediction process. Additionally, decision trees can accommodate both categorical and numerical data, as well as manage missing values.

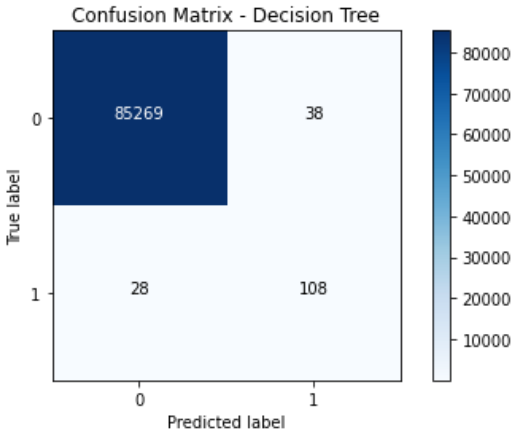
Nevertheless, decision trees have a tendency to overfit, particularly if the tree's depth is excessive or the feature count is substantial. Approaches like pruning and capping the tree's depth can alleviate overfitting. Furthermore, decision trees may not be the optimal selection for issues with interdependent features or situations with intricate, nonlinear decision boundaries.

As prediction of score is much important task according to our model therefore we are predicting the score on the basis of the given formula:

Score = 0.5 * TP + 0.5 * Deviation

Where, TP is True Positive value and Deviation is the deviation of outlier data from the standard data point.

On the basis of these score we made two classes 0 and 1. If the score is 1 it will move to class 1 and termed as legal transaction and if the score is 0 it will move to class 0 and termed as fraudulent transaction. At last, the accuracy is calculated on the basis of how many fraud transactions are there in our dataset and how many we predicted with the help of our model one without any replacement which results in creation of dataset for each tree with samples that are unique in nature.



VI. RESULTS AND DISCUSSION

We can see in the below table that Random Forest has the best performance in terms of accuracy and when compared to the Decision Tree and Isolation Random Forest which is proven to the best algorithm for the project until we have established the Credit Card Fraud Detection Using Machine Learning[13].

Algorithm	Accuracy
Isolation Forest	0.96
Random Forest	0.99
Decision Tree	0.98

Table1:Comparison of Performance of Regression Algorithms

VII. FUTURE SCOPE

There is a very strong possibility of the system being adopted as a norm for the major banking and financial services applications as fraud detection and prevention is the major checkpoint in financial and banking sector. The above system is also likely to be embedded in other applications based, modified as per platform-specific/application specific environment. The banks, financial and retail institutes have faced huge losses owing to cause of a robust and accurate system to predict and prevent the fraudulent transactions going on in an institution. This in-turn affects the business capabilities and

consumer trust of the company. Thus, the organizations have moved their focus onto implementing a system which can depict inconsistent transactions, providing banks a privilege to act upon it take necessary measures.

VI. CONCLUSION

In this model, we detected the fraudulent transactions and recognized which illustrates the robustness of the proposed system. This proposed model took the trained dataset and performed classification on basis of them, if the transaction was legal then it moved to class 0 and if the transaction was fraud then it moved to class 1, and significantly improve the detection accuracy. The proposed method works efficiently in various platform, vivid environment and is a full_fledged cross platform application. The system has depicted robust, scalable and accurate performance to the degree that efficiency is taken into consideration in the Credit Card Fraud Detection System. The system takes into consideration various factors and has been fulfilling or meeting all the project specifications documented.

References

[1] R. R. Subramanian, R. Ramar, “Design of Offline and Online Writer Inference Technique”, International Journal of Innovative Technology and Exploring Engineering, vol. 9, no. 2S2, Dec. 2019, ISSN: 2278-3075.

[2]https://www.kaggle.com/mlg-ulb/creditcardfraud database of cards

[3] Delamaire. L. Abdou, HAH and Pointon. J,”Credit card fraud and detection techniques”, Banks and Bank Systems, Volume 4, Issue 2, 2009,2014.

