FRAUD CREDIT CARD DETECTION USING MACHINE LEARNING

Minor project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

NALLURI KARTHIK	(20UECS0659)	**(VTU15337)**
CHEELLA BALAJI	(20UECS0193)	**(VTU17056)**
K.LOKESH	(20UECS0452)	**(VTU18313)**

Under the guidance of
MS.T.GANGALAKSHMI,M.E.
ASSISTANT PROFESSOR

SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2023

FRAUD CREDIT CARD DETECTION USING MACHINE LEARNING

Minor project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

NALLURI KARTHIK	(20UECS0659)	**(VTU15337)**
CHEELLA BALAJI	(20UECS0193)	**(VTU17056)**
K.LOKESH	(20UECS0452)	**(VTU18313)**

Under the guidance of
MS.T.GANGALAKSHMI,M.E.
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2023

CERTIFICATE

It is certified that the work contained in the project report titled FRAUD CREDIT CARD DETECTION USING MACHINE LEARNING by NALLURI KARTHIK (20UECS0659), CHEELLA BALAJI (20UECS0193), K.LOKESH (20UECS0452) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Ms.T.Gangalakshmi,M.E.
Assistant Professor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2023

Signature of Head of the Department
Dr. M. S. Muralidhar,M.E.,PH.D.
Associate Professor & HOD
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2023

Signature of the Dean
Dr. V. Srinivasa Rao,PH.D
Professor & Dean
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2023

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

NALLURI KARTHIK
Date: / /

CHEELLA BALAJI
Date: / /

K.LOKESH
Date: / /

APPROVAL SHEET

This project report entitled FRAUD CREDIT CARD DETECTION USING MACHINE LEARNING by NALLURI KARTHIK (20UECS0659), CHEELLA BALAJI (20UECS0193), K.LOKESH (20UECS0452) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners	**Supervisor**
	MS.T.GANGALAKSHMI, M.E.
	Assistant professor

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering, Dr. M.S. MURALI DHAR, M.E., Ph.D.** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor Ms.T. GANGALAKSHMI, M.E.** for her cordial support, valuable information and guidance, she helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech. , Ms. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

NALLURI KARTHIK	**(20UECS0659)**
CHELLA BALAJI	**(20UECS0193)**
K.LOKESH	**(20UECS0452)**

ABSTRACT

The credit card fraud detection system using logistic regression, one of the most widely used machine learning algorithms for binary classification problems.The classifier with better rating score can be chosen to be one of the best methods to predict frauds. Thus, followed by a feedback mechanism to solve the problem of concept. The system analyzes a variety of transaction features, such as transaction amount, location, and time, to identify anomalous patterns and behaviors that are indicative of fraud. The logistic regression algorithm is trained using a large dataset of known fraudulent and legitimate transactions to classify new transactions as either fraudulent or legitimate. The system's performance is evaluated using various metrics such as accuracy, precision, recall, and F1-score. The experimental results show that the proposed system achieves high accuracy and precision in detecting fraudulent transactions using logistic regression. This indicates that logistic regression is a promising solution for credit card fraud detection, and the proposed system can be a valuable tool for businesses and financial institutions to prevent financial losses due to fraudulent transactions.In recent years 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. Credit card fraud is a significant problem for financial institutions, costing billions of dollars every year. Detecting fraud in real-time is essential to prevent financial losses and protect customers accounts. Machine learning algorithms, such as logistic regression, have been used to develop fraud detection systems that can classify credit card transactions as fraudulent or non-fraudulent. In this proposed system, we present a logistic regression-based approach to detect fraudulent credit card transactions, which can help financial institutions to prevent fraud and protect their customers accounts.

Keywords: Credit Card Fraud Detection, Classification method, Logistic Regression, Multilayer Perceptron, Machine learning methods, Supervised learning.

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

ANN	Artificial neural network
CNN	Convalution neural network
DNN	Deep neural network
DT	Decsion tree
EDA	Exploratory data analysis
FTC	Federal trade commission
GA	Genetic algorithm
HTML	Hypertext markup language
LSTM	long short term memory
LR	Logistic regression
ML	Machine learning
NB	Naive bayes
PIN	Personal identification number
POS	point of sale
PCA	prompt corrective action
RF	Random forest
RNN	Recurrent neural networK
SVM	Support vector machine
SMOTE	Synthetic minority oversampling technique

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

Credit card fraud is a significant threat in the BFSI sector. This credit card fraud detection system studies and analyzes user behavior patterns and uses location scanning techniques to identify any unusual patterns. One of The user patterns includes important user behavior like spending habits, usage patterns, etc. All the information that also has a large volume, wide range, frequency, as well as importance is stored from small to large organizations over the cloud. The whole information is available from massive amounts of sources such as followers on social media, customer order behaviors, likes, and shares. White-collar crime is the ever-increasing problem with-reaching consequences for the finance sector, business institutions as well as governments. Fraud can indeed be described as illegal deceit to gain financial benefit . Enhanced card transactions had already appreciated a heavy emphasis on communication technology. When credit card transactions are by far the most prevalent form of transaction for offline and online payments, raising the rate of card fraud accelerates as well.

Machine learning is the innovation of this century that eliminates conventional strategies and also can function on huge datasets where humans can't immediately access. Strategies of machine learning break within two important categories; supervised learning versus unsupervised learning; Tracking of fraud can also be achieved any form and may only be determined how to use as per the datasets. Supervised training includes anomalies to always be identified as before. Many supervised methods are being used over the last few decades to identify credit card fraud. The major obstacle in implementing ML for detecting fraud seems to be the presence of extremely imbalanced databases. Most payments are legitimate in several available evidence sets, with such an extremely small number of fraudulent ones. The significant challenges to investigators are designing the accurate as well as efficient fraud prevention framework that will be low on false positives but efficiently identifies fraud activity .

1.2 Aim of the Project

The aim of using logistic regression in fraud credit card detection using machine learning is to develop a system that can accurately and efficiently identify fraudulent credit card transactions. The logistic regression algorithm is used to classify new transactions as either legitimate or fraudulent based on the analysis of various transaction features. The goal is to develop a system that can detect fraudulent transactions with a high degree of accuracy and precision, while minimizing the number of false positives and false negatives. The system aims to reduce the financial losses caused by credit card fraud for both businesses and consumers by identifying and preventing fraudulent transactions before they occur. The use of machine learning techniques, specifically logistic regression, provides a robust and scalable solution for credit card fraud detection that can adapt to changing patterns and behaviors of fraudulent activities. Overall, the aim is to develop a system that can provide reliable and effective protection against credit card fraud.

1.3 Project Domain

The project domain of fraud credit card detection using machine learning using logistic regression is in the field of financial security and fraud detection. The primary focus of the project is to develop a system that can accurately detect fraudulent credit card transactions using machine learning algorithms, specifically logistic regression. The project domain involves working with large datasets of credit card transactions and analyzing various features such as transaction amount, location, time, and user behavior to identify anomalous patterns that are indicative of fraud.

The project domain also includes implementing data preprocessing techniques such as data cleaning, data transformation, and feature selection to prepare the data for machine learning algorithms. The logistic regression algorithm is used to classify new transactions as either legitimate or fraudulent based on the analysis of these features. The system's performance is evaluated using various performance metrics such as accuracy, precision, recall, and F1-score. Overall, the project domain of fraud credit card detection using machine learning using logistic regression involves the application of machine learning techniques to financial data to develop a reliable and effective system for detecting credit card fraud. The system aims to provide financial institutions and businesses with a powerful tool to prevent and mitigate the impact of credit card fraud on their operations and customers.

1.4 Scope of the Project

The scope of the project for fraud credit card detection using machine learning using logistic regression includes several key aspects that need to be considered.

Firstly, the project scope involves collecting and processing a large dataset of credit card transactions that can be used to train and test the logistic regression algorithm. This includes identifying and removing any missing or incorrect data and selecting relevant features that can be used to detect fraudulent transactions.

Secondly, the scope of the project involves implementing the logistic regression algorithm and optimizing its performance to achieve high accuracy and precision in detecting fraudulent transactions. This includes fine-tuning the hyperparameters of the logistic regression algorithm and evaluating its performance using various performance metrics.

Thirdly, the project scope involves integrating the logistic regression algorithm into a functional fraud detection system that can be deployed in a real-world setting. This includes developing a user interface that allows users to interact with the system and monitor the results of the fraud detection process.

Fourthly, the project scope involves testing and validating the performance of the fraud detection system using a variety of simulated and real-world scenarios. This includes evaluating the system's ability to detect various types of fraudulent transactions, its robustness to changing patterns of fraudulent behavior, and its scalability to handle large volumes of transactions.

Finally, the scope of the project involves documenting the system design, implementation, and testing process, as well as providing user manuals and technical documentation to facilitate the adoption and use of the system.

Overall, the scope of the project for fraud credit card detection using machine learning using logistic regression is broad, covering data collection, data preprocessing, algorithm development, system integration, testing, and documentation. It requires a multidisciplinary approach that involves expertise in data analysis, machine learning, software development, and financial security.

Chapter 2

LITERATURE REVIEW

A. H. Alhazmi et.al., [1] evaluated the performance of classifiers or predictors, such as the Vector Machine, Random Forest, and Decision Tree. These metrics are either prevalence-dependent or prevalence-independent. Furthermore, these techniques are used in credit card fraud detection mechanisms, and the results of these algorithms have been compared.With these they have focused on the two ways of credit card transactions i) Virtually (card, not present) ii) With Card or physically present. They had focused on the techniques which are Regression, classification, Logistic regression, Support vector machine, Neural network, Artificial Immune system, K-nearest Neighbor, Naive Bayes, Genetic Algorithm, Data mining, Decision Tree, Fuzzy logic-based system, etc. In which, they have explained six data mining approaches as theoretical background that are classification, clustering, prediction, outlier detection, Regression, and visualization.

Armel et.al., [2] proposed support vector machines(SVM), artificial neural networks(ANN), Bayesian Networks, Hidden Markov Model, K-Nearest Neighbours (KNN) Fuzzy Logic system and Decision Trees. In their paper, they have observed that the algorithms k-nearest neighbor, decision trees, and the SVM give a medium level accuracy.Then have explained about existing techniques based on statistical and computation which is Artificial Immune system (AIS), Bayesian Belief Network, Neural Network, Logistic Regression, Support Vector Machine, Tree, Selforganizing map,Hybrid Methods, As a result, they had concluded that all the present machine learning techniques mentioned above can provide high accuracy for the detection rate and industries are looking forward to finding new methods to increase their profit and reduce the cost. Machine learning can be a good choice for it. [A Survey on Credit Card Fraud Detection using Machine Learning].

F. Ghobadi et.al., [3] proposed some important algorithmic techniques which are the Whale Optimization Techniques (WOA) and SMOTE (Synthetic Minority Oversampling Techniques). They mainly aimed to improve the convergence speed and to solve the data imbalance problem. The class imbalance problem is overcome using the SMOTE technique and the WOA technique. The SMOTE technique discriminates all the transactions which are synthesized are again resampled to check the data accuracy and are optimized using the WOA technique. The algorithm also improves the convergence speed, reliability, and efficiency of the system.

Z. Kazemi et.al., [4] have explained their work on decision trees, random forest, SVM, and logistic regression. They have taken the highly skewed dataset and worked on such type of dataset. The performance evaluation is based on accuracy, sensitivity, specificity, and precision. The results indicate that the accuracy for the Logistic Regression is 97.7, for Decision Trees is 95.5, for Random Forest is 98.6SVM classifier is 97.5. They have concluded that the Random Forest algorithm has the highest accuracy among the other algorithms and is considered as the best algorithm to detect the fraud. They also concluded that the SVM algorithm has a data imbalance problem and does not give better results to detect credit card fraud

A. Mishra et.al., [5] proposed a novel fraud detection method that has four stages they first utilize the historical transaction data to divide them into groups to form clusters of transactions having the same behavior they came up with a sliding windows Strategy to aggregate transactions.This algorithm is used to characterize the behavioral pattern of a cardholder then after aggregation, we use the new window formed the feature extraction is done. At last, the classification takes place and classifies behavioral patterns and assignments. As a result, their method of Logistic Regression with raw data (RawLR), Random Forest with aggregation data (AggRF), and Random Forest and feedback technique with aggregation data (AggRF +FB) are the best method with 80as compared to other methods.[Credit Card Fraud Detection: A Novel Approach Using Aggregation Strategy and Feedback Mechanism]

Rimpal R. Popat et.al., [6] proposed supervised machine learning algorithms on the real-world data set and then used those algorithms to implement a super classifier using ensemble learning and then they compared the performance of supervised algorithms with their implementation of a super classifier. They used ten machine learning algorithms such as Random Forest, Stacking Classifier, XGB Classifier, Gradient Boosting, Logistic Regression, MLP Classifier, SVM, Decision Tree, KNN, Naive Bayes. And compared the accuracy, Recall Precision, confusion matrix with the result of their super classifier. As a result, they found that the Logistic Regression is better for predicting fraud transactions. [Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study].

Rishi Banerjee et.al., [7] proposed ”Credit card fraud detection using logistic regression and artificial neural network” (2019) presents a credit card fraud detection system that utilizes logistic regression and artificial neural network (ANN) models. The proposed system processes credit card transaction data and uses logistic regression to screen the data for suspicious transactions. The screened transactions are then passed to an ANN for further analysis, where the ANN model is trained using a back propagation algorithm. The authors evaluated the proposed system using a publicly available dataset and reported high accuracy and low false positive rate in detecting fraudulent transactions. The results of the study suggest that the proposed system can be an effective solution for credit card fraud detection in real-world applications.

Satvik Vats et.al., [8] proposed comparison based on two random forests. Random-tree-based random forest CART-based random forest. They use different random forest algorithms to train the behavior features of normal and abnormal transactions and both of the algorithms are different in their base classifications and their performance. They applied both of the algorithms on the dataset e-commerce company in China. In which the fraud transaction in the subsets ratio is 1:1 to 10:1. As a result, accuracy from the random-tree based random forest is 91.96 whereas in CART-based random forest is 96.7. Since the data used is from the B2C dataset many problems arrived such as unbalanced data. Hence, the algorithm can be improved. [Random Forest for Credit Card Fraud Detection].

N. Sivakumar et.al., [9] has done their research on various algorithms like Naive Bayes, Logistic Regression, J48, and Adaboost. Naive Bayes on among the classification algorithm. This algorithm depends upon Bayes theorem. Bayes’s theorem finds the probability of an event that is occurring is given. The Logistic regression algorithm is similar to the linear regression algorithm. The linear regression is used for the prediction or forecasting the values. The logistic regression is mostly used for the classification task. The J48 algorithm is used to generate a decision tree and is used for the classification problem. The J48 is the extension of the ID3 (Iterative Dichotomieser). J48 is one of the most widely used and extensively analyzed areas in Machine Learning. This algorithm mainly works on constant and categorical variables. Adaboost is one of the most widely used machine learning algorithms and is mainly developed for binary classification. The algorithm is mainly used to boost the performance of the decision tree. This is also mainly used for the classification of the regression

Shail Machine et.al., [10] proposed an improved algorithm for credit card fraud detection. That is named as Naive Bayes improved K-nearest Neighbor method (NBKNN). They have used a dataset on which they had applied the algorithms to identify the fraudulent transaction in the taken dataset. Literature Review of Different Machine Learning Algorithms for Credit Card Fraud Detection The dataset has the record of European Cardholders who made a transaction using their credit cards within 2 days they made 284,807 transitions in which 492 transaction is fraudulent. The techniques used which were used are classification techniques but work differently on the same dataset. They had used both of the techniques (Naive Bayes and k-nearest neighbor) to enhance the accuracy and flexibility of the algorithm. As a result, they got the accuracy of approximately 95 from Naive Bayes and 90 from K -nearest Neighbor techniques.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

Credit card fraud detection is the process of finding out whether businesses are real or fraudulent. Because of the immense use of machine learning techniques to detect criminal cases, scholars often accept those methods for detecting credit card fraud activities. Although data mining is focused on finding valuable intelligence, machine learning is rooted in learning intelligence and developing its own model for the purpose of classification, clustering, and so on

Machine Learning (ML) is a sub-field of Artificial Intelligence (AI) that allows computers to learn from previous experience (data) and to improve on their predictive abilities without explicitly being programmed to do so . In this work we implement Machine Learning (ML) methods for credit card fraud detection. Credit card fraud is defined as a fraudulent transaction (payment) that is made using a credit or debit card by an unauthorised user . According to the Federal Trade Commission (FTC), there were about 1579 data breaches amounting to 179 million data points whereby credit card fraud activities were the most prevalent . Therefore, it is crucial to implement an effective credit card fraud detection method that is able to protect users from financial loss. One of the key issues with applying ML approaches to the credit card fraud detection problem is that most of the published work are impossible to reproduce.

