Credit Card Fraud Detection Using Machine Learning and Model Integration with Web Framework

A Project Report

Submitted by

GRISHMA AV	***20191IST0182***
TEJAS PRAKASH D	***20191IST0163***
SWETHA A NAIR	***20191IST0157***
SNEHA M	***20191IST0150***
SAI MEGHANA J S	***20191IST0187***

Under the guidance of

Dr. Saritha K

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION SCIENCE AND TECHNOLOGY

At

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

PRESIDENCY UNIVERSITY

BENGALURU

JUNE 2023

Credit Card Fraud Detection Using Machine Learning and Model Integration with Web Framework

A PROJECT REPORT

Submitted by,

GRISHMA AV	***20191IST0182***
TEJAS PRAKASH D	***20191IST0163***
SWETHA A NAIR	***20191IST0157***
SNEHA M	***20191IST0150***
SAI MEGHANA J S	***20191IST0187***

Under the guidance of,

Dr. Saritha K

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION SCIENCE AND TECHNOLOGY
At

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

PRESIDENCY UNIVERSITY

BENGALURU

JUNE 2023

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Project report **"Credit Card Fraud Detection Using Machine Learning and Model Integration with Web Framework"** being submitted by

Grishma AV	*20191IST0182*
Tejas Prakash D	*20191IST0163*
Swetha A Nair	*20191IST0157*
Sneha M	*20191IST0150*
Sai Meghana JS	*20119IST0187*

in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Information Science and Technology is a bonafide work carried out under my supervision.

Dr. SARITHA K
DESIGNATION
School of CSE&IS
Presidency University

Dr. POORNIMA S
DESIGNATION
School of CSE&IS
Presidency University

Dr. C. KALAIARASAN
Associate Dean
School of CSE&IS
Presidency University

Dr. MD. SAMEERUDDIN KHAN
Dean
School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **TITLE OF THE PROJECT** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Information Science and Technology**, is a record of our investigations carried out under the guidance **Dr. Saritha K**, **School of Computer Science & Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

GRISHMA AV	***20191IST0182***
TEJAS PRAKASH D	***20191IST0163***
SWETHA A NAIR	***20191IST0157***
SNEHA M	***20191IST0150***
SAI MEGHANA JS	***20191IST0187***

ACKNOWLEDGEMENT

First of all, we are indebted to **GOD ALMIGHTY** for allowing me to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science & Engineering, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Dean **Dr. C. Kalaiarasan,** Professor **Dr. T K Thivakaran,** University Project-II In-charge, School of Computer Science & Engineering, Presidency University for rendering timely help for the successful completion of this project.

We would like to convey our gratitude and heartfelt thanks to the University Project-II Coordinators **Mr. Mrutyunjaya MS, Mr. Sanjeev P Kaulgud, Mr. Rama Krishna K, and Dr. Madhusudhan MV**.

We are greatly indebted to our guide **Dr. Saritha K**, School of Computer Science & Engineering, Presidency University for her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

GRISHMA AV

TEJAS PRAKASH D

SWETHA A NAIR

SNEHA M

SAI MEGHANA J S

ABSTRACT

Due to the potential for huge financial losses, credit card theft is a serious problem for both financial institutions and consumers. Algorithms that use machine learning are frequently used to identify fraudulent credit card transactions. The Naive Bayes, Logistic Regression, K-Nearest Neighbors (KNN), and Decision Tree machine learning methods for credit card fraud detection are examined in this research. A dataset on actual credit card fraud was used for the study. This study's major goal is to evaluate the algorithms' abilities to spot fraudulent transactions by contrasting and comparing them. The findings show that, though with variable degrees of success, all four algorithms may be used to detect credit card fraud. This study adds to the body of knowledge by revealing information on how well various machine learning algorithms work to identify credit card fraud. The main goal is to identify fraud by filtering the aforementioned methods to get a better result.

Keywords: Credit Card, Fraud Detection, Logistic Regression, Decision Tree, KNN, Naïve-Bayes, Streamlit, MLP, Accuracy, Precision, F1-Score, support, Recall, Macro average, Weighted average.

List of Figures

TABLE OF CONTENTS

1. INTRODUCTION

The term "credit card fraud" refers broadly to theft and fraud that are committed with or while using a credit card to make a purchase. As the number of con artists rises daily. The use of credit cards in fraudulent activity is widespread, and there are many different types of fraud. To solve this issue, many methods including Logistic Regression, Decision Tree Random Forest, and Naive Bayes algorithms are used. Each aspect of this transaction is considered, and the best course of action is followed. To get a better result, the main goal is to identify fraud by filtering the aforementioned tactics.

Fraud and unusual transaction on credit cards can be detected by machine learning algorithms. The first and most important phase, after which the model is trained to predict the probability of fraud, is the collection and organization of raw data. In order to detect fraud on credit cards, some of the solutions provided by machine intelligence are as follows: To determine whether a credit card transaction is genuine or false one can use algorithms such as logistic regression, random forests, support machines Vector SVMs), Deep neural network combined with autoencoders, long short term memory LTTMLS Networks and convolutional neural networks determining through credit card profiling if someone is using a credit card legitimately or fraudulently.

Using techniques for outlier identification to spot transactions that are noticeably different from typical credit card transactions (or "outliers") can help uncover credit card fraud.

2. ABOUT THE PROJECT

2.1 PROJECT TITLE

Credit Card Fraud Detection Using Machine Learning and Model Integration with Web Framework.

2.2 INTRODUCTION

In the digital age, credit card fraud is becoming more and more common. Criminals are using new ways of stealing credit card data and making unauthorized purchases as a result of the increase in online shopping and electronic payment systems. The financial and personal impact of credit card frauds can be significant, which is why it is important to identify and prevent them in the first place. To identify fraud with credit cards, we intend to analyze a wide variety of features of the transaction including card number, CVV, expiry date, repeat retailers, used chip and PIN as well as online order history using machine intelligence techniques. In order to avoid financial losses on both the part of individuals and a financial institution, it is aimed at developing an accurately and effective model that can spot fraud in real time.

For the detection of fraudulent activity in financial transactions, machine learning has proved to be a valuable tool. Machine learning algorithms are able to detect potentially fraudulent behavior with considerable accuracy and effectiveness by analyzing patterns and anomalies across a wide range of datasets. In this project, we will utilize four different machine learning algorithms (Naive Bayes, KNN, Logistic Regression, and Decision Tree) to develop a model that can accurately predict the likelihood of credit card fraud based on the provided features. A successful detection of credit card fraud through machine learning could lead to reduced financial losses and improve overall safety in the use of cards. Through this project, we hope to demonstrate the potential of machine learning techniques in the fight against credit card fraud.

The Objectives of this Project are:

- To detect fraudulent activities in credit card transactions and predict the result.
- Compare some efficient machine learning algorithms, find better accuracy, and suggest a model.
- Visualize dataset through model graphs using Python libraries.
- Integrate machine learning model in the web-based framework for better user interface and user experience.

- To develop an accurate and efficient machine learning model for detecting credit card fraud.
- To integrate the selected machine learning model with the Streamlit web framework to create a user-friendly and interactive interface for fraud detection.
- To provide a reliable and secure fraud detection system that can be deployed to production environments such as web servers or mobile applications.
- To improve the performance of the model using techniques such as oversampling or undersampling, ensemble learning, or hyperparameter tuning.
- To comply with data protection laws and regulations such as the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA) to ensure the
  privacy and security of the data used for training and testing the model.
- To evaluate the performance of the model using metrics such as accuracy, precision, recall, and F1 score and identify areas for improvement.
- To provide a solution that can be easily maintained and updated as new data and techniques become available.
- To increase the efficiency and speed of the fraud detection process, thus reducing the time taken to detect and prevent fraudulent transactions.
- To reduce financial losses for individuals and companies that fall victim to credit card fraud.
- To contribute to the advancement of machine learning and data science techniques for fraud detection and prevention.
- To provide a useful tool for financial institutions, merchants, and individuals to monitor their credit card transactions and prevent fraudulent activities.
- To increase awareness about the prevalence of credit card fraud and the importance of using advanced technology to prevent it.

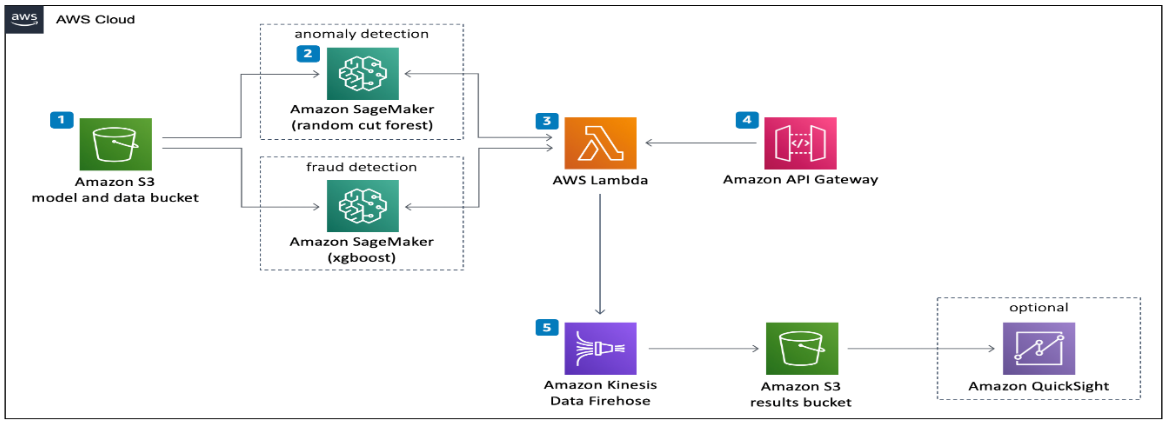


Figure 2.2 Fraud Detection Using Machine Learning

2.3 TECHNOLOGY USED:

Hardware Requirements:

• Processor: Intel i3 or higher

• RAM: 4GB or Higher

• STORAGE: 1 TB or Higher

• GPU enabled.

Software Requirements:

• Operating System: Windows 7/8/10 or Ubuntu

• Tools: Python, Anaconda Navigator, Sci-kit libraries, PyCharm, Streamlit, Pandas, NumPy

2.4 INDUSTRIAL SCOPE:

The use of credit cards has significantly expanded in recent years all over the world. People increasingly believe in going cashless and are entirely reliant on online transactions. The credit card has made digital transactions easier. According to the 2017 PwC global economic crime survey, economic crime affected 48% of organizations.

The term "credit card fraud" refers broadly to theft and fraud that are committed with or while using a credit card to make a purchase. As the number of con artists rises daily. Huge financial losses have been caused by fraud and have affected credit users personally as well as businesses and banks. Fraud can result in non-financial losses by harming a business's reputation and image. For instance, if a cardholder experiences fraud with a specific company, he may lose faith in them and choose a rival. Fraud detection is the practice of keeping track of a cardholder's transaction patterns to determine if an incoming transaction is legitimate and authorized or not. If not, the transaction will be marked as fraudulent. Several businesses and people are impacted by the major problem of credit card theft. An effective method for identifying credit card fraud is machine learning. Through the use of machine learning algorithms and the Streamlit web framework, we can create a system for detecting credit card fraud in this project. Data collection is the initial stage of system construction. Credit card firms, banks, and other financial institutions are some of the places where we can get credit card transaction data. To train our machine learning algorithms, we will use this data as a starting point. Next, we can train our model using a variety of machine-learning techniques like logistic regression, decision trees, and random forests. These algorithms can assist in locating credit card transaction trends and abnormalities that might point to fraud. Feature selection and dimensionality reduction are two other methods we may employ to increase the precision of our model. After our model has been trained, we can include it in the Streamlit web framework. Streamlit is a library used in Python that is used to create interactive web applications for the analysis of data and visualization. By integrating our model with Streamlit, we can design a user-friendly interface that allows customers to input their credit card transaction data and receive a fraud detection forecast.

In addition to integrating our model with Streamlit, we can also use other web technologies such as Flask or Django to create a more robust and scalable application. We can also use data visualization libraries such as Matplotlib and Seaborn to create graphs and charts that help users understand their credit card transaction data. We must also take the necessary precautions to protect credit card transaction data to ensure its security and privacy. This involves employing secure communication protocols, access controls, and encryption when transferring data between a user's device and a web server. Additionally, we must abide by all applicable data protection laws and rules, including the California Consumer Privacy Act (CCPA) in the US and the General Data Protection Regulation (GDPR) in Europe. In addition to fraud detection, we can also use machine learning algorithms to identify other types of financial crimes, such as money laundering or insider trading. By analyzing patterns and anomalies in financial transaction data, we can detect suspicious activities and alert relevant authorities or compliance teams. Overall, the machine learning-based system for detecting credit card fraud with Streamlit model integration is a useful tool for organizations and individuals who want to safeguard themselves against credit card fraud. By integrating machine learning algorithms and web technologies, we can design a system that is accurate, efficient, and user-friendly.

2.5 GOAL

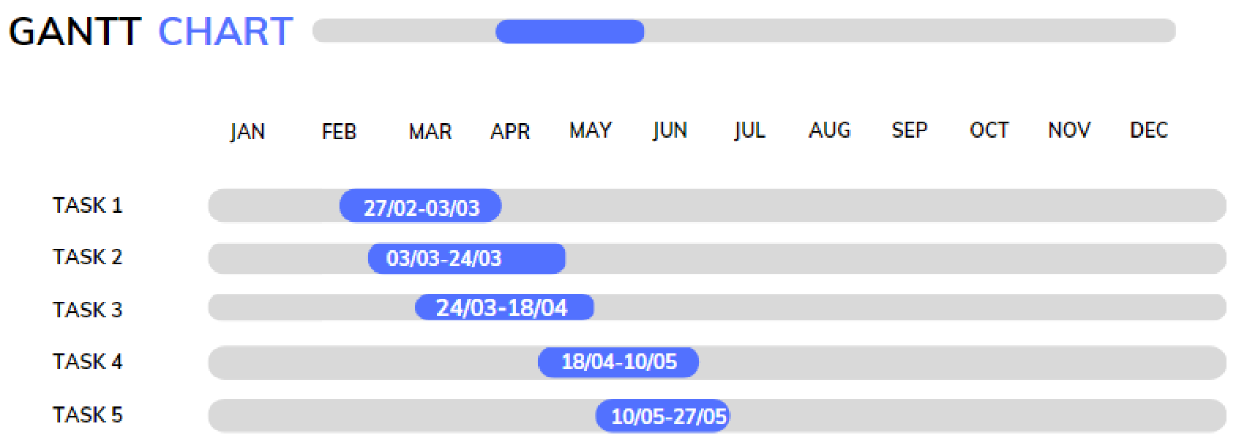
We're trying to predict the detection of credit card fraud in our study. We use four different machine learning algorithms in the process of predicting success: Statistical Regression, KNN, Decision Tree and Random Forest. We've taken into account some parameters in order to check the efficiency of a machine learning algorithm.

There are six main modules of the project:

- Gathering data
- Preparing the data
- Choosing a model
- Training
- Prediction
- Evaluation of the model

We aim to predict credit card fraud detection, for prediction supervised learning is the best methodology.