[4] R. R. Subramanian, B. R. Babu, K. Mamta and K. Manogna, "Design and Evaluation of a Hybrid Feature Descriptor based Handwritten Character Inference Technique," 2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Tamilnadu, India, 2019, pp. 1-5.

[5] Şahin, Y. G. and Duman, E. 2011. Detecting credit card fraud by decision trees and support vector machines.

[6] John Richard D. Kho, Larry A. Vea “Credit Card Fraud Detection Based on Transaction Behaviour” published by Proc. of the 2017 IEEE Region 10 Conference (TENCON), Malaysia, November 5-8, 2017.

[7] Yashvi Jain, Namrata Tiwari, ShripriyaDubey, Sarika Jain, “A Comparative Analysis of Various Credit Card Fraud Detection Techniques, Blue Eyes Intelligence Engineering and Sciences Publications 2019” .

[8] Learning Robert A. Sowah, Moses A. Agebure, Godfrey A. Mills, Koudjo M. Kaumudi, “New Cluster Undersampling Technique for Class Imbalance “of 2016 IJMLC.

[9] Baraneetharan, E. "Role of Machine Learning Algorithms Intrusion Detection in WSNs: A Survey." Journal of Information Technology 2, no. 03 (2020): 161-173.

[10] Mitra, Ayushi. "Sentiment Analysis Using Machine Learning Approaches (Lexicon based on movie review dataset)." Journal of Ubiquitous Computing and Communication Technologies (UCCT) 2, no. 03 (2020): 145-152.

[11] Mohamed Jaward Bah, Mohamed Hammad “Progress in Outlier Detection Techniques: A Survey” Hongzhi Wang, of the 2019 IEE. Aibus, S. et al. 2007. Application of Classification

Models on Credit Card Fraud Detection. IEEE

[12] Al Daoud, E. J. 1. 1. a. C. and Engineering, 1. 2019. Comparison between XGBoost, LightGBM and CatBoost Using a Home Credit Dataset. 13(1), pp. 6-10. Alghamdi, M. et al. 2017. Predicting diabetes mellitus using SMOTE and ensemble machine learning approach: The Henry Ford Exercise Testing (FIT) project. 12(7), p. 0179805.

[13] Awoyemi, J. O. et al, eds. 2017. Credit card fraud detection using machine learning techniques: A comparative analysis. 2017 International Conference on Computing Networking and Informatics (ICCNI) IEEE.

[14] Bahnsen, A. C. et al, eds. 2014. Improving credit card fraud detection with calibrated probabilities. Proceedings of the 2014 SIAM international conference on data mining SIAM.

[15] Barandela, R. et al, eds. 2004. The imbalanced training sample problem: Under or aver sampling? Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR). Springer.

CB5

by KoteswaraRao M

Submission date: 19-Mar-2023 06:26PM (UTC+1100)
Submission ID: 2040429567
File name: CB5_Final_Paper.pdf (163.75K)
Word count: 2412
Character count: 13429

Credit Card Fraud Detection Using Machine Learning

M.KoteswaraRao
Student
Department of Computer Science and Engineering
Narasaraopeta Engineering College
Narasaraopet, India.
mahankalikoteswararao78@gmail.com

J.SuryaNarayana
Student
Department of Computer Science and Engineering
Narasaraopeta Engineering College
Narasaraopet, India.
suryajakkula999@gmail.com

M.V.Aditya Kumar
Student
Department of Computer Science and Engineering
Narasaraopeta Engineering College
Narasaraopet, India.
adityakumarmudra@gmail.com

Y. Chandana
Asst. Professor
Department of Computer Science and Engineering
Narasaraopeta Engineering College
Narasaraopet, India.
chandana.nrtec@gmail.com