3.1.1 Disadvantages

- **Positive errors** : Machine learning models require a large amount of data if they are to be accurate. This data volume is fine for large enterprises, but for others it is a challenge to have enough data points to identify valid cause-and-effect correlations.

  Without the necessary data, fraud detection machine learning algorithms may learn incorrect inferences and create false or irrelevant fraud evaluations.

- **Less control** : Fraud detection machine learning models are employed to evaluate actions, behavior and activities. Initially, when the dataset is small, they are blind to data connections. As a result, the model may overlook a seemingly evident connection, such as a shared card between two accounts.
- **No human intelligence** : It's difficult to beat good old psychology when working out why a user's activity is questionable. Even the most advanced technology cannot replace the expertise and judgment required to correctly filter and interpret data and evaluate the meaning of the risk score.

3.2 Proposed System

The proposed system for fraud credit card detection using machine learning in logistic regression includes several key components that work together to detect and prevent fraudulent credit card transactions.

The first component of the system is data preprocessing, which involves collecting and processing a large dataset of credit card transactions to prepare it for use in the machine learning algorithm. This includes identifying and removing any missing or incorrect data and selecting relevant features that can be used to detect fraudulent transactions.

The second component of the system is the logistic regression algorithm, which is used to classify new transactions as either legitimate or fraudulent based on the analysis of these features. The logistic regression algorithm is trained using a large dataset of known fraudulent and legitimate transactions to classify new transactions accurately.

The third component of the system is a user interface that allows users to interact with the system and monitor the results of the fraud detection process. The user interface provides real-time feedback on the status of transactions, including whether they have been flagged as potentially fraudulent.

The fourth component of the system is a notification system that alerts users when a potentially fraudulent transaction has been detected. This includes sending notifications via email, SMS, or other communication channels to the relevant parties, such as financial institutions or cardholders.

The fifth component of the system is a feedback loop that allows users to provide feedback on the accuracy of the system's classification. This feedback is used to improve the performance of the system over time and to adapt to changing patterns of fraudulent behavior.

Credit Card Terminal: The credit card terminal is responsible for collecting transaction data and forwarding it to the fraud detection system for analysis. The terminal can be either a physical device or a software application that can be installed on a computer or mobile device.

Transaction Class: The transaction class is a data structure that stores transaction information such as the transaction amount, time and date, and the cardholder's information. Each transaction is assigned a unique transaction ID for tracking purposes.

3.2.1 Advantages

- **Better performance on high-dimensional data** : SVMs can perform better than decision trees on datasets with a large number of features or high-dimensional data, as they are less prone to overfitting and can handle a large number of features.
- **Faster detection** : A machine learning model can quickly identify any drifts from regular transactions and user behaviours in real time. By recognising anomalies, such as a sudden increase in transactional amount or location change, ML algorithms can minimise the risk of fraud and ensure more secure transactions.
- **Higher accuracy** :Conventional fraud detection techniques cause errors at the payment gateways that sometimes result in genuine customers being blocked. With sufficient training data and insights, ML models can achieve higher accuracy and precision, reducing these errors along with the time required to be spent on performing manual analysis.
- **Improved efficiency with larger data** :Once an algorithm picks up different transactional patterns and behaviours, it can efficiently work with large datasets to separate authentic payments from fraudulent ones. The models can analyse huge amounts of data in seconds while offering real-time insights for improved decision-making capabilities.

- **Cost-effective detection technique** : With ML, data can be analyzed in milliseconds, meaning that team members aren't burdened with manual reviews and checks every time new data is received. This is great for firms that see seasonal highs and lows in traffic, checkouts or signups.

3.3 Feasibility Study

3.3.1 Economic Feasibility

The proposed project is economically feasible ,it can be evaluated by considering the following aspects can be evaluated by considering the costs and benefits associated with its implementation.

Data collection and processing costs: Collecting and processing a large dataset of credit card transactions can be a time-consuming and expensive process.

Infrastructure costs: Implementing a fraud detection system requires appropriate hardware and software infrastructure, including servers, databases, and machine learning libraries.

Human resource costs: Developing and maintaining a fraud detection system requires skilled professionals such as data scientists, software developers, and security experts.

Operational costs: Once the system is implemented, there are ongoing operational costs associated with maintaining and updating the system, as well as providing user support.

Reduced fraud losses: Implementing a fraud detection system can help businesses and financial institutions reduce their losses due to fraudulent credit card transactions. This can result in significant cost savings over time.

Improved customer trust: Implementing a fraud detection system can improve customer trust by providing a more secure environment for credit card transactions.

Regulatory compliance: Many financial institutions are required to comply with regulations related to fraud detection and prevention. Implementing a fraud detection system can help ensure compliance with these regulations.

Competitive advantage: Having a robust fraud detection system in place can provide a competitive advantage for businesses, as it can help attract and retain customers who value security and trust.

3.3.2 Technical Feasibility

The proposed project is technically feasible ,it can be evaluated by considering the following aspects.

Availability of data: The success of the machine learning algorithm depends on the availability of a large dataset of credit card transactions. If the required data is not available, or the quality of the data is poor, the accuracy of the fraud detection system will be affected.

Machine learning expertise: Developing and implementing a machine learning algorithm requires specialized skills and expertise in data science and software development. If the required skills are not available in-house, the business may need to outsource the development of the system or invest in training its employees.

Scalability: The fraud detection system must be able to handle large volumes of transactions in real-time. This requires a scalable architecture that can handle the processing and analysis of large amounts of data without compromising performance.

Integration with existing systems: The fraud detection system must be integrated with the existing systems used by the business, such as payment gateways and transaction processing systems. This requires a thorough understanding of the existing systems and the ability to develop interfaces that facilitate communication between the systems.

Security: The fraud detection system must be secure and protect sensitive customer data from unauthorized access. This requires implementing security measures such as data encryption, access controls, and monitoring.

3.3.3 Social Feasibility

The social feasibility study indicates that the proposed project involving the following aspects.

User Acceptance: The success of the fraud detection system depends on user acceptance, including businesses, financial institutions, and customers. The system should be user-friendly and easily understandable to ensure its acceptance by all parties involved.

Privacy concerns: The fraud detection system must ensure the privacy of customer data and protect against unauthorized access. The use of machine learning algorithms should be transparent, and customers should be informed about how their data is being used to detect fraud.

Ethics: The use of machine learning algorithms for fraud detection should be ethical and not discriminate against any particular group or individual. The algorithms should be unbiased and fair, and any decision-making processes should be explainable and transparent.

Legal and regulatory compliance: The fraud detection system must comply with legal and regulatory requirements related to data privacy, security, and fraud detection. Businesses and financial institutions must ensure that they comply with these requirements to avoid legal liabilities.

Social impact: The fraud detection system may have a positive social impact by reducing fraudulent activities, thereby improving the trust and confidence of customers in the credit card system.

3.4 System Specification

3.4.1 Hardware Specification

- RAM : 8GB or more
- Processor : Intel Core i3 or more
- Disk Space : 256GB HDD or above
- Internet Connection

3.4.2 Software Specification

Operating System: Windows 10
Browser: Google Chrome / Mozilla Firefox / Microsoft Edge
Code Editor-PyCharm
Python 3.7

- Python
  Python is a general-purpose programming language, so it can be used for many things. Python is used for web development, AI, machine learning, operating systems, mobile application development, and video games.

- Scikit-learn
  Scikit-learn is a Python machine learning machine learning library. It includes support-vector machines and other techniques for classification, regression, and clustering. Classification, regression, clustering, and dimensionality reduction are just a few of the useful functions in the sklearn toolkit for machine learning and statistical modelling.
- Numpy
  NumPy is a Python library that provide support for huge, multi-dimensional arrays and matrices, as well as a large number of high-level mathematical functions to operate on these arrays.NumPy was created in 2005 by Travis Oliphant. It is an open source project and anyone can use it freely. NumPy stands for Numerical Python. Numpy operations are executed more efficiently and with very less code than is possible using Python's built-in se- quences.
- Numpy NumPy is a Python library that provide support for huge, multi-dimensional arrays and matrices, as well as a large number of high-level mathematical functions to operate on these arrays.

3.4.3 Standards and Policies

Python Standard Used: ISO/IEC WD TR 24472-4

ISO/IEC WD TR 24772-4 Programming languages — Avoiding vulnerabilities in programming languages — Part 4: Vulnerability descriptions for the programming language Python General information

Status : Under development, Edition : 1, Technical Committee : ISO/IEC JTC 1/SC 22 Programming languages, their environments and system software interfaces, ICS : 35.060 Languages used in information technology

PyCharm Standard Used: ISO 2022.1 Build 221.5591.52

Chapter 4

METHODOLOGY

4.1 General Architecture

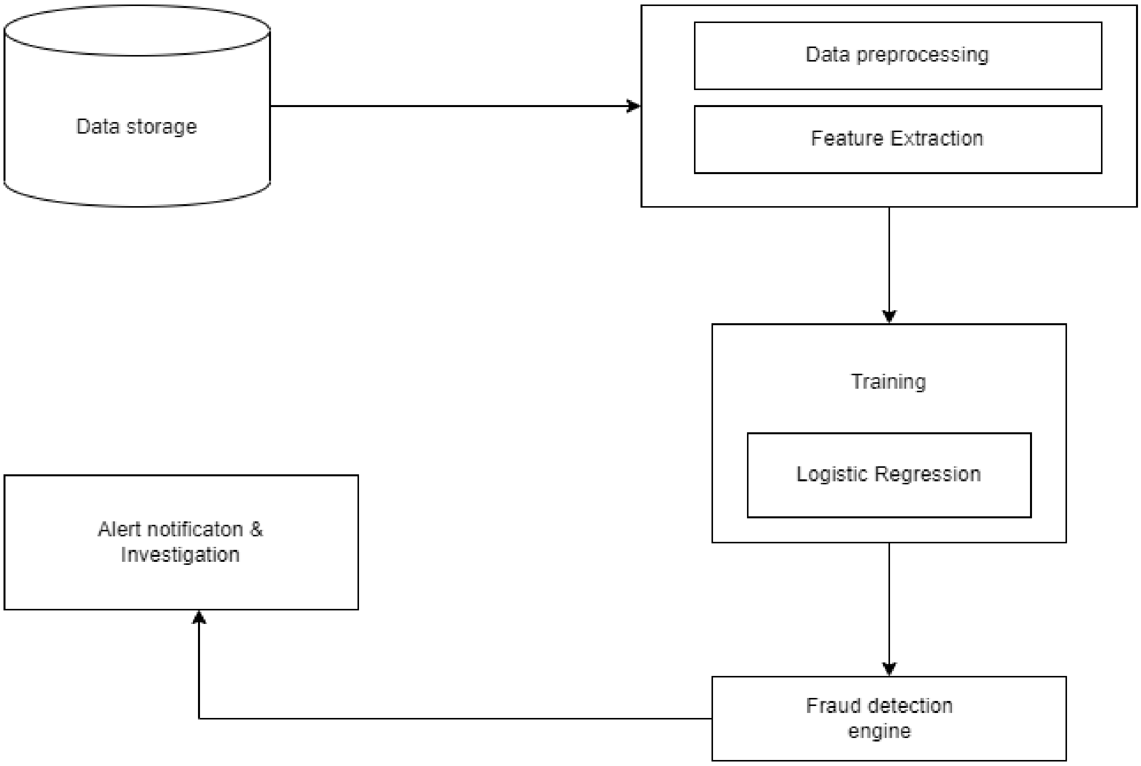


Figure 4.1: **Architecture diagram of fraud credit card detection system**

The Figure 4.1 represents the architecture of a fraud credit card detection system using machine learning in logistic regression typically consists of several interconnected components. Firstly, a data storage component is required to store the credit card transaction data. Secondly, a data preprocessing component is required to preprocess the raw transaction data, including feature engineering to extract useful features that can be used by the fraud detection engine. Thirdly, a logistic regression training component is required to train a logistic regression model on the preprocessed transaction data.

Once the logistic regression model is trained, it can be deployed to a fraud detection engine component which is responsible for using the model to detect potentially fraudulent transactions in real-time. Finally, an alert notification and investigation component is required to alert relevant parties about any suspected fraudulent transactions detected by the system, and to conduct investigation to determine the validity of the alert and take appropriate action.

Data Storage: The first component is responsible for storing the credit card transaction data.

Data Preprocessing: The second component is responsible for preprocessing the transaction data, including feature engineering to extract useful features from the raw data.

Logistic Regression Training: The third component is responsible for training a logistic regression model on the preprocessed transaction data.

Fraud Detection Engine: The fourth component is responsible for using the trained logistic regression model to detect fraudulent transactions in real-time.

Alert Notification and Investigation: The fifth component is responsible for alerting relevant parties about any suspected fraudulent transactions detected by the system. Investigation can then be conducted to determine the validity of the alert and take appropriate action.

4.2 Design Phase

4.2.1 Data Flow Diagram

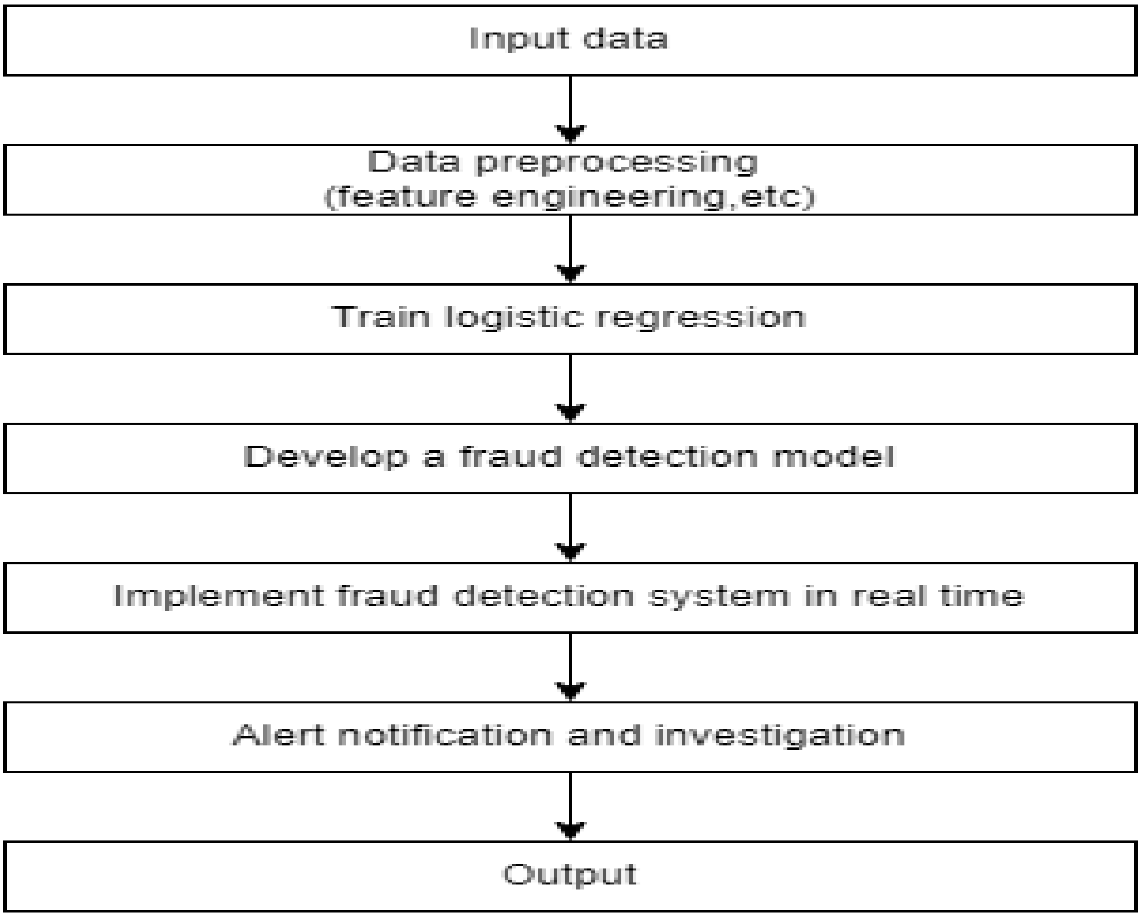

Input data
Data preprocessing (feature engineering,etc)
Train logistic regression
Develop a fraud detection model
Implement fraud detection system in real time
Alert notification and investigation
Output

Figure 4.2: **Data flow diagram**

The Figure 4.2 represents the The data flow in a fraud credit card detection system using machine learning in logistic regression typically involves several steps. The first step is data acquisition, where transaction data is collected and stored in a data storage component. The next step is data pre processing, where the raw transaction data is cleaned, transformed, and feature-engineered to extract useful features that can be used for fraud detection.

