CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING

A Project Report

Submitted in partial fulfillment of the

Requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

by

B.MUBASHERA KOUSER	**(2010104)**
G.KAVITHA	**(2010113)**
K.MANJULA	**(2010118)**

Under the esteemed guidance of

Sri. U.DHANUNJAYA,M.Tech, (Ph.D.)

Lecturer ,Department of CSE,SKUCET

DEPT. OF COMPUTER SCIENCE AND ENGINEERING

SRI KRISHNADEVARAYA UNIVERSITY

COLLEGE OF ENGINEERING AND TECHNOLOGY

ANANTAPUR - 515001

ANDHRA PRADESH

2020-2024

SRI KRISHNADEVARAYA UNIVERSITY

COLLEGE OF ENGINEERING AND TECHNOLOGY

ANANTAPUR – 515001

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that this is a bonafide record of the dissertation work entitled, "CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING", done by **B.MUBASHERAKOUSER(2010104),G.KAVITHA(2010113),K.MANJULA(2010118)** submitted to the faculty of **Computer Science and Engineering**, in partial fulfillment of the requirements for the Degree of **Bachelor of Technology with specialization in COMPUTER SCIENCE AND ENGINEERING from Sri Krishnadevaraya University College of Engineering and Technology, Ananthapuramu.**

Signature of the Supervisor

Sri. U.DHANUNJAYA, M.Tech, (Ph.D.)
Lecturer in Department of CSE,
S.K.U.College of Engg.& Tech,
Ananthapuramu

Signature of the Head of the Department

Sri. P.R.RAJESH KUMAR M.Tech, (Ph.D)
Head of the Department of CSE,
S.K.U.College of Engg.& Tech,
Ananthapuramu

SRI KRISHNADEVARAYA UNIVERSITY

COLLEGE OF ENGINEERING AND TECHNOLOGY

ANANTAPUR – 515001

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby declare that the project report entitled "CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING" submitted to the Department of Computer Science and Engineering, Sri Krishnadevaraya University, Anantapuramu for the partial fulfillment of the academic requirement for the degree for Bachelor of Technology in Computer Science and Engineering is an authentic record of our work carried out during the final year under the esteemed guidance of Sri. U.DHANUNJAYA, M.Tech.,(Ph.D), Lecturer Computer Science and Engineering Department, College of Engineering and Technology, Sri Krishnadevaraya University, Anantapuramu.

Signature of the students

1.

2.

3.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that I have now the opportunity to express my gratitude for all of them.

It is with immense pleasure that I would like to express my indebted gratitude to my Guide Sri. U.DHANUNJAYA, M.Tech.,(Ph.D) in CSE Department, who has guided me a lot and encouraged me in every step of the project work. His valuable moral support and guidance throughout the project helped me to a greater extent. I thank him for his stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

I wish to express my deep sense of gratitude to Sri.P.R.RAJESH KUMAR, M.Tech (Ph.D.) Lecturer and Head of the Department (I/C) of Computer Science and Engineering, for giving me the opportunity of doing the project and for providing a great support in completing my project work. I feel elated to thank him for inspiring me all the way by providing good lab facilities and helping me in providing such good environment.

I wish to convey my acknowledgment to Dr. R.RAMACHANDRA, M.Tech., Ph.D., Principal, SKU College of Engineering and Technology, Anantapuramu, for providingsuch a good environment and facilities.

My special thanks to the Faculty of CSE Department for giving the required information in doing my project work. Not to forget, I thank all the non-teaching staff, my friends and class mates who had directly or indirectly helped and supported me in completing my project in time.

Finally I wish to convey my gratitude to my parents who fostered all the requirements and facilities that I need.

B.MUBASHERA KOUSER	2010104
G.KAVITHA	2010113
K.MANJULA	2010118

ABSTRACT

Credit Card fraud happens when a person steals someone else's Credit Card information and uses it for their own financial gain. Credit Card fraud is presently the most frequently occuring problem in the present world. This is due to the rise in both online transactions and e-commerce platforms. Detecting fraud is crucial to protect users and financial institutions from financial losses.

The previous fraud detection systems mostly use basic rules and simple tricks to find bad transactions. But these old ways struggle to keep up with tricky fraud and sometimes mess up by thinking good transactions are bad or missing some tricky fraud ones. Our new system is different because it uses machine learning which means it learns and improves over time. The primary objective is to enhance the accuracy and efficiency of fraud detection in credit card transactions, thereby providing a more secure financial environment for users.

The proposed system integrates machine learning algorithms include random forest and logistic regression to analyze transaction data in real time. These models are trained on historical transactions data to learn patterns and relationships between various transaction features . By conducting extensive experiments on a large dataset of real-world credit card transactions we will assess the System's ability to accurately identify fraudulent transactions while minimizing disruption to legitimate card holders.

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

UML - Unified Modelling Language

ML - Machine Learning

RF - Random Forest

LR - Logistic Regression

PCA - Principal Component Analysis

CCFD - Credit Card Fraud Detection

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

In an age where electronic transactions have become the norm, ensuring the security of financial transactions, especially credit card transactions, has become a paramount concern. The exponential growth of online commerce and the increasing sophistication of fraudulent activities necessitate robust and efficient fraud detection mechanisms. This project aims to address this critical issue by leveraging the power of machine learning algorithms to detect and prevent credit card fraud in real-time.

The increase of credit card fraud poses significant challenges to financial institutions and consumers alike. Fraudulent activities range from stolen card details used for unauthorized transactions to more complex schemes involving identity theft and account takeover. Traditional rule-based systems struggle to keep pace with evolving fraud tactics, highlighting the need for more adaptive and intelligent approaches. By harnessing the capabilities of machine learning, this project seeks to enhance fraud detection accuracy and efficiency while minimizing false positives.

At the core of this project are two powerful machine learning algorithms: Random Forest and Logistic Regression. These algorithms are well-suited for the task of credit card fraud detection due to their ability to analyze large volumes of transaction data and identify patterns indicative of fraudulent behavior. Random Forest excels in handling high-dimensional data and capturing complex interactions, while Logistic Regression provides interpretable results and is effective in binary classification tasks.

The methodology involves several key steps, beginning with data preprocessing and feature engineering to extract meaningful information from raw transaction data. Feature selection techniques such as principal component analysis (PCA) may be employed to reduce dimensionality and enhance model performance. Subsequently, the dataset is divided into training and testing sets for model training and evaluation. Cross-validation techniques ensure the robustness of the models.

In addition to detecting fraudulent transactions, another crucial aspect of credit card fraud prevention is customer authentication and identity verification. Biometric authentication methods, such as fingerprint recognition or facial recognition, offer enhanced security by uniquely identifying individuals based on their physiological or behavioral characteristics. Combining machine learning-based fraud detection with biometric authentication systems can provide a multi-layered approach to fraud prevention, reducing the risk of unauthorized access and transactional fraud.

One of the challenges in credit card fraud detection is the imbalance between fraudulent and non-fraudulent transactions, where the number of legitimate transactions far exceeds fraudulent ones. This class imbalance can lead to skewed model performance and an increased likelihood of false positives. To address this issue, advanced sampling techniques such as oversampling of the minority class or undersampling of the majority class may be employed to create a balanced training dataset. Additionally, ensemble learning methods, such as combining multiple classifiers or using boosting techniques, can further improve model robustness and performance in handling imbalanced data.

Ultimately, the successful implementation of machine learning-based credit card fraud detection has the potential to mitigate financial losses for both financial institutions and consumers, safeguarding against fraudulent activities and enhancing trust in electronic payment systems. By leveraging advanced algorithms and analytical techniques, this project aims to contribute to the ongoing efforts to combat fraud and uphold the integrity of financial transactions in the digital age.

The project on credit card fraud detection using machine learning represents a multidimensional effort to combat financial fraud, enhance transaction security, and protect consumer interests. By leveraging advanced algorithms, innovative techniques, and ethical principles, this project aims to contribute to the development of scalable, accurate, and responsible fraud detection solutions. Through ongoing research, collaboration, and industry partnerships, the pursuit of effective fraud detection mechanisms remains pivotal in safeguarding the integrity of electronic payment systems and providing trust in digital transactions.

1.2 LITERATURE SURVEY

[1] L. Bhavya , V. Sasidhar Reddy , U. Anjali Mohan , S. Karishma, 2020, Credit Card Fraud Detection using Classification, Unsupervised, Neural Networks Models, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 09, Issue 04 (April 2020),

Online transactions have increased significantly in recent years. Online credit card transactions make up a sizable portion of them. Applications for detecting credit card theft are therefore greatly needed in the banking and financial industries. The intent of credit card fraud may be to obtain goods without paying for them or to withdraw money from an account without authorization. With the increase in desire for money, credit card fraud incidents increased. The cardholder suffers a significant financial loss as a result.

[2] Renjith, Shini. (2018). Detection of Fraudulent Sellers in Online Marketplaces using Support Vector Machine Approach. International Journal of Engineering Trends and Technology. 57. 48-53. 10.14445/22315381/IJETT-V57P210.

The amount of money spent on e-commerce globally has steadily increased over the years, indicating a clear shift in consumer attention away from brick-and-mortar stores and towards online retailers. Online marketplaces have emerged as one of the major forces driving this expansion in recent years. In-depth research is being done on fraudulent e-commerce buyers and their transactions, and various control and prevention measures are being considered. Merchant fraud refers to another type of fraud that occurs in marketplaces on the seller side. One straightforward example of this kind of fraud is the sale of goods or services at low prices but with no guarantee of delivery. This paper makes an effort to propose a framework using machine learning methods to identify such fraudulent sellers.

[3] Saputra, Adi & Suharjito, Suharjito. (2019). Fraud Detection using Machine Learning in e-Commerce. 10.14569/IJACSA.2019.0100943.

The number of internet users is growing, which is also prompting e-commerce transactions to grow. We also see that there is an increase in online transaction fraud. Machine learning will be used to develop strategies for preventing fraud in e-commerce. This study examines the best machine learning algorithms, including Decision Tree, Naive Bayes, Random Forest, and

Neural Network. The neural network has a 96 percent accuracy rate, followed by decision tree (91) percent, random forest (95 percent), in the evaluation using the confusion matrix. The Synthetic Minority Over-sampling Method (SMOTE) can raise the average F1-Score from 67.9 to 94.5 percent and the average G-Mean from 73.5 to 84.6 percent.

[4] A. K. Rai and R. K. Dwivedi, "Fraud Detection in Credit Card Data using Unsupervised Machine Learning Based Scheme," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2020, pp. 421-426, doi: 10.1109/ICESC48915.2020.9155615.

The rise of communication and e-commerce technologies has made credit cards the most widely used method of payment for both online and off-line purchases. Therefore, it is crucial that this system's security guards against fraudulent transactions. Each year, there are more fraudulent credit card data transactions. Researchers are experimenting with cutting-edge methods to identify and stop such scams in this direction. Nonetheless, there will always be a need for certain methods that can quickly and accurately identify these frauds. In this paper, a method for identifying credit card fraud using unsupervised learning based on neural networks is proposed. The proposed strategy performs better than the current K-Means clustering, Isolation Forest, Local Outlier Factor, and Auto Encoder approaches.

1.3 PROBLEM STATEMENT

Credit card fraud poses a significant threat to both financial institutions and consumers, leading to substantial financial losses and erosion of trust in electronic payment systems. Despite the implementation of various fraud prevention measures, fraudulent activities continue to evolve, necessitating advanced detection mechanisms. The problem statement for this project is to develop and implement a robust credit card fraud detection system using ML algorithms to accurately identify and prevent fraudulent transactions in real-time.

The key challenges addressed by this project include the detection of fraudulent activities amidst a vast volume of legitimate transactions, the dynamic nature of fraud tactics, and the need for timely intervention to mitigate financial losses. Traditional rule-based systems often struggle to adapt to evolving fraud patterns and exhibit limited scalability in handling large-scale transaction data. Consequently, there is a pressing need for adaptive and intelligent fraud detection solutions capable of analyzing complex transaction patterns and identifying anomalous behavior indicative of fraud.

Furthermore, the imbalanced nature of credit card transaction datasets, where the number of legitimate transactions far exceeds fraudulent ones, poses a significant obstacle to effective fraud detection. Class imbalance can lead to skewed model performance and an increased likelihood of false positives or false negatives, compromising the accuracy and reliability of fraud detection systems. Addressing class imbalance through appropriate sampling techniques and algorithmic approaches is essential to ensure the robustness and efficacy of the proposed fraud detection system.

Moreover, the deployment of the fraud detection system in real-world settings requires considerations of scalability, latency, and computational efficiency. The system must be capable of processing transactions in real-time, without introducing significant delays or disruptions to the payment processing workflow. Additionally, the system should be seamlessly integrated into existing transaction processing systems, enabling frictionless deployment and operation within financial institutions' infrastructure.

The problem statement encapsulates the need to develop a credit card fraud detection system that leverages machine learning algorithms to effectively combat fraudulent activities while minimizing false positives and false negatives. By addressing challenges of class imbalance, dynamic fraud tactics, the proposed system aims to enhance transaction security, protect consumer interests, and uphold the integrity of electronic payment systems.

1.4 PROJECT OBJECTIVE

The problem statement encapsulates the need to develop a credit card fraud detection system that leverages machine learning algorithms to effectively combat fraudulent activities while minimizing false positives and false negatives. By addressing challenges of class imbalance, dynamic fraud tactics, the proposed system aims to enhance transaction security, protect consumer interests, and uphold the integrity of electronic payment systems.

The primary objective of this project is to develop a robust credit card fraud detection system using machine learning algorithms to effectively identify and prevent fraudulent transactions in real-time. The project aims to achieve the following objectives.

Firstly, to leverage advanced machine learning algorithms, such as Random Forest and Logistic Regression, to analyze transaction data and detect patterns of fraudulent behavior. By using the power of these algorithms, the system seeks to enhance fraud detection accuracy and efficiency, thereby minimizing financial losses for both financial institutions and consumers.

Secondly, to address the challenge of class imbalance in credit card transaction datasets through appropriate sampling techniques and algorithmic approaches. By mitigating the effects of class imbalance, the system aims to improve model performance and reduce the incidence of false positives and false negatives, thereby enhancing the reliability and effectiveness of fraud detection.

Thirdly, to ensure the scalability and real-time processing capabilities of the fraud detection system to handle large volumes of transactions without introducing significant delays or disruptions to the payment processing workflow. The system will be designed to seamlessly integrate into existing transaction processing systems, enabling frictionless deployment and operation within financial institutions' infrastructure.