Abstract—Nowadays credit card became one of the essential parts of the people. Sudden increase in E-commerce, customer started using credit card for online purchasing therefore risk of fraud also increases. Instead of carrying a huge amount in hand it is easier to keep credit cards. But nowadays that too becomes unsafe. Now a days we are facing a big problem on credit card fraud which is increasing in a good percentage. The main purpose is the survey on the various methods applied to detect credit card frauds. From the abnormalities, in the transaction, the fraudulent one is identified. We address this issue in order to implement some machine learning algorithm like Isolation Random Forest Algorithm in order to detect this kind of fraud. In this paper we increase the efficiency in finding the fraud. However, we discussed and evaluated employee criteria. Currently, the issues of credit card fraud detection have become a big problem for new researchers. We implement an intelligent algorithm which will detect all kind of fraud in a credit card transaction. We handled the problem by finding a pattern of each customer in between fraud and legal transaction. Random Forest Algorithm and Decision Tree Algorithm are used to predict the pattern of transaction for each customer and a decision is made according to them. In order to prevent data from mismatching, all attribute are marked equally.

Keywords— CreditCard, Criminal Transactions.

I. INTRODUCTION

At Present Situations as we can see that there is a huge increase online payment and the payment is mostly done with the help of credit cards. It becomes a big problem for marketing company to overcome with the credit card fraudulent activities. Fraudulent can be done in many ways such as tax return in any other account, taking loans with wrong information etc. Therefore, we need an efficient fraudulent detection model to minimize fraudulent activity and to minimize their losses. There are a huge number of new techniques which provide different algorithms which help in detecting number of credit card fraudulent activity. Basic understanding of these algorithms will help us in making a significant credit card fraudulent detection model. This paper helps us in finding doubtful credit card transaction by proposing a machine learning algorithms. Credit Card Fraudulent detection comes under machine learning, and the objective is to reduce such type of fraudulent activity[6].

This type of fraud is happening from past, and till now not much research has done here in this particular area. The types of credit fraud in transactions are bankruptcy fraud, behavioral fraud, counterfeit fraud, application fraud[3]. There are experiments done before on credit card fraudulent activity on basis of meta-learning. There is certain limit of meta-learning. There are two features which is introduced here in our report is True Positive and False alarm. Both these features play an important role in catching fraudulent because the rate of determining fraudulent behavior is quick[7].

II. DATASET DESCRIPTION

The dataset holds information about credit card transactions which has been made in a span of two days. The number of frauds have been calculated as 492 out of 284,807 transactions[2]. The details have been given in form of positive and non-positive numerical values. The dataset contains 31 features which has been labelled as V1-V28 due to confidential reasons. The feature which has been revealed are Time and Amount of transaction. Here time denotes the number of seconds elapsed from the first transaction of Day 1. Amount of transaction consists of positive value denoting deposit and non-positive value denoting withdrawal[12].

III. DATA PREPROCESSING

Data preprocessing is a way of making raw data more suitable for analysis. It involves cleaning, transforming, and integrating data to make it more complete, consistent, and understandable. Data preprocessing helps to improve the accuracy and quality of data mining or machine learning results.

In this project we have performed various preprocessing techniques which have helped clean the dataset into useful format.

IV. DATA VISUALIZATION

Data visualization is a way of showing data using graphics, such as charts, plots, infographics, and animations[8],[15].

Fig.3:Number of occurrences of each class label.

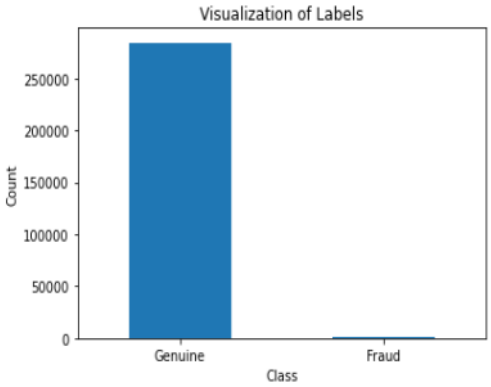


Fig.5:Approximate costing of online orders

V. MODEL BUILDING

The model is built in such way that the dataset we have is split into 70-30 using train_test_split().The model is built using Random Forest algorithm and Decision Tree Algorithm[9].

a)Isolation Random Forest (Existing system)

The previous detecting technique takes a long time to catch fraud which is basically depend on the database, not that much accurate and not give the result in-time. After that algorithm which is used for the detection of credit card fraudulent is generally on basis of analysis, fraudulent detection based on credit card transaction made by cardholder and the credit rate for cardholders.