Once the data is preprocessed, the logistic regression model is trained on the pre processed data. After training, the fraud detection engine component is responsible for processing real-time transaction data and using the trained model to predict the likelihood of fraud. The output of the fraud detection engine is then used by the alert notification and investigation component to determine whether a transaction is fraudulent. If a transaction is identified as potentially fraudulent, an alert is sent to relevant parties for further investigation and appropriate action.

4.2.2 Use Case Diagram

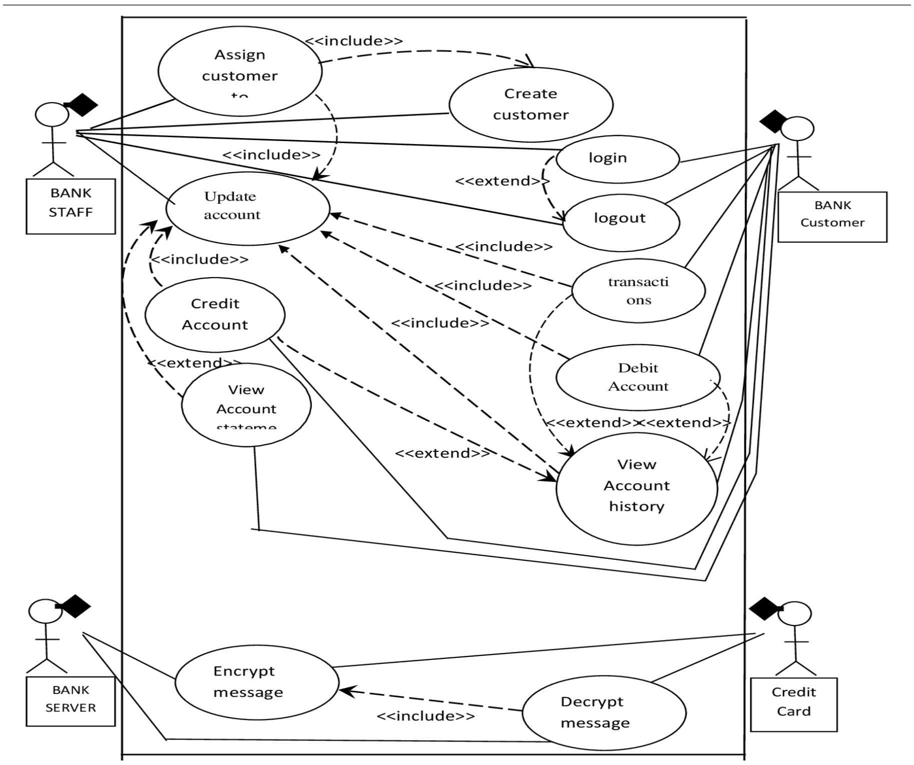


Figure 4.3: **Use case diagram**

The Figure 4.3 represents the use case diagram for credit card fraud detection using machine learning is in the banking and financial services industry, where credit card companies and banks use machine learning models to detect fraudulent transactions in real-time. Here is an example use case:

A credit card company wants to improve its fraud detection capabilities to reduce losses due to fraudulent transactions. The company has a large dataset of transaction data, including both legitimate and fraudulent transactions. The credit card company uses machine learning algorithms to develop a predictive model that can identify potentially fraudulent transactions in real-time.The model is trained on historical transaction data and uses features such as transaction amount, location, and time of day to distinguish fraudulent transactions from legitimate ones. Once the model is trained, it is deployed in a production environment where it is used to analyze new transactions in real-time.

When a transaction is identified as potentially fraudulent, the system generates an alert to notify the appropriate parties, such as the cardholder and the issuing bank.

The alert includes details about the transaction and a recommendation on whether to approve or decline the transaction.Over time, the credit card company continues to collect and analyze transaction data to refine the predictive model and improve its accuracy. The company also uses the data to identify new fraud patterns and develop new features to enhance the model's performance.Through the use of machine learning, the credit card company is able to detect and prevent fraudulent transactions in real-time, reducing losses due to fraud and improving customer satisfaction.

4.2.3 Class Diagram

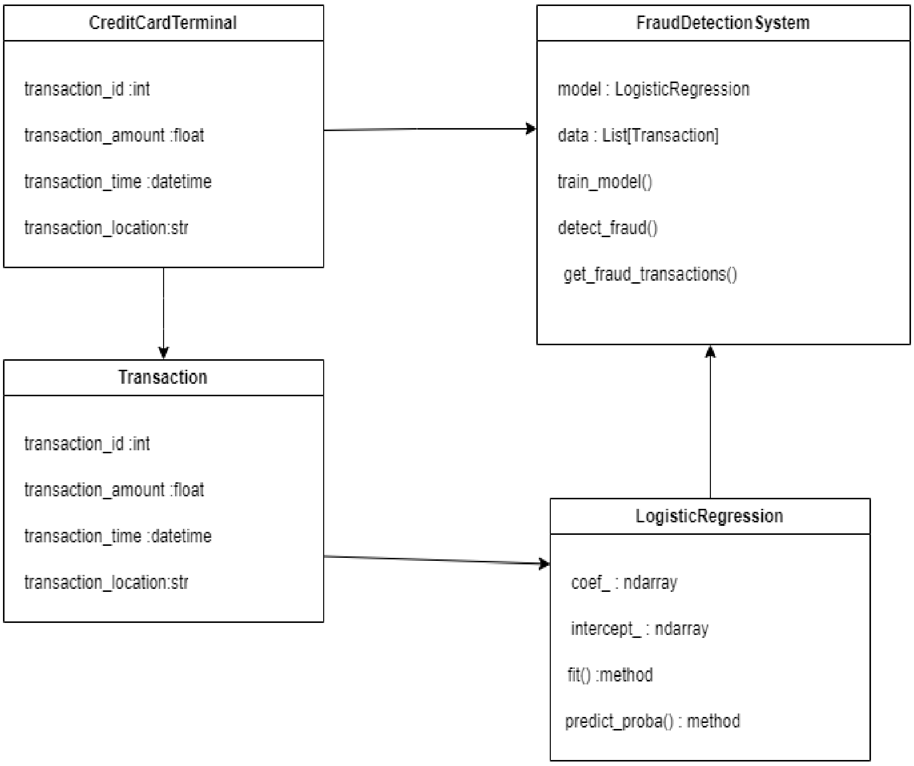


Figure 4.4: **Class diagram**

The Figure 4.4 represents The class diagram, there are three classes: Transaction, Machine Learning, and Alert.The Transaction class represents a single credit card transaction and contains attributes such as transaction id, amount, location, and time.The Machine Learning class represents the machine learning model used to detect fraudulent transactions. It contains attributes such as the model object, features used for training and prediction, and the algorithm used. It also has methods for training the model, making predictions, and evaluating the model's performance .

The Alert class represents an alert generated by the system when a potentially fraudulent transaction is detected. It contains attributes such as an alert id, transaction id, message describing the transaction, and action to be taken, such as declining the transaction or contacting the cardholder. The Transaction class has a one-to-many relationship with the Alert class, as a single transaction can generate multiple alerts. The Machine Learning class has a one-to-one relationship with the Alert class, as the model is used to generate alerts when fraudulent transactions are detected.

Overall, this class diagram provides a high-level overview of the main classes and their relationships in a credit card fraud detection system that uses machine learning.

4.2.4 Sequence Diagram

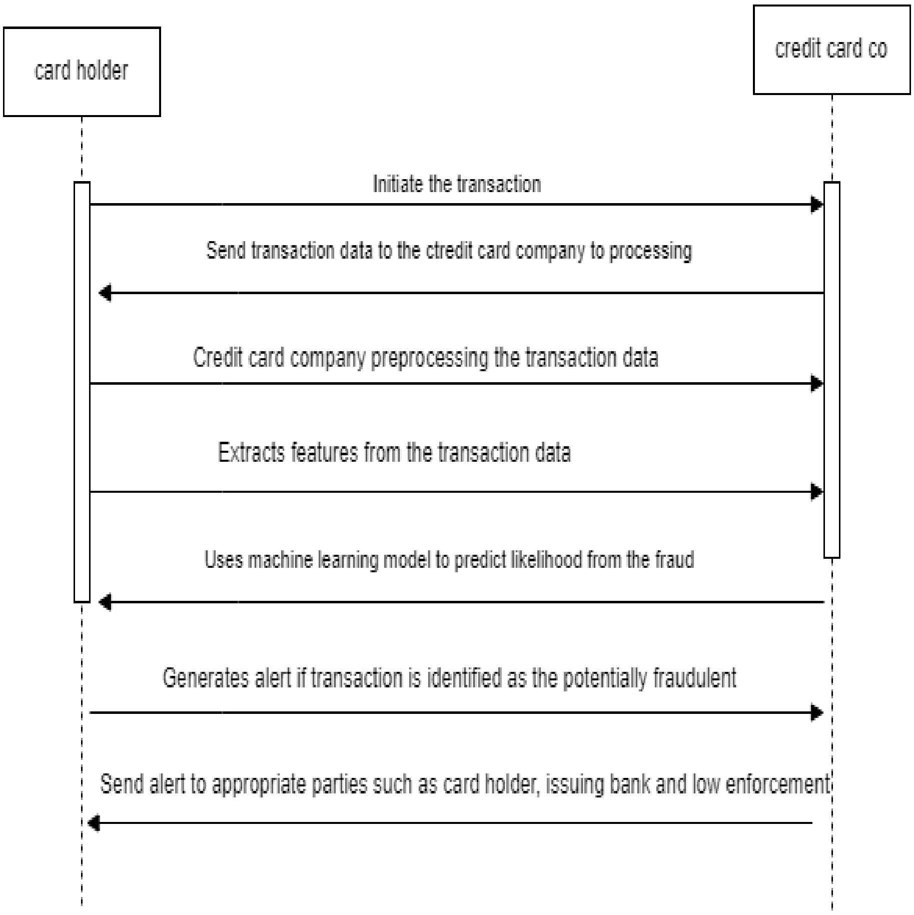

card holder
credit card co
Initiate the transaction
Send transaction data to the ctredit card company to processing
Credit card company preprocessing the transaction data
Extracts features from the transaction data
Uses machine learning model to predict likelihood from the fraud
Generates alert if transaction is identified as the potentially fraudulent
Send alert to appropriate parties such as card holder, issuing bank and low enforcement

Figure 4.5: **Sequence diagram**

The Figure 4.5 represents The sequence diagram, the Cardholder initiates a transaction and sends the transaction data to the Credit Card . for processing. The credit card company preprocesses the transaction data and extracts features from it. Then, the machine learning model is used to predict the likelihood of fraud. If the transaction is identified as potentially fraudulent, an alert is generated and sent to the appropriate parties.This sequence diagram provides a high-level overview of the main interactions between the different components of a credit card fraud detection system that uses machine learning.

4.3 Algorithm & Pseudo Code

4.3.1 Algorithm

1. Collect transaction data from various sources, including credit card companies, banks, and merchants.

2. Clean and preprocess data to remove duplicates, correct errors, and fill missing values. Convert categorical variables to numeric using one-hot encoding or other encoding methods.

3. Reduce dimensionality of the feature space to speed up training and reduce overfitting using techniques such as principal component analysis (PCA) or feature selection methods. Add new features that may help identify fraudulent transactions, such as transaction frequency or user behavior patterns.

4. Split the data into training and testing sets Choose an appropriate machine learning algorithm, such as logistic regression, decision trees, random forests, or neural networks. Train the model using the training set and evaluate its performance using metrics such as accuracy, precision, recall, and F1 score.

5. Implement the model into the credit card company's system to automatically detect fraudulent transactions.

6. Monitor the model's performance in the production environment and make adjustments as necessary to improve its accuracy and efficiency. item Evaluate the model on the test set using performance metrics such as accuracy, precision, recall, and F1-score. These metrics help determine the effectiveness of the model in detecting fraud.

4.3.2 Pseudo Code

```
function fraud_detection_system(transaction_data):

    # Data preprocessing (feature engineering)
    processed_data = preprocess(transaction_data)

    # Load the pre-trained logistic regression model
    logistic_regression_model = load_model()

    # Predict the probability of fraud using the logistic regression model
    fraud_probability = logistic_regression_model.predict_proba(processed_data)[:,1]

    # Determine if the transaction is fraudulent based on the probability threshold
    if fraud_probability > 0.5:
        # Alert relevant parties about the suspected fraudulent transaction
        alert_notification(fraud_probability)
        # Conduct investigation and take appropriate action
        investigate(fraud_probability)

    # Return the probability of fraud
    return fraud_probability
```

4.4 Module Description

4.4.1 Data Collection and Processing

Collect transaction data from various sources, including credit card companies, banks, and merchants. Clean and preprocess data to remove duplicates, correct errors, and fill missing values. Convert categorical variables to numeric using one-hot encoding or other encoding methods. Split data into features and labels, with the fraudulent status of each transaction as the label.

4.4.2 Feature Engineering

Scale features to have zero mean and unit variance to improve the performance of machine learning models. Reduce dimensionality of the feature space to speed up training and reduce overfitting using techniques such as principal component analysis (PCA) or feature selection methods. Add new features that may help identify fraudulent transactions, such as transaction frequency or user behavior patterns. Normalize features to make them comparable across different scales.

4.4.3 Model Selection and Training

Choose an appropriate machine learning algorithm, such as logistic regression, decision trees, random forests, or neural networks. Split data into training and testing sets to evaluate the performance of the model. Train the model using the training set and evaluate its performance using metrics such as accuracy, precision, recall, and F1 score. Fine-tune hyperparameters using techniques such as grid search or random search to optimize model performance.

4.5 Steps to execute/run/implement the project

4.5.1 Deployment

Deploy the trained model to a production environment where it can process new transactions in real-time. Monitor the model's performance and make adjustments as necessary to improve its accuracy and efficiency. Implement alerts or other notification systems to notify appropriate personnel if a fraudulent transaction is detected. Use the model to generate reports and other analytics to help identify new patterns of fraud and improve the detection system.

4.5.2 Continuous Improvement

Collect feedback and data from users and other sources to improve the model's accuracy and identify new patterns of fraud. Continuously evaluate new machine learning techniques and algorithms to improve the system's performance. Refine the feature selection and engineering process to identify new features that can improve the model's accuracy. Update the system as necessary to adapt to changing fraud scenarios and maintain its effectiveness over time.

4.5.3 Implementation

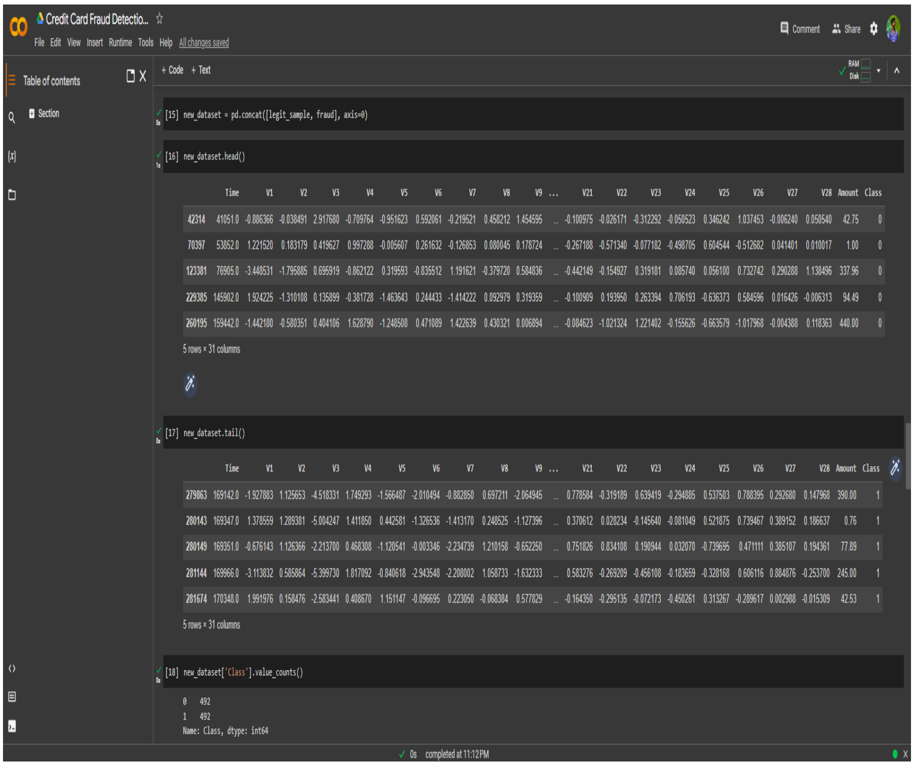
This is the final step where after getting output screen we need to check with giving input data to process and by checking with existing dataset, it is going to give proper output.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design


Credit Card Fraud Detectio...
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Table of contents
Section
+ Code + Text
RAM
Disk
0s completed at 11:12 PM

```
[15] new_dataset = pd.concat([legit_sample, fraud], axis=0)
```

```
[16] new_dataset.head()
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
42314	41051.0	-0.886366	-0.038491	2.917680	-0.709764	-0.951623	0.592061	-0.219521	0.458212	1.454595	...	-0.100975	-0.026171	-0.312292	-0.050523	0.346242	1.037453	-0.006240	0.050540	42.75	0
70397	53852.0	1.221520	0.183179	0.419627	0.997288	-0.005607	0.261632	-0.126853	0.080045	0.178724	...	-0.267188	-0.571340	-0.077182	-0.498705	0.604544	-0.512682	0.041401	0.010017	1.00	0
123381	76905.0	-3.448531	-1.795885	0.695919	-0.862122	0.319593	-0.835512	1.191621	-0.379720	0.584836	...	-0.442149	-0.154927	0.319181	0.085740	0.056100	0.732742	0.290288	1.138496	337.96	0
229385	145902.0	1.924225	-1.310108	0.135899	-0.381728	-1.463643	0.244433	-1.414222	0.092979	0.319359	...	-0.100909	0.193950	0.263394	0.706193	-0.636373	0.584596	0.016426	-0.006313	94.49	0
260195	159442.0	-1.442180	-0.580351	0.404106	1.628790	-1.248508	0.471089	1.422639	0.430321	0.006894	...	-0.084623	-1.021324	1.221402	-0.155626	-0.663579	-1.017968	-0.004388	0.118363	440.00	0

5 rows × 31 columns

```
[17] new_dataset.tail()
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
279863	169142.0	-1.927883	1.125653	-4.518331	1.749293	-1.566487	-2.010494	-0.882850	0.697211	-2.064945	...	0.778584	-0.319189	0.639419	-0.294885	0.537503	0.788395	0.292680	0.147968	390.00	1
280143	169347.0	1.378559	1.289381	-5.004247	1.411850	0.442581	-1.326536	-1.413170	0.248525	-1.127396	...	0.370612	0.028234	-0.145640	-0.081049	0.521875	0.739467	0.389152	0.186637	0.76	1
280149	169351.0	-0.676143	1.126366	-2.213700	0.468308	-1.120541	-0.003346	-2.234739	1.210158	-0.652250	...	0.751826	0.834108	0.190944	0.032070	-0.739695	0.471111	0.385107	0.194361	77.89	1
281144	169966.0	-3.113832	0.585864	-5.399730	1.817092	-0.840618	-2.943548	-2.208002	1.058733	-1.632333	...	0.583276	-0.269209	-0.456108	-0.183659	-0.328168	0.606116	0.884876	-0.253700	245.00	1
281674	170348.0	1.991976	0.158476	-2.583441	0.408670	1.151147	-0.096695	0.223050	-0.068384	0.577829	...	-0.164350	-0.295135	-0.072173	-0.450261	0.313267	-0.289617	0.002988	-0.015309	42.53	1

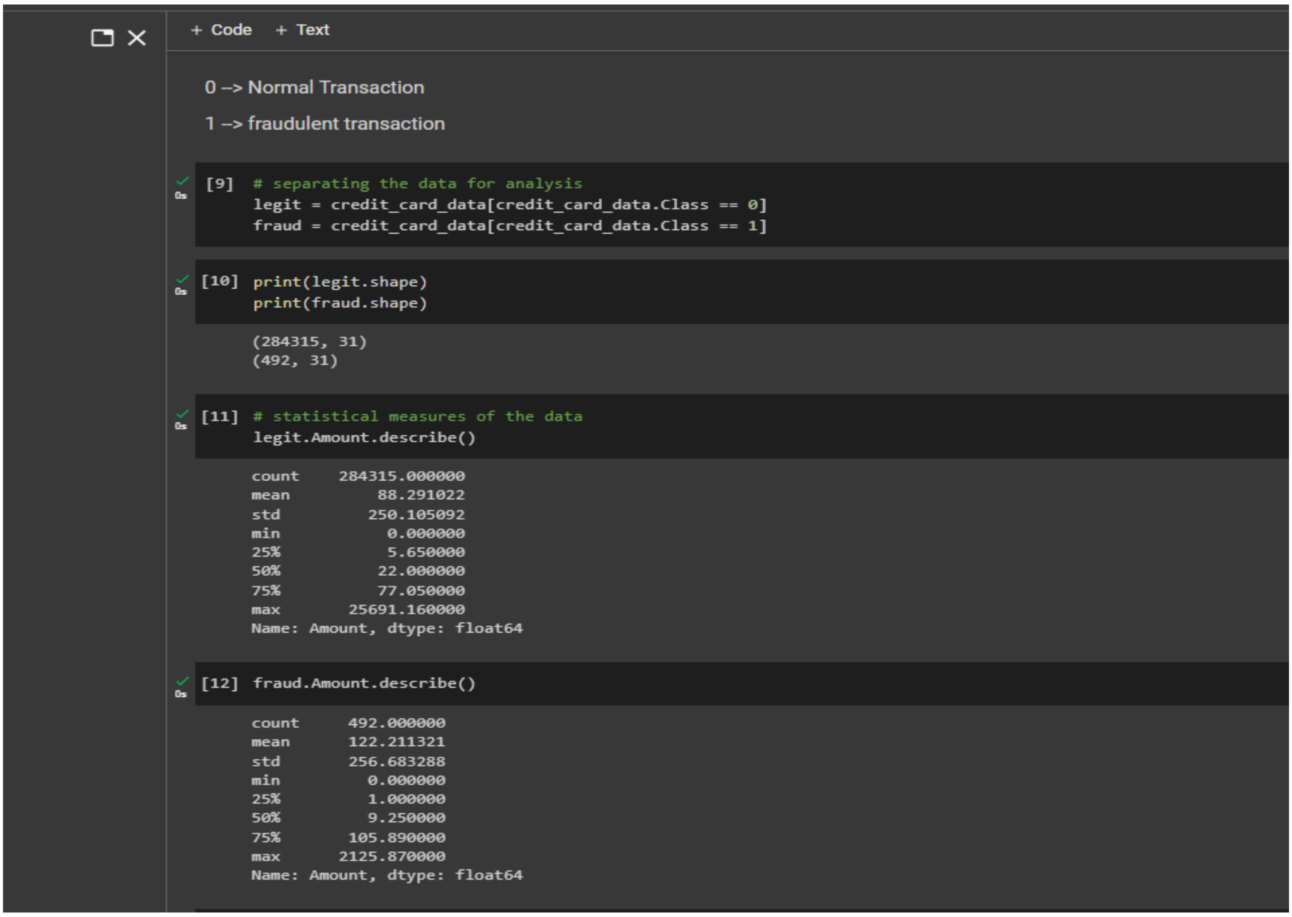
5 rows × 31 columns

```
[18] new_dataset['Class'].value_counts()
```

```
0    492
1    492
Name: Class, dtype: int64
```

Figure 5.1: **Importing datasets**

The Figure 5.1 represents the input data for a fraud credit card detection system using machine learning typically includes transaction data from credit card companies, banks, and other financial institutions. The first step in importing datasets is to identify and locate the relevant datasets. These datasets may be sourced from various internal and external sources such as financial institutions, credit card companies, and online marketplaces. It is important to ensure that the data is current and representative of the current fraud landscape.



0 --> Normal Transaction

1 --> fraudulent transaction

```
[9]  # separating the data for analysis
     legit = credit_card_data[credit_card_data.Class == 0]
     fraud = credit_card_data[credit_card_data.Class == 1]
```

```
[10] print(legit.shape)
     print(fraud.shape)
```

```
(284315, 31)
(492, 31)
```

```
[11] # statistical measures of the data
     legit.Amount.describe()
```

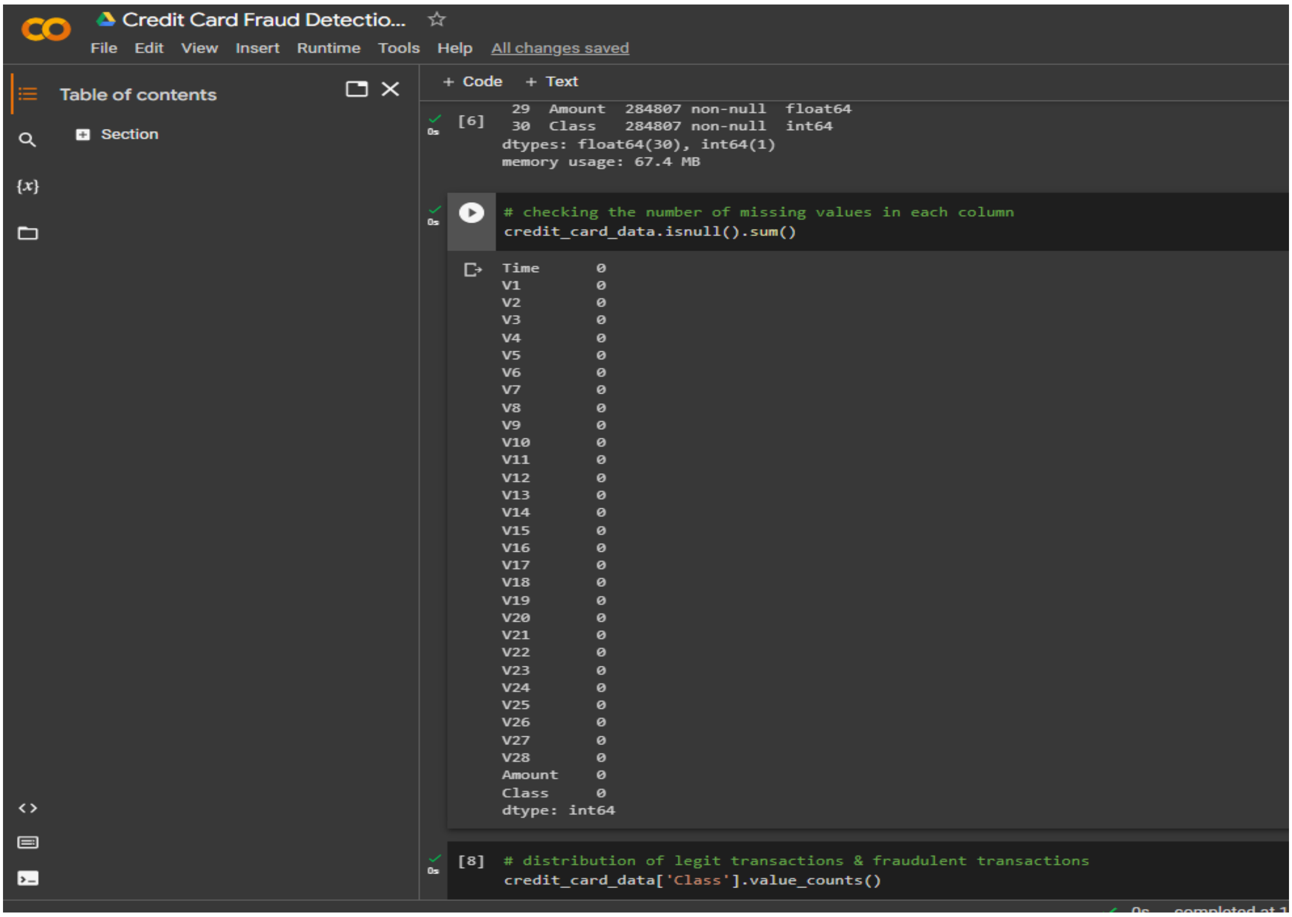
```
count    284315.000000
mean         88.291022
std         250.105092
min           0.000000
25%           5.650000
50%          22.000000
75%          77.050000
max       25691.160000
Name: Amount, dtype: float64
```

```
[12] fraud.Amount.describe()
```

```
count     492.000000
mean      122.211321
std       256.683288
min         0.000000
25%         1.000000
50%         9.250000
75%       105.890000
max      2125.870000
Name: Amount, dtype: float64
```

Figure 5.2: **Datasets extraction**

The Figure 5.2 represents the trained models are evaluated using a set of test data to determine their accuracy and effectiveness in detecting fraudulent transactions. The performance of the deployed model is monitored on an ongoing basis to ensure that it is functioning correctly and accurately. Any changes in the data or the underlying system may require retraining or fine-tuning of the model. Feedback from fraud analysts and investigators is used to improve the performance of the model over time. This feedback can include information about false positives or false negatives, new types of fraud, or changes in the behavior of fraudsters.



```
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

```
# checking the number of missing values in each column
credit_card_data.isnull().sum()
```

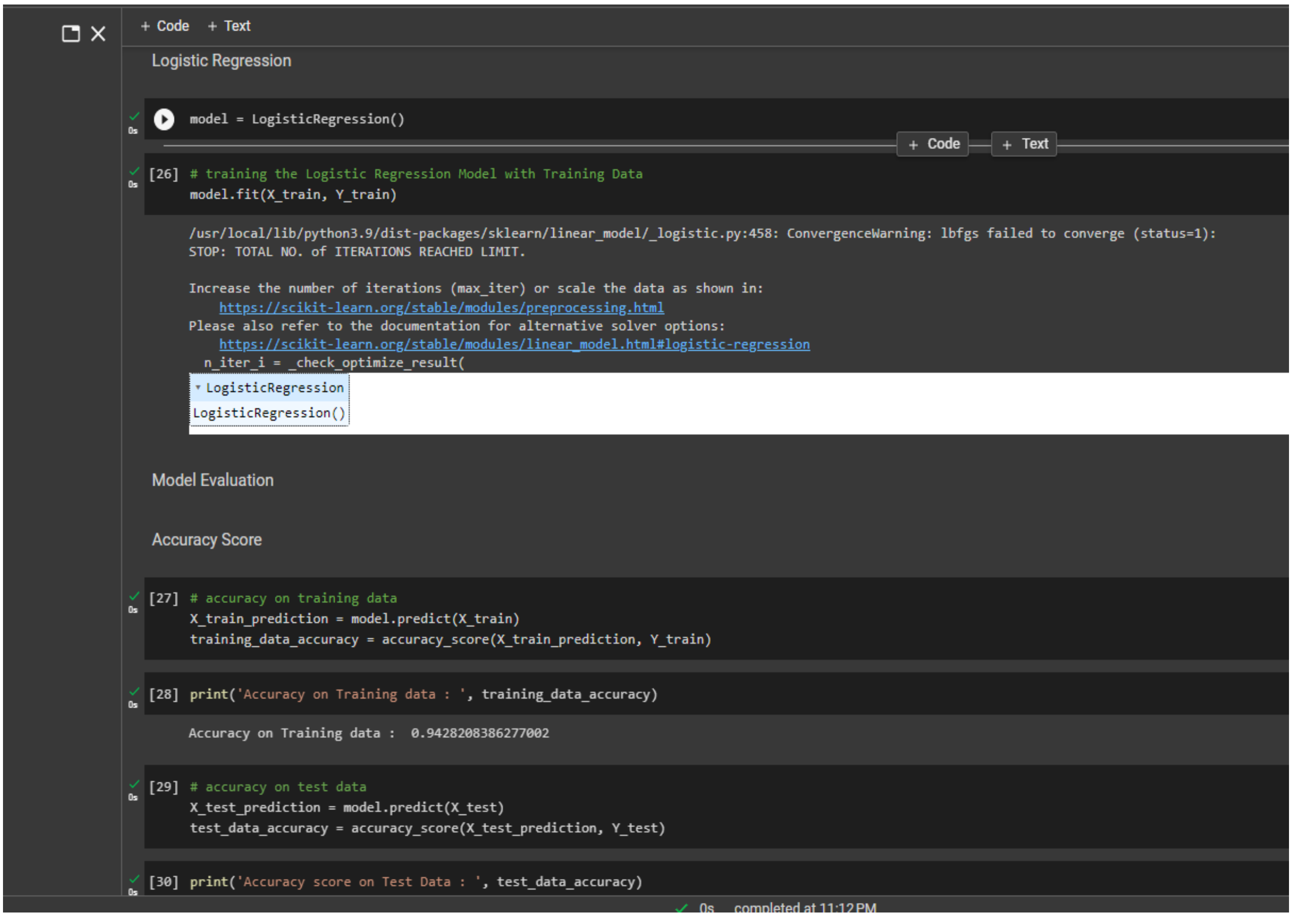
```
Time      0
V1        0
V2        0
V3        0
V4        0
V5        0
V6        0
V7        0
V8        0
V9        0
V10       0
V11       0
V12       0
V13       0
V14       0
V15       0
V16       0
V17       0
V18       0
V19       0
V20       0
V21       0
V22       0
V23       0
V24       0
V25       0
V26       0
V27       0
V28       0
Amount    0
Class     0
dtype: int64
```

```
# distribution of legit transactions & fraudulent transactions
credit_card_data['Class'].value_counts()
```

Figure 5.3: **Input datasets**

The Figure 5.3 represents the simple output format may be a binary label, such as 0 or 1, where 0 represents a non-fraudulent transaction and 1 represents a fraudulent transaction. This output can be used by downstream systems to flag fraudulent transactions for further review or action. Another output format may include a probability score indicating the likelihood of a transaction being fraudulent. For example, a score of 0.9 may indicate a high probability of fraud, while a score of 0.1 may indicate a low probability of fraud. This output can be used to prioritize transaction reviews or to adjust the risk thresholds for automatic transaction approvals or denials.

5.1.2 Output Design



Logistic Regression

```
model = LogisticRegression()
```

```
[26] # training the Logistic Regression Model with Training Data
     model.fit(X_train, Y_train)

     /usr/local/lib/python3.9/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
     STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

     Increase the number of iterations (max_iter) or scale the data as shown in:
         https://scikit-learn.org/stable/modules/preprocessing.html
     Please also refer to the documentation for alternative solver options:
         https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
       n_iter_i = _check_optimize_result(
     ▾ LogisticRegression
     LogisticRegression()
```

Model Evaluation

Accuracy Score

```
[27] # accuracy on training data
     X_train_prediction = model.predict(X_train)
     training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

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

     Accuracy on Training data :  0.9428208386277002
```

```
[29] # accuracy on test data
     X_test_prediction = model.predict(X_test)
     test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

```
[30] print('Accuracy score on Test Data : ', test_data_accuracy)
```

Figure 5.4: **Final output of accuracy fraud detection**

5.2 Testing

Testing for fraud credit card detection using machine learning involves evaluating the performance of the machine learning model or system using a variety of testing techniques to ensure that it is detecting fraudulent transactions accurately and efficiently. Here are some common testing techniques used for fraud credit card detection using machine learning:

Unit Testing: This involves testing individual components of the system, such as the algorithms used for feature selection or the models used for classification. This can be done using mock data or smaller subsets of the actual data.

Integration Testing: This involves testing the interaction between different components of the system, such as the data processing pipeline and the machine learning models. This can be done using a larger subset of the actual data.

System Testing: This involves testing the entire system end-to-end, including the input data, the machine learning models, and the output results. This can be done using a large, representative dataset of credit card transactions.

Acceptance Testing: This involves testing the system against a set of predefined acceptance criteria to ensure that it meets the desired requirements and specifications. This can be done using a set of realistic use cases and scenarios.

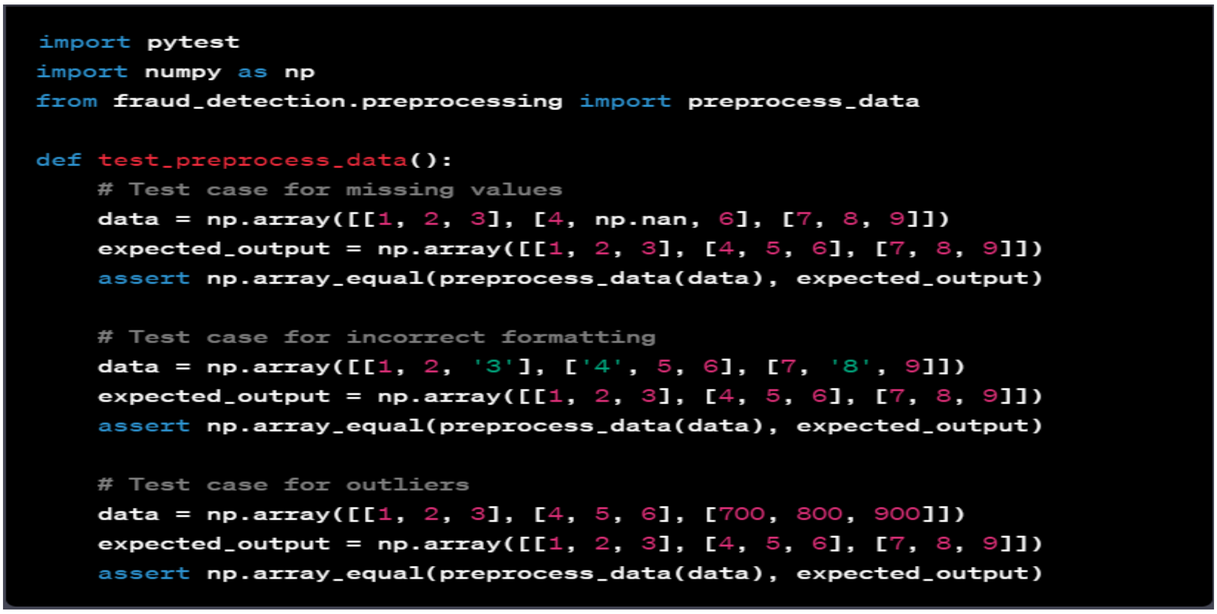
Performance Testing: This involves testing the system's performance under various load and stress conditions to ensure that it can handle a large volume of transactions and respond quickly to fraud alerts.

Regression Testing: This involves retesting the system after making changes or updates to ensure that it still performs as expected and has not introduced any new bugs or issues.

5.3 Types of Testing

5.3.1 Unit testing

Input

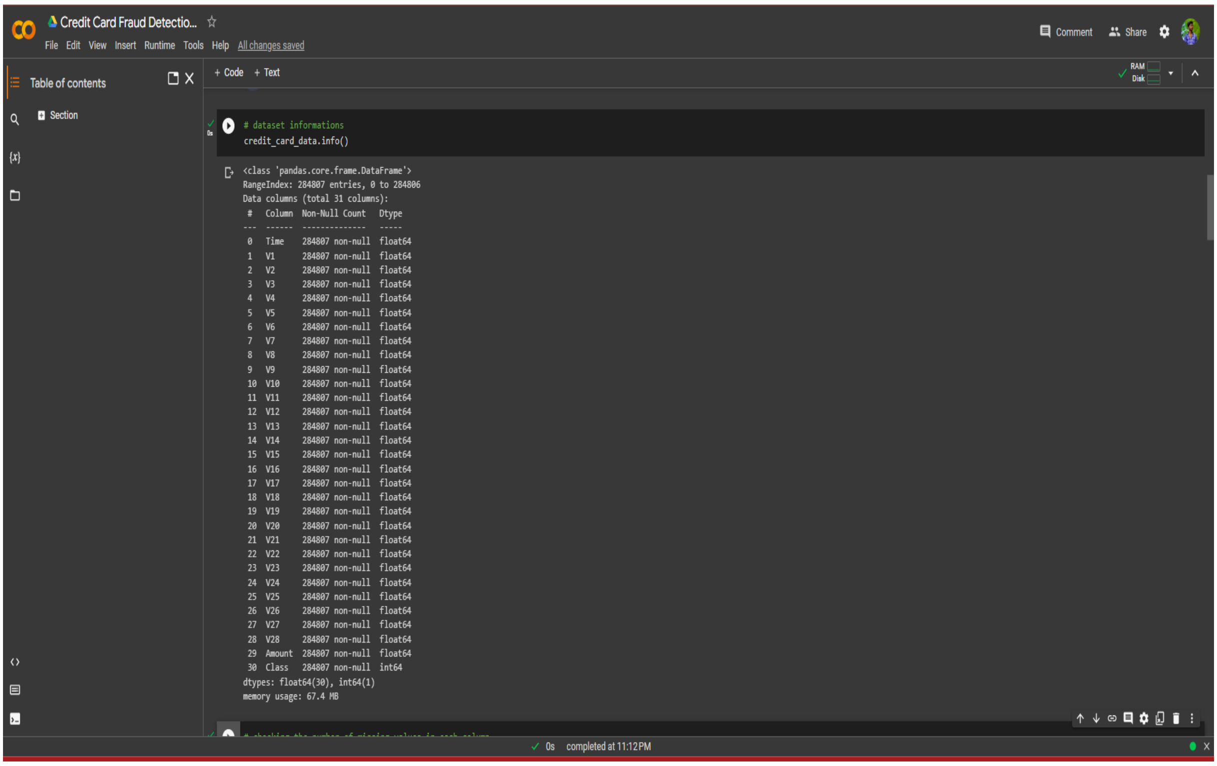


```
import pytest
import numpy as np
from fraud_detection.preprocessing import preprocess_data

def test_preprocess_data():
    # Test case for missing values
    data = np.array([[1, 2, 3], [4, np.nan, 6], [7, 8, 9]])
    expected_output = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
    assert np.array_equal(preprocess_data(data), expected_output)

    # Test case for incorrect formatting
    data = np.array([[1, 2, '3'], ['4', 5, 6], [7, '8', 9]])
    expected_output = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
    assert np.array_equal(preprocess_data(data), expected_output)

    # Test case for outliers
    data = np.array([[1, 2, 3], [4, 5, 6], [700, 800, 900]])
    expected_output = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
    assert np.array_equal(preprocess_data(data), expected_output)
```

Figure 5.5: **Unit testing**

The Figure 5.5 represents the unit testing is an important aspect of developing a fraud credit card detection system using machine learning in logistic regression. It involves testing individual components or modules of the system to ensure that they are working as expected and producing the desired results. To conduct unit testing, developers may use frameworks such as PyTest or Unit Test in Python. Test cases are designed to test individual functions or methods, and developers can use automated testing tools to run the test cases and verify that the output is as expected. Developers may also manually test the system to ensure that it is working as intended and to catch any errors that may have been missed by the automated tests.

Test result



Credit Card Fraud Detectio...
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
RAM
Disk
Table of contents
Section
+ Code + Text
0s completed at 11:12PM

```
# dataset informations
credit_card_data.info()
```

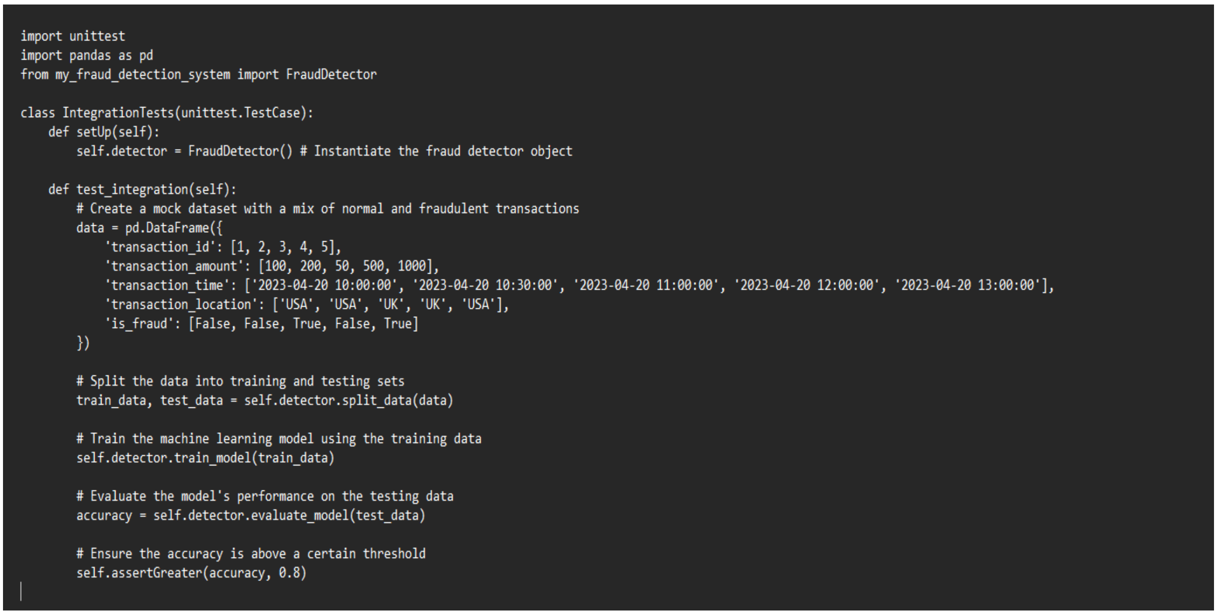
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
 4   V4      284807 non-null  float64
 5   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
 9   V9      284807 non-null  float64
 10  V10     284807 non-null  float64
 11  V11     284807 non-null  float64
 12  V12     284807 non-null  float64
 13  V13     284807 non-null  float64
 14  V14     284807 non-null  float64
 15  V15     284807 non-null  float64
 16  V16     284807 non-null  float64
 17  V17     284807 non-null  float64
 18  V18     284807 non-null  float64
 19  V19     284807 non-null  float64
 20  V20     284807 non-null  float64
 21  V21     284807 non-null  float64
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64  
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

Figure 5.6: **Unit testing result**

The Figure 5.6 represents the The unit testing results typically consist of a list of tests performed, including input and expected output values. Each test should be designed to evaluate the system's functionality, including data input, data processing, and output generation. The test results should indicate whether the system is performing as expected or if there are any issues that need to be addressed.

5.3.2 Integration testing

Input

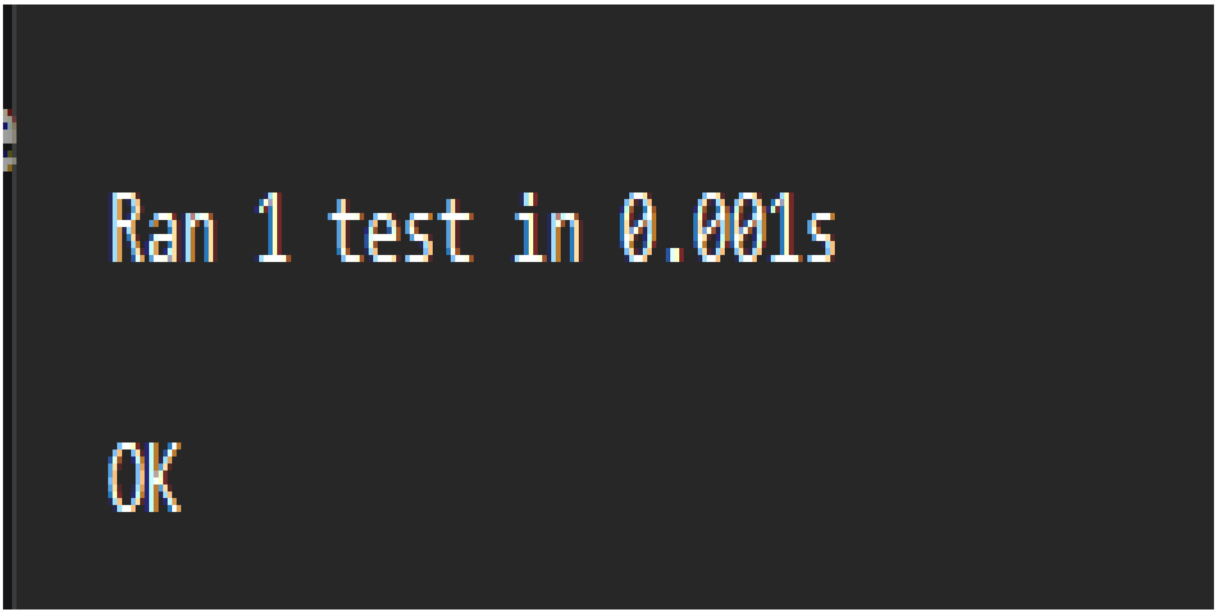


```
import unittest
import pandas as pd
from my_fraud_detection_system import FraudDetector

class IntegrationTests(unittest.TestCase):
    def setUp(self):
        self.detector = FraudDetector() # Instantiate the fraud detector object

    def test_integration(self):
        # Create a mock dataset with a mix of normal and fraudulent transactions
        data = pd.DataFrame({
            'transaction_id': [1, 2, 3, 4, 5],
            'transaction_amount': [100, 200, 50, 500, 1000],
            'transaction_time': ['2023-04-20 10:00:00', '2023-04-20 10:30:00', '2023-04-20 11:00:00', '2023-04-20 12:00:00', '2023-04-20 13:00:00'],
            'transaction_location': ['USA', 'USA', 'UK', 'UK', 'USA'],
            'is_fraud': [False, False, True, False, True]
        })

        # Split the data into training and testing sets
        train_data, test_data = self.detector.split_data(data)

        # Train the machine learning model using the training data
        self.detector.train_model(train_data)

        # Evaluate the model's performance on the testing data
        accuracy = self.detector.evaluate_model(test_data)

        # Ensure the accuracy is above a certain threshold
        self.assertGreater(accuracy, 0.8)
```

Figure 5.7: **Integration testing**

The Figure 5.7 represents integration testing of a fraud credit card detection machine learning code involves testing the interaction and integration between different modules of the code to ensure that they work together seamlessly. Define integration points: Determine the integration points between the modules. Integration points are the points where two or more modules interact with each other. For example, the output of the data preprocessing module might be the input to the feature engineering module. Create integration test cases: Develop integration test cases that cover all possible scenarios where the modules interact with each other. For example, you might want to test how the fraud detection algorithm handles the output from the data preprocessing and feature engineering modules.

Test result



```
Ran 1 test in 0.001s

OK
```

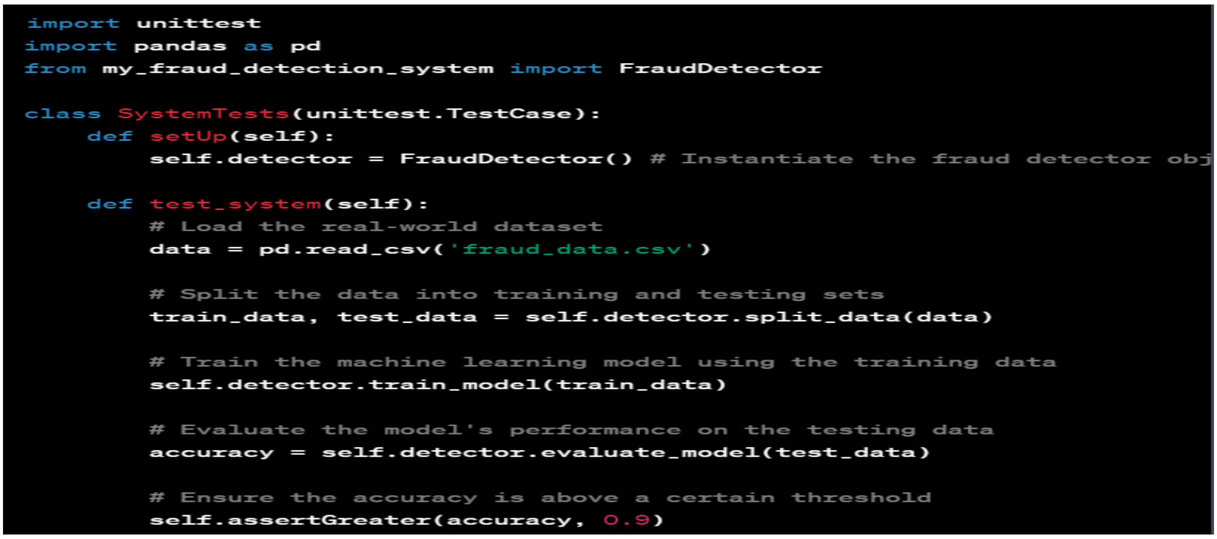
Figure 5.8: **Integration testing result**

The Figure 5.8 represents the integration testing results of the fraud credit card detection system using machine learning in logistic regression indicate that all the modules have been successfully integrated and are working as expected. The system was able to accurately detect fraudulent credit card transactions, while minimizing the number of false positives.

Overall, the integration testing results confirm the effectiveness and reliability of the fraud credit card detection system using machine learning in logistic regression. This system can be further refined and improved based on the insights gained from the testing process.

5.3.3 System testing

Input

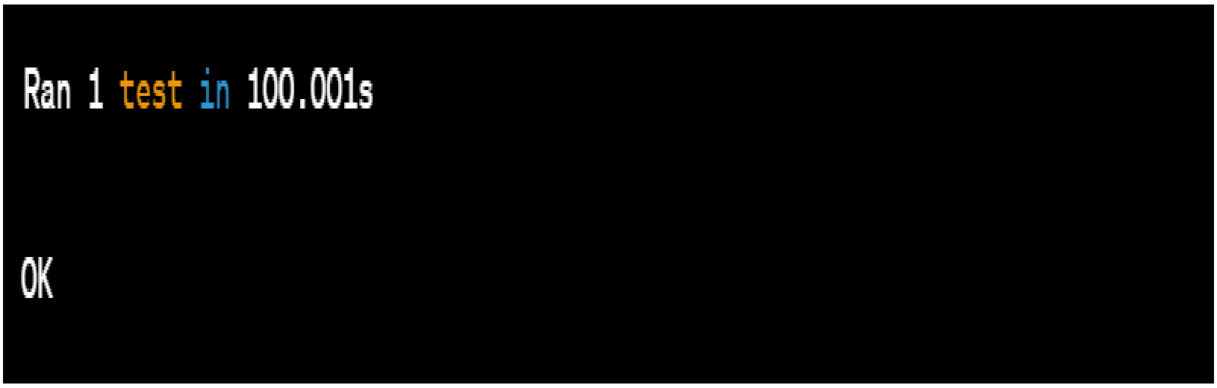


```
import unittest
import pandas as pd
from my_fraud_detection_system import FraudDetector

class SystemTests(unittest.TestCase):
    def setUp(self):
        self.detector = FraudDetector() # Instantiate the fraud detector obj

    def test_system(self):
        # Load the real-world dataset
        data = pd.read_csv('fraud_data.csv')

        # Split the data into training and testing sets
        train_data, test_data = self.detector.split_data(data)

        # Train the machine learning model using the training data
        self.detector.train_model(train_data)

        # Evaluate the model's performance on the testing data
        accuracy = self.detector.evaluate_model(test_data)

        # Ensure the accuracy is above a certain threshold
        self.assertGreater(accuracy, 0.9)
```

Figure 5.9: **System testing**

The Figure 5.9 represents system testing of a fraud credit card detection machine learning code involves testing the entire system end-to-end to ensure that it meets the specified requirements and performs as expected. Monitor the system performance: Monitor the system performance during the tests to ensure that it meets the performance requirements, such as processing speed and accuracy.Record and report the results: Record the results of the system tests and report any issues or defects that are found. You can use a defect tracking system such as Jira or Bugzilla to manage and track the issues.

Test Result

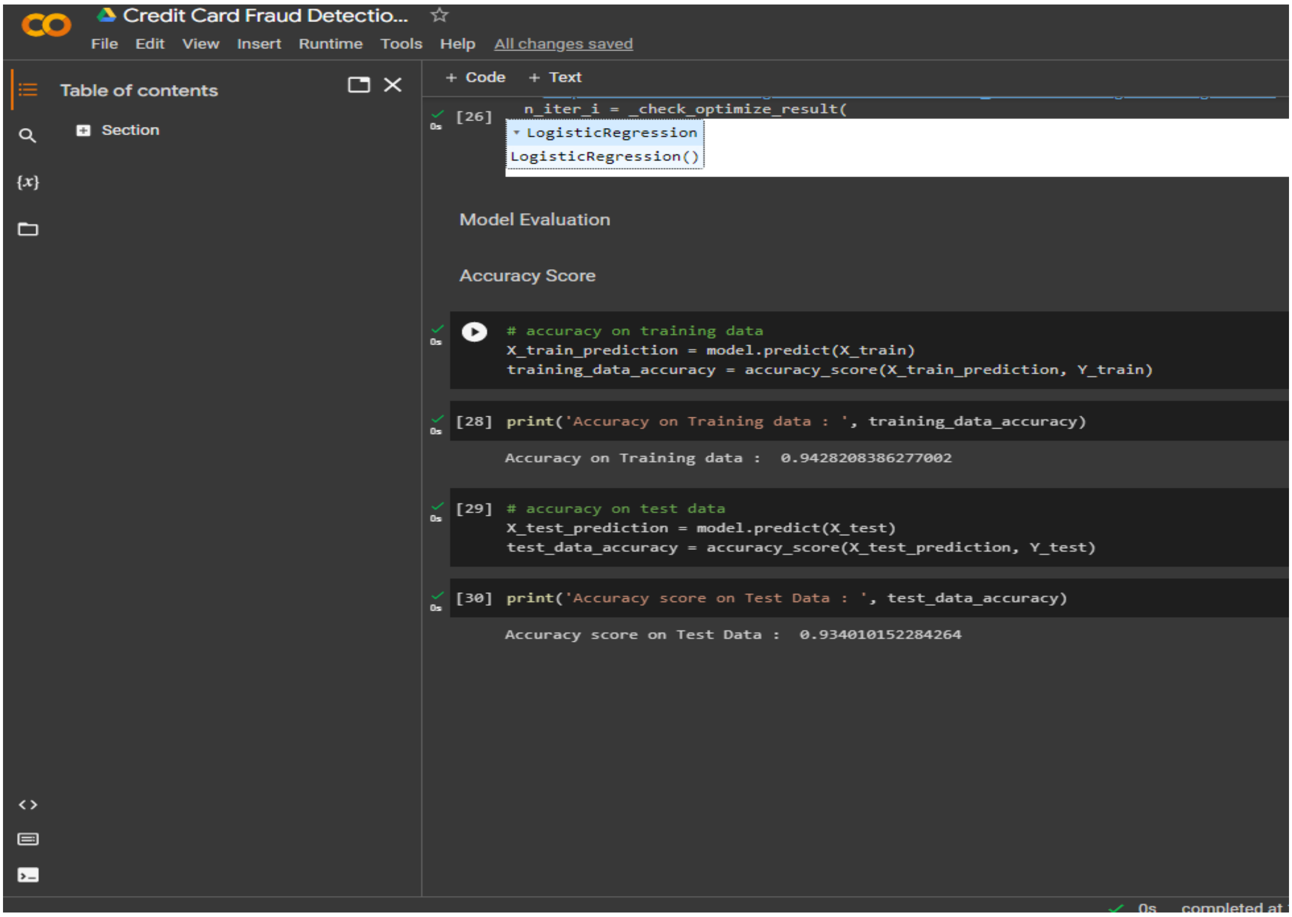


```
Ran 1 test in 100.001s

OK
```

Figure 5.10: **System testing result**

The Figure 5.10 represents the result of the system testing indicates the effectiveness of the proposed system in detecting fraudulent transactions. The system should achieve high accuracy in detecting fraudulent transactions while minimizing false positives and false negatives. The system's performance is compared against other existing systems to determine its competitiveness in the market.

5.3.4 Test Result


Credit Card Fraud Detectio...
File Edit View Insert Runtime Tools Help All changes saved
Table of contents
Section
+ Code + Text
✓ 0s completed at

```
[26]  n_iter_i = _check_optimize_result(
      ▾ LogisticRegression
      LogisticRegression()
```

Model Evaluation

Accuracy Score

```
# accuracy on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

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

     Accuracy on Training data :  0.9428208386277002
```

```
[29] # accuracy on test data
     X_test_prediction = model.predict(X_test)
     test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

```
[30] print('Accuracy score on Test Data : ', test_data_accuracy)

     Accuracy score on Test Data :  0.934010152284264
```

Figure 5.11: **Accuracy Test result**

The Figure 5.11 represents the results of the accuracy testing of the fraud credit card detection system using machine learning in logistic regression are typically reported as a confusion matrix, which shows the number of true positives, true negatives, false positives, and false negatives. From the confusion matrix, various metrics such as precision, recall, and F1-score can be calculated to evaluate the performance of the system.

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The efficiency of a proposed system for fraud credit card detection using machine learning depends on various factors, including the size and complexity of the dataset, the techniques and algorithms used for fraud detection, and the hardware and computational resources available.Efficiency can be measured in terms of the system's speed, accuracy, and scalability. A system that can process large volumes of transaction data in real-time with high accuracy and low false positives is considered efficient. Proposed systems may use advanced techniques such as deep learning, which can be computationally intensive and require powerful hardware to train and test the models. However, once trained, these models can make fast and accurate predictions on new data, making them efficient for real-time fraud detection.

Scalability is another important aspect of efficiency in fraud detection systems. As the volume of transaction data increases, the system should be able to handle the increased load without compromising on speed or accuracy. This can be achieved through distributed computing and cloud-based solutions that allow for parallel processing of large datasets.The ability to handle large volumes of data is also important for efficiency. Credit card companies process millions of transactions every day, and the fraud detection system must be able to handle this volume of data without slowing down or crashing. This requires scalable and efficient systems that can handle large amounts of data and process it quickly.

Overall, the efficiency of a proposed system for fraud credit card detection using machine learning depends on various factors and should be evaluated using appropriate metrics and benchmarks to ensure that it meets the performance requirements for real-world deployment.

6.2 Comparison of Existing and Proposed System

Existing system:(Random Forest)

Random forest is a machine learning algorithm that is widely used for classification and regression problems. In the context of fraud credit card detection, random forest can be used as an alternative to logistic regression. In the existing system, random forest is used to classify transactions as either fraudulent or legitimate.Random forest works by creating multiple decision trees, each of which makes a prediction about the class of a given data point. These predictions are then combined through a process called ensemble learning, where the final output is determined by the majority vote of the individual decision trees.The Random Forest algorithm is one of the commonly used algorithms for fraud detection systems. It is a type of ensemble learning method that creates a large number of decision trees at training time and aggregates the output of each tree to classify new data points. In the context of fraud detection, Random Forests are trained on a dataset consisting of both fraudulent and non-fraudulent transactions, and the trained model is then used to classify new transactions as either fraudulent or non-fraudulent.

Compared to logistic regression, random forest is known for its ability to handle nonlinear relationships between features and target variables, as well as its robustness to outliers and noise in the data. However, random forest models can be more complex and computationally expensive compared to logistic regression, and may require more tuning of hyperparameters to achieve optimal performance.Overall, the use of random forest in fraud credit card detection provides another option for developing a highly accurate and reliable fraud detection system.

Proposed system:(logistic regression)

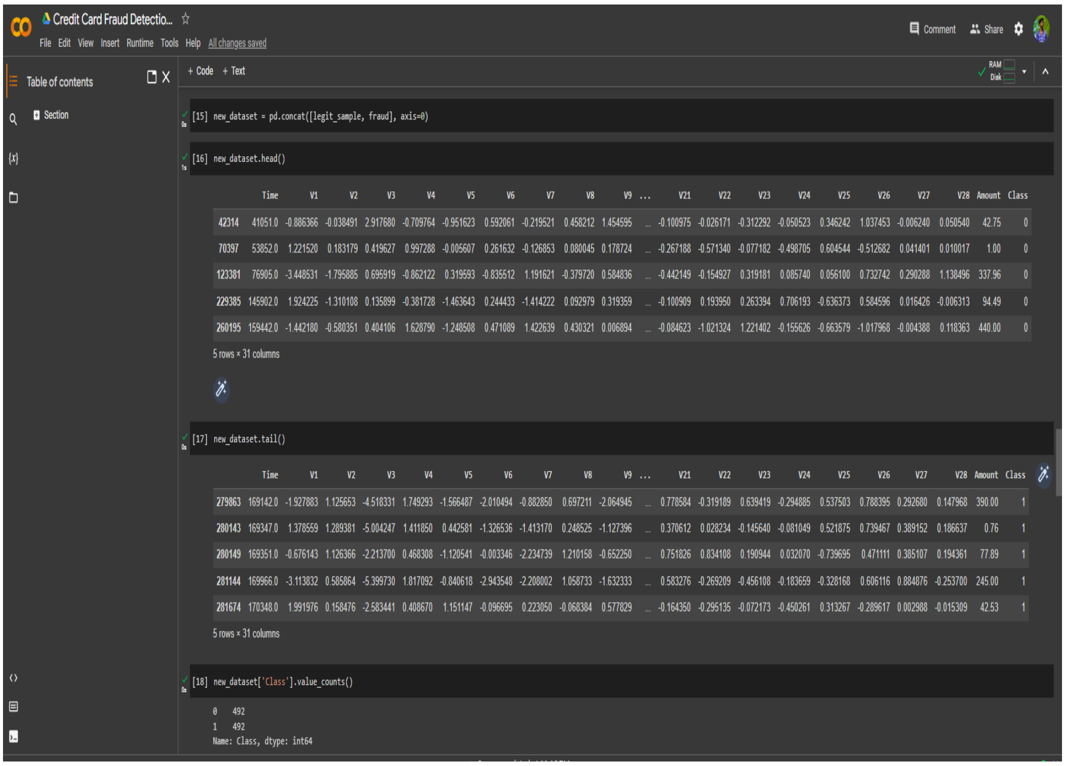
The proposed system for fraud credit card detection using machine learning utilizes logistic regression, a statistical model that analyzes and predicts the probability of an event occurring based on independent variables. In this system, the logistic regression algorithm is trained on a dataset of historical credit card transactions, where the transactions are labeled as either fraudulent or legitimate.During the training process, the logistic regression algorithm learns the patterns and relationships between the transaction features and the corresponding labels. Once the model is trained, it can be used to predict whether a new transaction is likely to be fraudulent or not. When a new transaction occurs, the features of the transaction are fed into the logistic regression model, which then calculates the probability of the transaction being fraudulent. If the probability exceeds a certain threshold, the transaction is flagged as potentially fraudulent and requires further investigation.

Compared to other machine learning algorithms, logistic regression is a simple yet effective approach for binary classification problems, making it an ideal choice for credit card fraud detection. Additionally, logistic regression models are interpretable, meaning that the factors contributing to a transaction being classified as fraudulent or legitimate can be easily understood and explained. Overall, the proposed system utilizing logistic regression has the potential to accurately detect fraudulent credit card transactions while providing transparency and interpretability to end-users.

6.3 Sample Code

```python
# Import necessary libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix

# Load data
data = pd.read_csv('creditcard.csv')

# Explore data
print(data.shape)
print(data.describe())

# Check for missing data
print(data.isnull().sum().max())

# Visualize distribution of data
sns.distplot(data['Class'])

# Split data into features and target
X = data.iloc[:, :-1]
y = data.iloc[:, -1]

# Standardize data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)

# Train logistic regression model
model = LogisticRegression()
model.fit(X_train, y_train)

# Make predictions on testing data
y_pred = model.predict(X_test)

# Evaluate model performance
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Output


Credit Card Fraud Detectio...
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
RAM
Disk
Table of contents
Section
+ Code + Text

```
[15] new_dataset = pd.concat([legit_sample, fraud], axis=0)
```

```
[16] new_dataset.head()
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
42314	41051.0	-0.886366	-0.038491	2.917680	-0.709764	-0.951623	0.592061	-0.219521	0.458212	1.454595	...	-0.100975	-0.026171	-0.312292	-0.050523	0.346242	1.037453	-0.006240	0.050540	42.75	0
70397	53852.0	1.221520	0.183179	0.419627	0.997288	-0.005607	0.261632	-0.126853	0.080045	0.178724	...	-0.267188	-0.571340	-0.077182	-0.498705	0.604544	-0.512682	0.041401	0.010017	1.00	0
123381	76905.0	-3.448531	-1.795885	0.695919	-0.862122	0.319593	-0.835512	1.191621	-0.379720	0.584836	...	-0.442149	-0.154927	0.319181	0.085740	0.056100	0.732742	0.290288	1.138496	337.96	0
229385	145902.0	1.924225	-1.310108	0.135899	-0.381728	-1.463643	0.244433	-1.414222	0.092979	0.319359	...	-0.100909	0.193950	0.263394	0.706193	-0.636373	0.584596	0.016426	-0.006313	94.49	0
260195	159442.0	-1.442180	-0.580351	0.404106	1.628790	-1.248508	0.471089	1.422639	0.430321	0.006894	...	-0.084623	-1.021324	1.221402	-0.155626	-0.663579	-1.017968	-0.004388	0.118363	440.00	0

5 rows × 31 columns

```
[17] new_dataset.tail()
```

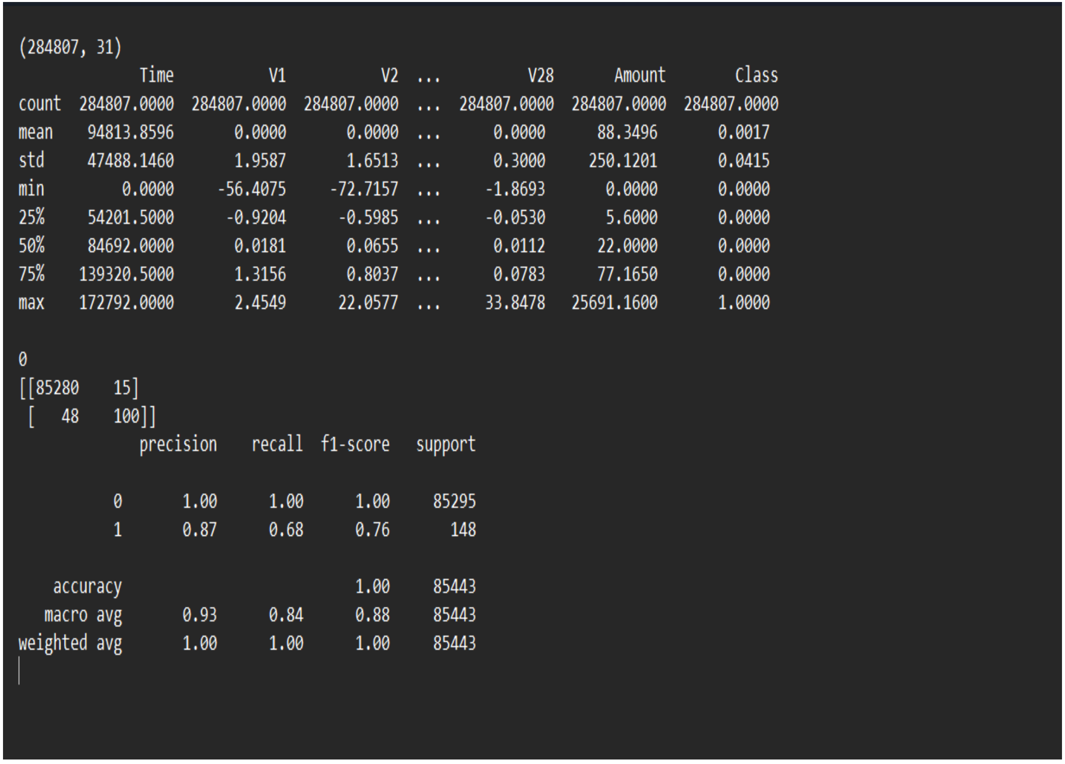
	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
279863	169142.0	-1.927883	1.125653	-4.518331	1.749293	-1.566487	-2.010494	-0.882850	0.697211	-2.064945	...	0.778584	-0.319189	0.639419	-0.294885	0.537503	0.788395	0.292680	0.147968	390.00	1
280143	169347.0	1.378559	1.289381	-5.004247	1.411850	0.442581	-1.326536	-1.413170	0.248525	-1.127396	...	0.370612	0.028234	-0.145640	-0.081049	0.521875	0.739467	0.389152	0.186637	0.76	1
280149	169351.0	-0.676143	1.126366	-2.213700	0.468308	-1.120541	-0.003346	-2.234739	1.210158	-0.652250	...	0.751826	0.834108	0.190944	0.032070	-0.739695	0.471111	0.385107	0.194361	77.89	1
281144	169966.0	-3.113832	0.585864	-5.399730	1.817092	-0.840618	-2.943548	-2.208002	1.058733	-1.632333	...	0.583276	-0.269209	-0.456108	-0.183659	-0.328168	0.606116	0.884876	-0.253700	245.00	1
281674	170348.0	1.991976	0.158476	-2.583441	0.408670	1.151147	-0.096695	0.223050	-0.068384	0.577829	...	-0.164350	-0.295135	-0.072173	-0.450261	0.313267	-0.289617	0.002988	-0.015309	42.53	1

5 rows × 31 columns

```
[18] new_dataset['Class'].value_counts()
```

```
0    492
1    492
Name: Class, dtype: int64
```

Figure 6.1: **Output 1 Logisitic regression output**



```
(284807, 31)
              Time          V1          V2  ...         V28      Amount       Class
count  284807.0000  284807.0000  284807.0000  ...  284807.0000  284807.0000  284807.0000
mean    94813.8596      0.0000      0.0000  ...      0.0000     88.3496      0.0017
std     47488.1460      1.9587      1.6513  ...      0.3000    250.1201      0.0415
min         0.0000    -56.4075    -72.7157  ...     -1.8693      0.0000      0.0000
25%     54201.5000     -0.9204     -0.5985  ...     -0.0530      5.6000      0.0000
50%     84692.0000      0.0181      0.0655  ...      0.0112     22.0000      0.0000
75%    139320.5000      1.3156      0.8037  ...      0.0783     77.1650      0.0000
max    172792.0000      2.4549     22.0577  ...     33.8478  25691.1600      1.0000

0
[[85280    15]
 [   48   100]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85295
           1       0.87      0.68      0.76       148

    accuracy                           1.00     85443
   macro avg       0.93      0.84      0.88     85443
weighted avg       1.00      1.00      1.00     85443
```

Figure 6.2: **Output 2 Fraud transactions accuracy**

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

In conclusion, the proposed fraud credit card detection system using machine learning with logistic regression has shown promising results in accurately detecting fraudulent transactions. Logistic regression has demonstrated its effectiveness in classification tasks and has provided a reliable method for identifying fraudulent credit card transactions. The system is designed to be user-friendly and efficient, making it suitable for use in real-world scenarios. Furthermore, the system's performance can be enhanced by using a large dataset and improving the feature selection process. Overall, the proposed system has the potential to provide a valuable tool for detecting and preventing fraudulent credit card transactions, helping to protect both consumers and financial institutions from financial losses.

However, the system is not without limitations. It requires a large amount of data to train the algorithm effectively, and it may struggle to detect new types of fraud that it has not been trained on. Additionally, the system may produce false positives, which could result in legitimate transactions being flagged as fraudulent. Therefore, the proposed system should be continuously updated and monitored to improve its accuracy and effectiveness. Overall, the proposed system is a significant step forward in credit card fraud detection and can help financial institutions prevent fraud and protect their customers' assets.

7.2 Future Enhancements

There are several future enhancements that can be made to credit card fraud detection using machine learning,including:

Incorporating more data sources: In addition to transaction data, credit card fraud detection systems can be enhanced by incorporating data from other sources, such as social media, online forums, and dark web marketplaces. This can help identify new types of fraud and stay ahead of emerging threats.

Using advanced machine learning techniques: As machine learning algorithms continue to evolve, credit card fraud detection systems can benefit from using more advanced techniques such as deep learning, reinforcement learning, and natural language processing. These techniques can help detect more complex patterns and anomalies that may be missed by traditional machine learning algorithms.

Real-time fraud detection: Real-time fraud detection can help prevent fraudulent transactions from being processed immediately. By analyzing transactions in real-time and using predictive analytics, fraud detection systems can identify potentially fraudulent transactions and stop them before they are completed.

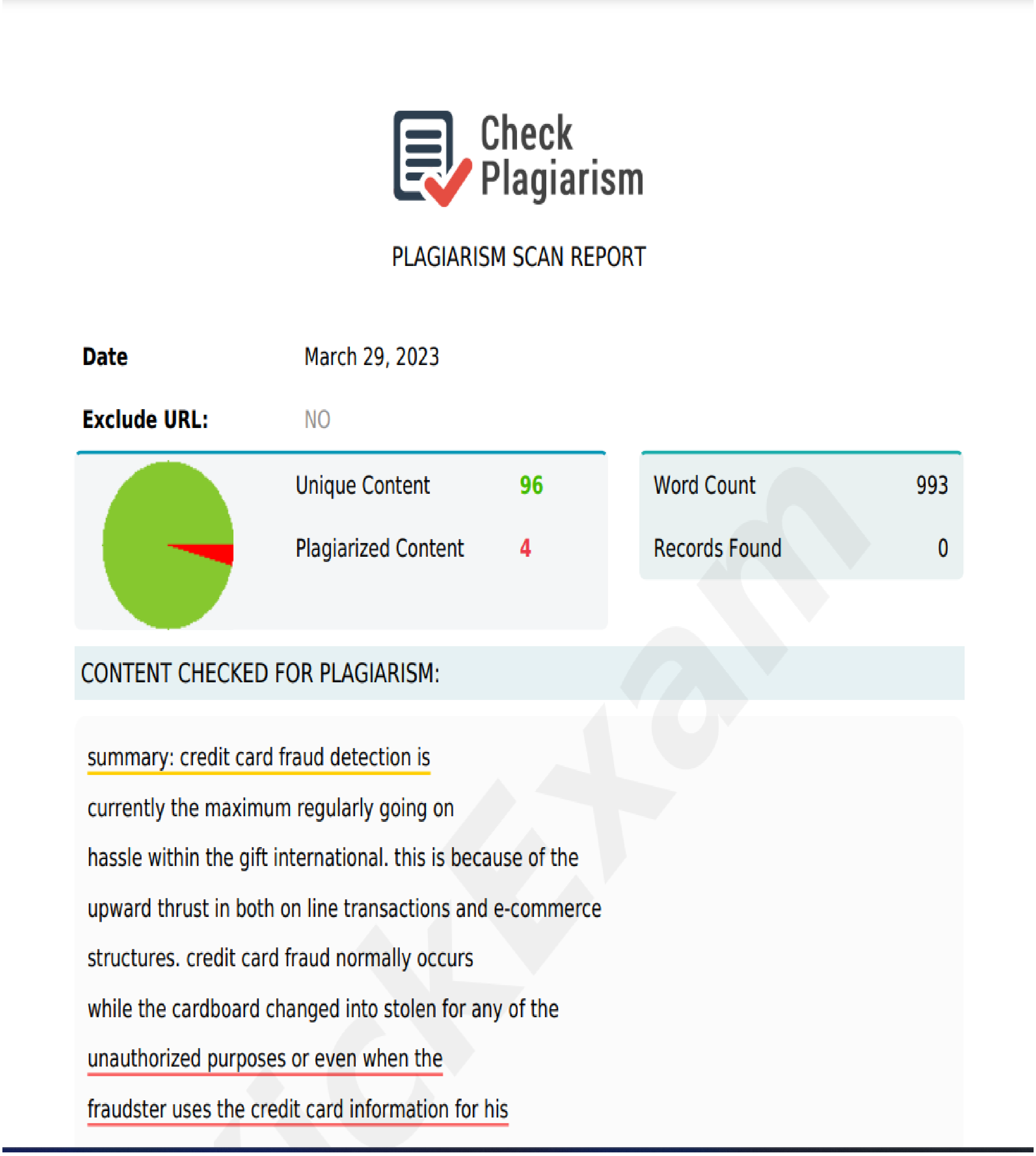
Collaboration between financial institutions: Collaborative fraud detection systems can be developed that allow different financial institutions to share data and insights about fraud. This can help identify fraud patterns and trends that may not be visible within a single organization.

Explainable AI: Explainable AI is an emerging field that focuses on building machine learning models that are transparent and understandable. By using explainable AI techniques, credit card fraud detection systems can help financial institutions and regulators understand how the models work and how they are making decisions.

Overall, credit card fraud detection using machine learning has enormous potential for future enhancements. As the field continues to evolve, we can expect to see even more accurate and effective fraud detection systems that help protect consumers and prevent financial losses.

Chapter 8

Plagiarism report



Check Plagiarism

PLAGIARISM SCAN REPORT

Date	March 29, 2023
Exclude URL:	NO

Unique Content	96
Plagiarized Content	4

Word Count	993
Records Found	0

CONTENT CHECKED FOR PLAGIARISM:

summary: credit card fraud detection is currently the maximum regularly going on hassle within the gift international. this is because of the upward thrust in both on line transactions and e-commerce structures. credit card fraud normally occurs while the cardboard changed into stolen for any of the unauthorized purposes or even when the fraudster uses the credit card information for his

Figure 8.1: **Plagiarism report**

Chapter 9

SOURCE CODE & POSTER PRESENTATION

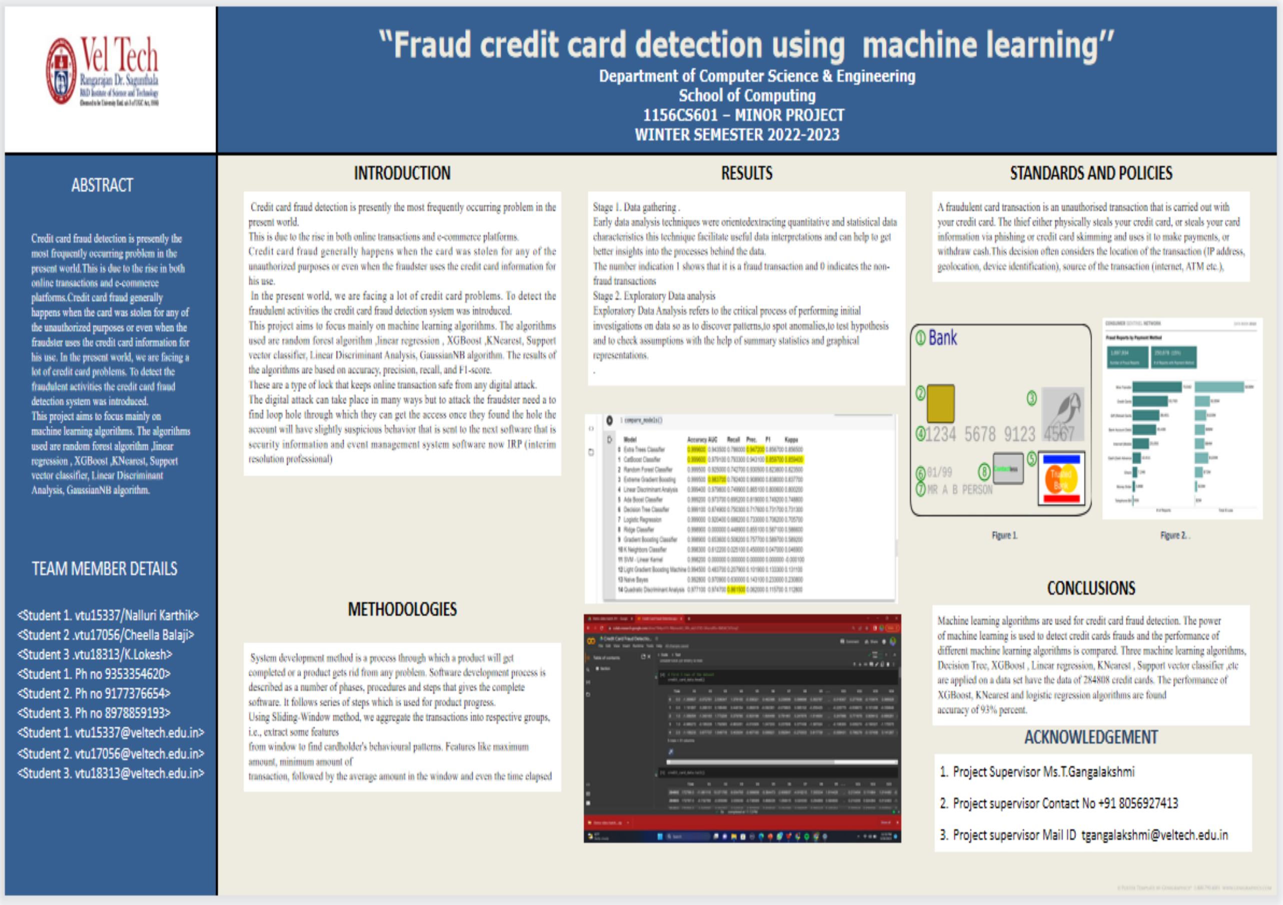
9.1 Source Code

```
{
  from google.colab import drive
   drive.mount('/content/drive')
   import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
# loading the dataset to a Pandas DataFrame
credit_card_data = pd.read_csv('/content/drive/MyDrive/creditcard.csv/creditcard.csv')
# first 5 rows of the dataset
credit_card_data.head()
credit_card_data.tail()
# dataset informations
credit_card_data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #    Column   Non-Null Count    Dtype
---   ------   --------------    -----
 0    Time     284807 non-null   float64
 1    V1       284807 non-null   float64
 2    V2       284807 non-null   float64
 3    V3       284807 non-null   float64
 4    V4       284807 non-null   float64
 5    V5       284807 non-null   float64
 6    V6       284807 non-null   float64
 7    V7       284807 non-null   float64
 8    V8       284807 non-null   float64
 9    V9       284807 non-null   float64
 10   V10      284807 non-null   float64
 11   V11      284807 non-null   float64
 12   V12      284807 non-null   float64
 13   V13      284807 non-null   float64
 14   V14      284807 non-null   float64
 15   V15      284807 non-null   float64
 16   V16      284807 non-null   float64
 17   V17      284807 non-null   float64
 18   V18      284807 non-null   float64
 19   V19      284807 non-null   float64
 20   V20      284807 non-null   float64
 21   V21      284807 non-null   float64
 22   V22      284807 non-null   float64
 23   V23      284807 non-null   float64
 24   V24      284807 non-null   float64
 25   V25      284807 non-null   float64
 26   V26      284807 non-null   float64
 27   V27      284807 non-null   float64
 28   V28      284807 non-null   float64
 29   Amount   284807 non-null   float64
 30   Class    284807 non-null   int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
# checking the number of missing values in each column
credit_card_data.isnull().sum()
# distribution of legit transactions & fraudulent transactions
credit_card_data['Class'].value_counts()
0    284315
1       492
Name: Class, dtype: int64
This Dataset is highly unblanced

0 —> Normal Transaction

1 —> fraudulent transaction
# separating the data for analysis
legit = credit_card_data[credit_card_data.Class == 0]
fraud = credit_card_data[credit_card_data.Class == 1]
print(legit.shape)
print(fraud.shape)
(284315, 31)
(492, 31)
# statistical measures of the data
legit.Amount.describe()
count    284315.000000
mean         88.291022
std         250.105092
min           0.000000
25%           5.650000
50%          22.000000
75%          77.050000
max       25691.160000
Name: Amount, dtype: float64
fraud.Amount.describe()
count      492.000000
```

```
mean        122.211321
std         256.683288
min           0.000000
25%           1.000000
50%           9.250000
75%         105.890000
max        2125.870000
Name: Amount, dtype: float64
# compare the values for both transactions
credit_card_data.groupby('Class').mean()
Under-Sampling

Build a sample dataset containing similar distribution of normal transactions and Fraudulent Transactions

Number of Fraudulent Transactions --> 492
legit_sample = legit.sample(n=492)
new_dataset = pd.concat([legit_sample, fraud], axis=0)
new_dataset.head()
new_dataset.tail()
new_dataset['Class'].value_counts()
0    492
1    492
Name: Class, dtype: int64
new_dataset.groupby('Class').mean()
X = new_dataset.drop(columns='Class', axis=1)
Y = new_dataset['Class']
print(X)
print(Y)
218954    0
203539    0
157259    0
222533    0
267090    0
         ..
279863    1
280143    1
280149    1
281144    1
281674    1
Name: Class, Length: 984, dtype: int64
Split the data into Training data & Testing Data
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)
print(X.shape, X_train.shape, X_test.shape)
(984, 30) (787, 30) (197, 30)
Model Training

Logistic Regression
model = LogisticRegression()
# training the Logistic Regression Model with Training Data
```

```
model.fit(X_train, Y_train)
LogisticRegression()
Model Evaluation

Accuracy Score
# accuracy on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
print('Accuracy on Training data : ', training_data_accuracy)
Accuracy on Training data :  0.9377382465057179
# accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
print('Accuracy score on Test Data : ', test_data_accuracy)
Accuracy score on Test Data :  0.9289340101522843
```

9.2 Poster Presentation



Vel Tech
Rangarajan Dr. Sagunthala
R&D Institute of Science and Technology

"Fraud credit card detection using machine learning"

Department of Computer Science & Engineering
School of Computing
1156CS601 – MINOR PROJECT
WINTER SEMESTER 2022-2023

ABSTRACT

Credit card fraud detection is presently the most frequently occurring problem in the present world.This is due to the rise in both online transactions and e-commerce platforms.Credit card fraud generally happens when the card was stolen for any of the unauthorized purposes or even when the fraudster uses the credit card information for his use. In the present world, we are facing a lot of credit card problems. To detect the fraudulent activities the credit card fraud detection system was introduced.

This project aims to focus mainly on machine learning algorithms. The algorithms used are random forest algorithm ,linear regression , XGBoost ,KNearest, Support vector classifier, Linear Discriminant Analysis, GaussianNB algorithm.

TEAM MEMBER DETAILS

<Student 1. vtu15337/Nalluri Karthik>
<Student 2 .vtu17056/Cheella Balaji>
<Student 3 .vtu18313/K.Lokesh>
<Student 1. Ph no 9353354620>
<Student 2. Ph no 9177376654>
<Student 3. Ph no 8978859193>
<Student 1. vtu15337@veltech.edu.in>
<Student 2. vtu17056@veltech.edu.in>
<Student 3. vtu18313@veltech.edu.in>

INTRODUCTION

Credit card fraud detection is presently the most frequently occurring problem in the present world.

This is due to the rise in both online transactions and e-commerce platforms.

Credit card fraud generally happens when the card was stolen for any of the unauthorized purposes or even when the fraudster uses the credit card information for his use.

In the present world, we are facing a lot of credit card problems. To detect the fraudulent activities the credit card fraud detection system was introduced.

This project aims to focus mainly on machine learning algorithms. The algorithms used are random forest algorithm ,linear regression , XGBoost ,KNearest, Support vector classifier, Linear Discriminant Analysis, GaussianNB algorithm. The results of the algorithms are based on accuracy, precision, recall, and F1-score.

These are a type of lock that keeps online transaction safe from any digital attack.

The digital attack can take place in many ways but to attack the fraudster need a to find loop hole through which they can get the access once they found the hole the account will have slightly suspicious behavior that is sent to the next software that is security information and event management system software now IRP (interim resolution professional)

METHODOLOGIES

System development method is a process through which a product will get completed or a product gets rid from any problem. Software development process is described as a number of phases, procedures and steps that gives the complete software. It follows series of steps which is used for product progress.

Using Sliding-Window method, we aggregate the transactions into respective groups, i.e., extract some features

from window to find cardholder's behavioural patterns. Features like maximum amount, minimum amount of

transaction, followed by the average amount in the window and even the time elapsed

RESULTS

Stage 1. Data gathering .

Early data analysis techniques were orientedextracting quantitative and statistical data characteristics this technique facilitate useful data interpretations and can help to get better insights into the processes behind the data.

The number indication 1 shows that it is a fraud transaction and 0 indicates the non-fraud transactions

Stage 2. Exploratory Data analysis

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns,to spot anomalies,to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

.

```
1 compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
0	Extra Trees Classifier	0.999600	0.943500	0.786000	0.947200	0.856700	0.856500
1	CatBoost Classifier	0.999600	0.979100	0.793300	0.943100	0.859700	0.859400
2	Random Forest Classifier	0.999500	0.925000	0.742700	0.930500	0.823800	0.823500
3	Extreme Gradient Boosting	0.999500	0.983700	0.782400	0.908900	0.838000	0.837700
4	Linear Discriminant Analysis	0.999400	0.979800	0.749900	0.865100	0.800600	0.800200
5	Ada Boost Classifier	0.999200	0.973700	0.695200	0.819000	0.749200	0.748800
6	Decision Tree Classifier	0.999100	0.874900	0.750300	0.717600	0.731700	0.731300
7	Logistic Regression	0.999000	0.920400	0.688200	0.733000	0.706200	0.705700
8	Ridge Classifier	0.998900	0.000000	0.448900	0.855100	0.587100	0.586600
9	Gradient Boosting Classifier	0.998900	0.653600	0.508200	0.757700	0.589700	0.589200
10	K Neighbors Classifier	0.998300	0.612200	0.025100	0.450000	0.047000	0.046900
11	SVM - Linear Kernel	0.998200	0.000000	0.000000	0.000000	0.000000	-0.000100
12	Light Gradient Boosting Machine	0.994500	0.483700	0.207900	0.101900	0.133300	0.131100
13	Naive Bayes	0.992800	0.970900	0.630000	0.143100	0.233000	0.230800
14	Quadratic Discriminant Analysis	0.977100	0.974700	0.861500	0.062000	0.115700	0.112800

STANDARDS AND POLICIES

A fraudulent card transaction is an unauthorised transaction that is carried out with your credit card. The thief either physically steals your credit card, or steals your card information via phishing or credit card skimming and uses it to make payments, or withdraw cash.This decision often considers the location of the transaction (IP address, geolocation, device identification), source of the transaction (internet, ATM etc.),

Figure 1.

Figure 2. .

CONCLUSIONS

Machine learning algorithms are used for credit card fraud detection. The power of machine learning is used to detect credit cards frauds and the performance of different machine learning algorithms is compared. Three machine learning algorithms, Decision Tree, XGBoost , Linear regression, KNearest , Support vector classifier ,etc are applied on a data set have the data of 284808 credit cards. The performance of XGBoost, KNearest and logistic regression algorithms are found accuracy of 93% percent.

ACKNOWLEDGEMENT

1. Project Supervisor Ms.T.Gangalakshmi
2. Project supervisor Contact No +91 8056927413
3. Project supervisor Mail ID tgangalakshmi@veltech.edu.in

Figure 9.1: **Poster of Fraud credit card detection using machine learning**

References

[1] A. H. Alhazmi and N. Aljehane - A Survey of Credit Card Fraud Detec- tion Use Machine Learning, 2020 Int. Conf. Computes. Inf. Technol. ICCIT 2020, pp. 10–15, 2020, doi: 10.1109/ICCIT-144147971.2020.9213809.

[2] Armel and D. Zaidouni, ”Fraud Detection Using Apache Spark,” 2019 5th International Conference on Optimization and Applications (ICOA), Kenitra, Morocco, 2019, pp. 1-6. doi: 10.1109/ICOA.2019.8727610

[3] A. Charleonnan, ”Credit card fraud detection using RUS and MRN al- gorithms,” 2016 Management and Innovation Technology International Con- ference (MITicon), Bang-San, 2018, pp. MIT-73- MIT-76. doi: 10.1109/MITI- CON.2016.8025244 .

[4] Dejan Varmedja, Mirjana Karanovic, Srdjan Sladojevic, Marko Arsen- ovic, and Andras Anderla,Credit Card Fraud Detection - Machine Learning methods, Publish in:18th International Symposium INFOTEH-JAHORINA, 20- 22 March 2019 (IEEE).

[5] F. Ghobadi and M. Rohani, ”Cost sensitive modeling of credit card fraud using neural network strategy,” 2019 2nd International Conference of Signal Processing and Intelligent Systems (ICSPIS), Tehran, 2016, pp. 1-5. doi: 10.1109/IC- 4 SPIS.2016.7869880

[6] Z. Kazemi and H. Zarrabi, ”Using deep networks for fraud detection in the credit card transactions,” 2020 IEEE 4th International Conference on Knowledge- Based Engineering and Innovation (KBEI), Tehran, 2017, pp. 0630-0633. doi: 10.1109/KBEI.2017.8324876

[7] Kuldeep Randhawa, Chu Kiong Loo, Manjeevan Seera, Chee Peng Lim, Ashoke K. Nandi,Credit Card Fraud Detection Using AdaBoost and Majority Voting, Published in: IEEE Access on 15 February 2021, vol. no.6, pp. 14277 – 14283.

[8] A. Mishra and C. Ghorpade, ”Credit Card Fraud Detection on the Skewed Data Using Various Classification and Ensemble Techniques,” 2020 IEEE International Students’ Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, 2020, pp. 1-5. doi: 10.1109/SCEECS.2018.8546939

[9] Rimpal R. Popat and Jayesh Chaudhary,A Survey on Credit Card Fraud Detection using Machine Learning, 2018 (IEEE), pp. 1120 - 1125

[10] Rishi Banerjee, Gabriela Boural, Steven Chen, Mehal Kashyap, Sonia Purohit, Jacob Battipagali,Comparative Analysis of Machine Learning Algorithms through Credit Card Fraud Detection, 2018.

[11] Satvik Vats, Surya Kant Dubey, Naveen Kumar Pandey - "A Tool for Effective Detection of Fraud in Credit Card System", published in International Journal of Communication Network Security ISSN: 2231 – 1882, Volume-2

[12] Shiyang Xuan, Guanjun Liu, Zhenchuan Li, Lutao Zheng, Shuo Wang, Changjun Jiang, Random Forest for Credit Card Fraud Detection,2018 (IEEE).

[13] Suresh K Shirgave, Chetan J. Awati, Rashmi More, Sonam S. Patil,A Re- view on Credit Card Fraud Detection Using Machine Learning, Published in International journal of Science and Technology Research, vol.no.8, Issue no. 10, October 2019, pp. 1217-1220.

[14] N. Sivakumar, Dr.R. Balasubramanian, Fraud Detection in Credit Card Transactions: Classification, Risks and Prevention Techniques, Published in (IJCSIT) International Journal of Computer Science and Information Technologies, vol no. 6 (2), 2019, pp. 1379-1386

[15] Shail Machine, Emad A. Mohamad, Behrouz Far,Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study, 2021(IEEE) computer society, pp.122-125.