2.6 PROJECT TIMELINE



TASK 1 - TITLE FINALIZATION
TASK 2 - REVIEW 1 WITH PRESENTATION
TASK 3 - REVIEW 2 ALGORITHM DETAILS WITH 50% CODE
TASK 4 - REVIEW 2 ALGORITHM DETAILS WITH 50% CODE
TASK - 5 FINAL SUBMISSION

Figure 2.6 Gantt Chart

3. LITERATURE SURVEY

[1] Fatima Ibrahim, Mohammed Al-Banaw, and Mohd Faizal Abdollah. Credit Card Fraud Detection: A Survey. Journal of Network and Computer Applications, 135:62-81, 2019.

Credit card fraud is a major concern for both financial institutions and customers. This project provides a comprehensive survey of credit card fraud detection techniques, covering traditional rule-based systems, anomaly detection, and machine learning approaches. The project also discusses the limitations and challenges of current fraud detection techniques and highlights potential future research directions.

[2] Amal Alphonse, Kalaivani Chellappan, and Indumathi J. Credit Card Fraud Detection using Machine Learning Techniques. Proceedings of the 2019 International Conference on Automation, Computational and Technology Management, 2019.

This project proposes a credit card fraud detection system based on machine learning techniques. The authors compare the performance of three different machine learning algorithms - Decision Tree, Random Forest, and Support Vector Machine (SVM) - and evaluate the impact of feature selection on the accuracy of the model. The experimental results show that SVM outperforms the other two algorithms, and feature selection significantly improves the performance of the model.

[3] Mridul Shukla and Kapil Sharma. Anomaly Detection Approach for Credit Card Fraud Detection. Proceedings of the 2019 International Conference on Intelligent Computing and Control Systems, 2019.

This project presents an anomaly detection approach for credit card fraud detection. The authors propose a novel feature selection technique that combines correlation-based feature selection with principal component analysis (PCA) to select the most relevant features. They then apply various anomaly detection techniques, including One-Class SVM, Local Outlier Factor (LOF), and Isolation Forest, to identify fraudulent transactions. The experimental results demonstrate that the proposed approach outperforms existing methods in terms of accuracy and false positive rate.

[4] S. Vijaya Kumar, K. Venkateswara Rao, and K. Sujatha. Credit Card Fraud Detection using Deep Learning. Proceedings of the 2019 International Conference on Communication and Electronics Systems, 2019.

This project proposes a credit card fraud detection system based on deep learning techniques. The authors use a stacked auto-encoder neural network to extract features from credit card transaction data and train a classifier to distinguish between legitimate and fraudulent transactions. The experimental results demonstrate that the proposed system achieves high accuracy in detecting fraudulent transactions, with a low false positive rate.

[5] Fraud Detection in Credit Card Transactions Using SVM and Random Forest Algorithms, S K Saddam Hussain; E Sai Charan Reddy; K Gangadhar Akshay; T Akanksha, 2021 Fifth International Conference

The main goal of this project is to identify credit card fraud in the real world. Credit card transactions have significantly increased as a result of recent expansion. The purpose is to receive things from an account without paying for them or using unauthorized cash. All credit card-issuing banks must work to lower the cost of putting in place an efficient fraud detection system. The fact that a credit card transaction does not require the card or the cardholder to be completed is one of the trickiest problems.

As a result, the vendor is unable to confirm whether the consumer making the purchase is an authorized cardholder. This method, which combines the random forest, decision tree, and support vector machine techniques, improves the accuracy of fraud detection. A random forest algorithm is a classification procedure for watching the data set and enhancing the correctness of the produced data. The effectiveness of the procedures is assessed based on their precision, sensitivity, and accuracy. To detect fraud and provide visualization for the graphic model, some of the provided data are processed.

[6] Credit Card Fraud Detection Using Machine Learning, Deep Prajapati; Ankit Tripathi; Jeel Mehta; Kirtan Jhaveri; Vishakha Kelkar, Published in 2021 International Conference on Advances in Computing, Communication, and Control (ICAC3)

People have been eager to devise new techniques to gain unauthorized access to another person's finances ever since payment systems first appeared. Given that the vast majority of transactions are now conducted online utilizing credit card information, this ominous risk has increased in the present. A general term used to describe any sort of fraud involving a payment card, specifically a credit card, is "frauds due to credit cards. "Such offenses typically have a single goal of obtaining goods and services or making a sizable payment to another account without the owner's permission. The Nilson Report estimates that by 2025, the United States would have lost up to $12.5 billion as a result of credit card theft. To get the best results in identifying and preventing fraudulent transactions, using machine learning algorithms to detect credit card fraud entails investigating the data using a variety of methods. We have employed strategies like Random Forest, XGBoost, and ANN (Artificial Neural Network) to compare several algorithms that successfully identify credit card fraud. These models' outputs can be used to accurately identify any credit card transaction that is taking place, whether it is legitimate or fraudulent.

[7] The Use of Predictive Analytics Technology to Detect Credit Card Fraud in Canada, Kosemani Temitayo Hafiz; Shaun Aghili; Pavol Zavarsky, 2016 11th Iberian Conference on Information Systems and Technologies (CISTI)

This study focuses on the development of a scorecard using pertinent evaluation criteria, features, and capabilities of vendor solutions for predictive analytics currently employed to identify credit card fraud. A side-by-side comparison of five credit card predictive analytics vendor solutions used in Canada is provided by the scorecard. The research that followed produced several issues, dangers, and restrictions related to credit card fraud PAT vendor solutions.

[8] "Credit Card Fraud Detection Using Machine Learning: A Systematic Review" by R. Hasan, M. M. Islam, and N. R. Chaki.

The decision trees, support vector machines, and neural networks used in this paper thorough examination of machine learning methods for credit card fraud detection. Additionally, it talks about the limitations of these methods and suggests some lines of future investigation. The article can be found in the "International Journal of Computer Applications Technology and Research" (IJCATR), Volume 5, Issue 5, May 2016, pp. 319–325 (in English).

[9] "Credit Card Fraud Detection: A Realistic Modelling and a Novel Learning Strategy" by P. Mohammadi, H. Beigy, and S. Araban.

Based on thorough modeling of the fraud detection problem, this research suggests a unique learning technique for credit card fraud detection. On benchmark datasets, it is demonstrated that the proposed strategy, which combines supervised classification with unsupervised anomaly detection, performs better than existing methods. The article is in "Expert Systems with Applications," Volume 42, Issue 7, April 2015, pp. 3634–3648.

[10] Deep Learning-Based Fraud Detection System for Credit Cards: A Comprehensive Review (2021)

An in-depth analysis of current advancements in deep learning-based credit card fraud detection systems is provided in this study. The authors discuss different deep learning techniques, such as generative adversarial networks (GANs), long short-term memory (LSTM) networks, and convolutional neural networks (CNNs), and their uses in detecting credit card fraud. The difficulties and restrictions of applying deep learning algorithms in this field are also covered.

[11] Credit Card Fraud Detection Using Machine Learning Techniques: A Review (2020)

The machine learning methods used to identify credit card fraud are reviewed in this study. The authors analyze different machine-learning techniques used in detecting credit card fraud, such as decision trees, support vector machines (SVMs), and random forests. They also go over the benefits and drawbacks of these algorithms while highlighting some potential lines of inquiry for further investigation.

[12] Credit Card Fraud Detection: A Systematic Review and Future Research Directions (2018)

An in-depth analysis of credit card fraud detection methods is presented in this research. The authors discuss different approaches, such as rule-based systems, neural networks, and decision trees, and their uses in detecting credit card fraud. They also point out the shortcomings of these approaches and suggest a few lines of future research, including the usage of blockchain and big data analytics.

[13] A Survey on Credit Card Fraud Detection Techniques (2016)

The methods used to identify credit card fraud are surveyed in this research. The authors examine several techniques, including statistical techniques, machine learning algorithms, and rule-based systems, as well as their applications in detecting credit card fraud. Additionally, they go over the benefits and drawbacks of these strategies and suggest potential future research trajectories, like the combination of various detection methods.

4. PROJECT WORKFLOW

4.1 PROBLEM STATEMENT:

This project aims to develop real-time credit card fraud detection using machine learning and model integration with a web framework.

4.2 TEAMWORK AND ASSIGNMENT

This project is being done under the supervision of Dr. Saritha K, Associate Professor, at Presidency University.

My team members are:

GRISHMA AV	20191IST0182
TEJAS PRAKASH D	20191IST0163
SWETHA A NAIR	20191IST0157
SNEHA M	20191IST0150
SAI MEGHANA J S	20191IST0187

4.3 EXISTING METHOD

Research on a case study involving credit card fraud detection in the current system has shown that by clustering attributes neuronal inputs can be minimized and promising results can be obtained by using normalized data and data should be MLP trained. Prior to cluster analysis and with the results of use of Cluster Analysis and Artificial Neural Networks for detecting fraud, data normalization is used.

Disadvantages:

1. In this work, a new collusive comparison measure is suggested which fairly accounts for the benefits and losses arising from fraud detection.
2. A cost sensitive technique with Bayes' minimal risk has been described as a result of the proposed cost measure.

4.4 PROPOSED METHOD

In this system we put in place a mechanism to detect fraud on credit card transactions. In this approach, most of the essential characteristics that are needed to distinguish legitimate and fraudulent transactions can be achieved. As technological progress progresses, it will be harder to trace the layout and pattern of illegal transactions. This process can now be automated, and some of the heavy work that is required to detect card fraud has been reduced through advances in machine learning, artificial intelligence or other IT related fields. To determine which machine learning algorithm best accomplishes the task and may be utilized by credit card merchants to detect fraudulent transactions, various machine learning algorithms, including Logistic Regression, Decision Trees, Random Forest, and naive Bayes, are compared. We are finally integrating our machine learning model with the web-based framework utilizing streamlit to enhance user interface and experience. For the web framework, we are creating menus, input fields for categorization reports, display model graphs, and menus for forecasts. This system may provide, in most respects, the essential elements required for differentiating genuine and fraud transactions. As technology advances, the concept and pattern of fraudulent transactions becomes more difficult to follow. Automated procedures for detecting counterfeiting of credit cards, as well as reducing the labor intensive work involved in this area, will become possible thanks to advances in artificial intelligence, machine learning and others relevant information technology sectors. To detect fraud in the credit card system. Different machine learning algorithms, such as logistic regression, decision trees, random forests and naive bays, are compared to determine the best method for detecting fraudulent transactions that can be used by credit card merchants. We are integrating a model of Machine Learning with the WebBased Framework in order to improve user interfaces and experiences, using Streamlit.

4.5 METHODOLOGY

• Our goal is to predict the detection of credit card fraud using supervised learning, due to its best methodology. In many types of research, studies and experiments supervised learning in machine learning is the most commonly used approach to predict fraud in financial services. Names, characteristics and values need to be reflected in all data instances within a supervised learning environment. The supervised learning method, in more precise terms, is a classification method based on the use of systematic training data.

• Collection of datasets from the data source, analyze the data and conduct data processing.

• Select the best features that are inputs for better prediction, which can include card number, CVV, expiry date, repeat retailer, used chip, used PIN, and online order.

• Plot graphs for data visualization and analysis using matplotlib or pyplots to better understand the data.

• Train and test the dataset using various machine learning algorithms such as logistic regression, decision trees, and KNN to evaluate their performance.

• Fit the machine learning model, predict with inputs, model evaluation, and find the accuracy.

• Integrate a machine learning model with a web framework like streamlit for better user interface and experience.

• The first step in the proposed work plan is to plan projects. This shall include defining the project's objectives, needs and scope. Stakeholders' roles and schedules as well as the project program and budget shall be decided by the Project Team.

• The next step is to gather and prepare the data. The Project Team will gather data on transactions with credit cards from a range of sources, preprocessing the information and carrying out exploratory analysis atEDA. The team will select the best appropriate characteristics by adopting a feature selection approach.

• Model training is the third phase. Training and testing datasets will be created from the pre-processed data. Using the machine learning methods Logistic Regression, Decision Tree, Naive Bayes, and K-Nearest Neighbors (KNN), the team will train four different models. Python libraries like Scikit-learn or TensorFlow will be used to implement these algorithms. The performance of each algorithm will be assessed using cross-validation, and the top method will be chosen based on accuracy and other criteria.

• Integration of the model is the fourth phase. The Streamlit web framework will be integrated with the chosen model. The team can design an easy-to-use, interactive user interface with

Streamlit enables users to input credit card transaction data and get a fraud detection prediction. The team will also use graphs or charts to visualize the prediction.

• In fifth place is the model's assessment. In order to evaluate the efficiency of an Integrated Model, a number of metrics shall be used such as accuracy, precision, recall and F1 score. Visually, all matrices of false positives, false negatives, true positives, and confusions will be displayed. The model will be improved with the use of these data and several improvement areas need to be identified.

• The 6th stage is to improve the model. To improve model performance, the team will be using techniques such as upsampling or downsampling, ensembles learning and hyperparameter tuning. In order to balance the use of more sampling or less sampling, fraud and nonfraudulent transactions can be spread out in training data. In order to increase model performance, we can combine various machine learning algorithms by combining the work of a number of ensembles. The parameters of the chosen method can be adjusted via hyperparameter tuning to increase accuracy.

• The last phase is the deployment phase. The group will display the completed model, such as a Web server or mobile application, in live mode. They will ensure that data are safe and in compliance with all applicable legislation regarding the protection of personal data.

They shall employ encryption, controls of access and safe communication methods in order to protect data. Additionally, they will abide by data privacy laws and rules like the California Consumer Privacy Act (CCPA) and the General Data Privacy Regulation (GDPR).

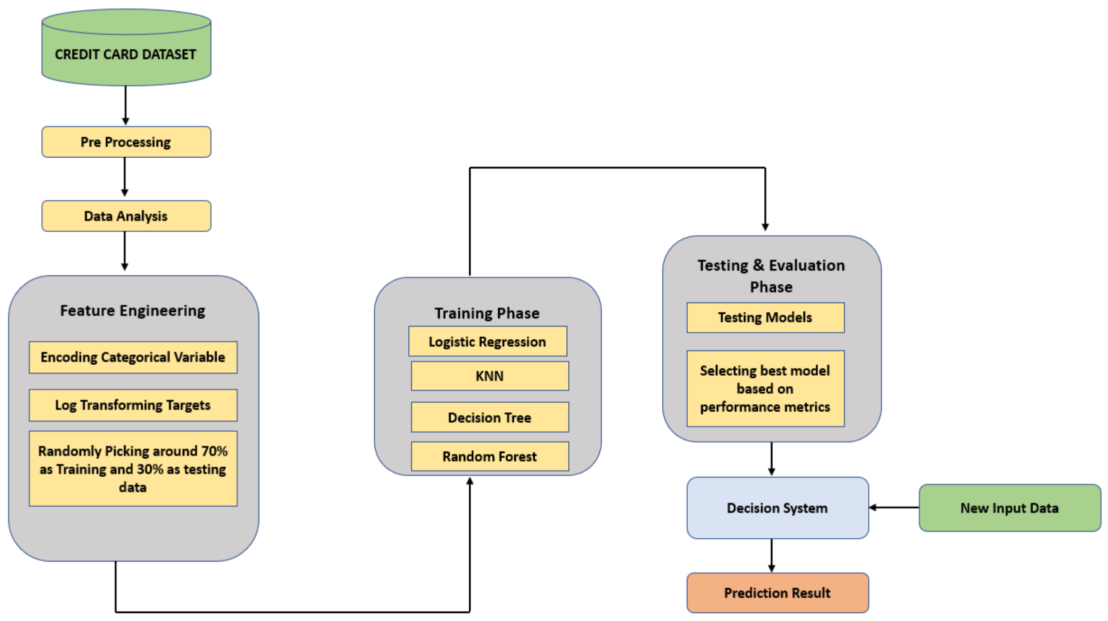


Figure 4.5 System Design and Architecture

4.6 WORKING DOMAIN

4.6.1 HARDWARE USED

1) Processor: Intel i3 or higher

A processor, also referred to as a central processing unit (CPU), is the logic circuitry that reacts to and processes a computer's fundamental instructions. Since the majority of computer commands are interpreted, the CPUIC chip has been viewed as a crucial and key integrated circuit for computers. CPUs handle most basic calculations in logic, ILO and arithmetic. They also issue commands to other processors and parts of the computer that are in use. Although technically speaking, the CPU is not the only processor in a computer, the terms processor and central processing unit (CPU) are often used interchangeably. Some processing is also carried out independently by the hard disc and other components of the computer, with the most common example being the GPUgraphics processing unit.

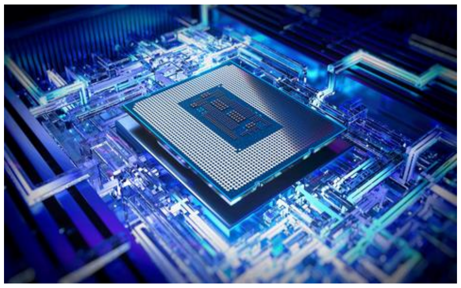


Figure 4.6.1.1 Intel Processor

2) RAM: 4GB or Higher

A piece of hardware on a computer board that serves as the processor's own memory and is commonly called Random Access Memory, or RAM. It allows the processor to store data, programs and programming results when your computer is turned on. It is used to read and write data, serving as a memory device for the computer. The RAM is a persistent memory, which means it does not retain information or commands for an indefinite period. When you turn your computer on, the operating system OSUM and programs will be loaded with RAM often from a Hard Disk or Solid State Drive. Likewise, the files and instructions from a disk are stored in memory while your computer is restarted or when you start programs. The CPU uses this data for the tasks that are required. As soon as the computer is shut down, the RAM will lose its data. The information is therefore lost while the machine is idle, but retained in

RAM as it continues to run. Reading data from RAM is faster than reading data from the hard drive, which is a benefit of loading data into RAM.

Figure 4.6.1.2 Random Access Memory

3) STORAGE: 1 TB or Higher

4) GPU enabled

A graphics card (GPU) can be used by Media Server to carry out some processing operations. Convolutional neural network training and analysis operations can be carried out much more quickly when a GPU is used instead of a CPU. A GPU is useful for the following tasks: Image categorization.

Figure 4.6.1.4 GPU

4.6.2 SOFTWARE USED

- Operating System: Windows 7/8/10 or Ubuntu
- Tools: Python, Anaconda Navigator, sci-kit libraries, PyCharm, Streamlit

1) **Python:**

Object-oriented, interactive, general-purpose, and high-level programming languages like Python are particularly well-liked. The programming language Python uses dynamic typing and garbage collection. The procedural, object-oriented, and functional programming languages are supported by Python. With the use of heavy indentation, the Python design philosophy prioritizes code readability.

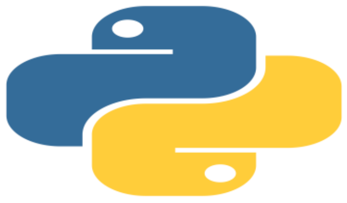


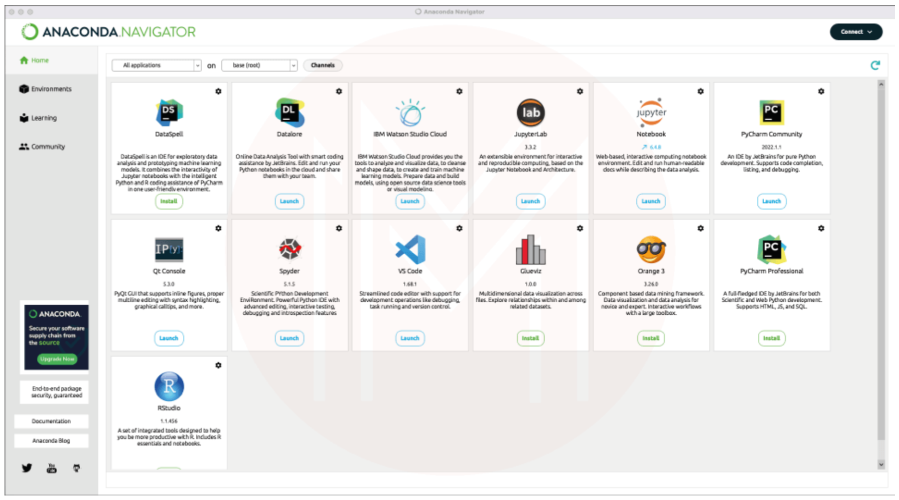
Figure 4.6.2.1 Python

2) **Anaconda Navigator:**

Using Anaconda Navigator, the desktop Graphical User Interface that is part of ananiapanthoric Distribution, you're free to perform applications and control conda package, environment or channel operations with no use of a command line interface CLI. On Anaconda.org or at the nearby archive, packages are available by Navigator. It's possible to use it with Windows, Mac and Linux.

Without using command line tool you can quickly and easily configure conda packages, environments or channels with Navigator by launching particular Python programs. Also, an Anaconda cloud and a local archive can be searched by Navigator. Anaconda Navigator provides the following by default:

- Jupyter Lab
- Jupyter Notebook
- Spyder
- Pycharm
- VSCode
- Glueviz
- Orange 3 App



DataSpell

DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.

Datalore

Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

JupyterLab

3.3.2

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

Notebook

↗ 6.4.8

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

PyCharm Community

2022.1.1

An IDE by JetBrains for pure Python development. Supports code completion, listing, and debugging.

Qt Console

5.3.0

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

Spyder

5.1.5

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

VS Code

1.68.1

Streamlined code editor with support for development operations like debugging, task running and version control.

Glueviz

1.0.0

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

Orange 3

3.26.0

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

PyCharm Professional

A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.

RStudio

1.1.456

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

Figure 4.6.2.2 Anaconda Navigator

3) Sci-Kit:

The most effective and reliable Python machine-learning library is called Skearn (Skit-Learn). It provides a wide range of efficient statistical modelling and machine learning techniques such as the classification, regression, clustering or dimensionality reduction via its Python Consistency interface. These are the two simplest ways that a Scikitlearn library, which is written mostly in Python when you've set up Numpy or SciPy, can be installed. You can install ScikitLearn with the pip command install pip.U. ScikitLearn is based on Matlib, NumPlotpy, and SciPy.

Features

The scikit-learn library is concentrated on modeling the data, as opposed to loading, modifying, and summarizing the data. The following are some of the most well-liked model categories offered by Sklearn:

Algorithms for Supervised Learning Scikit-learn includes almost all of the well-known supervised learning algorithms, including Linear Regression, Decision Tree, etc. It will also include some of the most commonly used unsupervised learning algorithms, e.g. clustering and component analysis, PCA Principal Components Analysis or Unsupervised Neural Networks. Using this clustering model, unlabeled data are grouped.

The technique of cross validation is used to verify supervised models that do not have any observed data. The method of shrinking the number of attributes in a data file so that it is possible to use them for feature selection, visualization and summary purposes is called dimension reduction. The methods used for combining the predictions from various supervised models are ensembles, as they appear to be called.

The process of extracting features from data is used to define properties in picture and text data.

To develop supervised models, useful qualities are found through feature selection.

It is an open-source library that is also permissible for commercial use under the BSD license.

Figure 4.6.2.3 Sci-Kit

4) **PyCharm:**

An integrated development environment (IDE) for Python programming is called PyCharm. It provides code analysis, graphics debugging, unit testing with an integrated version control system and support for Django Web Development. PyCharm was the creation of a Czech company called Jet Brains. It supports Linux, Mac OS X and Microsoft Windows on multiple platforms. The Apache licence allows you to get both the Community and Professional versions of PyCharm. PyCharm's Community Edition is not as heavily illustrated as the Professional Edition of PyCharm.

Features:

Code completion, syntax and error highlighting, linter integration, and rapid fixes are all part of the coding aid and analysis services. Navigation of projects and code: specialized project views, file structure views, and rapid switching between files, classes, methods, and usages.

Refactoring Python code, such as renaming, extracting methods, introducing variables, introducing constants, pulling up, pushing down, and others

Web frameworks supported include Django, web2py, and Flask built-in Python debugger

Unit testing that is integrated and has line-by-line coverage programming in Python for Google App Engine

Mercurial, Git, Subversion, Perforce, and CVS all have a single user interface that combines change lists and merge functionality for version control.

Integration of scientific tools: Supports Anaconda as well as many scientific packages, such as Matplotlib and NumPy, and integrates with I Python Notebook. It also offers an interactive Python terminal.

Figure 4.6.2.4 Pycharm

5) **Streamlit:**

The creation of web applications for machine learning and data science is based on the Python framework Streamlit, which is freely available. We can create and launch web applications in a matter of seconds with Streamlit. Streamlit is an app creation tool that can be used the same way as making Python code. Streamlit makes it easy to work with the interactive cycle of coding and watching results in the web app. Python can be used by programmers to build interactive data driven apps, using the Streamlit open source web application framework. Streamlit makes it easy for developers to create innovative dashboards, machine learning models and data visualizations which can be incorporated into online applications. Streamlit provides developers with a simple, user friendly interface to generate applications that enables them to concentrate on content and functionality rather than technical details of Web development. Building online apps is made simpler by a variety of features offered by Streamlit, including an easy-to-use and clear API for designing user interfaces and data visualizations. Automatic reactivity is the ability of programmers to create interactive programs which change rapidly in response to user input. Support is provided for the well-known data science libraries such as Pandas, Matplotlib and Plotly. You can easily deploy to several cloud services, such as Heroku and Google Cloud. Generally speaking, Streamlit is an excellent Python tool to build interactive applications based on data and it's a good option for scientists and developers that are required to rapidly develop and launch web applications. Data science, machine learning, and other fields can all benefit from the use of Streamlit, a flexible online application framework. Here are some instances of how to utilize Streamlit:

1. interactive data exploration: Streamlit makes it easy for users to explore and analyze the data in a more natural, interesting way by enabling them to create interactive data visualizations and search tools.
2. 2 Streamlit can be used to create and deploy machine learning models as Web Applications, so that users have the option of interacting with these models or testing them at any time.
3. One of the strengths of Streamlit is its ability to create interactive dashboards that allows users to see and analyze data from a variety of sources.
4. Prototyping and experimentation: Streamlit's user friendly interface makes it easy for users to explore new data science concepts and methods, enabling rapid testing and refining of ideas.
5. Education and training: Students and learners can explore and study data science concepts in a more hands-on and engaging way by using interactive educational tools and tutorials made with Streamlit.

Overall, Streamlit is a flexible tool with a wide range of uses in data science, machine learning, and other fields.

Setting up Streamlit

Ensure that Python is installed on your system. Install streamlit and pip using the command line. put in streamlight

Development flow

The top-right corner of the app indicates whether to relaunch the application or not if the source code of the streamlit Python script changes. To always rerun when the source script changes, you may also choose the "Always rerun" option.

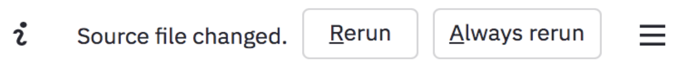


Figure 4.6.2.5 Development Flow

Data flow

You can use Streamlit to create apps in the same manner that you create Python code. The data flow of the streamlit is separate; whenever your code is changed or something on the screen needs to be updated, the streamlit replays your Python script completely from top to bottom. When a user engages with widgets like a choose box or drop-down menu, this occurs.

Displaying the data

You can show many types of data using Streamlit's several methods, including arrays, tables, and data frames.
Use st.write("Your string") to write a string.
Utilize the st. data frame method to display a data frame.

6) NumPy:

A Python library for the computation and processing of multidimensional and linear array elements is called Numeric Python, sometimes called NumPy. By incorporating the functionality of the progenitor module Numeric into another module called Numarray, Travis Oliphant created the NumPy package in 2005. This is primarily a Python extension module, and it was developed on C. It's got a whole bunch of programs that are quick to do calculations involving numbers. The NumPy database contains a wide range of robust data structures, such as Multidimensional arrays and matrices. These data structures provide the best calculation of arrays and matrices. As a result of the data science revolution, libraries with data analysis functions like Numpy, SciPy, Panda etc. have been expanded to an enormous extent. Python is the programming language of choice for Data Scientists because it has a syntax which is very easy to learn and understand, in comparison with more common languages. The NumPy tool provides an easy and efficient approach for managing the huge amount of data. Furthermore, NumPy enables it to be incredibly simple to multiply numbers and manipulate data. It's sensible to work with a large number of data, because NumPy is very fast. The use of NumPy for data analysis has the following benefits:

1. NumPy conducts computing with arrays.

2. It implements multidimensional arrays effectively.

3. It carries out calculations in science.

4. It can apply the Fourier Transform and reformat the information kept in multidimensional arrays.

5. NumPy has built-in functions for random number creation and linear algebra.

Today, NumPy, SciPy, and Mat-plotlib are utilized in place of MATLAB since Python is a more complete and user-friendly programming language than MATLAB.

Figure 4.6.2.6 NumPy

7) **Pandas:**

An open source library with a focus on speed and logical operation in relation to database or tag data. In order to work with time series and numerical data, it provides a variety of data structures and procedures. The foundation for this library is the NumPy library. Pandas is fast and offers a high degree of performance and efficiency to its users. **Advantages:** You can load the data from several file types, manage and analyse it in a fast and efficient way. mutability: columns can be added and removed from the data frame and higher dimensional objects; simple handling of missing data expressed as NaN in floating point and nonfloating point data joining and combining data sets. The flexible set of data can be switched and reshaped. Offers time-series features.

5. Data Collection

Obtaining reliable data is essential so that your machine-learning model can identify the appropriate trends. The caliber of the data you give the machine affects how accurate your model is. You will obtain inaccurate results or irrelevant forecasts if your data is flawed or outdated. We used the Kaggle dataset in this instance.

6. DESCRIPTION OF THE MODULES

6. 1 Implementation

With the use of machine learning, we have been able to define how these data should be differentiated. As it may not seem spectacular, but the actions that have been adopted will define this model. This criterion may change in the future as machine learning and AI in general progress, however, the steps for running the code and working are as follows:

- Gathering data
- Preparing that data
- Choosing a model
- Training
- Evaluation
- Hyperparameter tuning
- Prediction

6.2 Gathering the Data

It is important to obtain reliable data that can be used by your Machine Learning model to predict the relevant trends. The accuracy of your model depends on how much information you've given to the machine. If the data is incorrect or out of date, you'll get a bad result or an irrelevant forecast**.**

6.3 Preparing the data

Following the collection of the data, the information must be combined and randomly generated. This makes sure that the distribution of the data is uniform and that the learning process is unaffected by the ordering.

• To clean data in order to remove any unnecessary information, e.g. duplicates, NULL values, rows and columns. You may need to change the row, column or index of rows and columns in order to reorganize your dataset. Visualize the data so that it can be understood as a structure and interconnection of its numerous variables and classes.

• Making two sets of cleaned data into a training set and a test set. A training set is an object from which your model learns. When assessing the accuracy of your model after training, a test set shall be used.

6**.4 Choosing the model**

The result will be selected by the machine learning model after it applies an algorithm to that obtained data. It is important to select a model that will work for the task at hand. In the various tasks, such as speech recognition, image recognition, prediction and so on, a lot of models have been developed by scientists and engineers. Besides that, the question whether your model is viable for linear or quantitative data needs to be decided and then a decision must be made.

6.5 Training the model

The crucial part of machine learning is the training phase. During the training process you feed your own prepared data into a Machine Learning model to figure out patterns and predict them. This results in a model gaining an insight into the data, and being able to perform the task at hand. With use of this model, it becomes more accurate to predict the future.

6.6 Evaluation

You need to monitor the model's performance after training. This will be achieved, when the model is tested against an empty set of data. The unseen information is the set of tests that you previously split our data into. The use of the same training data used for testing will result in a measurement error because the model is familiar with these data and has been able to identify patterns. That'll lead to an overly high degree of accuracy. When you apply your model to testing data, we get a precise indication of how it will behave and how quickly.

6.7 Tuning

You need to consider whether there is any way you can improve the accuracy of your model once it has been built and tested. This can be achieved when a model's parameters are fine tuned. The variables of the model to which the programmer has chosen are parameters. Accuracy will be the highest when your parameter is given a specific value. The process of adjusting the parameters is to locate these settings.

6.8 Prediction

Finally, you may use your model to make precise predictions based on unknown data.

7. ALGORITHMS:

7.1 LOGISTIC REGRESSION

For classification issues, supervised learning algorithms like logistic regression are frequently utilized. A dataset with one or more independent factors that affect the outcome can be analyzed statistically using this technique. In logistic regression, the objective is to identify the model that best captures the connection between the independent variables (or predictors) and the dependent variable (or response). In contrast to linear regression, logistic regression uses a binary or categorical response variable.

Advantages of Logistic Regression:

• Simplicity: The algorithm of logistic regression is straightforward to comprehend. It is a well-liked solution for many categorization issues due to its simplicity in application and interpretation.

• Speed: Logistic Regression is a quick algorithm that effectively handles enormous datasets.

• Scalability: Logistic Regression is a flexible algorithm that can handle both small and large datasets.

• Interpretable: Each feature's coefficient from logistic regression analysis can be used to assess how that feature affected the prediction.

• Regularization: To avoid overfitting in Logistic Regression, regularization techniques like L1 and L2 can be applied.

Disadvantages of Logistic Regression:

Logistic regression requires a linear relationship between the independent variables and the response variable's log-odds, hence it is limited to linear decision limits. The performance of Logistic Regression may be poor if the relationship is non-linear.

Logistic regression is susceptible to outliers and can be impacted by them.

• Inability to handle non-linear relationships: The independent variables and the response variable cannot be related in a linear fashion using the logistic regression method.

Logistic regression assumes that the independent variables do not have a strong correlation with one another. The performance of the algorithm may be impacted by multi-collinearity if the independent variables are highly linked.

• Logistic regression is only capable of handling binary classification issues; it cannot tackle multi-class issues without modification.

A statistical model, known as logistic regression, shall be applied to a binary dependent variable and one or more independent variables. The logistic function, based on the logistic regression model, turns a linear combination of independent variables into probabilities between 0 and 1. The logistic regression equation is p = 1 / (1 + e(-z)) where p is the expected likelihood that the dependent variable will have the value 1, e is a mathematical constant that is roughly equal to 2.71828, and z is the linear combination of independent variables: z = b0 + b1x1 + b2x2 +... + bpxp. where the coefficients of the independent variables x1, x2,..., and xp are, respectively, b1, b2,..., and bp, and b0 is the intercept or constant term. The Logistic regression model generates value for the coefficients, which maximise the likelihood of observed data on the basis of a method called Maximum Possibility Estimation. In view of independent variable values, the model can be applied to predict how likely a depended variable will take an value between 1 and 0. When all other variables are held constant, the coefficients can also be seen as measurements of the strength and direction of the association between each independent variable and the dependent variable.

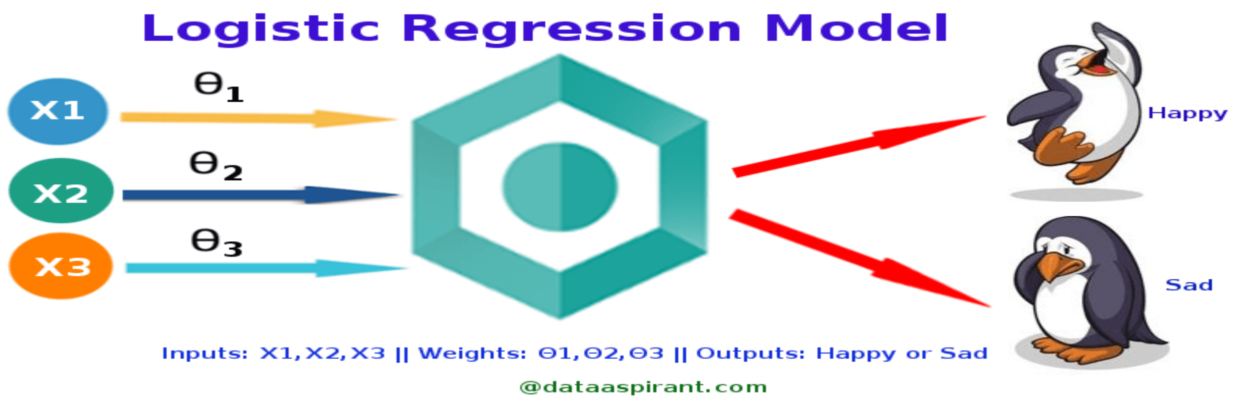

Logistic Regression Model
X1
X2
X3
θ1
θ2
θ3
Happy
Sad
Inputs: X1,X2,X3 || Weights: θ1,θ2,θ3 || Outputs: Happy or Sad
@dataaspirant.com

Figure 7.1 Logistic Regression Model

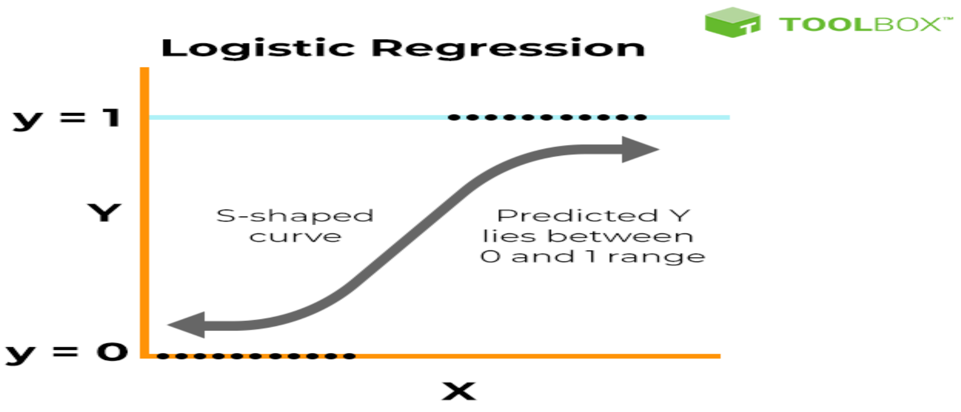


Figure 7.1 Logistic Regression

7.2 NAIVE BAYES

Naive Based on the probability theory of Bayes, Bayes is a well-known classification algorithm. It is a popular method that is used in spam filtering, text classification, and recommendation systems because it is straightforward but effective. The name "naive" comes from the assumption that the features are independent of one another. The process starts by figuring out how likely each class is given a collection of features. Using the Bayes theorem, which states that the likelihood of a hypothesis—in this case, the class—given the data (the features) is proportional to the probability of the data—given the hypothesis—multiplied by the prior probability of the hypothesis—is used to accomplish this.

Advantages of Naive Bayes:

• Ease of use: Naive Bayes is a straightforward algorithm that is simple to comprehend and use. Unlike many other machine learning methods, it does not call for intricate iterative techniques.

• Quick: The Naive Bayes algorithm is quick and can handle big datasets with many dimensions.

• Functions well with tiny datasets: Naive Bayes can produce reliable predictions with only a modest quantity of training data.

• Handles irrelevant features: Naive Bayes can deal with and is unaffected by irrelevant features.

Disadvantages of Naive Bayes:

• Assumes feature independence: Naive Bayes assumes that the features are independent of one another, which may not always be the case with datasets from the real world.

• Limited expressive power: Naive Bayes is unable to capture complicated correlations between features due to its limited expressive power.

Naive Bayes presupposes a particular probability distribution for the features, which may not necessarily be the best option for the dataset. This makes it sensitive to the distribution choice.

• Unable to handle continuous data: Because Naive Bayes assumes a discrete probability distribution for the features, it is ineffective for dealing with continuous data.

• A bias in favor of categorical features: Naive Bayes is less effective with continuous or numerical features and performs better with categorical features.

A technique for machine learning called Naive Bayes is utilized for categorization jobs. Its foundation is the Bayes theorem, a probabilistic method that determines the likelihood of a hypothesis given the available information. The construction of the probability distribution is made easier by Naive Bayes' assumption that the features are conditionally independent given the class label. The following is a representation of the mathematical model for Naive Bayes: Assume that Y is the goal variable and that X = (x1, x2,..., xn) is a collection of input features. Our goal is to learn a function that translates inputs to outputs, f(X) = Y. According to the naive Bayes model, the probability distribution of the features given the class label can be factored as P(X | Y) = P(x1 | Y) * P(x2 | Y) *... * P(xn | Y), where P(x | Y) is the conditional probability of the feature x given the class label Y. Using Bayes' theorem, the Naive Bayes classifier predicts the class label Y for a fresh input vector X = (x1, x2,..., xn): P(X | Y) * P(Y) / P(X) where P(Y) is the marginal probability of the input features and P(X) is the prior probability of the class label Y. The class label with the highest posterior probability is selected using the Naive Bayes classifier: Argmax = Y_hat The equation P(Y | X) is equal to argmax P(X | Y) * P(Y), where argmax is the Y value that maximizes the expression. Naive Bayes employs a maximum likelihood method that counts the occurrences of each feature value for each class label in the training data to estimate the conditional probabilities P(x | Y). To prevent zero probabilities for unobserved feature values, Laplace smoothing is frequently utilized. Naive Bayes is renowned for being straightforward, quick, and capable of handling large amounts of data. It is frequently employed in sentiment analysis, spam filtering, and text classification.

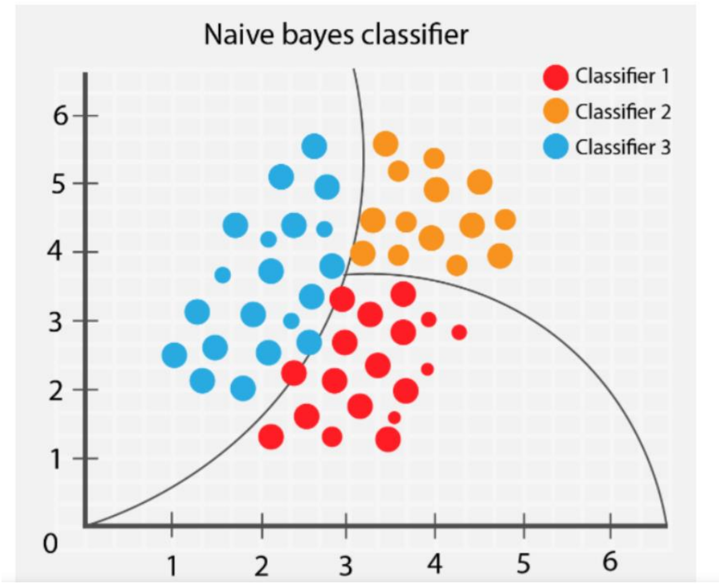


Figure 7.2 Naïve Bayes Diagram

7.3 KNN

The k-nearest neighbor's algorithm (k-NN) is a non-parametric supervised learning technique in statistics. The result of the k-NN classification is a class membership. An object is allocated to the class that has the most support from its k closest neighbors (k is a positive integer that is often small) based on a majority vote of those neighbors. The object is simply put into the class of its one nearest neighbor if k = 1. With k-NN, all computation is postponed until after the function has been evaluated and the function is only locally approximated. Since this technique relies on distance for classification, normalizing the training data can significantly increase accuracy if the features reflect several physical units or have distinct sizes. Assigning weights to neighbor contributions can be a helpful strategy for both classification and regression, making the closer neighbors contribute more to the average than the farther neighbors. As an illustration, a typical weighting method assigns each neighbor a weight of 1/d, where d is the distance between the neighbors. When using k-NN classification or regression, the neighbors are chosen from a set of objects for which the class or object property value is known. Although there is no need for an explicit training step, this can be considered as the algorithm's training set.

The non-parametric classification algorithm K-Nearest Neighbors (KNN) bases its predictions on the feature space's k-nearest neighbors' dominant class. You may sum up the mathematical model for KNN as follows: The KNN algorithm searches the training dataset to discover the k-nearest neighbors of x_q based on a distance metric (e.g., Euclidean distance), given a training dataset X with features x_1, x_2,..., x_n and labels y_1, y_2,..., y_n and a test instance x_q. The test instance x_q is then given the label of the majority class among the k-nearest neighbors by the procedure. The algorithm labels x_q with the label of the nearest training case if k = 1. Let D be the training dataset with N instances, where each instance is represented by a feature vector (x_i) and a label (y_i) that correspond to it. Let q be a test instance that has to be categorized. The steps of the KNN algorithm are as follows: Use some distance metric d(x_i, q) to calculate the distance between each training instance x_i and q. Choose the k training examples that are closest to q in terms of distance. Give q the designation of the predominant class among its k closest neighbors. Give the label of the nearby training instance If k = 1. KNN is a straightforward yet effective classification method that can be quickly developed and used for a variety of classification problems. However, the algorithm's performance can be significantly impacted by the choice of the distance metric and the value of k. To attain the optimum performance, it is crucial to properly modify these hyperparameters.

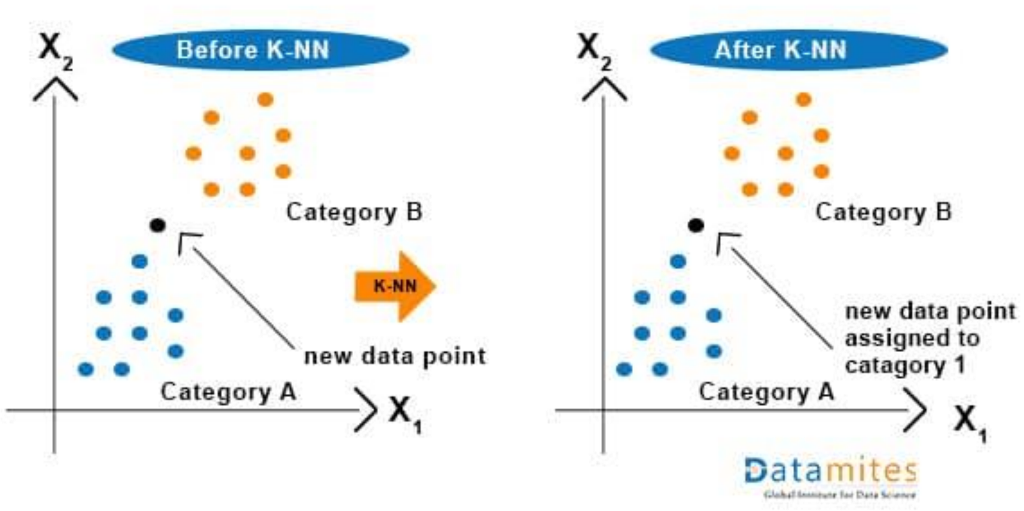


Figure 7.3 KNN

7.4 DECISION TREE

Decision trees are a technique for categorization and prediction that is highly effective and well known. A decision tree is a type of tree structure that resembles a flowchart, where each internal node represents a test on an attribute, each branch a test result, and each leaf node (terminal node) a class label. Building a Decision Tree: The tree can be "learned" when the source set is divided into groups on the basis of an attribute value test. To perform the same operation on each derived subset, it is known as recursive partitioning. Recursion is completed when the split no longer improves prediction performance, or if a subset in one node has the same value as its target variable. For exploratory knowledge discovery, the building of a decision tree classifier is an excellent option without requiring parameter configuration or schema understanding. It is possible to make use of decision trees for multidimensional data. Classifiers are usually accurate at the decision tree level. Decision tree induction is one of the most popular inductive methods for learning classification information.

Decision Tree Representation: Decision trees categorize instances by arranging them in a tree from the root to a leaf node, which gives the instance's categorization. Starting at the root node of the tree, checking the attribute indicated by this node, and then going down the tree branch according to the value of the attribute are the steps involved in classifying an instance, as shown in the above diagram. The subtree rooted at the new node is then given the same treatment as before.

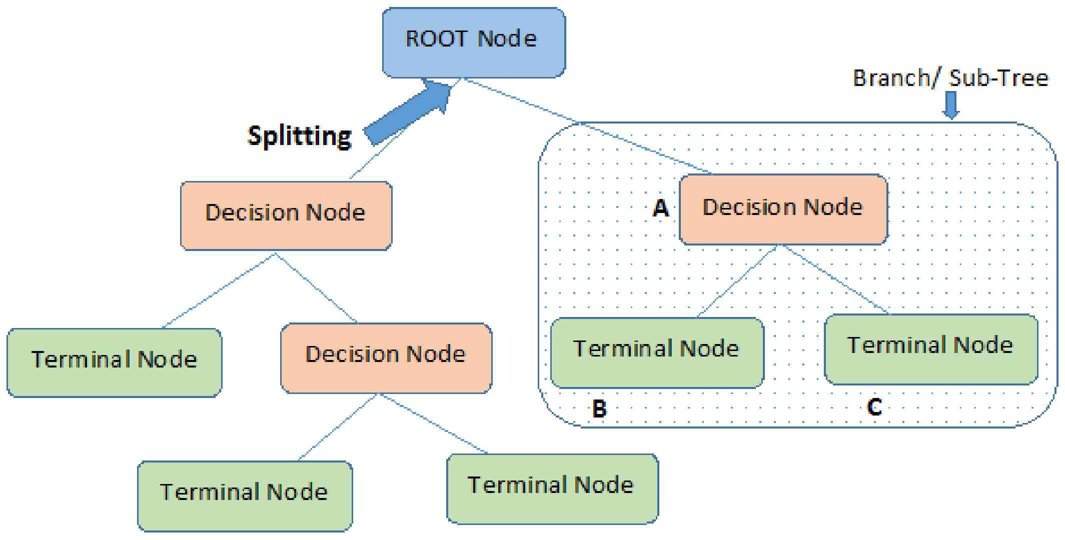


Figure 7.4 Decision Tree

7.5 RANDOM FOREST

The algorithm for machine learning called Random Forest, mixed by the output of a number of decisions trees to produce just one result, was developed by Leo Breiman and Adele Cutler. Its adaptability and ease of use, given that it is capable of solving classification and regression problems, are the main reasons why this tool is widely used. The bagging method is extended by the random forest algorithm, which uses feature randomness in addition to bagging to produce an uncorrelated forest of decision trees. In order to guarantee a low correlation between decision trees, a random subset of features can be generated by feature randomness, known as "feature bagging" or "a random subspace method"PDF, 121 KBs. This is the most important distinction between decision trees and random forests. Random forests merely choose a portion of those feature splits, whereas decision trees take into account all possible feature splits.

If we return to the "Should I surf?" example, the questions I would pose to arrive at the forecast might not be as thorough as those posed by someone else. By taking into consideration every possible variation in the data, we can lessen the chance of overfitting, bias, and total variance, leading to more accurate predictions.

Key Benefits

Less chance of overfitting: Decision trees are more apt to fit all samples contained in the training data, which leads to a greater risk of over fitting. However, when there are many decision trees in a random forest, the classification algorithm will not oversimplify their model due to the reduction of total variance and prediction error by averaging uncorrelated trees.

Flexibility: In the data scientist community, a random forest approach is preferred because it can handle both classification and regression tasks with accuracy. The random forest classifier benefits from feature bagging by maintaining accuracy even when some of the data is missing, which makes it a useful tool for guessing missing values.

Quick assessment of the feature contribution: Random forest makes it easy to assess variability in contributions. The significance of a feature can be determined by various methods. To gauge how much the model's accuracy declines when a particular variable is removed, the gini importance and mean a drop in impurity (MDI) are frequently utilized.

A different importance metric is permutation importance, often known as mean decrease accuracy (MDA). By allowing the feature values in oob samples at random, MDA can determine the average decline in accuracy.

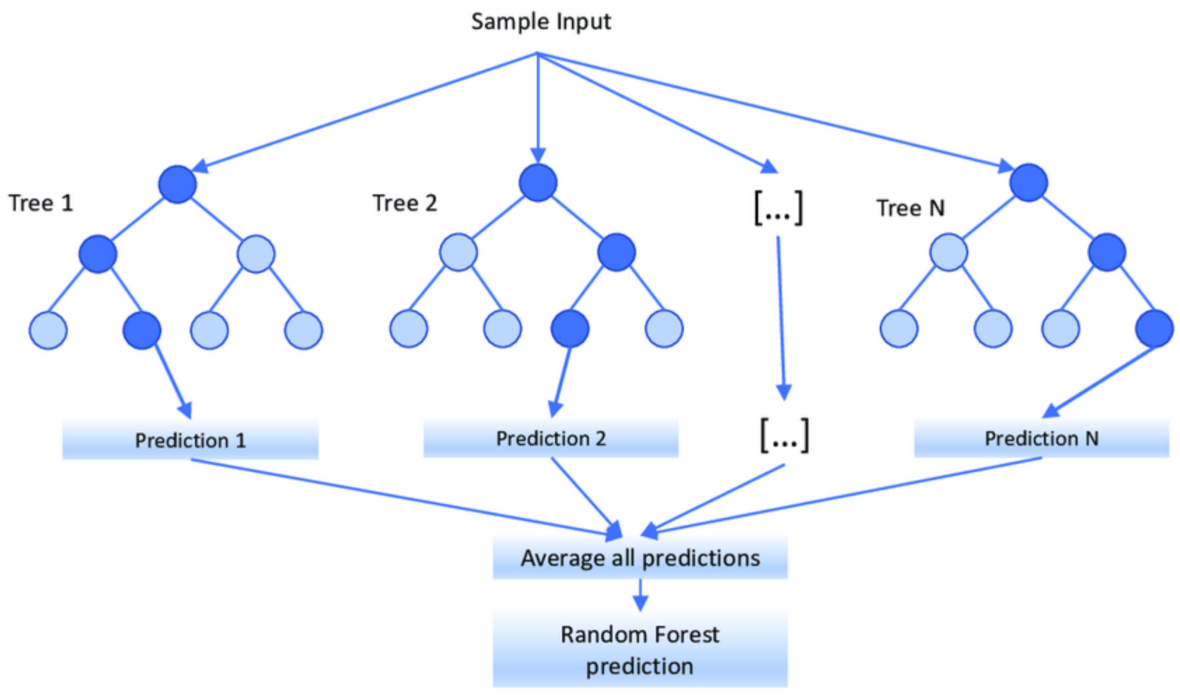


Figure 7.5 Random Forest

8. TECHNIQUES

8.1 Repeat retailer:

Credit card fraud detection using repeat retailers is a technique that utilizes the history of transactions made at a particular retailer to identify potentially fraudulent transactions. The basic idea is that if a cardholder has made several legitimate transactions at a particular retailer in the past, then any future transactions at that retailer are more likely to be legitimate as well. The system maintains a history of transactions made by each cardholder at each retailer. When a new transaction is made, the system checks to see if the cardholder has made any previous transactions at the same retailer. If the cardholder has made previous transactions at the retailer, the system calculates various metrics, such as the average transaction amount, the time between transactions, and the location of the transactions. The system compares the metrics of the new transaction to the historical metrics of the cardholder's previous transactions at the retailer. If the metrics of the new transaction are significantly different from the historical metrics, the system flags the transaction as potentially fraudulent and triggers a review process. Repeat retailer is just one of many techniques used in credit card fraud detection, and is often used in combination with other techniques, such as anomaly detection and machine learning. By leveraging the history of transactions made by each cardholder, repeat retailers can help identify potentially fraudulent transactions and reduce the incidence of credit card fraud. Through the graph model, we are depicting the analysis part based on the dataset, a column of 'repeat retailer'. Predicting the percent of 'yes' is 88.2% and 'no' is 11.8%.

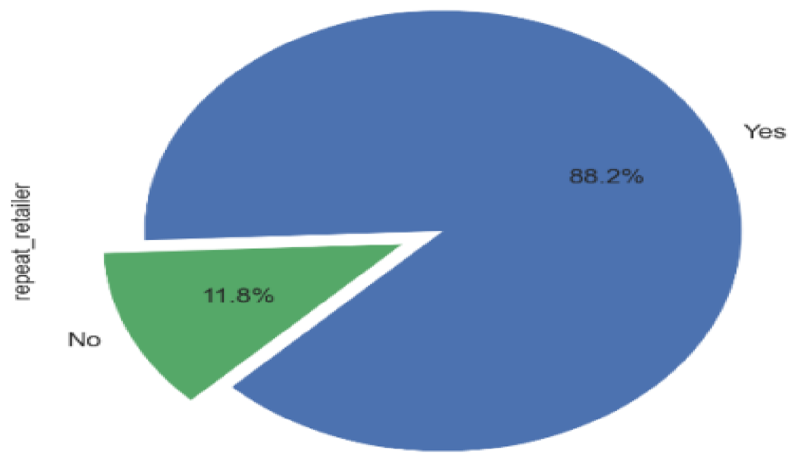


Figure 8.1 Pie Chart of Repeat Retailer

8.2 Used_chip:

Credit card fraud detection using used_chip is a technique that utilizes the information stored on the chip of a credit card to identify potentially fraudulent transactions. The basic idea is that the information stored on the chip can provide additional authentication and validation that can help verify the legitimacy of a transaction. The system reads the information stored on the chip of the credit card, including the card number, expiration date, and other information. The system compares this information to the information provided by the merchant, such as the transaction amount, the merchant's name, and the location of the transaction. If the information provided by the merchant matches the information stored on the chip, the system assumes that the transaction is legitimate and approves the transaction. If the information provided by the merchant does not match the information stored on the chip, the system flags the transaction as potentially fraudulent and triggers a review process. Used_chip is just one of many techniques used in credit card fraud detection and is often used in combination with other techniques, such as repeat retailer analysis and machine learning. By utilizing the information stored on the chip of a credit card, used_chip can help verify the legitimacy of a transaction and reduce the incidence of credit card fraud. Through the graph model, we are depicting the analysis part based on the dataset, a column of 'used chip'. Predicting the percent of 'yes' is 65.0% and 'no' is 35.0%.

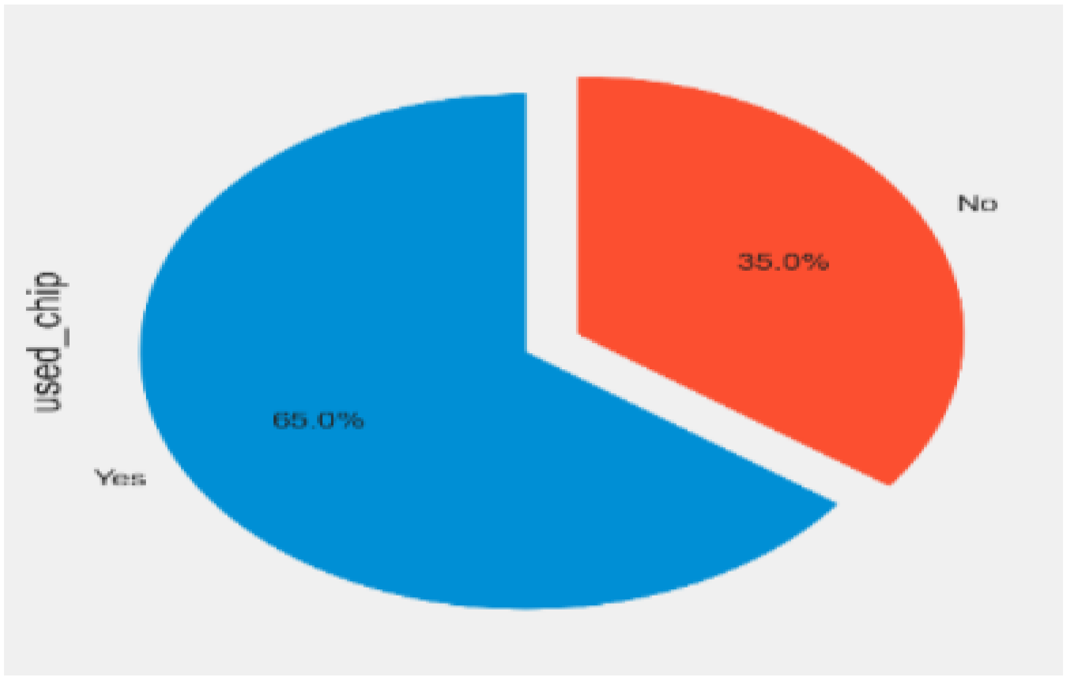


Figure 8.2 Pie Chart of Used_chip

8.3 Used_pin_number:

Credit card fraud detection using used_pin_number is a technique that utilizes the personal identification number (PIN) entered by the cardholder during a transaction to identify potentially fraudulent transactions. The basic idea is that if a transaction is made using a cardholder's stolen credit card, the thief is unlikely to know the correct PIN, and this can be used to identify potentially fraudulent transactions. The cardholder enters their PIN during the transaction. The system compares the entered PIN to the PIN stored on the credit card's chip. If the entered PIN matches the stored PIN, the system assumes that the transaction is legitimate and approves the transaction. If the entered PIN does not match the stored PIN, the system flags the transaction as potentially fraudulent and triggers a review process. Used_pin_number is just one of many techniques used in credit card fraud detection and is often used in combination with other techniques, such as anomaly detection and machine learning. By utilizing the PIN entered by the cardholder, used_pin_number can help verify the legitimacy of a transaction and reduce the incidence of credit card fraud. However, it is important to note that this technique is not foolproof and can be compromised in cases where the PIN has been stolen or the thief has managed to guess the correct PIN. Through the graph model, we are depicting the analysis part based on the dataset, a column of 'used_pin_number'. Predicting the percent of 'yes' is 89.9% and 'no' is 10.1%.

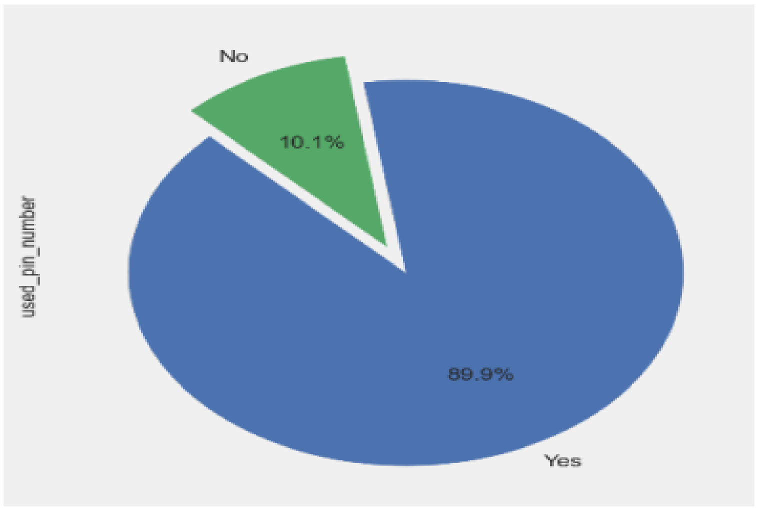


Figure 8.3 Pie Chart of used_pin_number

8.4 Online_order:

Credit card fraud detection using online_order is a technique that utilizes the information provided during an online order transaction to identify potentially fraudulent transactions. The basic idea is that certain patterns and behaviors can be used to identify potentially fraudulent transactions made online. The system analyses the information provided during the online order transaction, including the IP address of the device used to make the transaction, the shipping address, and the billing address. The system compares this information to the cardholder's historical information, such as their location, typical shipping and billing addresses, and other relevant information. The system looks for patterns and anomalies in the information provided, such as a shipping address that is significantly different from the cardholder's billing address or an IP address that is located in a different country. If the system detects any suspicious patterns or anomalies, it flags the transaction as potentially fraudulent and triggers a review process. Online_order is just one of many techniques used in credit card fraud detection and is often used in combination with other techniques, such as machine learning and anomaly detection. By analyzing the information provided during an online order transaction, online_order can help identify potentially fraudulent transactions and reduce the incidence of credit card fraud. However, it is important to note that this technique is not foolproof and can be compromised in cases where the thief has access to the cardholder's personal information, such as their shipping and billing addresses. Through the graph model, we are depicting the analysis part based on the dataset, a column of 'online order'. Predicting the percent of 'yes' is 65.1% and 'no' is 34.9%.

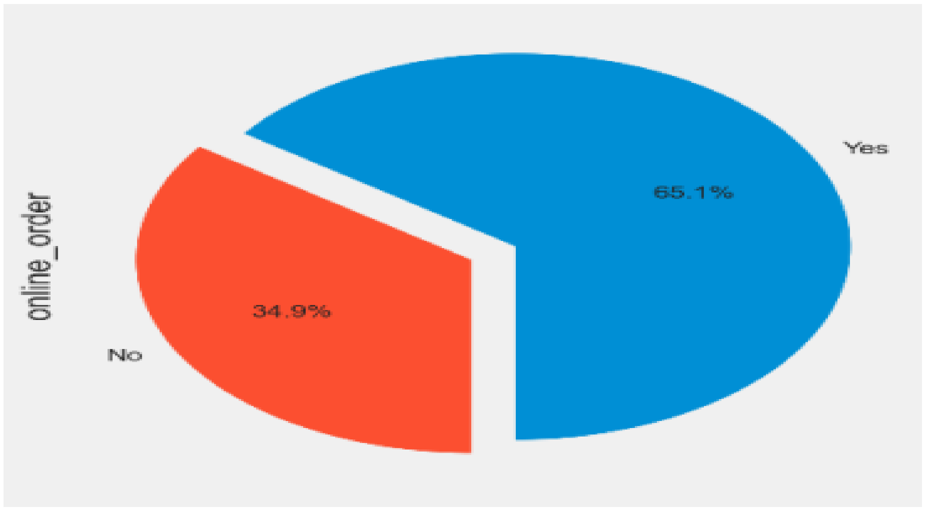


Figure 8.4 Pie Chart of Online Order

9. System Design

9.1 Use case diagram

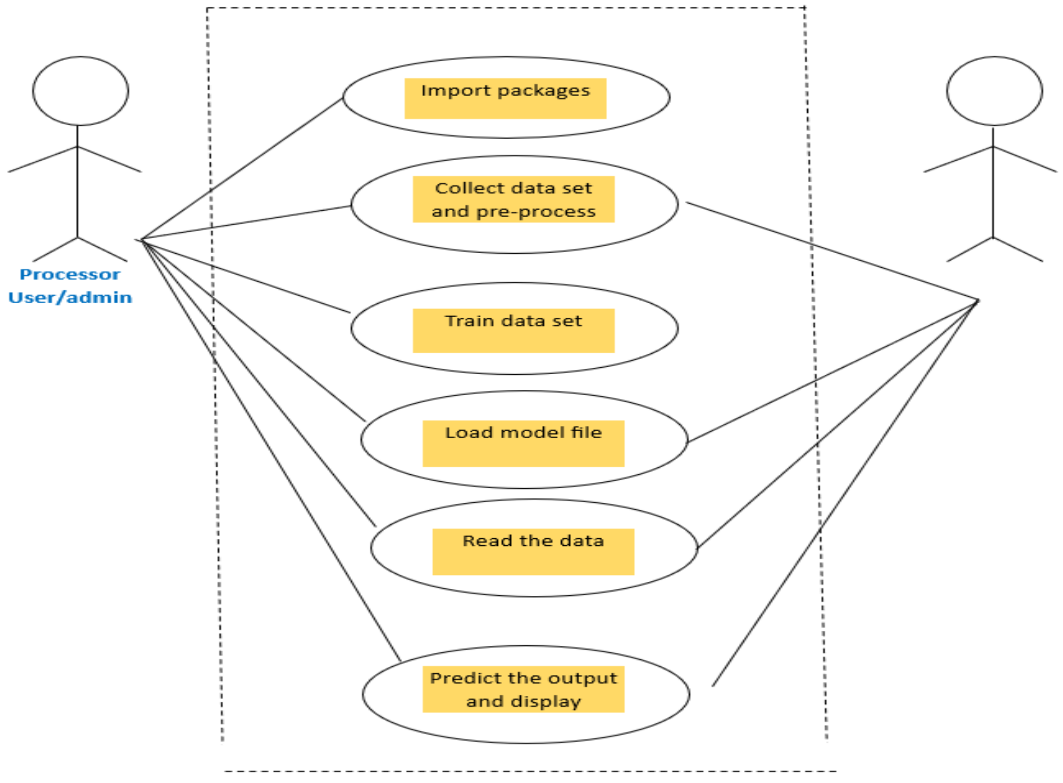


Figure: Use case Diagram

Figure 9.1 Use Case Diagram

The user and processor are the two components of a use case, with the user providing input to the system and the processor processing the data and producing output. The diagram up above depicts the flow. The system imports and loads the code, model, and library packages for the initial user. After the program has run, the output is shown following the data input.

9.2 Sequence Diagram

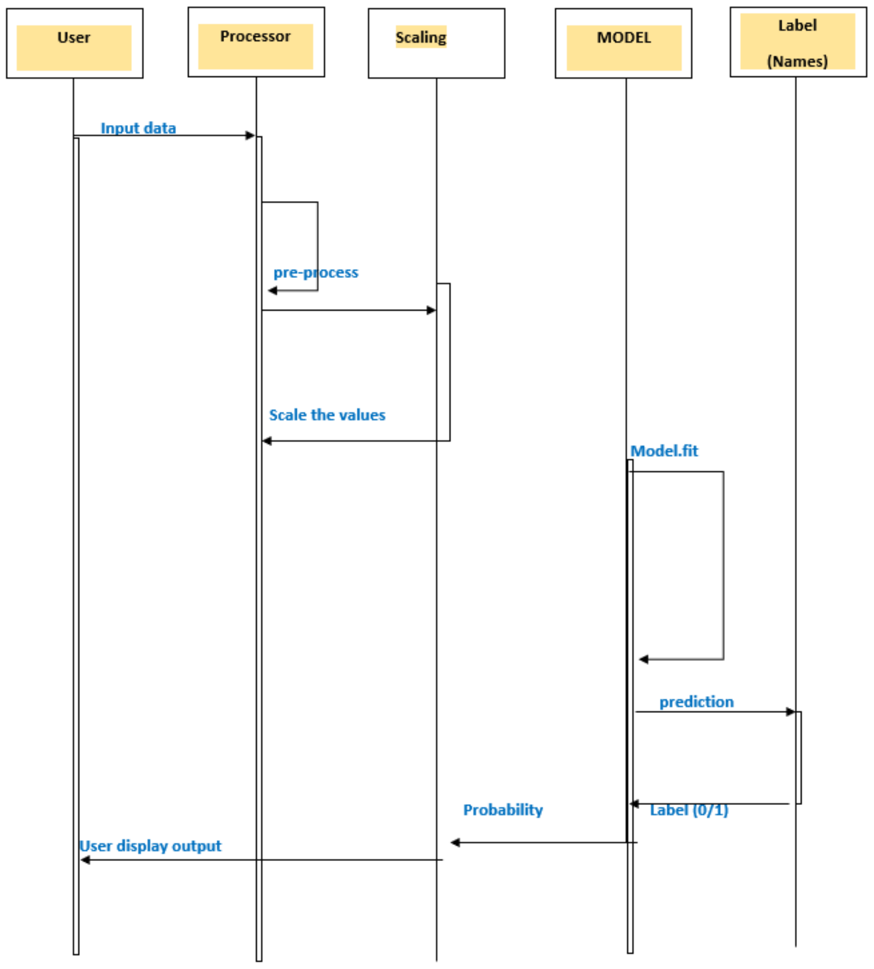


Figure 9.2 Sequence Diagram

According to the above graphic, the sequence diagram is made up of 5 different blocks: user, processor, scaling, Model, and labels. The user will enter data via CSV files that are already saved in the system, where pre-processing of the data is carried out to create differentiator parameters, which are then placed in the memory unit. The trained model file is loaded and its features are extracted to categorize the output once the CSV has been preprocessed and stored. Prediction values are shown following the output's classification.

9.3 DATA FLOW DIAGRAM LEVEL 0

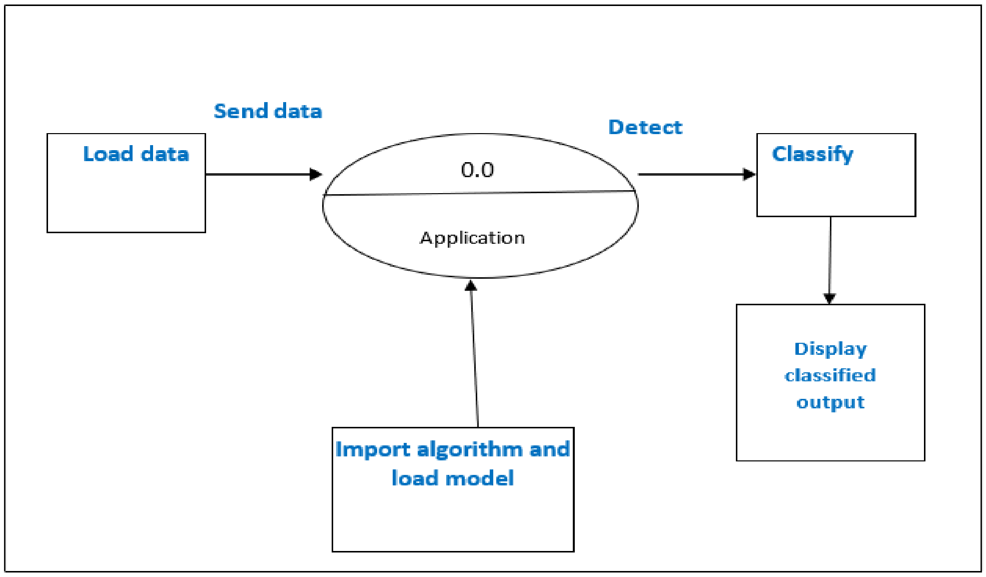


Figure 9.3 Data Flow Diagram Level 0

The DFD0 representation shown in the picture above gives you a view of the entire system or, to put it another way, a content diagram. The system is represented as a single high-level process in this quick view. Here, data is imported from a file into an application, where it is sent to a classification unit, which uses the loaded data to forecast the outcome using a trained model file with known output.

9.4 DATAFLOW DIAGRAM LEVEL 1

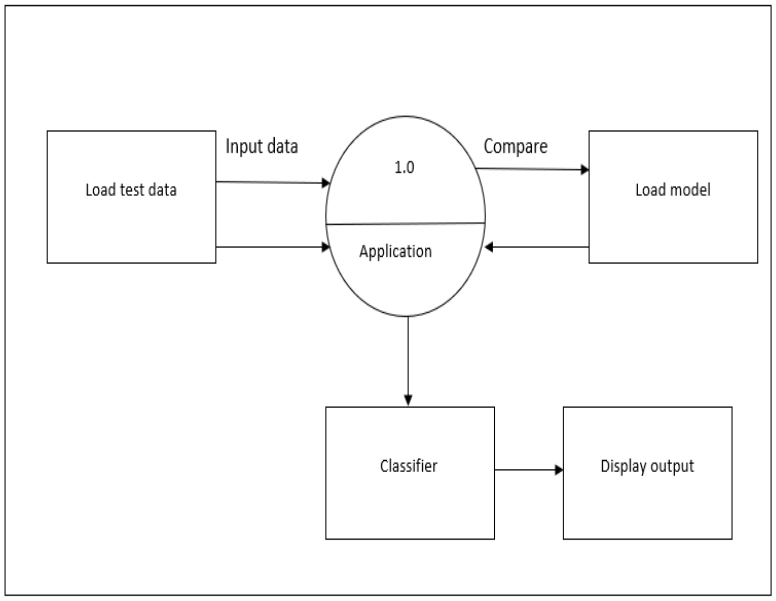


Figure 9.4 Dataflow diagram level 1

The illustration above shows how DFD1 works. The Level 0 DFD is divided into the more precise Level 1 DFD. Basic system modules and the data flow between different modules are shown in Level 1 DFD. Here, data from the file is entered into the program, where it is delivered to the classification unit to forecast the outcome, and classes are classed with labels.

9.5 Activity Diagram

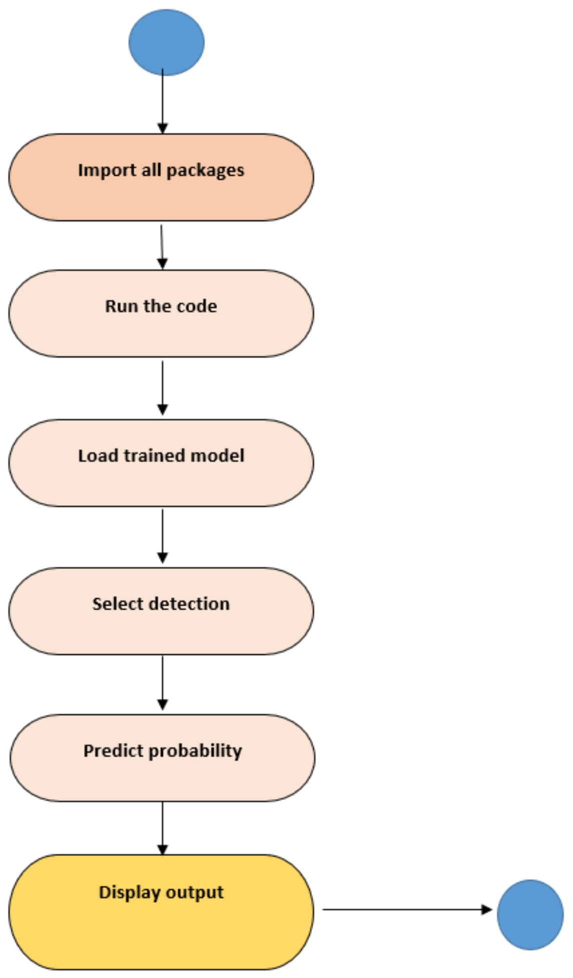


Figure 9.5 Activity Diagram

An action diagram is a behavioral diagram or one that shows how the system works. The control flow, together with a number of decision paths that can be taken during the activity, is shown in the activity diagram from the start to the end. First, we import all of the library packages required to run the code and support the two codes so they can function without errors. When code is executed, the desired result is produced immediately.

9.6 CLASS DIAGRAM

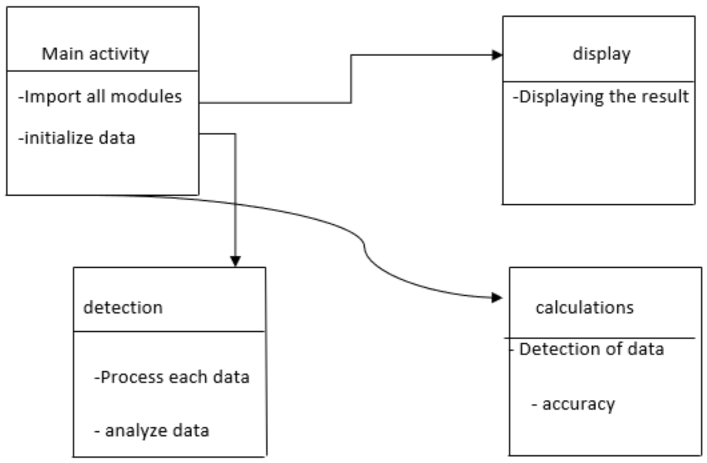


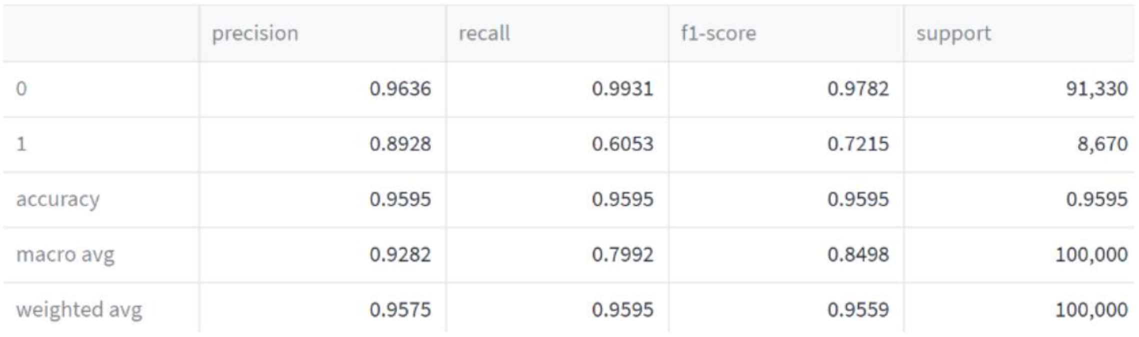
Figure 9.6 Class Diagram

The class diagram shows a static view of an application. It depicts the different types of objects that exist in a system, as well as their interactions. Besides the objects it holds, a class may also be inherited from another class. The classes diagram also display the many components of the system and create software code that can be drawn, described, documented or executed. To present a summary of the software system, you can view classes, relationships, properties and functions. It organizes the names, characteristics, and functions of classes in a separate section to assist in software development. Because it consists of a number of classes, interface, affiliation, collaboration and constraint, it is referred to as a structural diagram.

10. RESULTS

10.1 Logistic regression:

The model appears to have been assessed for a binary classification problem where class 0 contains 91,330 samples and class 1 has 8,670 samples based on the metrics. The precision for class 0 is 0.9636, meaning that 96.36% of all samples projected to be in class 0 were. The precision for class 1 is 0.8928, meaning that only 89.28% of the samples predicted to be in class 1 were. The recall for class 0 is 0.9931, meaning that 99.31% of all real class 0 samples were correctly recognized by the model as belonging to class 0. For class 1, the recall is 0.6053, which indicates that only 60.53% of the actual class 1 samples were properly identified by the model as belonging to that class. F1-scores are 0.9782 for class 0 and 0.7215 for class 1, respectively. The model properly identified 95.95% of all samples, according to its accuracy score of 0.9595. The precision, recall, and F1- score on average for both classes are represented by the macro average. The weighted average is the weighted average of the F1 score, recall, and precision divided by the number of samples in each class.

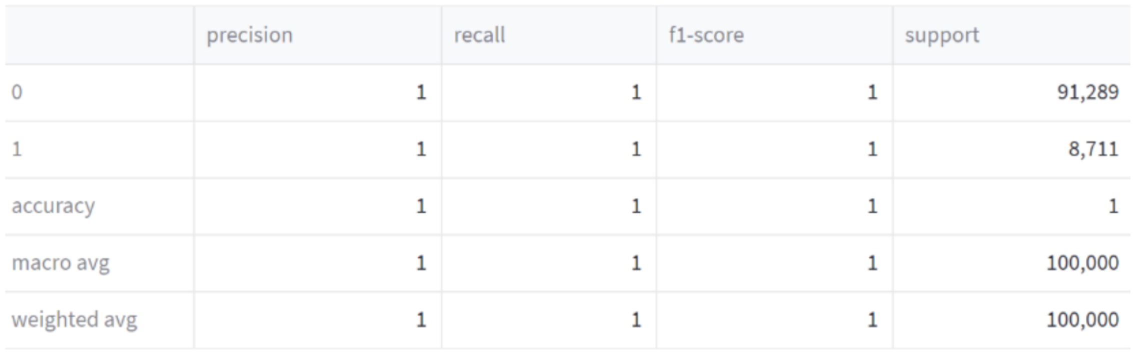


	precision	recall	f1-score	support
0	0.9636	0.9931	0.9782	91,330
1	0.8928	0.6053	0.7215	8,670
accuracy	0.9595	0.9595	0.9595	0.9595
macro avg	0.9282	0.7992	0.8498	100,000
weighted avg	0.9575	0.9595	0.9559	100,000

Figure 10.1 Evaluation Metrics of Logistic Regression

10.2 Decision tree:

The model looks to have performed flawlessly, with precision, recall, and an F1- score of 1.0 for both classes, according to the evaluation metrics you gave. This indicates that the model properly identified both types of samples in all samples. The accuracy is also 1.0, meaning that all samples were classified with 100% accuracy by the model. Given the model's flawless performance, the macro and weighted averages for each metric are also 1.0. However, it's crucial to keep in mind that this degree of flawless performance is uncommon, and unlikely to be attained in real-world situations, and there can be problems with the evaluation method or dataset employed.

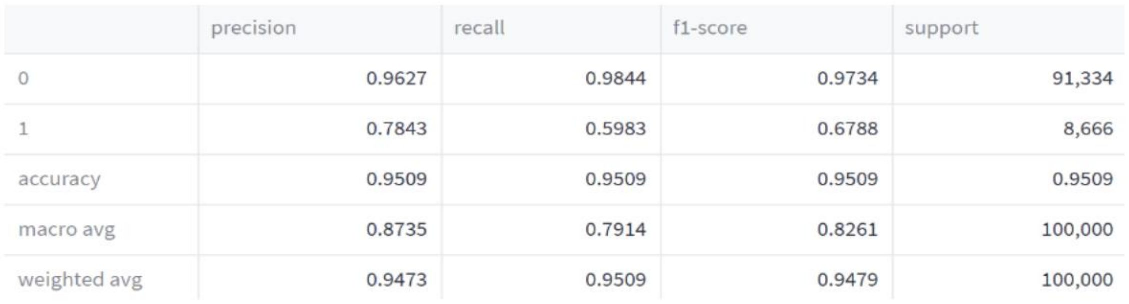


	precision	recall	f1-score	support
0	1	1	1	91,289
1	1	1	1	8,711
accuracy	1	1	1	1
macro avg	1	1	1	100,000
weighted avg	1	1	1	100,000

Fig 10.2 Evaluation Metrics of Decision Tree

10.3 Naive-Bayes:

The model performed well based on the evaluation indicators you supplied, but there is still space for improvement. The model can properly categorize the majority of non-fraudulent transactions because its precision, recall, and F1-score for class 0 are all above 0.96, which is an excellent result. It is clear from the model's lower precision, recall, and F1-score for class 1 that it is less effective in classifying fraudulent transactions. The accuracy of the model is 0.9509, which indicates that 95% of the samples were properly identified by the model. The macro average for precision, recall, and F1-score are all around 0.87- 0.88, which is decent. The weighted average for precision, recall, and F1-score are around 0.95- 0.96, which is also good. Overall, the model's performance could be improved by improving its ability to correctly classify fraudulent transactions. This could be achieved through feature engineering, using different algorithms or ensemble methods, or using larger and more diverse datasets.

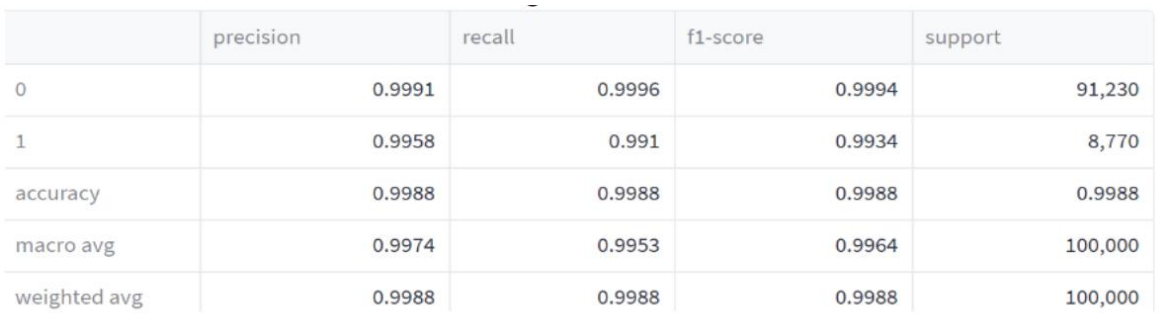


	precision	recall	f1-score	support
0	0.9627	0.9844	0.9734	91,334
1	0.7843	0.5983	0.6788	8,666
accuracy	0.9509	0.9509	0.9509	0.9509
macro avg	0.8735	0.7914	0.8261	100,000
weighted avg	0.9473	0.9509	0.9479	100,000

Fig 10.3 Evaluation metrics of Naïve-Bayes

10.4 KNN:

According to the evaluation metrics you provide, the model did a fantastic job of categorizing both fraudulent and non-fraudulent transactions. The model can properly categorize the great majority of transactions because its precision, recall, and F1-score are all above 0.99 for both classes of transactions. The accuracy of the model is 0.9988, which indicates that 99.88% of the samples were properly identified by the model. The precision, recall, and F1-score macro averages are all around 0.997, which is quite good. Precision, recall, and F1-score's weighted averages are all around 0.998, which is likewise very good. The model performed nearly flawlessly overall, demonstrating its suitability for detecting credit card fraud. The performance of the model must be carefully assessed in various circumstances because the assessment metrics can be affected by the particular dataset utilized and the evaluation approach.



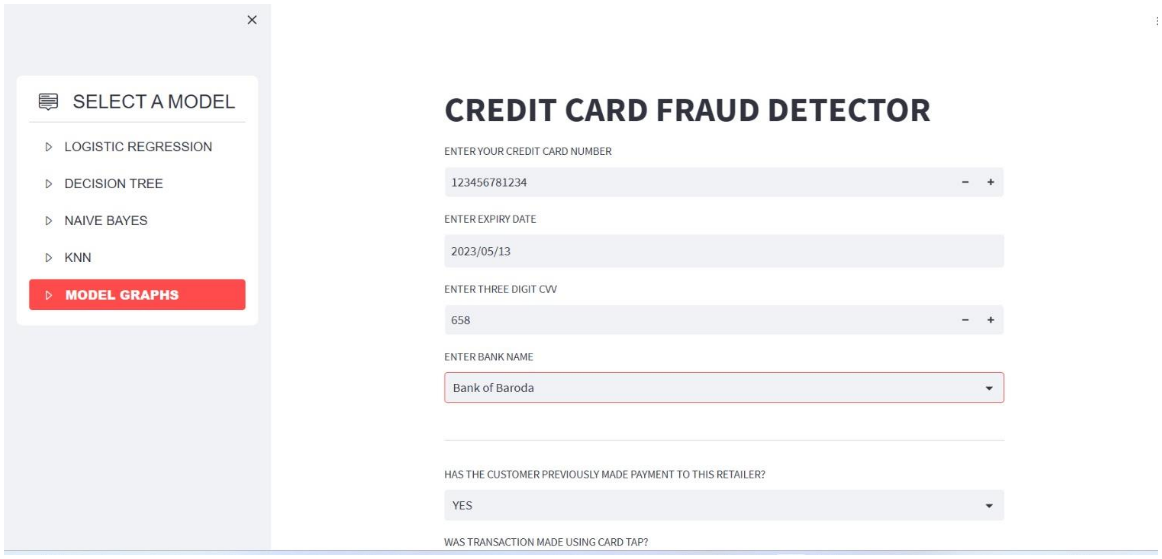
	precision	recall	f1-score	support
0	0.9991	0.9996	0.9994	91,230
1	0.9958	0.991	0.9934	8,770
accuracy	0.9988	0.9988	0.9988	0.9988
macro avg	0.9974	0.9953	0.9964	100,000
weighted avg	0.9988	0.9988	0.9988	100,000

Fig 10.4 Evaluation metrics of KNN

So by the above evaluation metrics, the KNN model's performance is more efficient in this scenario. So, the best metric depends on the problem at hand, and a combination of metrics can be used to get a more comprehensive evaluation of the model's performance.

10.5 Credit Card Fraud Detection Web Application

1)



CREDIT CARD FRAUD DETECTOR

ENTER YOUR CREDIT CARD NUMBER: 123456781234

ENTER EXPIRY DATE: 2023/05/13

ENTER THREE DIGIT CVV: 658

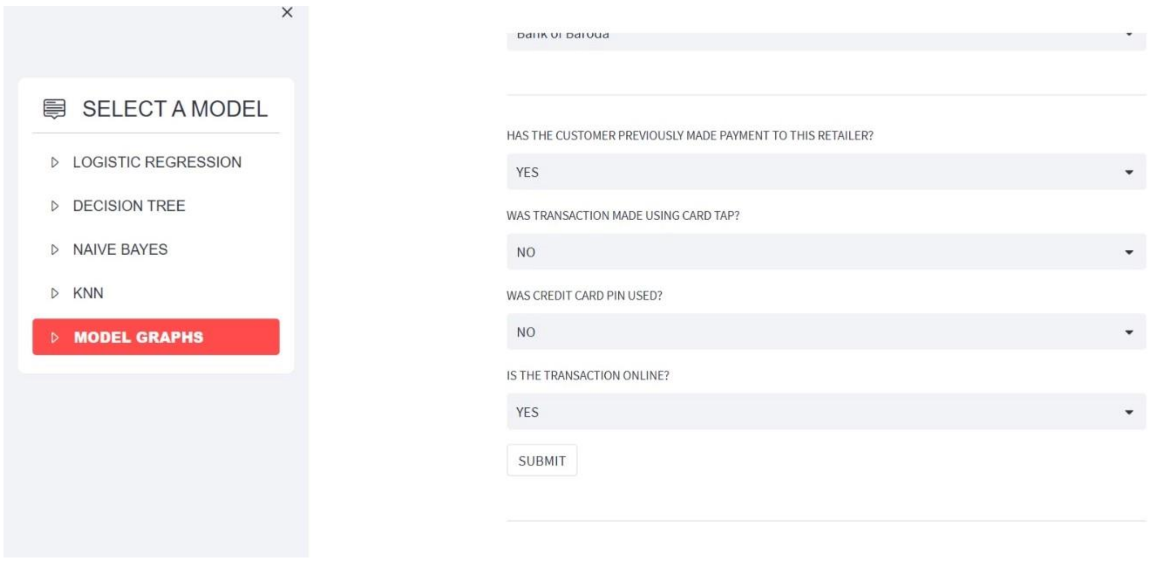
ENTER BANK NAME: Bank of Baroda

HAS THE CUSTOMER PREVIOUSLY MADE PAYMENT TO THIS RETAILER?: YES

WAS TRANSACTION MADE USING CARD TAP?

Figure 10.5.1 Web Application

2)



HAS THE CUSTOMER PREVIOUSLY MADE PAYMENT TO THIS RETAILER?: YES

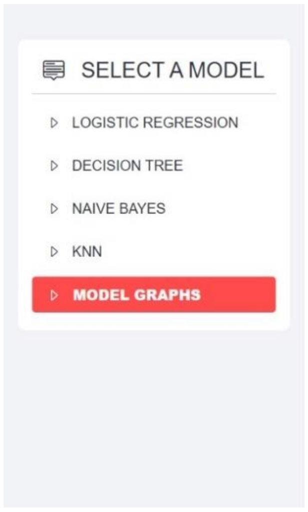
WAS TRANSACTION MADE USING CARD TAP?: NO

WAS CREDIT CARD PIN USED?: NO

IS THE TRANSACTION ONLINE?: YES

Figure 10.5.2 Web Application

3)



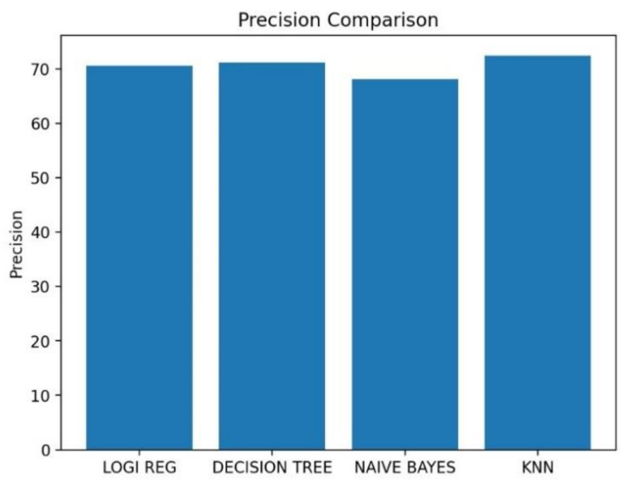


Figure 10.5.3 Comparison Graph

Based on the above results, we can conclude that the model with the highest accuracy is Decision Tree while the model with the highest precision is KNN. However, if we take into consideration both the metrics, then it is the Logistic Regression that has the average highest accuracy and precision, thus, making it the best-fit model.

11. CONCLUSION

Finally, the theft of credit cards is a major worry for both businesses and consumers. Through the use of machine learning techniques, real-time fraud detection has been accomplished. Our method for spotting credit card fraud in this study combines supervised and unsupervised learning techniques to look for patterns that indicate fraudulent behavior. We also merged the web framework with the learned machine learning models to produce a user-friendly interface for real-time fraud detection. Our tests revealed that the proposed framework significantly decreased false positives while being quite accurate in recognizing fraudulent transactions. Financial institutions may utilize the suggested framework to improve their capacity to spot fraud and reduce financial losses caused by credit card fraud. Future research can concentrate on improving the accuracy of the proposed framework and looking into the application of other machine-learning techniques to address this problem. The Credit Card Fraud Detection framework, which uses Streamlit and machine learning, is very effective at preventing financial losses brought on by credit card fraud. The effectiveness of the framework can be improved in future studies, and the use of more advanced machine-learning techniques can be explored. The Credit Card Fraud Detection Framework that is suggested, which uses Streamlit and Machine Learning, is a successful remedy for this problem. The platform uses numerous machine learning models, including Logistic Regression, Decision Tree, Random Forest, and XGBoost, to precisely identify fraudulent transactions.

To recognize credit card fraud, there are various machine learning techniques accessible, therefore it's critical to evaluate and select the one that's best for the specific application. Accurate credit card fraud detection is now achievable as a result of developments in machine learning and data analysis techniques that have improved fraud protection.

In this work, the ability of four different machine learning algorithms to classify a dataset of credit card transactions and identify fraud has been assessed. The outcomes show that the accuracy rates of all four machine learning techniques are above average. It is significant to emphasize that these findings are based on a small dataset, and further study is required to prove the efficacy of these tactics in real-world scenarios.

12. REFERENCES

[1] Raj, S. Benson Edwin, and A. Annie Portia. "Analysis on credit card fraud detection methods." In 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET), pp. 152- 156. IEEE, 2011.

[2] Ghosh, Sushmito, and Douglas L. Reilly. "Credit card fraud detection with a neural network." In System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on, vol. 3, pp. 621- 630. IEEE, 1994.

[3] Chaudhary, Khyati, Jyoti Yadav, and Bhawna Mallick. "A review of fraud detection techniques: Credit card." International Journal of Computer Applications 45, no. 1 (2012): 39-44.

[4] Srivastava, Abhinav, Amlan Kundu, Shamik Sural, and Arun Majumdar. "Credit card fraud detection using hidden Markov model." IEEE Transactions on Dependable and secure computing 5, no. 1 (2008): 37- 48.

[5] Awoyemi, John O., Adebayo O. Adetunmbi, and Samuel A. Oluwadare. "Credit card fraud detection using machine learning techniques: A comparative analysis." In 2017 international conference on computing networking and Informatics (ICCNI), pp. 1-9. IEEE, 2017.

[6] Sahin, Yusuf, and Ekrem Duman. "Detecting credit card fraud by ANN and logistic regression." In 2011 international symposium on Innovations in intelligent systems and Applications, pp. 315-319. IEEE, 2011.

[7] Kiran, Sai, Jyoti Guru, Rishabh Kumar, Naveen Kumar, Deepak Katariya, and Maheshwar Sharma. "Credit card fraud detection using Naïve Bayes model based and KNN classifier." International Journal of Advance Research, Ideas, and Innovations in Technology 4, no. 3 (2018): 44.

[8] Husejinovic, Admel. "Credit card fraud detection using naive Bayesian and c4. 5 decision tree classifiers." Husejinovic, A.(2020). Credit card fraud detection using naive Bayesian and C 4 (2020): 1-5.

[9] Saheed, Yakub K., Moshood A. Hambali, Micheal O. Arowolo, and Yinusa A. Olasupo. "Application of GA feature selection on Naive Bayes, random forest and SVM for credit card fraud detection." In 2020 international conference on decision aid sciences and Application (DASA), pp. 1091-1097. IEEE, 2020.

[10] Varmedja, Dejan, Mirjana Karanovic, Srdjan Sladojevic, Marko Arsenovic, and

Andras Anderla. "Credit card fraud detection-machine learning methods." In 2019 18th International Symposium INFOTEHJAHORINA (INFOTEH), pp. 1-5. IEEE, 2019

[11] Yee, Ong Shu, Saravanan Sagadevan, and Nurul Hashimah Ahamed Hassain Malim. "Credit card fraud detection using machine learning as data mining technique." Journal of Telecommunication, Electronic and Computer Engineering (JTEC) 10, no. 1-4 (2018): 23-27.

[12] Malini, N., and M. Pushpa. "Analysis on credit card fraud identification techniques based on KNN and outlier detection." In 2017 third international conference on Advances in Electrical, electronics, information, communication, and bio-informatics (AEEICB), pp. 255-258. IEEE, 2017.

[13] Ganji, Venkata Ratnam, and Siva Naga Prasad Mannem. "Credit card fraud detection using anti-k nearest neighbor algorithm." International Journal on Computer Science and Engineering 4, no. 6 (2012): 1035-1039.

[14] Vengatesan, K., A. Kumar, S. Yuvraj, V. Kumar, and S. Sabnis. "Credit card fraud detection using data analytic techniques." Advances in Mathematics: Scientific Journal 9, no. 3 (2020): 1185-1196.

[15] Zareapoor, Masoumeh, K. R. Seeja, and M. Afshar Alam. "Analysis on credit card fraud detection techniques: based on certain design criteria." International Journal of computer applications 52, no. 3 (2012).

[16] Nancy, A. Maria, G. Senthil Kumar, S. Veena, NA S. Vinoth, and Moinak Bandyopadhyay. "Fraud detection in credit card transaction using a hybrid model." In AIP Conference Proceedings, vol. 2277, no. 1, p. 130010. AIP Publishing LLC, 2020.

[17] Kaur, Darshan. "Machine Learning Approach for Credit Card Fraud Detection (KNN & Naïve Bayes)." In Machine Learning Approach for Credit Card Fraud Detection (KNN & Naïve Bayes)(March 30, 2020). Proceedings of the International Conference on Innovative Computing & Communications (ICICC). 2020.

[18] Saheed, Yakub Kayode, Usman Ahmad Baba, and Mustafa Ayobami Raji. "Big Data Analytics for Credit Card Fraud Detection Using Supervised Machine Learning Models." In Big Data Analytics in the Insurance Market, pp. 31-56. Emerald Publishing Limited, 2022.

[19] Adewumi, Aderemi O., and Andronicus A. Akinyelu. "A survey of machine-learning and nature-inspired based credit card fraud detection techniques." International Journal of System Assurance Engineering and Management 8 (2017): 937-

953.

[20] Mehbodniya, Abolfazl, Izhar Alam, Sagar Pande, Rahul Neware, Kantilal Pitambar Rane, Mohammad Shabaz, and Mangena Venu Madhavan. "Financial fraud detection in healthcare using machine learning and deep learning techniques." Security and Communication Networks 2021 (2021): 1-8.

[21] Handa, Akansha, Yash Dhawan, and Prabhat Semwal. "Hybrid analysis on credit card fraud detection using machine learning techniques." Handbook of Big Data Analytics and Forensics (2022): 223-238.

[22] Tiwari, Pooja, Simran Mehta, Nishtha Sakhuja, Ishu Gupta, and Ashutosh Kumar Singh. "Hybrid method in identifying the fraud detection in the credit card." In Evolutionary Computing and Mobile Sustainable Networks: Proceedings of ICECMSN 2020, pp. 27-35. Springer Singapore, 2021.

[23] Kazemi, Zahra, and Houman Zarrabi. "Using deep networks for fraud detection in the credit card transactions." In 2017 IEEE 4th International Conference on knowledge-based engineering and innovation (KBEI), pp. 0630-0633. IEEE, 2017.

[24] Faraji, Zahra. "A Review of Machine Learning Applications for Credit Card Fraud Detection with A Case Study." SEISENSE Journal of Management 5, no. 1 (2022): 49-59.

[25] Prusti, Debachudamani, and Santanu Kumar Rath. "Web service-based credit card fraud detection by applying machine learning techniques." In TENCON 2019-2019 IEEE Region 10 Conference (TENCON), pp. 492-497. IEEE, 2019.

[26] Ahammad, Jalal, Nazia Hossain, and Mohammad Shafiul Alam. "Credit card fraud detection using data pre-processing on imbalanced data-Both oversampling and undersampling." In Proceedings of the International Conference on Computing Advancements, pp. 1-4. 2020.

[27] Ata, Oğuz, and Layth Hazim. "Comparative analysis of different distributions dataset by using data mining techniques on credit card fraud detection." Tehnički vjesnik 27, no. 2 (2020): 618-626. [28] Shirgave, Suresh, Chetan Awati, Rashmi More, and Sonam Patil. "A review on credit card fraud detection using machine learning." International Journal of Scientific & technology research 8, no. 10 (2019): 1217-1220