There are certain limits of meta-learning. There are two features which is introduced here in our report is True Positive and False alarm. Both these features play an important role in catching fraudulent because the rate of determining fraudulent behavior is quick. For the better performance of model, we need a better classifier. Different classifier can be combined together with help of meta-learning.

Previously attempts have been made to work out Credit Card Fraud Detection system using SVM (Select Vector Machine). SVM makes use of hyperplane to classify the data points in a collection. A good hyperplane associates greater number of data points within its margin[5].

Processing a large amount of data sets can be inefficient due to the possibility of redundant data, which increases processing time. Therefore, it usually delayed in calculating the fraud or there might be probability to not calculate in time.

b) Random Forest(Proposed System)

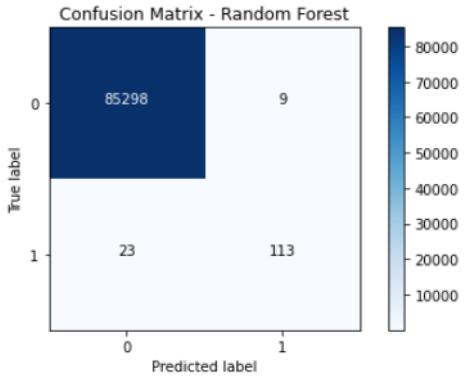
Belonging to the family of ensemble methods, Random Forest is a widely used machine learning algorithm. Ensemble methods combine multiple models to improve the accuracy and robustness of predictions. By utilizing decision trees- simple models that segment the feature space into smaller regions based on input feature values, Random Forest combines multiple models to form a powerful machine learning algorithm.

Multiple decision trees are created by the Random Forest algorithm, where each tree is trained on a random subset of input data and input features. The algorithm then aggregates the predictions of these individual trees to produce a final prediction. In order to aggregate results, two methods can be used depending on the problem type: majority vote for classification problems or averaging for regression problems.

The idea behind Random Forest is that by combining multiple decision trees, the model can capture a wider range of relationships between the input features and the target variable. Additionally, by using random subsets of the input data and features, By preventing models from becoming overly complex and overfitting to the training data, the algorithm mitigates the risk of poor performance on new and unseen data.

Random Forest outperforms other machine learning algorithms in multiple aspects, such as its capability to handle a large number of input features, its resilience to noisy data, and its ability to generate feature importance rankings that help uncover relationships between input features and target variables.

Random Forest is a versatile and powerful algorithm that can produce highly accurate predictions with relatively little tuning. Using this Random Forest We got 99.6% better Accuracy than Decision tree. And we are also performed train and evaluation of Dataset over Confusion Matrix-Random Forest.



c)Decision Tree (Proposed System)

A decision tree is a type of machine learning algorithm that is used for classification and regression analysis. It is a tree-like model where each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label or a numerical value.

Decision trees are commonly employed in problems involving classification, where the objective is to anticipate a categorical output variable. The algorithm constructs a tree by dividing the data into smaller subsets repeatedly, based on the feature values. The splits are chosen to maximize the separation of the classes, usually based on metrics like information gain or Gini impurity.

Decision trees possess several benefits, including their simplicity in interpretation and visualization, which facilitates human comprehension of the model's prediction process. Additionally, decision trees can accommodate both categorical and numerical data, as well as manage missing values.