Lastly, to contribute to the ongoing efforts to combat credit card fraud and uphold the integrity of electronic payment systems. By developing an adaptive, intelligent, and scalable fraud detection system, the project aims to safeguard consumer interests, enhance transaction security, and foster trust in electronic payment systems among an evolving threat landscape.

The project objectives encompass the development of a comprehensive credit card fraud detection solution that leverages machine learning algorithms to detect and prevent fraudulent activities in real-time, thereby mitigating financial losses and protecting the interests of both financial institutions and consumers.

1.5 FEASIBILITY STUDY

Before embarking on the development of a credit card fraud detection system using machine learning algorithms, it is essential to conduct a comprehensive feasibility study to assess the project's viability and potential challenges. This feasibility study encompasses three main types: economical, technical, and procedural feasibility.

1.5.1 Economical Feasibility

The economical feasibility of the project involves evaluating the financial viability and cost-effectiveness of developing and implementing the fraud detection system. This includes estimating the initial investment required for acquiring necessary hardware, software, and expertise, as well as ongoing operational costs such as maintenance, training, and infrastructure upgrades. Additionally, the potential cost savings resulting from reduced fraud losses and improved transaction security must be weighed against the project's expenses to determine its economic feasibility. Cost-benefit analysis and return on investment calculations are essential tools in assessing the project's economic viability and determining whether it aligns with the organization's budgetary constraints and long-term financial goals.

1.5.2 Technical Feasibility

Technical feasibility focuses on assessing the project's technical requirements, capabilities, and constraints. This includes evaluating the availability of requisite technology infrastructure, computing resources, and expertise needed to develop, deploy, and maintain the fraud detection system. Factors such as data availability, quality, and compatibility with existing systems must be considered to ensure seamless integration and interoperability. Additionally, the scalability and performance of machine learning algorithms in handling large volumes of transaction data in real-time need to be evaluated to determine the system's technical feasibility. Prototyping and proof-of-concept experiments can help validate the technical feasibility of the project and identify potential technical challenges that may need to be addressed during implementation.

1.5.3 Procedural Feasibility

Procedural feasibility assesses the feasibility of implementing the fraud detection system within the organization's existing processes and regulatory frameworks. This includes evaluating legal and regulatory compliance requirements, data privacy and security considerations, and organizational policies governing fraud detection. Furthermore, the project's alignment with organizational objectives, stakeholders support, and the availability of human resources and expertise play crucial roles in determining procedural feasibility. Conducting stakeholder consultations, risk assessments, and impact analyses and some other tasks can help identify procedural challenges and mitigate potential barriers to project implementation.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

In the existing credit card fraud detection landscape, financial institutions typically rely on rule-based systems and statistical models to identify potentially fraudulent transactions. These systems analyze transactional data based on predefined rules and thresholds to flag suspicious activities for further investigation by fraud analysts. While these rule-based systems provide a basic level of fraud detection, they often struggle to adapt to evolving fraud tactics and may generate a high number of false positives.

Moreover, the existing system faces challenges related to the imbalanced nature of credit card transaction datasets, where the number of legitimate transactions far exceeds fraudulent ones. This class imbalance can lead to skewed model performance and an increased likelihood of false positives or false negatives, compromising the accuracy and reliability of fraud detection.

Additionally, the existing system may lack real-time processing capabilities and scalability to handle large volumes of transactions in a timely manner. Delays in fraud detection and response times can result in increased financial losses and undermine consumer confidence in electronic payment systems.

While the existing credit card fraud detection system provides a foundational framework for detecting and preventing fraudulent activities, it is limited in its ability to adapt to dynamic fraud tactics, address class imbalance, and provide real-time processing capabilities. There is a need for more advanced and adaptive fraud detection solutions that leverage machine learning algorithms to enhance accuracy, scalability, and efficiency in detecting and preventing credit card fraud.

Furthermore, the existing system may also lack the capability to effectively integrate domain-specific knowledge and contextual information into the fraud detection process. Incorporating additional features such as merchant category codes, transaction types, and geographical information can provide valuable insights into transactional behavior and aid in distinguishing between legitimate and fraudulent transactions. Without robust feature engineering and domain expertise integration, the existing system may overlook crucial indicators of fraudulent activity, leading to undetected fraud instances and increased financial losses. Thus, there is a pressing need for more sophisticated fraud detection systems that can leverage advanced machine learning techniques to integrate domain knowledge effectively and enhance the accuracy and effectiveness of fraud detection algorithms.

2.1.1 Disadvantages of Existing System

- Low Accuracy
- Lack of Adaptability to Evolving Fraud Tactics
- Historical rule dependency
- Limited Scalability
- Inability to Handle Class Imbalance
- Lack of Real-Time Processing Capabilities

2.2 PROPOSED SYSTEM

The proposed credit card fraud detection system builds upon the limitations of the existing system by integrating advanced machine learning algorithms to enhance accuracy, adaptability, and scalability in detecting fraudulent transactions. Leveraging the power of Random Forest (RF) and Logistic Regression(LR) algorithms, the proposed system aims to improve fraud detection capabilities by analyzing transactional data and identifying complex patterns indicative of fraudulent behavior.

In contrast to the rule-based and statistical models used in the existing system, the proposed system utilizes machine learning algorithms to dynamically adapt to evolving fraud tactics. By continuously learning from historical transaction data and identifying emerging fraud patterns, the system can better anticipate and respond to fraudulent activities in real-time, thereby improving overall detection accuracy and reducing false positives.

Furthermore, the proposed system addresses the challenge of class imbalance inherent in credit card transaction datasets by implementing advanced sampling techniques and algorithmic approaches. By balancing the representation of fraudulent and non-fraudulent transactions in the training dataset, the system can mitigate the effects of class imbalance and improve model performance in accurately identifying fraudulent transactions while minimizing false alarms.

The scalability of the proposed system is also enhanced through efficient data processing and model deployment strategies. By leveraging cloud-based computing resources and distributed processing frameworks, the system can handle large volumes of transactions with minimal latency, ensuring timely detection and response to fraudulent activities even during peak transaction periods.

Moreover, the proposed system offers real-time processing capabilities, enabling rapid detection and intervention in fraudulent transactions as they occur. By integrating with existing transaction processing systems and leveraging streaming data processing technologies, the system can analyze transactions in real-time and trigger alerts or interventions as soon as suspicious activity is detected, thereby minimizing potential financial losses for both financial institutions and consumers.

By effectively detecting and preventing fraudulent transactions in real-time, the proposed system instills greater confidence among consumers in the security of electronic payment systems. With reduced instances of fraudulent activities and minimized financial losses, consumers can trust that their credit card transactions are protected, fostering stronger relationships between financial institutions and their customers.

In summary, the proposed credit card fraud detection system represents a significant advancement over the existing system by harnessing the capabilities of machine learning algorithms to improve accuracy, adaptability, scalability, and real-time processing capabilities. By addressing the limitations of the existing system and leveraging advanced techniques, the proposed system aims to enhance transaction security, protect consumer interests, and uphold the integrity of electronic payment systems in the face of evolving fraud threats.

2.2.1 Advantages of Proposed System

- Enhanced Accuracy
- Adaptability to Evolving Fraud Tactics
- Reduced False Positive Rate
- Improved Scalability

2.3 HARDWARE REQUIREMENTS

• Operating System	: Windows 10
• Processor	: Intel processor(I5)
• Hard disk	: 120 GB
• RAM	: 8 GB

2.4 SOFTWARE REQUIREMENTS

• Technology	: Machine Learning
• Language	: python
• Platform	: Jupyter notebook

CHAPTER 3

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

The system architecture for the fraud detection module follows a well-structured approach designed to handle the complexities of detecting fraudulent transactions efficiently. It comprises several interconnected components, each serving a specific purpose within the fraud detection pipeline.

At the core of the architecture lies the "Importing Libraries and Datasets" module, responsible for initializing the system by importing necessary libraries and loading the transaction dataset into memory. This step lays the foundation for subsequent data preprocessing and model training stages.

Following data import, the "Data Preprocessing" module takes charge of preparing the raw transaction data for analysis. This involves tasks such as handling missing values, removing duplicates, and scaling numerical features to ensure uniformity across the dataset. Additionally, feature engineering techniques may be employed to derive new insights from the data, enhancing the effectiveness of the subsequent machine learning models.

Once the data is preprocessed, the focus shifts to the "Model Training and Evaluation" module, where machine learning models are trained on the prepared dataset. Multiple algorithms, such as logistic regression, random forest, or gradient boosting, are trained and fine-tuned using techniques like cross-validation and hyperparameter optimization. The trained models are then evaluated using standard performance metrics to assess their efficacy in identifying fraudulent transactions accurately.

With the trained models in place, the system moves towards real-time transaction detection through the "Transaction Detection" module. Deployed models process incoming transaction data in real-time, scoring each transaction to determine its likelihood of being fraudulent. A predefined threshold is applied to classify transactions as either fraudulent or legitimate, triggering alerts for further investigation when necessary.

In summary, the system architecture for the fraud detection module provides a comprehensive framework for identifying and mitigating fraudulent transactions. By leveraging advanced data preprocessing techniques, machine learning algorithms, and real-time transaction analysis, the architecture empowers organizations to safeguard their financial assets and preserve the integrity of their operations in an ever-evolving threat landscape.

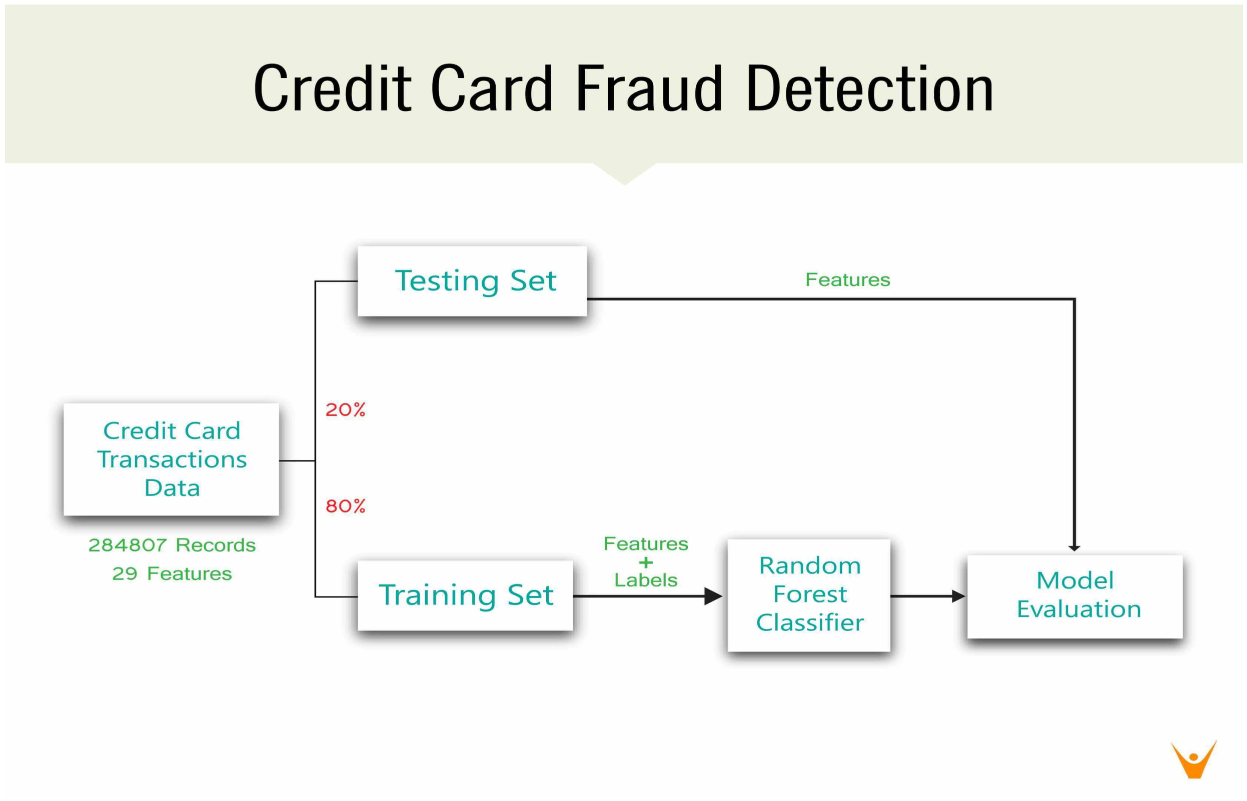


Fig.3.1 System Architecture

3.2 MODULES

The system architecture comprises four primary modules designed to facilitate efficient fraud detection processes. These modules encompass key functionalities from data ingestion to model training and real-time transaction analysis.

The first module, "Importing Libraries and Datasets," serves as the system's foundation, initializing necessary dependencies and loading the transaction dataset into memory. This step ensures that the system has access to essential tools and data required for subsequent processing.

Following data import, the "Data Preprocessing" module takes charge of preparing the raw transaction data for analysis. This involves tasks such as handling missing values, scaling numerical features, and performing feature engineering to enhance the dataset's quality and relevance for model training.

Once the data is preprocessed, the focus shifts to the "Model Training and Evaluation" module, where machine learning models are trained and assessed for their effectiveness in identifying fraudulent transactions. Various algorithms are explored, and model hyperparameters are optimized to achieve the best possible performance.

With trained models in place, the system proceeds to the "Transaction Detection" module, where real-time transaction data is analyzed to detect potentially fraudulent activities. Deployed models process incoming transactions, scoring each transaction based on its likelihood of being fraudulent and triggering alerts for further investigation when necessary.

These four modules collectively form the backbone of the fraud detection system, providing a structured approach to data processing, model training, and real-time analysis, ultimately enabling organizations to mitigate the risks associated with fraudulent transactions.

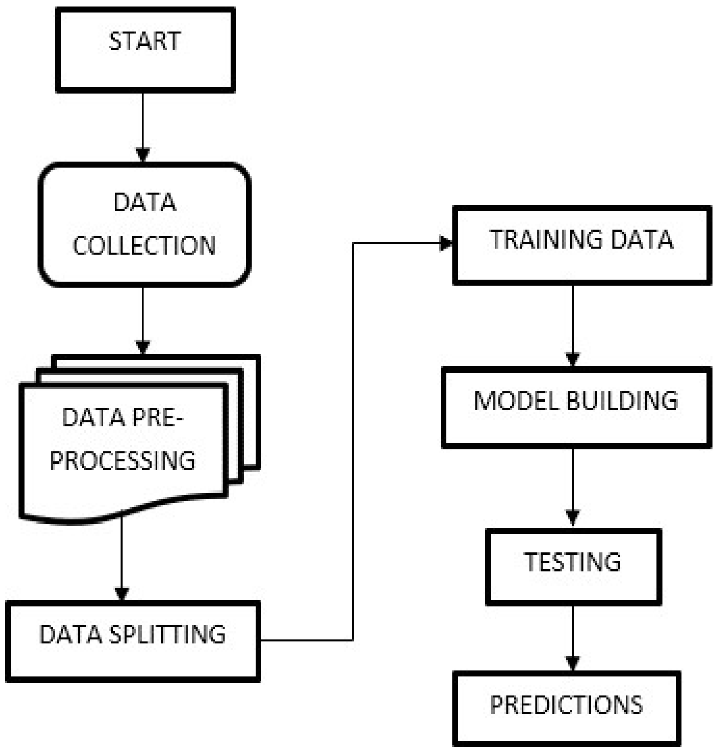


Fig.3.2.1 Block Diagram

3.3 UML DIAGRAMS

Unified Modeling Language is referred to as UML. In the area of object-oriented software engineering, UML is a recognised general-purpose modelling language. The Object Management Group oversees the standard and is responsible for its creation.

The objective is for UML to establish itself as a standard language for modelling object oriented computer programmes. UML now consists of a meta-model and a notation as its two main parts. In the future, UML may potentially be coupled with or added to in the form of a method or process.

The Unified Modeling Language is a standard language for business modelling, non-software systems, and describing, visualising, building, and documenting the artifacts of software systems.

The UML is an one of best engineering approaches that have been effective in simulating huge, complicated systems. While creating objects-oriented software and going through the software development process, the UML is a crucial component. The design of software projects is expressed mostly using graphical notations of UML.

The Unified modeling language diagram, is a standardized general-purpose visual modeling language in the field of Software Engineering. It is used for specifying, visualizing, constructing, and documenting the primary artifacts of the software system. It helps in designing and characterizing, especially those software systems that incorporate the concept of Object orientation. It describes the working of both the software and hardware systems.

Goals:

The following are the UML's primary design objectives:

- Provide consumers access to a ready-to-use, expressive visual modelling language so they can create and trade meaningful models.
- Provide techniques for extending and specialising the fundamental ideas.
- Be sure on certain development methodologies and programming languages.
- Give the modelling language a formal foundation.
- Foster the expansion of the market for Object Oriented tools .
- Encourage notions of higher-level development including partnerships, frameworks, patterns, and components.

Class diagram :

Class diagrams provide a high-level overview of a system’s design, helping to communicate and document the structure of the software. They are a fundamental tool in object-oriented design and play a crucial role in the software development lifecycle.

The class diagram for the fraud detection system illustrates the organization of classes and their relationships within the system. Key classes include "DataPreprocessor" responsible for handling data preprocessing tasks such as handling missing values and feature engineering, "ModelTrainer" for training machine learning models and evaluating their performance, "TransactionDetector" for real-time analysis of transaction data and detection of fraudulent activities using deployed models, based on detected fraudulent transactions.

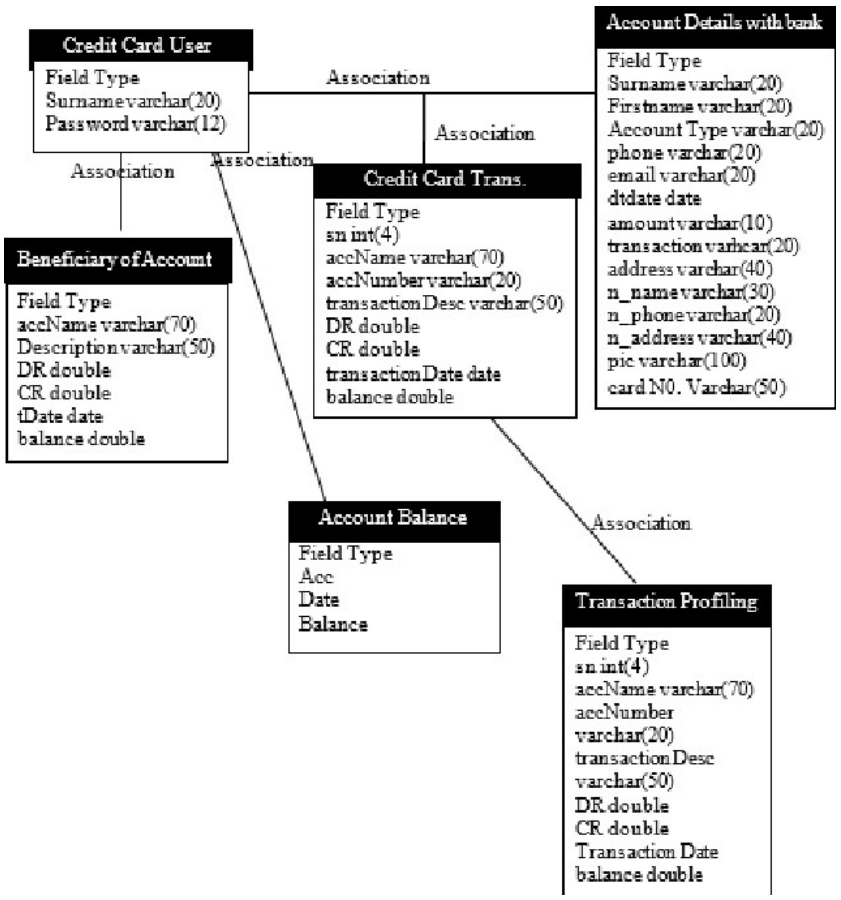


Fig . 3.3.1 Class Diagram

Activity Diagram :

Activity Diagrams are used to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. It is a type of behavioral diagram and we can depict both sequential processing and concurrent processing of activities using an activity diagram ie an activity diagram focuses on the condition of flow and the sequence in which it happens.

The activity diagram illustrates the workflow of the fraud detection process, depicting various activities and their sequential flow. It begins with the "Data Preprocessing" activity, where raw transaction data is processed to handle missing values, scale features, and perform feature engineering. Upon completion, the process moves to the "Model Training" activity, where machine learning models are trained using the preprocessed data. Following model training, the system proceeds to the "Real-time Transaction Analysis" activity, where deployed models analyze incoming transaction data to detect potential fraud. If a transaction is classified as fraudulent, the system triggers the "Alert Generation" activity, generating alerts for further investigation by fraud analysts or security teams. Simultaneously, the "Reporting" activity compiles comprehensive reports summarizing fraud detection activities and performance metrics. Throughout the process, the activities are interconnected by control flows, guiding the sequential execution of tasks within the fraud detection system.

Additionally, the activity diagram emphasizes the iterative and cyclical nature of the fraud detection process, reflecting the continuous refinement and adaptation required to effectively combat fraudulent activities. Feedback loops are incorporated to ensure ongoing optimization of data preprocessing techniques, model training methodologies, and alert generation strategies based on evolving fraud patterns and detection performance. Moreover, the diagram underscores the collaborative nature of fraud detection, highlighting the coordination between automated systems and human intervention for timely response and resolution of fraudulent incidents. By providing a visual representation of the system's workflow and interactions, the activity diagram offers a comprehensive understanding of the fraud detection process, enabling stakeholders to streamline operations, enhance decision-making, and bolster defenses against fraudulent activities.

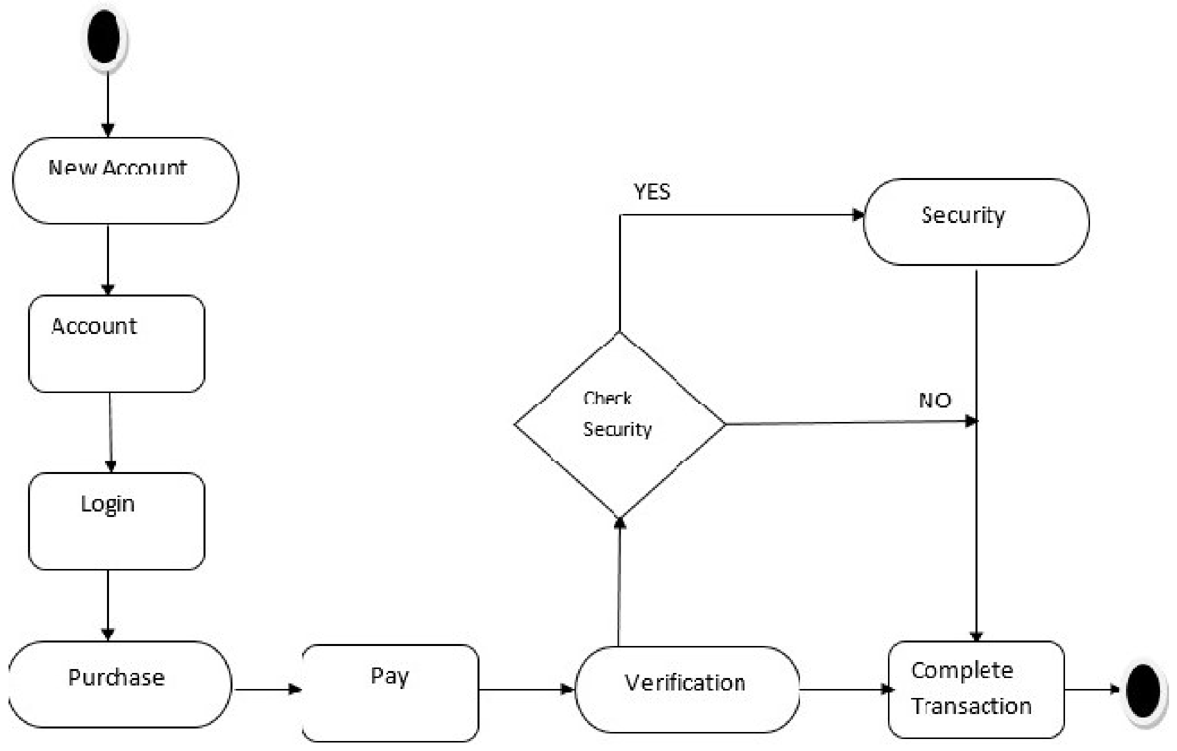


Fig.3.3.2 Activity Diagram

Sequence Diagram:

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

The sequence diagram illustrates the interactions between various components of the fraud detection system during the processing of a transaction. It begins with the "Client" sending a transaction request to the "Preprocessor," indicating the transaction details. Upon receiving the request, the Preprocessor initiates the data preprocessing sequence by handling missing values, scaling features, and performing feature engineering as necessary. Once the data is preprocessed, the Preprocessor forwards the processed data to the "Model Trainer." The Model Trainer then triggers the model training sequence, where machine learning models are trained using the preprocessed data. After training, the trained models are deployed for real-time transaction analysis, indicated by the interaction between the "Transaction Detector" and the deployed models. The Transaction Detector receives incoming transaction data and scores each transaction using the deployed models to detect potential fraud. If a transaction is classified as fraudulent, the Transaction Detector generates an alert and notifies the Alerting System. The Alerting System receives the alert and triggers the alerting sequence, which may involve sending notifications to relevant stakeholders or initiating further investigation procedures. Throughout the sequence, interactions are depicted using numbered arrows, indicating the flow of messages between components and the sequence of actions performed during the fraud detection process.

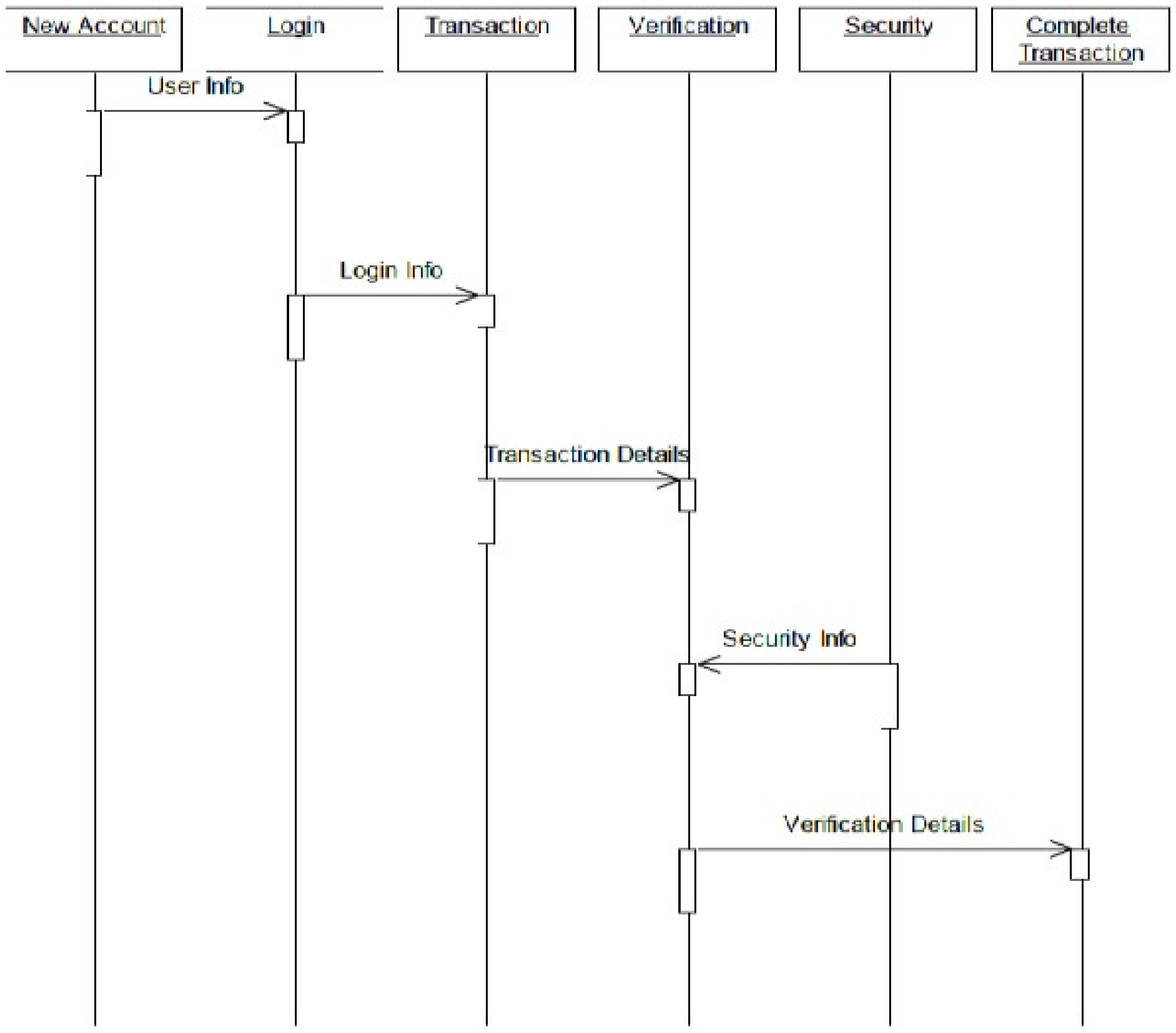


Fig 3.3.3 Sequence Diagram

Component Diagram :

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node.

It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function.

The component diagram provides a high-level overview of the system architecture, illustrating the various components and their interconnections. At the center of the diagram is the Fraud Detection System component, representing the overarching system responsible for detecting and preventing fraudulent activities. This component is further divided into several sub-components, including Data Preprocessing, Model Training, and transaction detection. Each sub-component encapsulates specific functionalities related to data processing, model training, real-time transaction analysis.

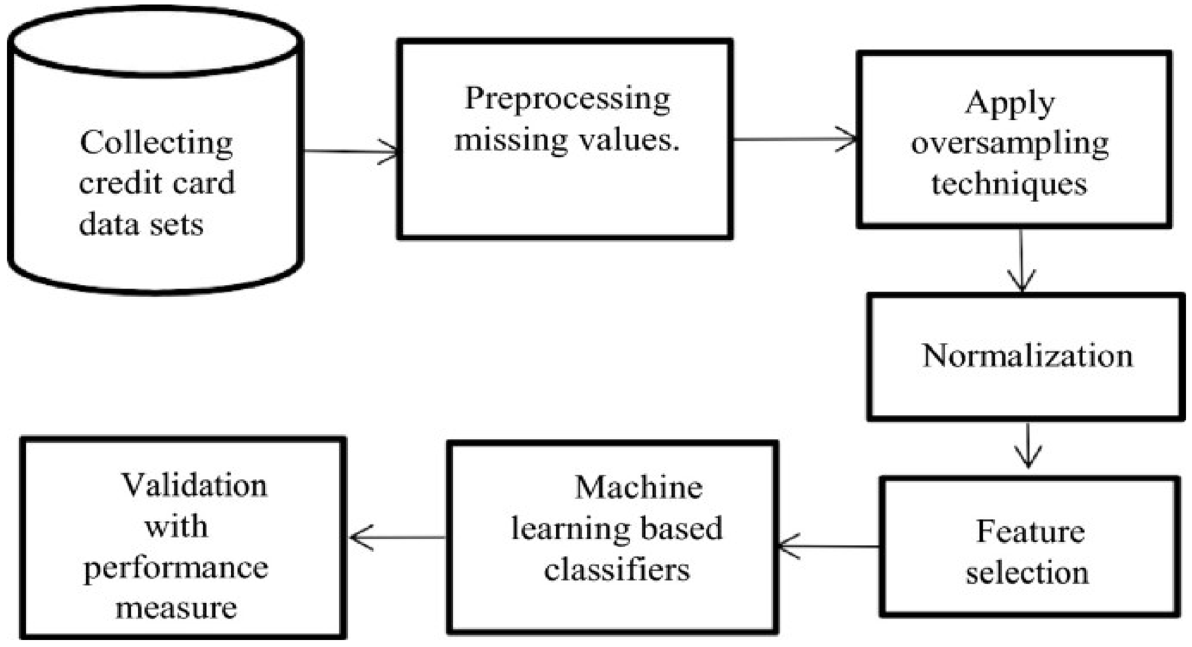

Collecting credit card data sets
Preprocessing missing values.
Apply oversampling techniques
Normalization
Feature selection
Machine learning based classifiers
Validation with performance measure

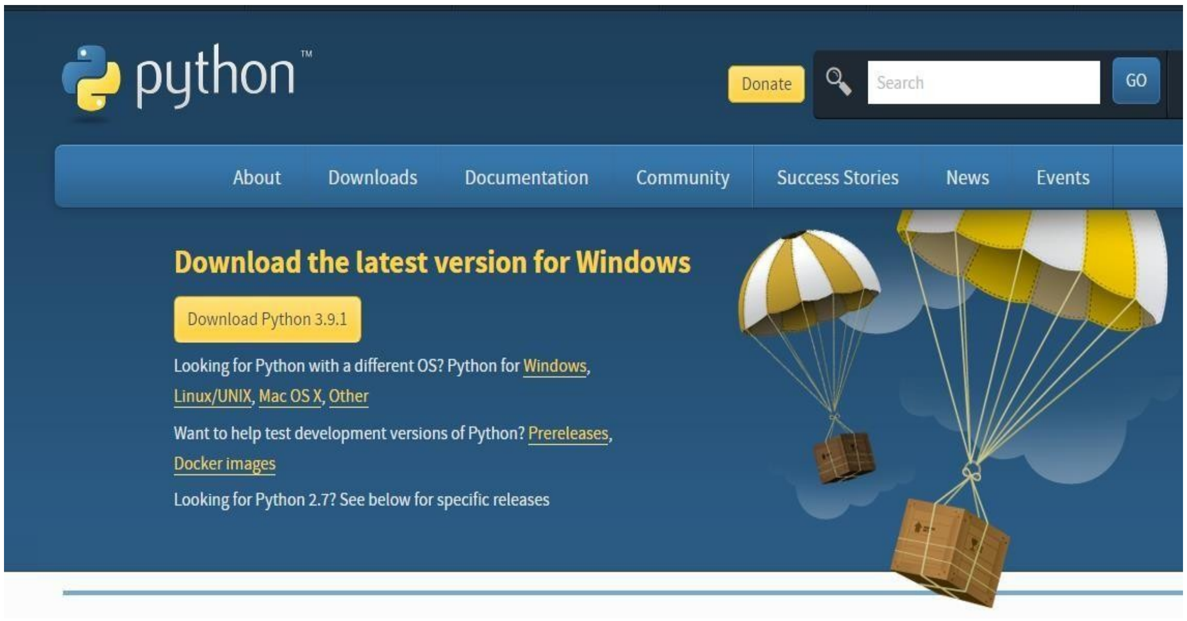
Fig.3.3.4 Component Diagram

CHAPTER 4

IMPLEMENTATION

SOFTWARE INSTALLATION FOR MACHINE LEARNING PROJECTS:

4.1 PYTHON INSTALLATION



Download the latest version for Windows

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Prereleases, Docker images

Looking for Python 2.7? See below for specific releases

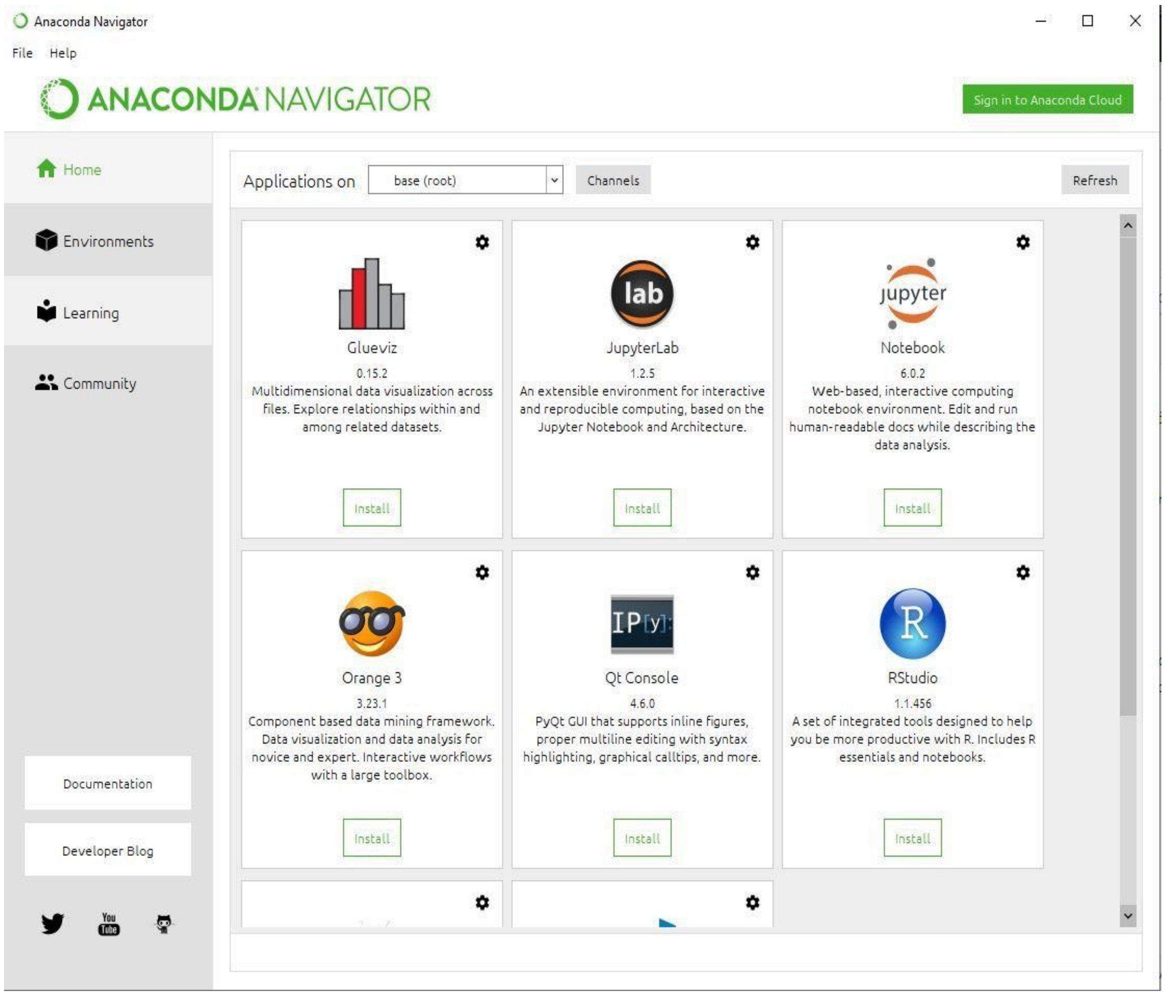
Fig.4.1 Python Installation

1. To download and install Python visit the official website of Python https://www.python.org/downloads/ and choose your version.
2. Once the download is complete, run the exe for install Python. Now click on Install Now.
3. You can see Python installing at this point.
4. When it finishes, you can see a screen that says the Setup was successful. Now click on "Close"
5. Python is successfully installed.

4.2 JUPYTER INSTALLATION

Anaconda is an open-source software that contains Jupyter, spyder, etc. that are used for large data processing , data analytics, heavy scientific computing. Anaconda works for R and python programming languages. Open cv for python will work in spyder. Package versions are managed by the package management system called conda

Launch Anaconda Navigator:



Glueviz
0.15.2
Multidimensional data visualization across files. Explore relationships within and among related datasets.

JupyterLab
1.2.5
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
6.0.2
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

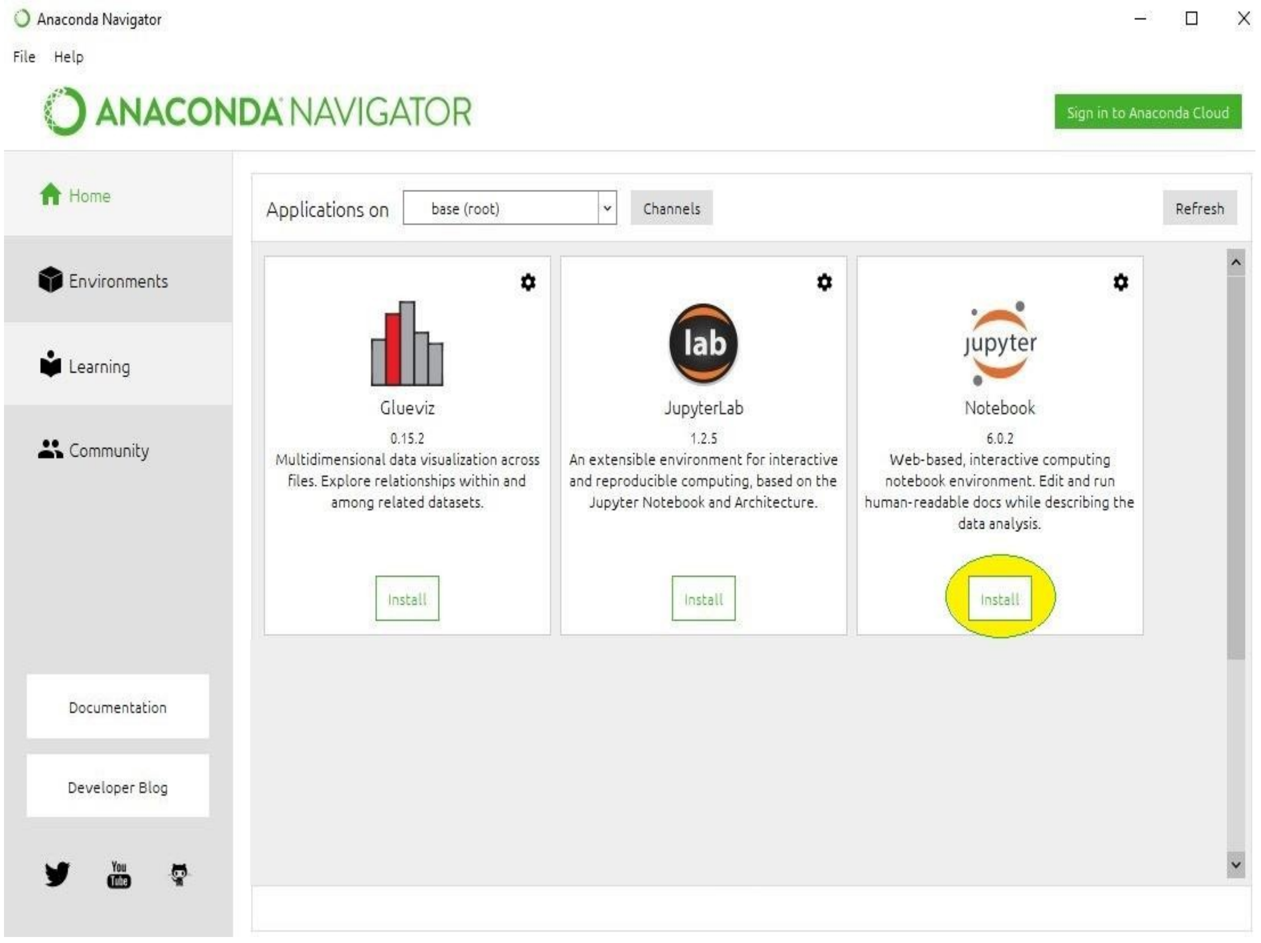
Orange 3
3.23.1
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Qt Console
4.6.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Fig.4.2.1 Launch Anaconda Navigator

Click on the install jupyter notebook button:



Glueviz
0.15.2
Multidimensional data visualization across files. Explore relationships within and among related datasets.

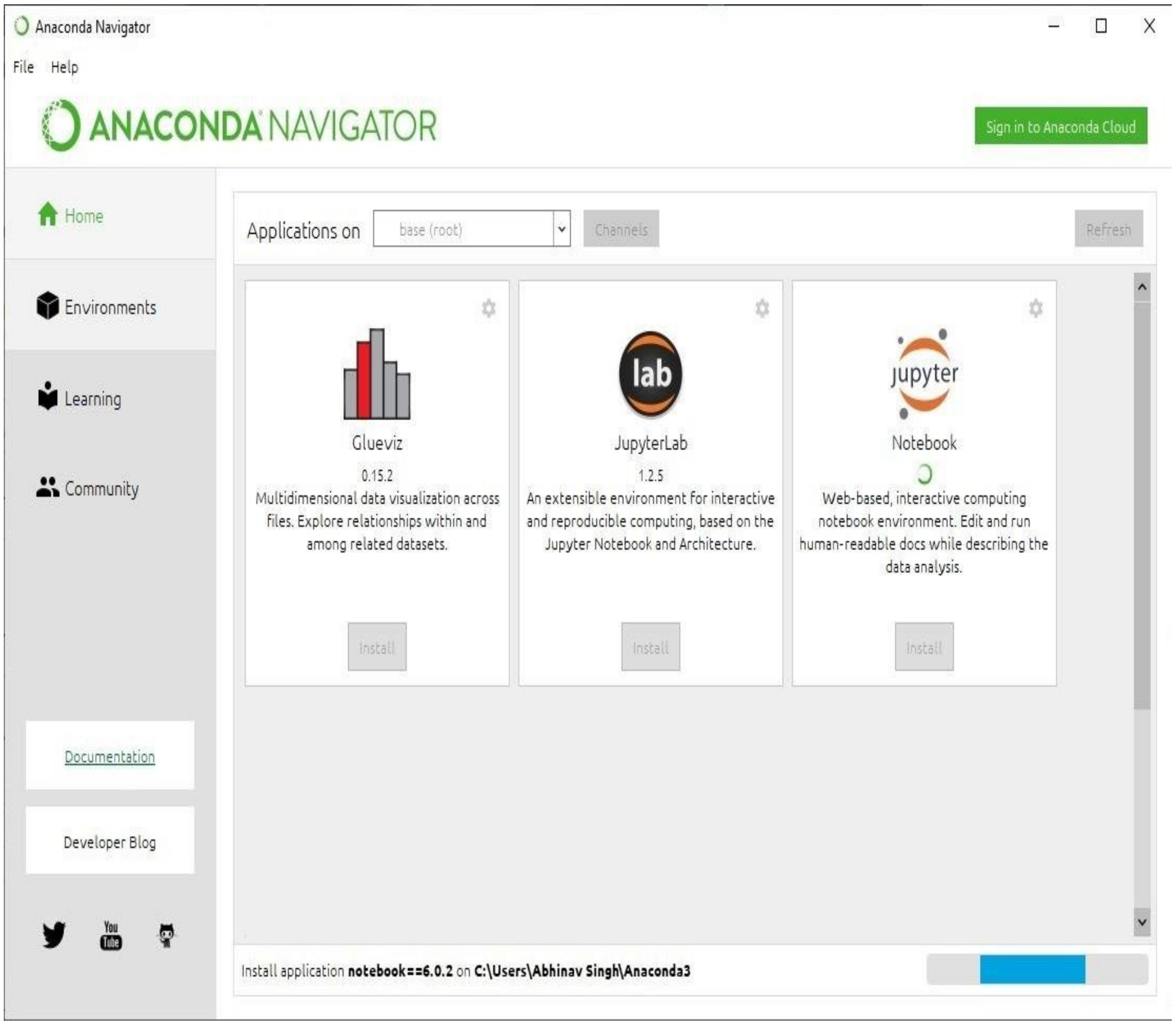
JupyterLab
1.2.5
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
6.0.2
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Fig.4.2.2 Install Jupyter notebook

To install Jupyter Notebook, start by ensuring Python is installed on your system. Then, use pip, Python's package manager, to install Jupyter Notebook with a simple command. Once installed, verify the installation by running Jupyter Notebook from the command line. If successful, a new notebook can be created by selecting "Python 3" from the "New" dropdown menu in the Jupyter interface.

Beginning the installation :



Multidimensional data visualization across files. Explore relationships within and among related datasets.

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

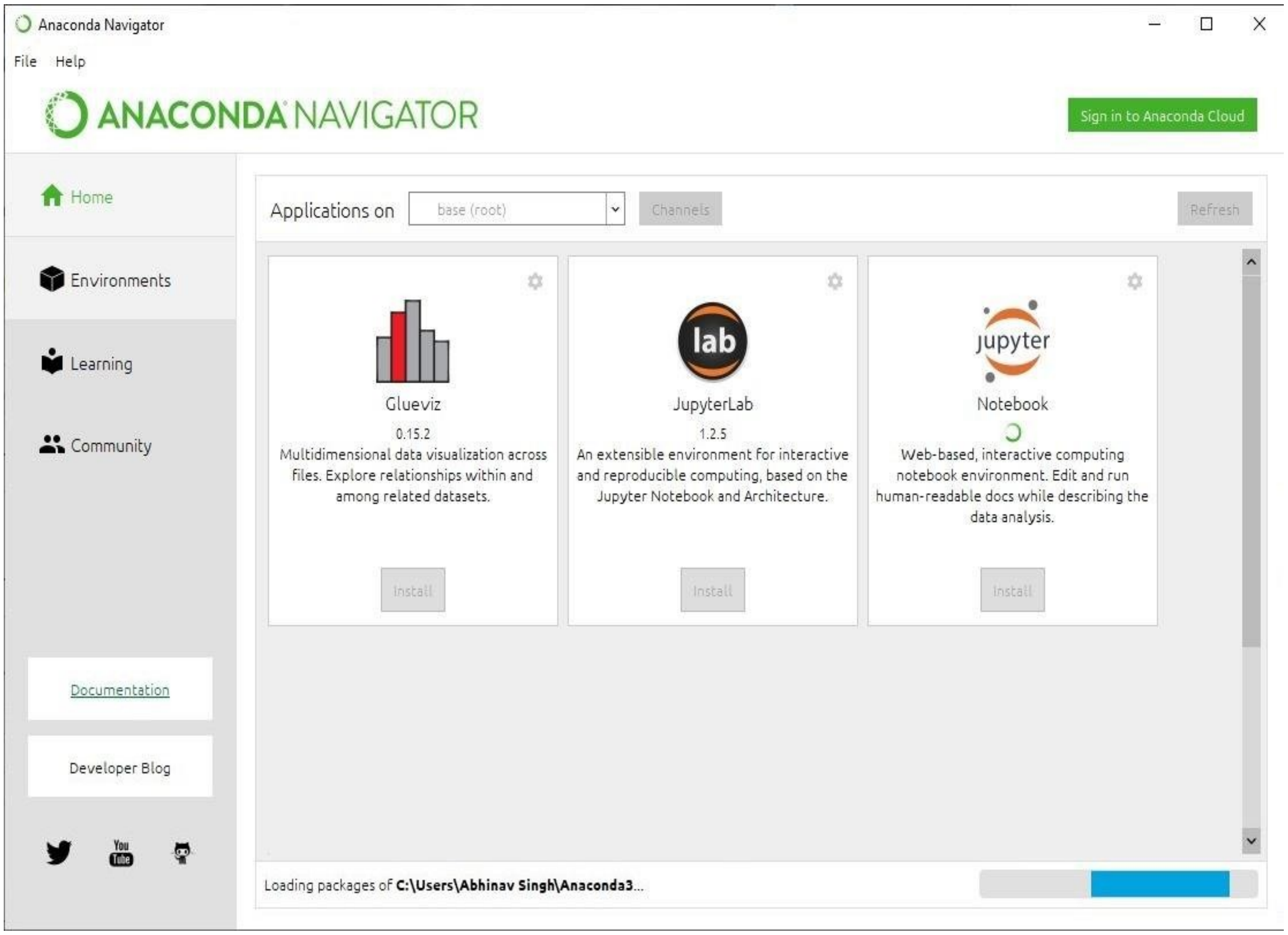
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Install application **notebook==6.0.2** on **C:\Users\Abhinav Singh\Anaconda3**

Fig.4.2.3 Beginning Installation

To install Jupyter Notebook, start by ensuring Python is installed on your system. Then, use pip, Python's package manager, to install Jupyter Notebook with a simple command. Once installed, verify the installation by running Jupyter Notebook from the command line. If successful, a new notebook can be created by selecting "Python 3" from the "New" dropdown menu in the Jupyter interface.

Loading Packages :

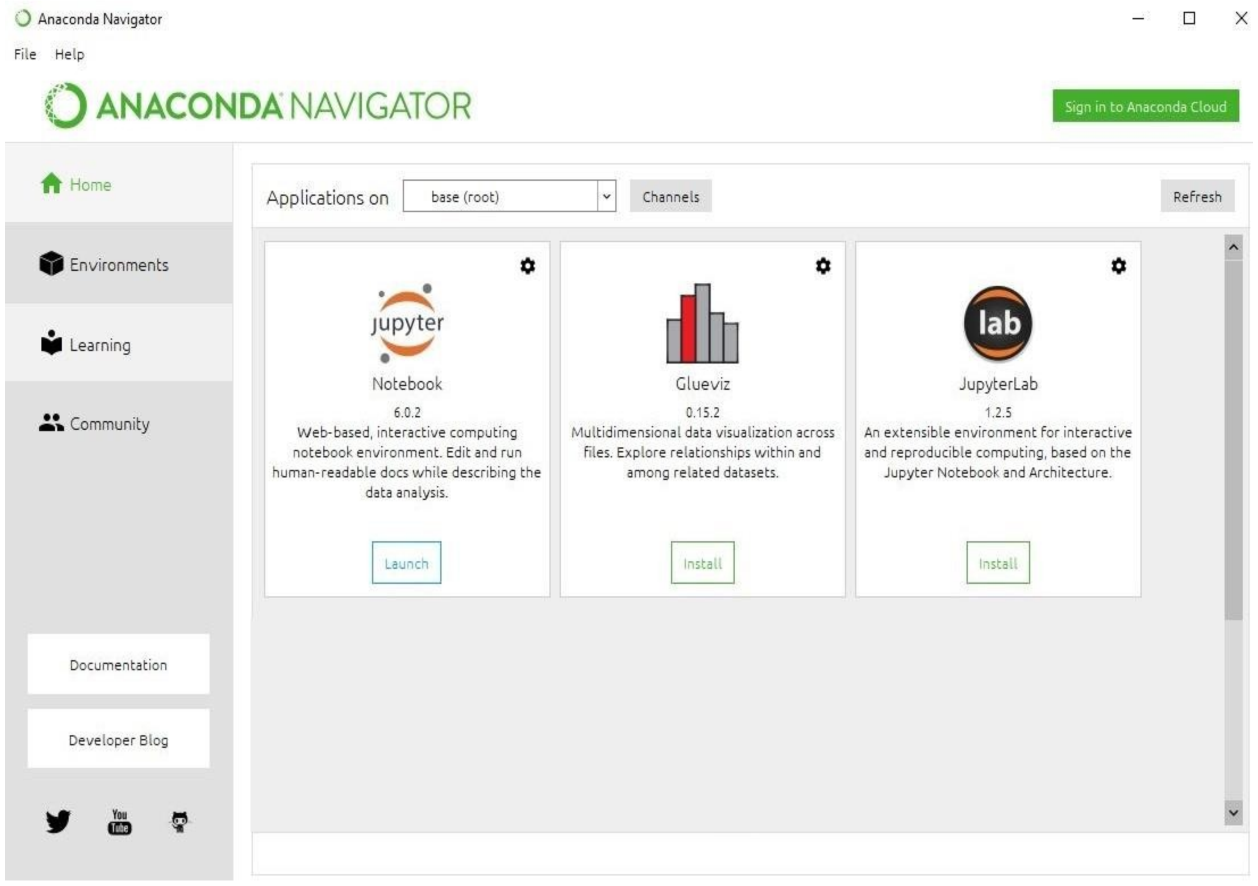

Anaconda Navigator
File Help
ANACONDA NAVIGATOR
Sign in to Anaconda Cloud
Home
Environments
Learning
Community
Documentation
Developer Blog
Applications on
base (root)
Channels
Refresh
Glueviz
0.15.2
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install
JupyterLab
1.2.5
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Install
Notebook
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Install

Loading packages of **C:\Users\Abhinav Singh\Anaconda3**...

Fig.4.2.4 Loading Packages

Loading packages in Python involves importing the necessary libraries or modules into your script or notebook to access their functionalities. This step is crucial as it provides access to pre-written code that simplifies complex tasks, such as data manipulation, visualization, and machine learning. By importing packages like pandas for data manipulation, matplotlib for plotting, and scikit-learn for machine learning, you can leverage their capabilities to analyze data effectively and build predictive models efficiently. This process streamlines development and allows you to focus on solving the problem at hand rather than reinventing the wheel.

Finishing the installation :



Notebook
6.0.2
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

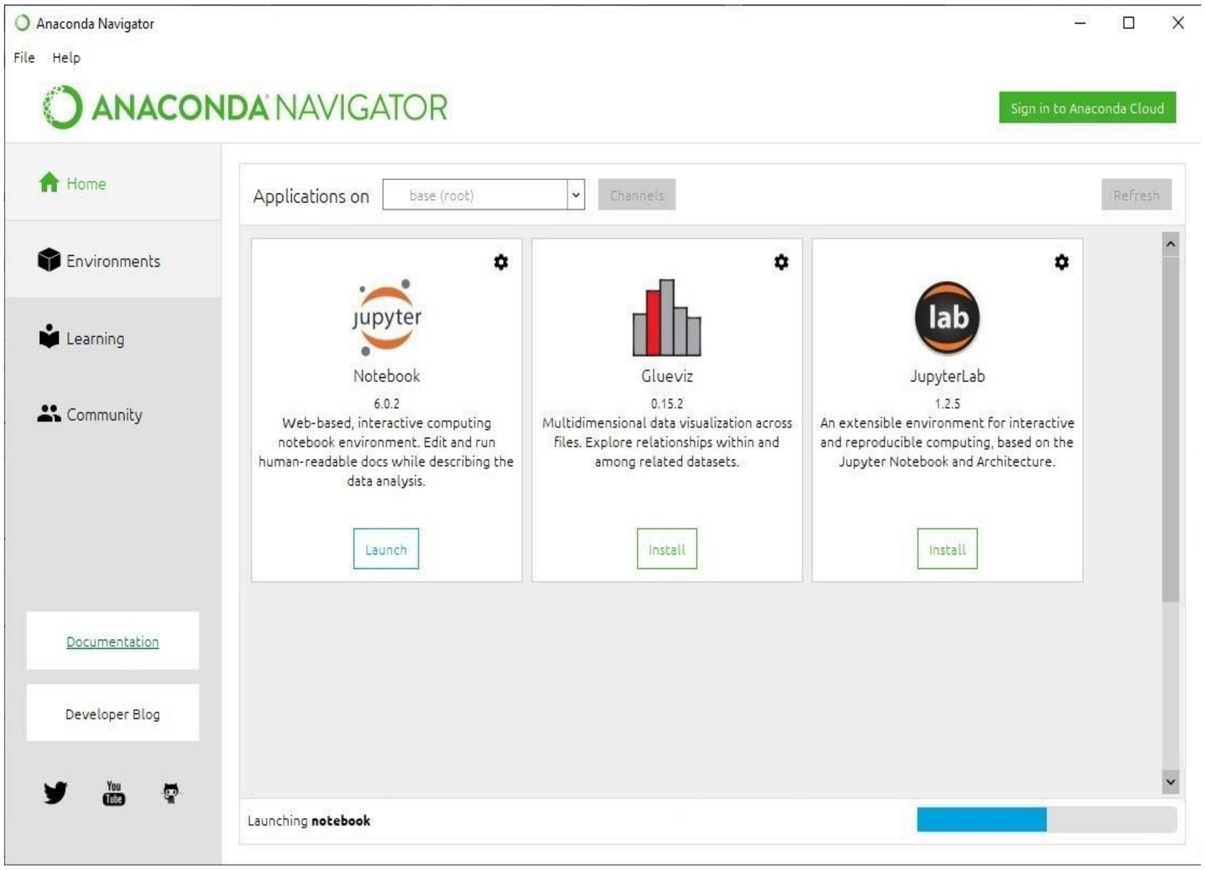
Glueviz
0.15.2
Multidimensional data visualization across files. Explore relationships within and among related datasets.

JupyterLab
1.2.5
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Fig.4.2.5 Finishing Installation

Completing the installation involves verifying that Jupyter Notebook is properly installed and ready to use on your system. After installation, you can ensure everything is set up correctly by launching Jupyter Notebook from the command line or terminal. If successful, this will open the Jupyter interface in your default web browser, confirming that the installation process was completed without errors. With Jupyter Notebook running, you can begin creating and running notebooks to write and execute Python code, enabling you to explore data, prototype algorithms, and collaborate with others seamlessly.

Launching Jupyter :



Notebook
6.0.2
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Glueviz
0.15.2
Multidimensional data visualization across files. Explore relationships within and among related datasets.

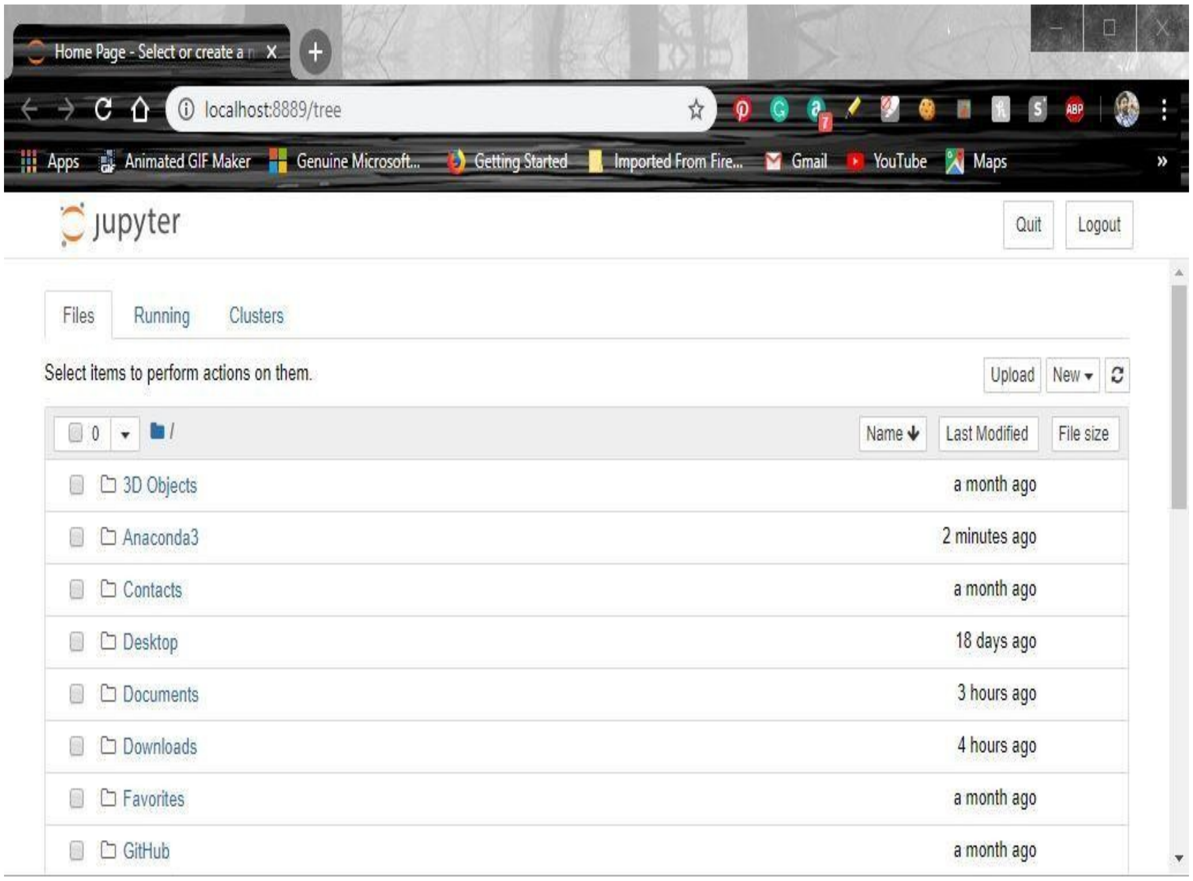
JupyterLab
1.2.5
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Launching **notebook**

Fig.4.2.6 Launching Jupyter

Launching Jupyter Notebook initiates the interactive computing environment, enabling you to create and work with notebooks containing code, visualizations, and explanatory text. By running the command "jupyter notebook" in your terminal or command prompt, the Jupyter server starts, and your default web browser automatically opens to display the Jupyter interface. From here, you can navigate your file system, create new notebooks, and open existing ones. This interface serves as the gateway to your Python environment, providing a user-friendly platform for data exploration, analysis, and collaboration.

Opening Files :



Select items to perform actions on them.

Name	Last Modified
3D Objects	a month ago
Anaconda3	2 minutes ago
Contacts	a month ago
Desktop	18 days ago
Documents	3 hours ago
Downloads	4 hours ago
Favorites	a month ago
GitHub	a month ago

Fig.4.2.7 Open files

Jupyter Notebooks are interactive documents that allow users to write and execute code, view visualizations, and document their analysis in a single environment. To open a Jupyter Notebook file, simply launch the Jupyter Notebook application or use the JupyterLab interface. Once opened, users can explore the contents, execute code cells, and interact with the document to understand the analysis or code provided.

4.3 PROJECT CODE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
credit_card = pd.read_csv('Downloads/creditcard.csv')
credit_card.head()
credit_card.shape
credit_card.tail()
credit_card.info()
credit_card.isnull().sum()
credit_card.duplicated().sum() # checking no. of duplicate values
credit_card.duplicated().any() # checking if theres any duplicate value
df = credit_card.drop_duplicates()
df.duplicated().any()
df.columns
df.shape
df['Class'].value_counts()
df['Class'].unique()
df['Class'].value_counts().plot(kind='bar')
legitimate, fraud = df['Class'].value_counts()
print(f"Legitimate Transactions : {legitimate}")
print(f"Fraud Transactions : {fraud}")
print("This is imbalanced data")
legit = df[df['Class'] == 0]
fraud = df[df['Class'] == 1]
legit
fraud.head()
```

```
legit_new = legit.sample(n=473)
new_df = pd.concat([legit_new, fraud], axis=0)
new_df.shape
new_df['Class'].value_counts()
new_df.groupby('Class').mean()
# spliting the data in  x and y
X = new_df.drop('Class', axis=1)
Y = new_df['Class']
X.shape
Y.shape
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.2, random_state=2)
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train, Y_train)
y_pred = model.predict(X_test)
accuracy_score(y_pred, Y_test)
#Creating X and y labels using the dataset
X=credit_card.drop(['Class'],axis=1)
Y=credit_card['Class']
print("Shape of X:", X.shape)
print("Shape of Y:" , Y.shape)
xData=X.values
yData=Y.values
#Using Skicit-learn to split data into training and testing data
from sklearn.model_selection import train_test_split
xTrain,xTest,yTrain,yTest=train_test_split(xData,yData,test_size=0.2,random_state=42)
print("Shape of Training Data:",xTrain.shape)
print("Shape of Testing Data:",xTest.shape)
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier(n_estimators=10,random_state=0)
rfc.fit(xTrain,yTrain)
```

```
#Predicting on Test set using above trained models
yPred=rfc.predict(xTest)
#Evaluating the Random Forest Classifier
from sklearn.metrics import classification_report,accuracy_score
from sklearn.metrics import precision_score,recall_score
from sklearn.metrics import f1_score,matthews_corrcoef
from sklearn.metrics import confusion_matrix
n_outliers = len(fraud)
n_errors = (yPred != yTest).sum()
print("The model used is Random Forest classifier")
acc = accuracy_score(yTest, yPred)
print("The accuracy is {}".format(acc))
prec = precision_score(yTest, yPred)
print("The precision is {}".format(prec))
rec = recall_score(yTest, yPred)
print("The recall is {}".format(rec))
f1 = f1_score(yTest, yPred)
print("The F1-Score is {}".format(f1))
credit_card.drop(['Class'],axis=1, inplace = True)
credit_card
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=10, random_state=0)
model.fit(xTrain, yTrain)
input_values=input("Enter values seperated by commas :")
test_list=input_values.split(',')
test_list=[float(value) for value in test_list]
test_df = pd.DataFrame(test_list)
test = test_df.transpose()
test_pred = model.predict(test)
if test_pred == 1:
    print("Fraudulent Transaction Detected.")
```

```
else:
    print("Non-Fraudulent Transaction.")
```

App.py :

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
#from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import streamlit as st
# Load data
data = pd.read_csv(r'C:\Users\Mubashera Kouser\Downloads\Credit_Card_Fraud_Detection-
main\Credit_Card_Fraud_Detection\creditcard.csv')
# Separate legitimate and fraudulent transactions
legit = data[data['Class'] == 0]
fraud = data[data['Class'] == 1]
legit_new = legit.sample(n=len(fraud), random_state=2)
new_df = pd.concat([legit_new, fraud], axis=0)
# Split the data into X and Y
X = new_df.drop('Class', axis=1)
Y = new_df['Class']
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.2,stratify=Y, random_state=2)
# Train logistic regression model
model = LogisticRegression()
model.fit(X_train, Y_train)
#Check model performance
train_data_acc = accuracy_score(model.predict(X_train), Y_train)
test_data_acc = accuracy_score(model.predict(X_test), Y_test)
```

```
# Train Random Forest model
#model = RandomForestClassifier(n_estimators=100, random_state=0)
#model.fit(X_train, Y_train)
# Check model performance
#train_data_acc = accuracy_score(model.predict(X_train), Y_train)
#test_data_acc = accuracy_score(model.predict(X_test), Y_test)
# Set page title and favicon
st.set_page_config(page_title="Credit Card Fraud Detection", page_icon=":credit_card:")
# Web app
st.title("Credit Card Fraud Detection Model")
st.markdown("---")
# Input text box
input_data = st.text_input("**Enter all required features values**")
input_splited = input_data.split(',')
# Detect button
detect = st.button("Detect", key="detect_btn", help="Click to detect transaction")
# Perform detection
if detect:
    # Get input feature values
    features = np.array(input_splited, dtype=np.float64)
    # Make prediction
    prediction = model.predict(features.reshape(1,-1))
    # Display result
    if prediction[0] == 0:
        st.success("**Non Fraudulent Transaction**")
# Display result
else:
        st.error("**Fraudulent Transaction**")

# Custom CSS for styling
#credit card fraud detection model
```

```
st.markdown(
    """
    <style>
    body {
        background: black;
        color: white;
    }
    .stTextInput input {
        border: 3px solid orange;
    }
    .stbutton button{
        background-color: black;
        color: white;
        font-weight: bold;
        border-radius: 5px;
    }
    .css-14bw00s button:hover {
        background-color: #333 !important;
    }
    .stMarkdown {
        font-size: 24px;
    }
    </style>
    """,
    unsafe_allow_html=True
)
```

CHAPTER 5

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionalityof components, sub-assemblies, assemblies and/or a finished product It is the process of exercisingsoftware with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

5.1 TYPES OF TESTS :

Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Unit testing is a crucial aspect of software development where individual units or components of a system are tested in isolation to ensure their correctness and reliability. In the context of a fraud detection system, unit testing involves verifying the functionality of individual modules such as data preprocessing, model training, and transaction analysis. Test cases are created to validate the behavior of each unit against expected inputs and outputs, covering both normal and edge cases. By isolating components and testing them independently, unit testing helps identify bugs early in the development cycle, promotes code quality, and facilitates easier debugging and maintenance. Additionally, automated unit tests can be integrated into the continuous integration pipeline, enabling developers to detect regressions quickly and ensure the overall robustness of the fraud detection system.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Integration testing is a critical phase in software development where individual modules or components of a system are combined and tested as a group to ensure they interact correctly and function as intended. In the context of a fraud detection system, integration testing involves verifying the interoperability and data flow between components such as data preprocessing, model training, real-time analysis, and alerting systems. Test cases are designed to validate the integration points and ensure that data is passed correctly between modules, dependencies are resolved accurately, and system interactions behave as expected. By identifying and addressing integration issues early in the development process, integration testing helps mitigate risks associated with system complexity, promotes seamless collaboration between components, and enhances the overall reliability and effectiveness of the fraud detection system.

Functional testing :

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation. Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.
Systems/Procedures	: interfacing systems or procedures must be invoked

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current testsis determined.

5.2 SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions andflows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is the purpose. Itisused to test areas that cannot be reached from a black box level

White box testing is a software testing technique that focuses on examining the internal structure and logic of a system's code. In white box testing, testers have access to the source code and use this knowledge to design test cases that exercise different paths, conditions, and branches within the code. This allows for thorough coverage of code statements, branches, and paths, ensuring that all aspects of the code are tested. Techniques such as statement coverage, branch coverage, and path coverage are commonly used in white box testing to ensure comprehensive testing of the codebase. White box testing is particularly useful for detecting logical errors, ensuring code quality, and validating the correctness of individual functions or modules within the system to increase efficiency.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Black box testing is a software testing approach where testers evaluate the functionality of a system without knowledge of its internal implementation details. Instead of focusing on the internal code structure, black box testers examine the system's inputs and outputs, as well as its behavior in response to different inputs. Test cases are designed based on the system's requirements and specifications, with the goal of uncovering any deviations from expected behavior. Black box testing is effective at identifying issues such as incorrect functionality, missing features, and usability problems from the end user's perspective. It allows testers to assess the system's functionality independently of its internal design, making it suitable for validating system behavior against specified requirements and ensuring that the system meets user expectations.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases. Test strategy and approach Field testing will be performed manually and functional tests will be written in detail. Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Integration testing is a critical phase in software development where individual modules or components of a system are combined and tested as a group to ensure they interact correctly and function as intended. In the context of a fraud detection system, integration testing involves verifying the interoperability and data flow between components such as data preprocessing, model training, real-time analysis, and alerting systems. Test cases are designed to validate the integration points and ensure that data is passed correctly between modules, dependencies are resolved accurately, and system interactions behave as expected. By identifying and addressing integration issues early in the development process, integration testing helps mitigate risks associated with system complexity, promotes seamless collaboration between components, and enhances the overall reliability and effectiveness of the fraud detection system.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

Acceptance testing is the final phase of the software testing process, where the software is evaluated to determine whether it meets the acceptance criteria set by stakeholders and end users. This type of testing verifies that the software fulfills the specified requirements and is ready for deployment. Acceptance testing can take various forms, including user acceptance testing , business acceptance testing , and operational acceptance testing , depending on the stakeholders involved and the scope of testing. During acceptance testing, real-world scenarios and data are used to validate the software's functionality, usability, performance, and compliance with business objectives. Successful completion of acceptance testing signifies that the software is fit for its intended purpose and ready to be released to users or customers. It serves as a critical validation step, ensuring that the software meets user expectations and delivers value to the organization.

Acceptance testing serves as a bridge between the development team and stakeholders, providing a platform for communication and collaboration. During this phase, stakeholders actively participate in testing the software to ensure that it aligns with their needs and expectations. Feedback gathered from acceptance testing helps identify any discrepancies between the software's functionality and the stakeholders' requirements, enabling necessary adjustments to be made before deployment. Additionally, acceptance testing fosters trust and confidence in the software among stakeholders, as it validates that their input has been incorporated and their concerns addressed. By facilitating open communication and mutual understanding, acceptance testing promotes a smoother transition from development to production and ultimately contributes to the success of the software project.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

5.3 SCREENSHOTS

In our project, we are detecting whether the transaction is fraud or non fraud

5.3.1 Home Page

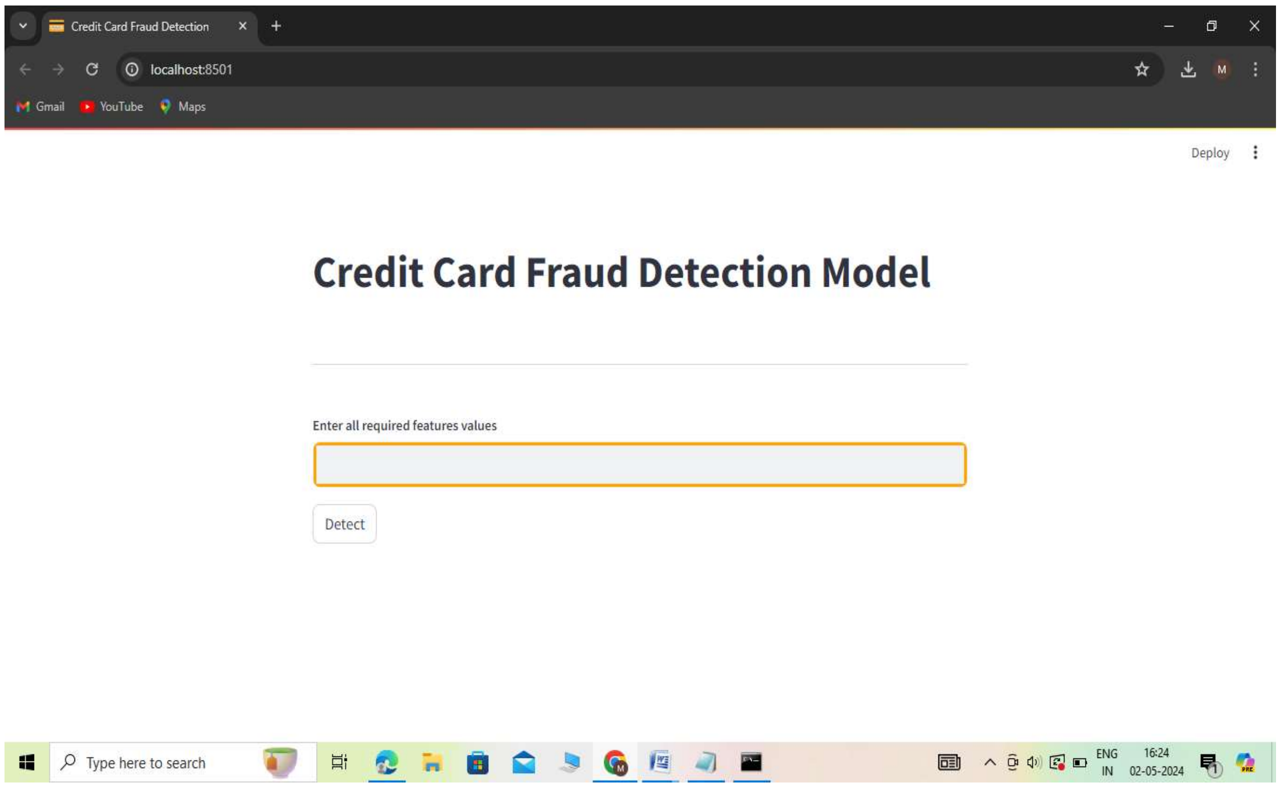
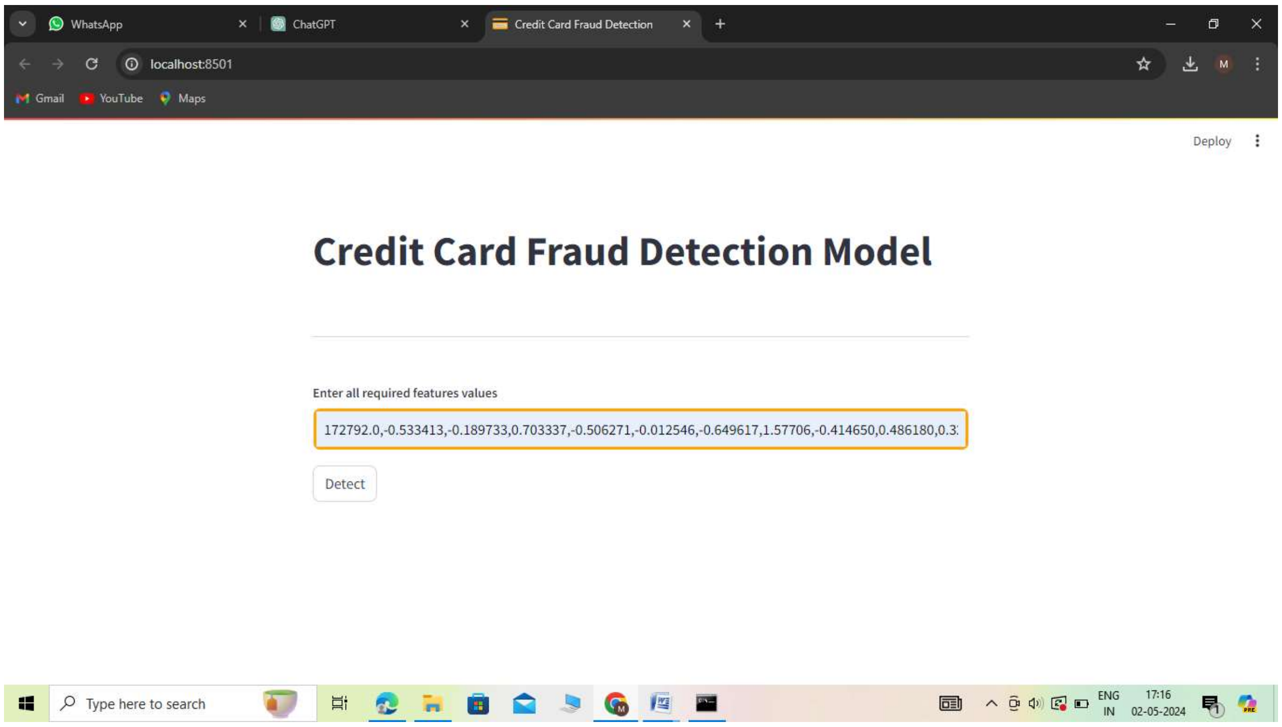

Credit Card Fraud Detection
localhost:8501
Gmail
YouTube
Maps
Deploy
Credit Card Fraud Detection Model
Enter all required features values
Detect
Type here to search
ENG IN
16:24
02-05-2024

Fig.5.3.1 Home Page

5.3.2 Input Page



Credit Card Fraud Detection Model

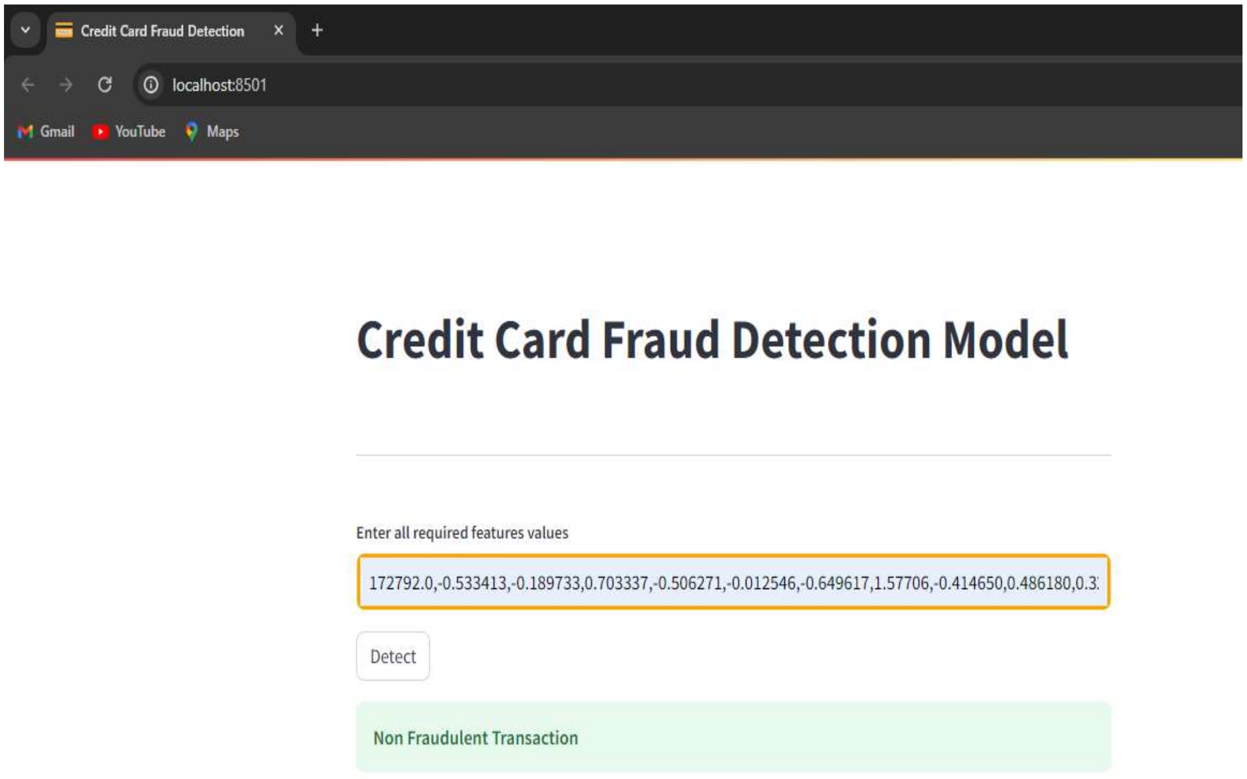
Enter all required features values

172792.0,-0.533413,-0.189733,0.703337,-0.506271,-0.012546,-0.649617,1.57706,-0.414650,0.486180,0.3

Detect

Fig.5.3.2 Input Page

5.3.3 Output page (Non Fraud)



Credit Card Fraud Detection Model

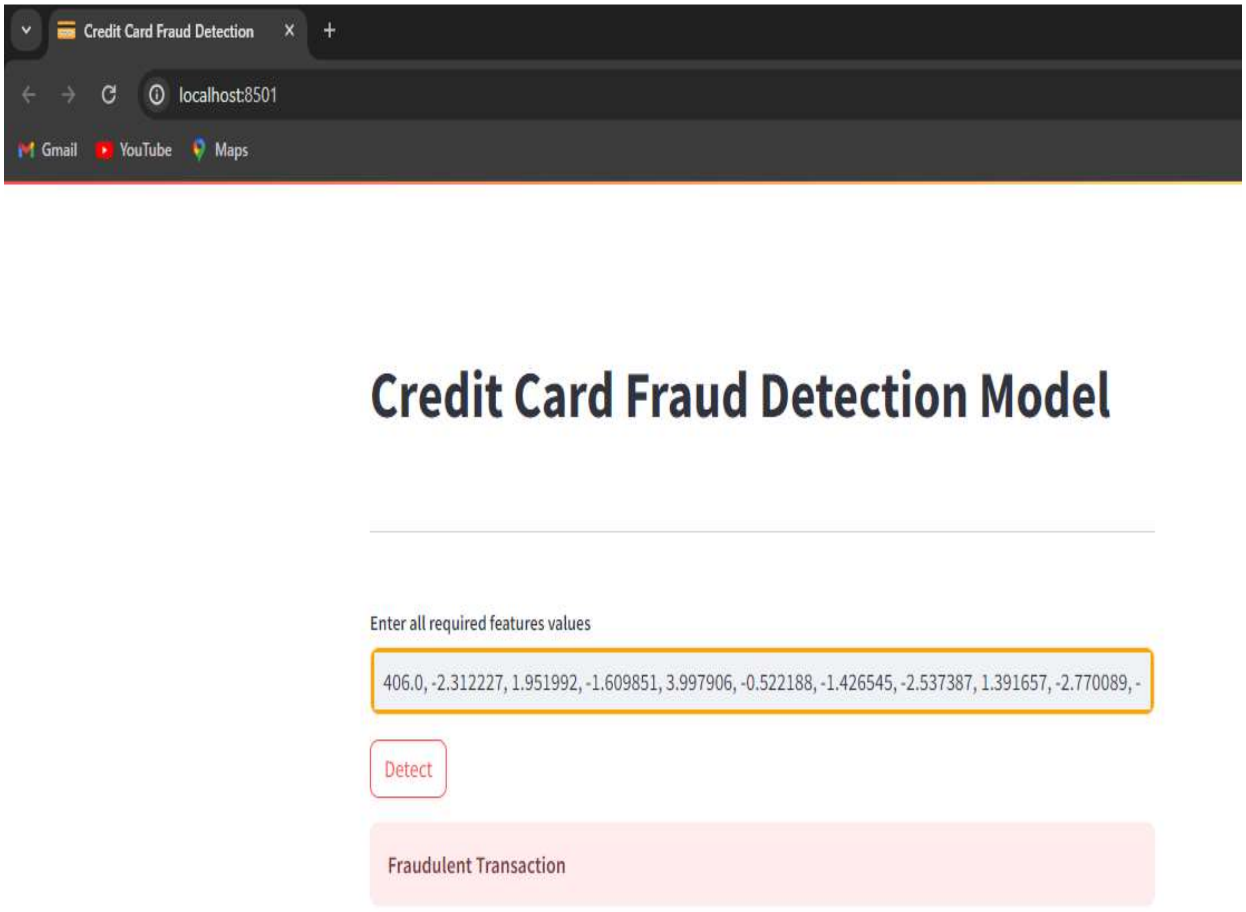
Enter all required features values

172792.0,-0.533413,-0.189733,0.703337,-0.506271,-0.012546,-0.649617,1.57706,-0.414650,0.486180,0.3

Non Fraudulent Transaction

Fig.5.3.3 Output Page(Non-fraud)

5.3.4 Output Page (Fraud)



Credit Card Fraud Detection Model

Enter all required features values

406.0, -2.312227, 1.951992, -1.609851, 3.997906, -0.522188, -1.426545, -2.537387, 1.391657, -2.770089, -

Fraudulent Transaction

Fig.5.3.4 Output Page(Fraud)

CHAPTER 6

CONCLUSION

6.1 CONCLUSION

In conclusion, the development of a fraud detection system represents a crucial endeavor in safeguarding financial transactions and preserving the integrity of business operations. Through the meticulous design, implementation, and testing of various components, our project has addressed the pressing need for robust mechanisms to detect and prevent fraudulent activities effectively. By leveraging advanced techniques in data preprocessing, machine learning, and real-time analysis, we have created a comprehensive solution capable of identifying suspicious transactions with high accuracy and reliability.

Throughout the project, a strong emphasis has been placed on collaboration and communication among team members, stakeholders, and end users. Continuous engagement with stakeholders has ensured that the system's requirements are accurately captured and translated into actionable functionalities. Moreover, user feedback and testing have played a pivotal role in refining the system's capabilities and enhancing its usability, thereby fostering a sense of ownership and trust among stakeholders.

The successful implementation of the fraud detection system underscores the importance of employing a systematic approach to software development, encompassing requirements analysis, design, implementation, testing, and deployment. Each phase of the development lifecycle has been meticulously executed, with careful attention to detail and adherence to best practices. As a result, the system is well-positioned to meet the evolving needs of the organization and adapt to changing fraud patterns and regulatory requirements.

Looking ahead, the fraud detection system holds immense potential for further enhancements and refinements. Continuous monitoring and evaluation of the system's performance will be essential to identify areas for improvement and ensure its continued effectiveness in detecting fraudulent activities. Additionally, ongoing collaboration with stakeholders and end users will facilitate the incorporation of new features, integration with external systems, and compliance with emerging standards and regulations.

Beyond its immediate impact on fraud prevention, the fraud detection system has broader implications for organizational efficiency, risk management, and customer trust. By proactively identifying and mitigating fraudulent activities, the system helps minimize financial losses, reduce operational disruptions, and enhance the overall security posture of the organization. Moreover, its role in promoting transparency, accountability, and compliance underscores its value as a strategic asset for the organization.

In conclusion, the successful completion of the fraud detection project marks a significant milestone in our journey towards enhancing financial security and integrity. Through collaborative efforts, innovative solutions, and a commitment to excellence, we have developed a robust and scalable system that empowers organizations to combat fraud effectively. As we celebrate this achievement, we remain dedicated to advancing the capabilities of the fraud detection system and delivering tangible value to our stakeholders and end users.

In closing, I would like to express my gratitude to all team members, stakeholders, and supporters who have contributed to the success of this project. Your dedication, expertise, and collaboration have been instrumental in bringing our vision to fruition and making a positive impact in the fight against fraud. As we embark on the next phase of our journey, let us continue to innovate, collaborate, and strive for excellence in all our endeavors. Together, we can build a safer, more secure future for organizations and individuals alike.

6.2 SCOPE FOR THE FUTURE WORK

As we envision the future scope of the fraud detection system with our algorithms at the forefront, several avenues for further development and enhancement emerge. Firstly, through continuous refinement and optimization of our machine learning algorithms, such as logistic regression, random forest, and gradient boosting, we can improve the system's accuracy and efficiency in detecting fraudulent activities. By fine-tuning model hyperparameters and exploring novel feature selection techniques, we can enhance the discriminatory power of our algorithms and reduce false positive rates, thereby minimizing the impact of fraudulent transactions on organizations.

Moreover, advancements in deep learning techniques, such as convolutional neural networks and recurrent neural networks , present opportunities to augment our fraud detection capabilities. By leveraging the hierarchical representations learned by deep neural networks, we can extract intricate patterns and relationships from transactional data, enabling more precise detection of sophisticated fraud schemes. Additionally, the integration of anomaly detection algorithms, such as autoencoders and Isolation Forests, can complement our existing models by identifying unusual patterns and outliers indicative of fraudulent behavior.

Furthermore, the incorporation of ensemble learning techniques, such as stacking and boosting, can enhance the robustness and resilience of our fraud detection system. By combining multiple base learners, each with different strengths and weaknesses, we can create ensemble models that outperform individual algorithms and exhibit greater generalization ability. Additionally, the utilization of model interpretability techniques, such as SHapley Additive exPlanations values and Local Interpretable Model-agnostic Explanations, can provide insights into the decision-making process of our algorithms, enabling stakeholders to understand and trust the system's predictions.

Additionally, the integration of real-time streaming analytics capabilities can enable our fraud detection system to process transactional data in real-time, allowing for immediate detection and response to fraudulent activities as they occur. By leveraging streaming algorithms, such as the Vowpal online learning framework and Apache Flink's stream processing engine, we can analyze incoming data streams and adapt our models dynamically to changing patterns and trends. Moreover, the integration of graph-based algorithms, such as PageRank and community detection algorithms, can enhance our ability to detect organized fraud rings and collusion networks operating across multiple accounts and entities.

Furthermore, the exploration of federated learning techniques can enable collaborative model training across distributed datasets without compromising data privacy and security. By leveraging federated learning frameworks, such as TensorFlow ,we can aggregate model updates from multiple sources while preserving the confidentiality of sensitive transactional data.

The future scope of the fraud detection system with our algorithms lies in continuous innovation and adaptation to emerging technologies and trends. By advancing our machine learning algorithms, exploring new avenues such as deep learning and ensemble learning, and embracing real-time streaming analytics and federated learning, we can enhance the accuracy, efficiency, and scalability of our fraud detection system very efficiently.

REFERENCES:

[1] Taha, Altyeb & Malebary, Sharaf. (2020). An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine. IEEE Access. 8. 25579-25587.

[2] Assaghir, Zainab & Taher, Yehia & Haque, Rafiqul & Hacid, Mohand-Said & Zeineddine, Hassan. (2019). An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection. IEEE Access.

[3] L. Meneghetti, M. Terzi, S. Del Favero, G. A Susto, C. Cobelli, “DataDriven Anomaly Recognition for Unsupervised Model-Free Fault Detection in Artificial Pancreas”, IEEE Transactions On Control Systems Technology, (2018) pp. 1-15

[4] F. Carcillo, Y.-A. Le Borgne and O. Caelen et al., “Combining unsupervised and supervised learning in credit card fraud detection”, Information Sciences, Elsevier (2019), pp. 1-15.

[5] Ashphak, Mr. & Singh, Tejpal & Sinhal, Dr. Amit. (2012). A Survey of Fraud Detection System using Hidden Markov Model for Credit Card Application Prof. Amit Sinhal.

[6] Renjith, Shini. (2018). Detection of Fraudulent Sellers in Online Marketplaces using Support Vector Machine Approach. International Journal of Engineering Trends and Technology. 57. 48-53. 10.14445/22315381/IJETT-V57P210.

[7] Saputra, Adi & Suharjito, Suharjito. (2019). Fraud Detection using Machine Learning in eCommerce. 10.14569/IJACSA.2019.0100943.

[8] A. K. Rai and R. K. Dwivedi, "Fraud Detection in Credit Card Data using Unsupervised Machine Learning Based Scheme," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2020, pp. 421-426, doi: 10.1109/ICESC48915.2020.9155615.

[9] John O. Awoyemi, Adebayo O. Adetunmbi, Samuel A. Oluwadare et al., “Credit card fraud detection using Machine Learning Techniques: A Comparative Analysis”, IEEE, 2017.

APPENDICES

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

B: https://ieeexplore.ieee.org/document/9121114

C:https://www.researchgate.net/publication/336800562_Credit_Card_Fraud_Detection_using_Machine_Learning_and_Data_Science

D: https://www.sciencedirect.com/science/article/pii/S187705092030065X

E: https://github.com/topics/credit-card-fraud-detection

F: https://www.ijert.org/credit-card-fraud-detection-using-machine-learning-algorithms

STUDENT BIO-DATA

STUDENT - 1

Name: B.MubasheraKouser

Father Name : B.Shabbir

Roll.No : 2010104

Date of Birth : 01-06-2003

Nationality : Indian

Communication Address:

Town/Village : Anantapur	**Mandal: Anantapur**	**District: Anantapur**

PIN Code : 515001

Ph.No : 6302418976

e-mail : mubasherakouser111@gmail.com

Permanent Address:

Town/Village : Anantapur	**Mandal: Anantapur**	**District: Anantapur**

PIN Code :515001

Ph.No :6302418976

e-mail : mubasherakouser111@gmail.com

Qualifications :

B.tech in Computer Science and Engineering

Technical Skills :

C, Java, Html, CSS,JavaScript

Area of Interest :

Problem solving, Leadership

Declaration :

I hereby declared that the information furnished above is true to the best of my knowledge.

B.Mubashera Kouser

Signature

STUDENT - 2

Name	**: G. Kavitha**
Father Name	**: G.Chinnappa**
Roll.No	**: 2010113**
Date of Birth	**: 27-09-2002**
Nationality	**:Indian**

Communication Address:

Town/Village : Nagiri Mandal: Sambepalli District: Kadapa

PIN Code	**:516257**
Ph.No	**:7815926086**
e-mail	**:galiveetikavitha287@gmail.com**

Permanent Address:

Town/Village : Nagiri Mandal: Sambepalli District: Kadapa

PIN Code	**:516257**
Ph.No	**:7815926086**
e-mail	**: galiveetikavitha287@gmail.com**

Qualifications :

B.tech in Computer Science and Engineering

Technical Skills :

Python, C, Web development.

Area of Interest :

Reading books, Learning new skills

Declaration :

I hereby that declare that the information furnished above is true to the best of my knowledge.

G.Kavitha
Signature

STUDENT-3

Name	**: K.Manjula**
Father Name	**: K.VenkataRamanappa**
Roll.No	**: 2010118**
Date of Birth	**: 16-08-2003**
Nationality	**: Indian**

Communication Address:

Town/Village : Sanipalli Mandal: Roddam District: Anantapur

PIN Code	**: 515124**
Ph.No	**: 9703046329**
e-mail	**: varshinijcteam@gmail.com**

Permanent Address:

Town/Village : Sanipalli Mandal: Roddam District: Anantapur

PIN Code	**: 515124**
Ph.No	**: 9703046329**
e-mail	**: varshinijcteam@gmail.com**

Qualifications :

B.Tech in Computer Science and Engineering

Technical Skills :

C, Python, Web development

Area of Interest :

Learning new skills,Reading books.

Declaration :

I hereby declared the above information is true to the best of my knowledge

K.Manjula
Signature