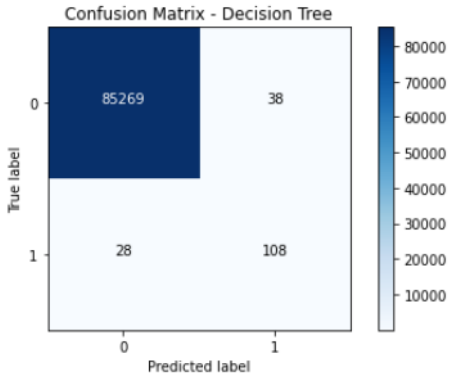
Nevertheless, decision trees have a tendency to overfit, particularly if the tree's depth is excessive or the feature count is substantial. Approaches like pruning and capping the tree's depth can alleviate overfitting. Furthermore, decision trees may not be the optimal selection for issues with interdependent features or situations with intricate, nonlinear decision boundaries.

As prediction of score is much important task according to our model therefore we are predicting the score on the basis of the given formula:

Score = 0.5 * TP + 0.5 * Deviation

Where, TP is True Positive value and Deviation is the deviation of outlier data from the standard data point.

On the basis of these score we made two classes 0 and 1. If the score is 1 it will move to class 1 and termed as legal transaction and if the score is 0 it will move to class 0 and termed as fraudulent transaction. At last, the accuracy is calculated on the basis of how many fraud transactions are there in our dataset and how many we predicted with the help of our model one without any replacement which results in creation of dataset for each tree with samples that are unique in nature.



VI. RESULTS AND DISCUSSION

We can see in the below table that Random Forest has the best performance in terms of accuracy and when compared to the Decision Tree and Isolation Random Forest which is proven to the best algorithm for the project until we have established the Credit Card Fraud Detection Using Machine Learning[13].

Algorithm	Accuracy
Isolation Forest	0.96
Random Forest	0.99
Decision Tree	0.98

Table1:Comparison of Performance of Regression Algorithms

VII. FUTURE SCOPE

There is a very strong possibility of the system being adopted as a norm for the major banking and financial services applications as fraud detection and prevention is the major checkpoint in financial and banking sector. The above system is also likely to be embedded in other applications based, modified as per platform-specific/application specific environment. The banks, financial and retail institutes have faced huge losses owing to cause of a robust and accurate system to predict and prevent the fraudulent transactions going on in an institution. This in-turn affects the business capabilities and

consumer trust of the company. Thus, the organizations have moved their focus onto implementing a system which can depict inconsistent transactions, providing banks a privilege to act upon it take necessary measures.

VI. CONCLUSION

In this model, we detected the fraudulent transactions and recognized which illustrates the robustness of the proposed system. This proposed model took the trained dataset and performed classification on basis of them, if the transaction was legal then it moved to class 0 and if the transaction was fraud then it moved to class 1, and significantly improve the detection accuracy. The proposed method works efficiently in various platform, vivid environment and is a full_fledged cross platform application. The system has depicted robust, scalable and accurate performance to the degree that efficiency is taken into consideration in the Credit Card Fraud Detection System. The system takes into consideration various factors and has been fulfilling or meeting all the project specifications documented.

References

[1] R. R. Subramanian, R. Ramar, “Design of Offline and Online Writer Inference Technique”, International Journal of Innovative Technology and Exploring Engineering, vol. 9, no. 2S2, Dec. 2019, ISSN: 2278-3075.

[2]https://www.kaggle.com/mlg-ulb/creditcardfraud database of cards

[3] Delamaire. L. Abdou, HAH and Pointon. J,”Credit card fraud and detection techniques”, Banks and Bank Systems, Volume 4, Issue 2, 2009,2014.

[4] R. R. Subramanian, B. R. Babu, K. Mamta and K. Manogna, "Design and Evaluation of a Hybrid Feature Descriptor based Handwritten Character Inference Technique," 2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Tamilnadu, India, 2019, pp. 1-5.

[5] Şahin, Y. G. and Duman, E. 2011. Detecting credit card fraud by decision trees and support vector machines.

[6] John Richard D. Kho, Larry A. Vea “Credit Card Fraud Detection Based on Transaction Behaviour” published by Proc. of the 2017 IEEE Region 10 Conference (TENCON), Malaysia, November 5-8, 2017.

[7] Yashvi Jain, Namrata Tiwari, ShripriyaDubey, Sarika Jain, “A Comparative Analysis of Various Credit Card Fraud Detection Techniques, Blue Eyes Intelligence Engineering and Sciences Publications 2019” .

[8] Learning Robert A. Sowah, Moses A. Agebure, Godfrey A. Mills, Koudjo M. Kaumudi, “New Cluster Undersampling Technique for Class Imbalance “of 2016 IJMLC.

[9] Baraneetharan, E. "Role of Machine Learning Algorithms Intrusion Detection in WSNs: A Survey." Journal of Information Technology 2, no. 03 (2020): 161-173.

[10] Mitra, Ayushi. "Sentiment Analysis Using Machine Learning Approaches (Lexicon based on movie review dataset)." Journal of Ubiquitous Computing and Communication Technologies (UCCT) 2, no. 03 (2020): 145-152.

[11] Mohamed Jaward Bah, Mohamed Hammad “Progress in Outlier Detection Techniques: A Survey” Hongzhi Wang, of the 2019 IEE. Aibus, S. et al. 2007. Application of Classification

Models on Credit Card Fraud Detection. IEEE

[12] Al Daoud, E. J. 1. 1. a. C. and Engineering, 1. 2019. Comparison between XGBoost, LightGBM and CatBoost Using a Home Credit Dataset. 13(1), pp. 6-10. Alghamdi, M. et al. 2017. Predicting diabetes mellitus using SMOTE and ensemble machine learning approach: The Henry Ford Exercise Testing (FIT) project. 12(7), p. 0179805.

[13] Awoyemi, J. O. et al, eds. 2017. Credit card fraud detection using machine learning techniques: A comparative analysis. 2017 International Conference on Computing Networking and Informatics (ICCNI) IEEE.

[14] Bahnsen, A. C. et al, eds. 2014. Improving credit card fraud detection with calibrated probabilities. Proceedings of the 2014 SIAM international conference on data mining SIAM.

[15] Barandela, R. et al, eds. 2004. The imbalanced training sample problem: Under or aver sampling? Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR). Springer.

CB5

ORIGINALITY REPORT

10%	8%	8%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.thefreelibrary.com Internet Source	2%
2	Submitted to Berlin School of Business and Innovation Student Paper	2%
3	www.ijres.org Internet Source	1%
4	www.researchgate.net Internet Source	1%
5	"Machine Intelligence and Soft Computing", Springer Science and Business Media LLC, 2021 Publication	1%
6	"Advances in Computing and Data Sciences", Springer Science and Business Media LLC, 2018 Publication	1%
7	www.ijraset.com Internet Source	1%
	Submitted to Coventry University	

8	Student Paper	1%
9	Hamed Ghoddusi, Germán G. Creamer, Nima Rafizadeh. "Machine learning in energy economics and finance: A review", Energy Economics, 2019 Publication	<1%
10	ijcat.com Internet Source	<1%
11	www.pce.ac.in Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

PAPER ID
NECICAIEA2K23041

International Conference on
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K23
17th & 18th March, 2023
Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that KoteswaraRao Mahankali , Narasaraopeta Engineering College has presented the paper title Credit Card Fraud Detection Using Machine Learning in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

PAPER ID
NECICAIEA2K23041

International Conference on
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K23
17th & 18th March, 2023
Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that J.SuryaNarayana , Narasaraopeta Engineering College has presented the paper title Credit Card Fraud Detection Using Machine Learning in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

PAPER ID
NECICAIEA2K23041

International Conference on
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K23
17th & 18th March, 2023
Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that M.V.Aditya Kumar , Narasaraopeta Engineering College has presented the paper title Credit Card Fraud Detection Using Machine Learning in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

PAPER ID
NECICAIEA2K23041

International Conference on
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K23
17th & 18th March, 2023
Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that Y. Chandana , Narasaraopeta Engineering College has presented the paper title Credit Card Fraud Detection Using Machine Learning in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar