CHAPTER-01

INTRODUCTION

This chapter of the project report is the beginning of the content of this report. It contains the building up of the plot of this report. The problem statement along with the main objectivesof this project are discussed here. The significance of this project and the real motivation behind the intentions to take up this topic as our project are also listed in detail in this chapter. The organization of this project report is also listed in this very chapter.

INTRODUCTION

“Fraud” in credit card transaction is unauthorized and unwanted usage of an account by someone other than the owner of the account. Necessary prevention measures can be taken to stop this abuse and the behavior of such fraudulent practices can be defined as a case wherea person uses someone else’s credit card for personal reasons while the owner and the card issuing authorities are unaware of the fact that the card is being used.
In today’s era, with the widespread use of credit cards for online transactions, the risk of fraudulent activities has increased significantly. Addressing this challenge demands sophisticated methods that can swiftly and accurately detect fraudulent transactions to safeguard financial assets and uphold customer trust.

FIG.1: Credit Card fraud detection image

Fraud detection involves monitoring the activities of populations of users in order to estimate,perceive or avoid objectionable behavior, which consist of fraud, intrusion, and defaulting.

Machine learning algorithms are employed to analyses all the authorized transactions and report the suspicious ones. These reports are investigated by professionals who contact the cardholders to confirm if the transaction was genuine or fraudulent.

Some of the currently used approaches to detection of such fraud are:

- Artificial Neural Network (ANN)
- Fuzzy Logic
- Genetic Algorithm
- Logistic Regression
- Decision Tree
- Support Vector Machines (SVM)
- Bayesian Networks
- Hidden Markov Model (HMM)
- K-Nearest Neighbor

PROBLEM STATEMENT

Creating and implementing efficient fraud detection strategies while handling additional fraud crimes presents the true challenge in credit card fraud detection. One of the numerous issues facing today's fraud detection systems is the requirement to identify between fraudulent and authentic content fast and accurately. The prediction model was disliked in the majority of classes due to the stark disparity between the actual market and the fraud in the data set. Concerns have also been raised concerning the model's generalizability in spotting novel fraudtendencies and adjusting to evolving ones. Stakeholders and regulators are the only ones who can comprehend machine learning models used in fraud detection because they are still tough to understand.

Furthermore, since fraud is always evolving, flexible systems that can stop scams without compromising user privacy or experience are needed. Last but not least, a persistent problemin the sector is maintaining tight security protocols and legal requirements while skillfully incorporating these fraud detection systems into the current financial infrastructure.

OBJECTIVES

There are some proposed methods to develop a mechanism to determine that the upcoming transaction is fraud or not. The fraud transaction will be recognized with the help of location where the transaction took place, Frequency the interval of the time between two transactions, Amount what was the amount that was withdrawn from the transaction. And the comparison of different Machine Learning algorithms will be shown. The figure below showsthe overall system framework.

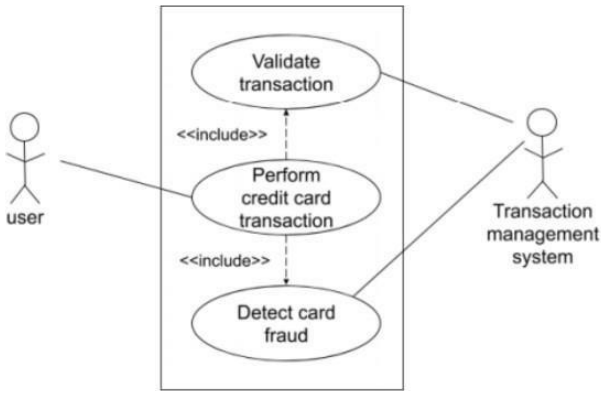


FIG.2: System framework

The main objectives which we try to aim during the completion of this project are all listed below –

- Get Credential Information.
- To balance the dataset which is unbalanced using SMOTE technique.
- To create a machine learning model using Logistic Regression, XG Boost, DecisionTree.
- Faster detection and higher accuracy.

SIGNIFICANCE AND MOTIVATION OF THE PROJECT WORK

The significance and motivation behind undertaking a project “Credit Card Fraud Detection Using Machine Learning” is the increasing number of frauds. In today’s era of technology, it become of piece of cake for the fraudulent to conduct credit card fraud. So, in order to minimize the fraud, it is important to build a system which help us to minimize the credit card fraud. It holds critical importance in the realm of financial security and customer trust. Beloware some points that highlight their significance:

- **Financial Protection:** Credit card fraud poses a significant threat to financial institutions and individuals, leading to substantial monetary losses. Detecting fraudulent losses for both the financial institution and customers.
- **Customer Trust and Satisfaction:** Effective fraud detection using machine learning techniques enhances customer confidence in the security measures provided by financial institutions. Protecting customers financial assets fosters trust and loyalty, contributing to overall customer satisfaction.
- **Adaptive and Agile Solutions:** In response to novel fraudulent tendencies, machine learning algorithms are able to change and adapt. These models get better at spotting new fraud schemes by constantly absorbing new data, which makes the system more adaptable and resilient to changing threats.
- **Minimization of False Positives:** Conventional fraud detection techniques may raise false alerts, which would annoy customers by preventing valid transactions. It is possible to optimize machine learning models to reduce false positives, which will facilitate and uninterrupted transactions for real customers.
- **Efficiency and Scalability:** Machine learning-based fraud detection systems provide scalability and efficiency as transaction volumes increase. They are able to quickly and efficiently detect possible fraudulent activity by handling and processing massive volumes of data in real-time.

- **Technological Advancements:** Using machine learning approaches makes it possible to apply sophisticated algorithms that can identify irregularities and subtle patterns that rule-based systems or manual inspection can miss.
- **Compliance and Regulatory Requirements:** In the financial sector, adherence to regulatory norms and criteria is essential. Meeting compliance requirements and regulatory expectations is facilitated by the use of machine learning to implement strong fraud detection mechanisms.
- **Industry Competitiveness:** By showcasing their dedication to client security, financial institutions that implement cutting-edge fraud detection systems acquire a competitive advantage. It demonstrates their aptitude for technology and commitment to bringing cutting-edge solutions to the market.
- **Data-Driven Decision Making:** By offering insights into transaction patterns and possible hazards, machine learning-driven fraud detection systems enable financial institutions to make data-driven decisions. This facilitates the development of proactive fraud mitigation strategies.

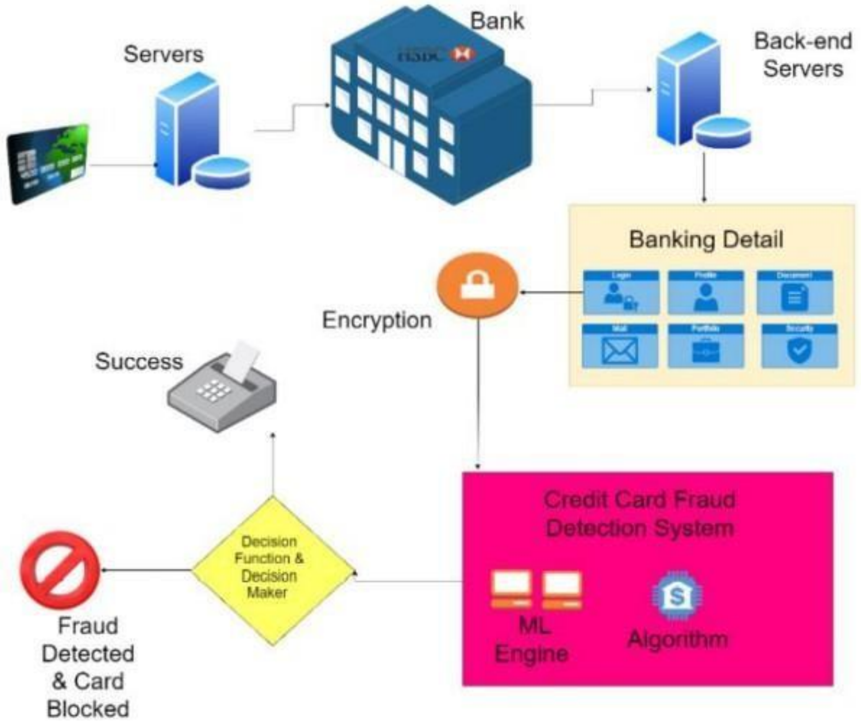


FIG.3: Data-Driven decision making

ORGANIZATION OF PROJECT REPORT

Project report introduction:

- Serve as the report’s opening section.
- Write the foundational text for upcoming chapters.
- Describes the essential elements of the project, including the problemdescription, primary goals, and significance.
- Explain the significance of implementing robust fraud detection systems, emphasizingthe need to protect user’s financial assets and maintain trust.
- Define the scope of the project by specific objectives, goals and limitation in developing a machine learning-based fraud detection system.

Context and setting:

- **Technical context:** In this digital age, online trading has become an important part of daily life. The increase in credit card use in online shopping has led to an increase in fraud and has necessitated the development of fraud detection systems.
- **Security and trust:** The program operates in an environment where ensuring security and building trust among stakeholders is important. The project aims to improve security measures and restore trust in online payments by improving fraud detection methods.

Overview of Credit Card Fraud Detection using ML:

- **Purpose:** Use machine learning to detect and prevent credit card fraud.
- **Problem Statement:** Identification of fraud in credit card information resulting from legal transactions.
- **Dataset:** Get comprehensive data on fraud cases related to the credit card industry.

Project Objectives:

The objective of credit card fraud using machine learning is to create an effective system that can identify and prevent fraud in credit card information. Aim of our project is:

- **Fraud Detection:** Build a ML models that can show the difference between not fraudulent and fraudulent credit card transactions. That is which transaction is fraud and which transactions is valid.
- **Model Accuracy:** Provide models for fraud detection that minimize false positives and negatives while offering accurate, true, recall, and F1 scores.
- **Balancing the Data:** The data we use was highly unbalanced, so we need to balance the data using the data balancing technique SMOTE (Synthetic Minority Over- Sampling Technique).

Inspiring and Vital:

- **Financial Security:** In today's digital world, protecting financial transactions are veryimportant. Protecting people and businesses from financial losses through fraud is crucial to financial stability and trust in the financial system.
- **Technological Innovation:** To address real-world issues, this program makes use of technologies like data science and machine learning. The ability of these technologies to enhance security measures is demonstrated.
- **Social Impact:** Customer trust in internet enterprises can be raised by identifying and stopping credit card fraud. By promoting the upkeep of consumer, financial institution, and company trust, it fosters financial stability.

The information flows clearly from this well-organized organization, giving a thorough grasp ofthe project objectives and context.

CHAPTER-02

LITERATURE REVIEW

OVERVIEW OF RELEVANT LITERATURE

The aim of the Credit Card Fraud Detection Machine Learning (ML) project is to develop a reliable system that can detect credit card fraud. He acknowledged that financial fraud linked to electronic payments and e-commerce platforms is increasing and emphasized the need for effective detection methods. The limitations of existing security methods, such as tokenization and encryption, require the use of machine learning (ML) methods becausethese methods often fail to protect new information from fraud.

[1] Credit Card Fraud Detection using Machine Learning Algorithms (2020):

Overall, this paper presents research on various machine learning, challenges, and newtechniques to improve credit card fraud, detection systems. The plan will involve a group of cardholders, training different employees and using strategies to learn more about fraud. These studies aim to analyze the customer's details through the transaction, extract behavioral patterns in the cardholder group according to transaction costs, and then introduce different people to this group.

[2] Credit card fraud detection using machine learning techniques A comparative analysis (2017): This article focuses on the challenges of credit card fraud, highlighting the vulnerability of credit card fraud as well as the ever-changing nature of fraudulent behavior and fraud-related data. financial information fraud. It investigates the performance of three machine learning classifiers (Naïve Bayes, K-Nearest Neighbors (KNN), and logistic regression) on credit card fraud profiles obtained from residents of Europe(with284,807transactions).The results show the best accuracy achieved by Naive Bayes (97.92%), KNN (97.69%) and logistic regression (54.8%) classifiers.

[3] Credit Card Fraud Detection Using Machine Learning(2022):

This article addresses the problem of credit card fraud that has arisen due to the increasing use of credit cards around the world. The authors cite statistics from 2019 and 2020 that show an increase in credit card fraud due to the creation of new illegal accounts or unauthorized use of existing accounts. This warning led the authors to consider an analysis to address the problem, specifically using various machine learning (ML) methods to detect fraud in many credit card transactions. Overall, this article focuses on the use of machine learning techniques to solve the growing problem of credit card fraud to determine the most appropriate and effective methods for detecting fraud based on comparisons and insights from previous research.

[4] Anomaly Detection in Credit Card Transactions using Machine Learning(2020):

This research paper focuses on the development of an automatic and effective method to detect credit card fraud using machine learning techniques, specifically the search forest classification algorithm with the help of H2O.ai. This study aims to solve the fundamental problem of credit card fraud, which has become an important problem in the age of digital money. This research specifically investigates the classification forest algorithm, which is not very useful in detecting anomalies, especially credit card fraud. Performance evaluation of forest classification models is often based on widely accepted criteria such as precision and recall. The test data used in this study was taken from the data science competition platform Kaggle. Overall, this articlewill focus on the use of machine learning, specifically the classification forest algorithm, to create a powerful fraud detection system for the credit card industry.The importance of this research lies in its ability to help create automated systems thatcan prevent credit card fraud, there by protecting the interests of consumers and banks.

[5] Selection Features and Support Vector Machine for Credit CardRisk Identification(2020):

In this case study, machine learning is used to address the credit card risk detection (CCR) issue. It illustrates the probability of credit card fraud in the digital age as well as the financial consequences of fraud, according to IC3 (Internet Crime Complaint Centre). The investigator lacks the requisite information, and guidance creates the context of the design by raising the likelihood of fraud. Using RFC in conjunction with SVM to extract the most salient characteristics is the main focus, and it underscores the importance of effectively identifying tiny abnormalities in large datasets. In order to shed light on fraud detection strategies, this article analyses and examines prior research as well as publications that use supervised and unsupervised algorithms, among other techniques and methodologies. Examples include Contrast Miner, Enhanced Fraud Miner, Principal Component Analysis (PCA), Hidden Markov Models (HMM), and cost-aware neural network fraud models. Overall, this article focuses on the development of advanced credit card fraud models using machine learning algorithms, with particular attention to specialized options such as random forest classifiers and support vector machines to improve the classification of large files.

[6] Credit card fraud detection in the era of disruptive technologies: Asystematic review(2022): The issue of addressing information in consistencies in credit card theft is thoroughly examined in this paper. It investigates numerous data augmentation strategies and presents a novel approach known as K-CGAN in order toaddress this issue. The purpose of this study is to assess the efficacy of different information management techniques and determine how they affect the credit card fraud detection system. In consistencies in credit card information, which make up a very little portion of the fraud problem, are the primary issue that must be addressed. In order to improve performance, machine learning models require balanced data, which is why this article examines different data augmentation strategies to close the gap. It introduce K-CGAN a new augmentation model, as well as other methods such as SMOTE, B-SMOTE, and CGAN, which generate synthetic data to balance the dataset. This article discuss the limitations of some oversampling techniques such as SMOTE and GANs as a new solution. It shows the advantage and potential pitfalls of these strategies when dealing with different data. It demonstrates the flexibility and advantage of GAN in real-world.

[7] Improving Classification Performance in Credit Card Fraud Detection by Using New Data Augmentation (2023): This article provides a comprehensive investigation of the problem of resolving information inconsistencies in credit card fraud. It explores various data augmentation techniques to solve this problem and introduce a new approach called K-CGAN. This study evaluates the effectiveness of variousdatacurationmethodstounderstandtheirimpactonclassificationsystemsfor credit card fraud detection. The main issue that needs to be addressed is the inconsistency of credit card information, which is a small part of the fraud problem. Since machine learning models require balanced data for performance, this article explores various data augmentation techniques to balance the gap. It introduces and introduces K- CGAN, a new data augmentation model, as well as other established methods such as SMOTE, B-SMOTE, and CGAN, which generate synthetic data to balance the dataset. This article discusses the limitations of some oversampling techniques suchas SMOTE and the introduction of GANs as a new solution. It shows the advantages and potential pitfalls of these strategies when dealing with different data. It demonstrates the flexibility and advantages of GANs in real-world predictions and presents K-CGAN as a potential development that could overcome the limitations of existing methods.

[8] Credit Card Fraud Detection using Machine Learning and Data Science(2017): This article centers around the utilization of information science andAI procedures to dissect charge cards. It accentuates the significance of recognizing fake strategic policies to forestall unlawful charges against purchasers. The objective is to foster a model that precisely distinguishes deceitful exchanges while limiting misclassification. The examination included investigating and focusing on informational indexes utilizing fair-minded search strategies, for example, residential areas backwoods prohibition from PCA exchange charge card exchanges.

[9] Survey Paper On Credit Card Fraud Detection(2014): The paper examines credit card theft and demonstrates how common it is in the digital age, particularly in light of the expansion of e-commerce and online businesses. It stresses that credit cardfraud should be eradicated in order to prevent this kind of fraud and defines creditcard fraud as the unauthorized use of an account by someone other than the account owner. The introduction gives a general summary of the difficulties in detecting fraud, emphasizing class disparities and the evolving character of the transfer model.It also illuminates the procedures that are engaged in the field of fraud detection systems, wherein machine learning algorithms examine approved transactions and identify transactions that are questionable so that specialists can look into themfurther. The paper discusses methods for precisely predicting commercial fraud, such as hybrid data mining, outlier mining, and sophisticated network classification algorithms. To find out how well nontraditional techniques like genetic algorithms work in lowering false alarms in fraud detection, more research has been done on them. This article emphasizes the significance of credit card fraud, the issues surrounding it, and the several methods and algorithms that have been employed in the field of study to address it. aims to improve fraud prevention by offering a thorough summary of the most recent information and procedures pertaining to credit card theft.

[10] Credit Card Fraud Detection Using Local Outlier Factor And Isolation Forest(2019): Since credit cards are used in both online and offline cashless transactions, this article focuses on the risk of credit card fraud in e-commerce and online commerce. It draws attention to the vulnerabilities of credit cards and the growing annual losses incurred by consumers and financial institutions as a result of fraud. Credit card fraud is on the rise and a major threat to consumers and businesses alike as technology and online commerce advance. The initial step towards fraud is the theft of the actual card or card data. The Credit Card Transactions in Europe in September 2013 data used in this article comes from Kaggle. Transactions are classified as fraudulent or non-fraudulent in the data, which creates questions that need to be looked into. An experimental comparison between the distributed forest method and the local outlier factor is presented in this work. By utilising machine learning, these systems examine anomalous behaviour to identify fraudulent activity. For nearby settlements, the accuracy was 97%, but for distant woodlands, it was76%.This article explores the possible losses resulting from credit card fraud and emphasizes how urgent is it to completely alter fraud. This highlight show crucial it isto evaluate and select the most effective algorithm for detecting fraud.

[11] Application of Classification Models on Credit Card Fraud Detection(2007): In this paper, the effectiveness of three classification models—logistic regression, decision trees, and neural networks—in credit card analysis is examined. With data on monetary losses brought on by fraud in nations like the United States and the United Kingdom, it tackles the largest credit card fraud issue credit card issuers face globally. In light of China's high-risk economy, this study emphasizes the value of smart fraud models as a weapon in the fight against fraud. For financial organizations, credit cards are crucial. The two types of credit card fraud discussed in this article are international and domestic. This study's major goal is to assess and contrast the outcomes of logistic regression, decision trees, and neural networks in external card analysis. Analyze the accuracy of your predictions using these classification techniques. Section 2 of the paper's structure consists of a review of pertinent studies, while Section 3 describes the research procedure. The research data and experiments are presented in Section4, and the results are described and their implications are analyzed in Section 5. Section 6 brings the article to a close by presenting research findings and ideas regarding theusefulness of classification models in credit card theft.

KEY GAPS IN THE LITERATURE

- **Specificity of discrepancies:** Does not specify or describe the type of discrepancies present in credit card transactions. Understanding the nuances of inconsistencies (e.g., incorrect or missing data, differences in data exchange, changing data input)is critical to creating problem solving.

- **The Impact of Conflict on Crime Investigations:** The narrative does not clearly show that conflicting data directly affects the accuracy or reliability of the scam. A detailed understanding of how these inconsistencies affect the performance of machine learning models is critical to resolving these issues.

- **Detailed evaluation of augmentation techniques:** Although many data augmentation techniques (SMOTE,B-SMOTE,CGAN,K-CGAN) are mentioned, They are not comprehensive on how each approach handles inequality. These methods do not providea comparison or detailed evaluation of the effectiveness of these methods in reducing data inconsistencies.

- **Limitations of Electricity and GANs as Solutions:** Briefly explains the limitations of some previous approaches (such as SMOTE) and presents GANs as new solutions.However, it did not explicitly address the shortcomings of the current system in processing such different information. There is also no detailed research on how GANs can solve these limitations in real situations.

- **Limited Discussion on Comparative Analysis:** A brief summary of past fraud detection techniques and techniques, but no comparisons or evaluations are delvedinto. The importance of this process affects the proposed method. Offering good comparisons or differences will improve understanding of the novelty of the plan.

- **Not paying enough attention to dataset problems:** Not paying enough attention to specific problems arising from the credit card fraud data set and not paying enough attention to the models proposed to solve the problems no. Detailed information on issues such as class inconsistencies, data inconsistencies, or complex credit card features can help define solutions more clearly.

- **Clarify classification improvement:** While the definition states the goal of improving classification performance using machine learning algorithms, it is not clear what improvement needs or performance metrics the proposed model targets.It is important to demonstrate the need for existing methods in terms of accuracy, precision, recall, or other metrics.

CHAPTER-03

SYSTEM DEVELOPMENT

In this chapter, a thorough discussion regarding the requirements (both functional and non-functional) is done. The architecture on which these language models are build is the transformer architecture which was launched in 2017. The various steps along with the diagram are also briefed in this chapter. The implementation part of this project report is also shown in this very chapter. The major key challenges that we came across while working on this project are also mentioned in the end.

REQUIREMENTS AND ANALYSIS

FUNCTIONAL REQUIREMENTS

The functional requirements of the project on "Credit Card Fraud Detection Using Machine Learning" outline the specific features and capabilities that the system should possess to meet its objectives effectively. These requirements are crucial for the successful development,implementation, and utilization of project in the context of cyber security testing and simulation. Functional requirements for developing a credit card fraud detection model revolve around the system's capabilities and functionalities. Here are the key functionalrequirements:

1. **Data Collection and Integration:**

   Requirement: The system's job is to gather transaction data from multiple sourcesand combine it into a single, central database.

   Justification: Accurate fraud detection requires transaction data to be readily available and accessing and combining data from several sources guarantees this.

2. **Feature Engineering and Selection:**

   Requirement: In order to train the model, the system must extract pertinent features and choose those that are instructive.

Justification: The process of extracting and selecting features enhances the predictivepower of the model by incorporating relevant data and removing super fluousor noisy qualities.

3. **Model Development and Training:**

   Requirement: Capacity to use past data to build machine learning models that classify transactions as fraudulent or not.

   Justification: In order to develop prediction algorithms that reliably discern between authentic and fraudulent transactions, model training is essential.

4. **Real-time Prediction and Scoring:**

   Requirement: Put in place a system that can score and process transactions instantly in order to quickly spot possible fraudulent activity.

   Justification: Processing in real time guarantees that suspicious transactions may be handled quickly, reducing the possibility of losses.

5. **Model Evaluation and Validation:**

   Requirement: Utilize relevant metrics to assess the accuracy, precision, and recall of the model.

   Justification: Frequent assessment and validation support enhancements and validate the model's efficacy, guaranteeing accurate fraud detection.

6. **Alert Generation and Action Triggers:**

   Requirement: Create alerts for fraudulent transactions that have been reported, and then take the necessary measures to handle those incidents.

   Justification: Alerts make it possible to respond quickly and mitigate possible fraudulent activity by facilitating timely intervention.

7. **Model Retraining and Adaptability:**

   Requirement: Establish methods for ongoing learning and retrain models with fresh data so they can adjust to changing fraud trends.

   Justification: The system is kept up to date and flexible to new fraud strategies and emerging patterns through constant retraining.

8. **User Interface and Reporting:**

   Requirement: Create an intuitive user interface and produce thorough reports for regulatory compliance and stakeholders.

   Justification: Monitoring system performance, offering insights, and guaranteeing regulatory compliance are made easier with the help of an easy-to-use interface and educational reports.

9. **Security and Consistence:**

   Requirement: Guarantee vigorous safety efforts to safeguard delicate exchange information and conform to information assurance guidelines.

   Justification: Safety efforts shield private data, keeping up with client trust andmeeting administrative prerequisites.

10. **Transaction Monitoring and Profiling:**

    Requirement: Profile client conduct and set unique limits to identify deviations fromordinary exchange designs.

    Justification: Observing way of behaving and setting limits further develop precisionby recognizing unusual exchanges well defined for individual clients or gatherings.

NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements describe the characteristics, attributes, and constraints that define how a system performs rather than what it does. Non-Functional requirements list out the client expectations from product design, security, accessibility, and reliability or performance viewpoint.

1. **Performance:**

Requirement: The framework ought to handle exchanges with negligible inactivity, holdingback nothing or close constant handling.

Justification: Low inactivity guarantees ideal recognition and reaction to possible extortion,limiting monetary misfortunes.

2. **Scalability:**

Requirement: The framework should deal with expanding exchange volumes withoutcompromising execution or exactness.

Justification: Adaptability guarantees the framework stays successful and responsive asexchange loads develop over the long haul.

3. **Reliability:**

Requirement: The framework ought to keep up with high accessibility, going for the goldabove industry guidelines.

Justification: High accessibility guarantees constant extortion location capacities,forestalling disturbances in monetary administrations.

4. **Security:**

Requirement: Carry out powerful safety efforts to protect delicate exchange information andguarantee consistence with industry guidelines.

Justification: Solid safety efforts safeguard private data, keeping up with client trust andmeeting administrative prerequisites.

5. **Usability:**

Requirement: Foster an instinctive UI for framework overseers and examiners to easilyinterface with the framework.

Justification: An instinctive connection point up grades client efficiency, working withproficient checking and the board of the misrepresentation recognition framework.

6. **Maintainability:**

Requirement: Guarantee the framework is particular, indisputable, and simple to keep up with,with clear rendition control and updates.

Justification: Simplicity of support lessens margin time and works with consistent updates,further developing framework dependability and life span.

7. **Interoperability:**

Requirement: The framework ought to incorporate consistently with existing misrepresentationcounter action instruments and frameworks utilizing normalized points of interaction or APIs.

Justification: Interoperability empowers durable usefulness with different instruments, utilizingtheir capacities for improved misrepresentation recognition.

8. **Compliance:**

Requirement: Comply with information insurance guidelines and consistence principlespertinent to monetary exchanges and misrepresentation identification frameworks.

Justification: Consistence guarantees legitimate adherence, mitigates chances, and keeps upwith the establishment's standing.

9. **Exactness and Heartiness:**

Requirement: Guarantee high precision in extortion location while limiting misleading up-sides and negatives to keep up with framework vigor.

Justification: High precision lessens wrong activities on authentic exchanges while actuallydistinguishing deceitful exercises.

10. **Versatility and Learning:**

Requirement: Foster components for cease less learning and transformation to new misrepresentation designs and changing exchange ways of behaving.

Justification: Versatility permits the framework to develop and stay viable against arisingextortion strategies.

PROJECT DESIGN AND ARCHITECTURE

Project Design

Designing the architecture for the "Credit Card Fraud Detection Using Machine Learning"involves several components and stages.

1. **Data Collection:**

- **Data Sources:** Accumulate Master card exchange information from differentsources like financial foundations, monetary data sets, or APIs.
- **Data Ingestion:** Foster components to ingest and gather value-based informationprogressively or clusters.

2. **Data Preprocessing and Feature Designing:**

- **Data Cleaning:** Perform information cleaning to deal with missing qualities, exceptions, and irregularities in the dataset.
- **Feature Extraction:** Separate pertinent highlights from exchange information, including exchange sum, time, area, shipper subtleties, and standards of conduct.
- **Feature Transformation:** Normalize, scale, or encode feature as expected for machine learning model input.

3. **Model Development:**

- **Choosing of ML Algorithm:** Use different machine learning like Logistic Regression, Decision Tree, XG Boost Classifier, SMOTE and Outfit Techniques.

- **Training:** Train these models using the preprocessed dataset, partitioned into preparing and validation sets.

4. Model Evaluation and Hyperparameter Tuning:

- **Cross-validation:** Use procedures like k-fold cross-validation to assess and approve model execution.

- **Hyperparameter Tuning:** Upgrade model hyperparameters utilizing techniques like matrix search or randomized search to improve model exactness.

5. Evaluation Metrics and Validation:

- **Performance Metrics:** Assess models utilizing proper measurements (precision, accuracy, recall, F1-score, ROC curve, AUC) to measure their effectiveness in fraud detection.

6. Real-time Scoring and Deployment:

- **Scoring Engine:** Foster a framework for ongoing scoring of new transaction using thetrained models.

- **Deployment:** Coordinate the chose model(s) into a creation climate for constant misrepresentation recognition in monetary frameworks.

7. Monitoring and Maintenance:

- **Performance Monitoring:** Persistently screen model execution, track measurements,and set up cautions for deviations or drops in execution.

- **Model Updates:** Plan for intermittent model updates and retraining in view of newinformation or changing misrepresentation designs.

System Architecture:

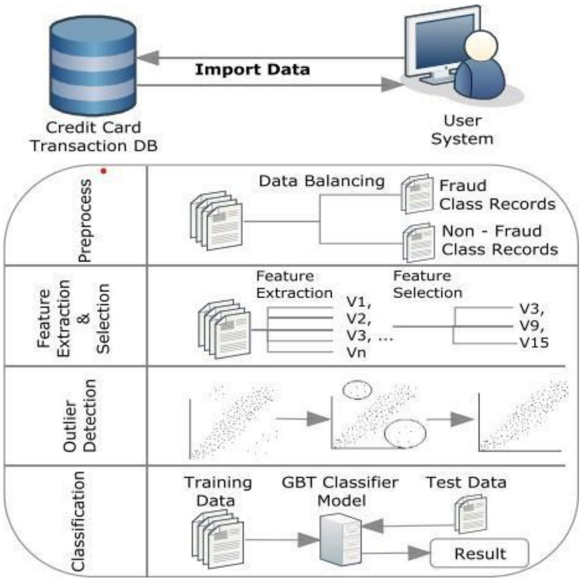


FIG.4: System Architecture

The above figure shows the process of CCFDS. This system model accepts a real time customer credit card transaction database. It is more important to find the fraud rate of credit cards.

Use Case Diagram

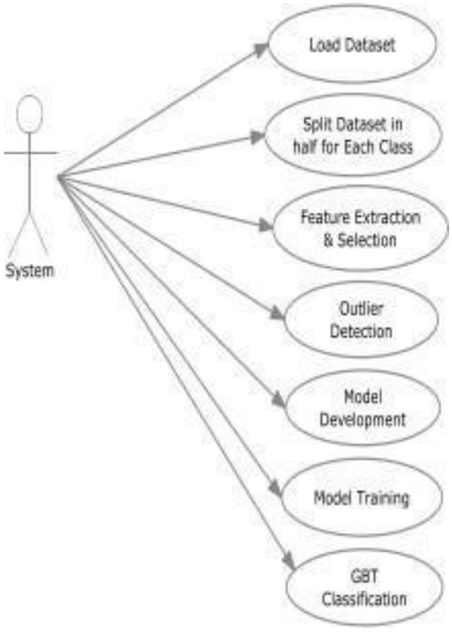


FIG.5 Use case Diagram

DFD (Data Flow Diagram)

The DFD used as communication system an user. It is simple representation of the completeproject process. Transaction detection activity follows three phrases.

1. Data Exploration
2. Data Preprocessing
3. Data Classifications

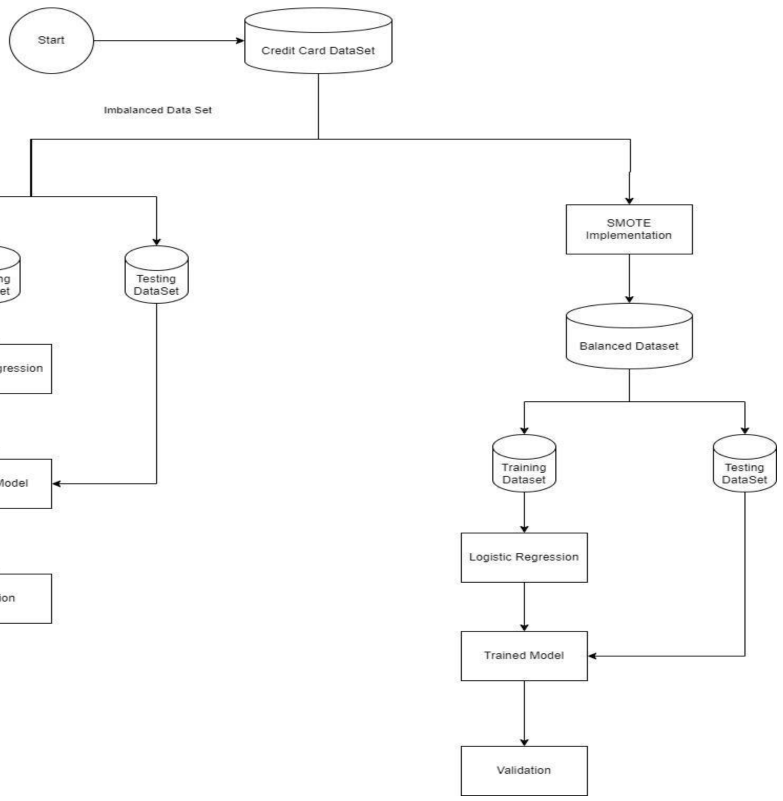


FIG.6 Data Flow Diagram

DATA PREPARATION

Below figures show the structure of the dataset where all attributes are shown, with their type, in addition to glimpse of the variables within each attribute, as shown at the end of the figure the class type is integer which needed to change to factor and identify the 0 as not- fraud to ease the process of creating the model and obtain visualizations.

IMPLEMENTATION

Tools and Technologies used:

- Google Colab
- Matplotlib
- Scikit
- Pandas

Algorithm Used:

- **Logistic Regression:** Logistic regression is a simple and widely used technique in binary distribution problems. Unlike linear regression, which predicts a continuous outcome, logistic regression is suitable for situations where the variable is categorical and has two groups. The algorithm works by predicting the probability that a given entryfalls into a particular category. It models the relationship between independent variables and the probability of a particular event occurring. The logistic regression model calculates the log difference of the probability and then converts it to the probability using the logistic function (also known as the sigmoid function). This function produces output 0 and 1, which is a list of different numbers for efficiency. Logistic regression can make it suitable for many fields such as finance, healthcare and business because it can control the relationship between different input and output distributions. Logistic Function:

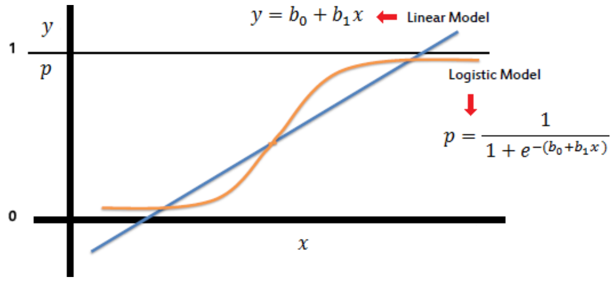


FIG. 7 Logistic Regression function

- **XG BOOST CLASSIFIER:** XG Boost classifier is a robust machine learning algorithmthat can help you understand your data and make better decisions. XGBoost is an implementation of gradient-boosting decision trees. It has been used by data scientists and researchers worldwide to optimize their machine learning models. XGBoost stands for “Extreme Gradient Boosting” and is has become one of the most popular and widely used machine learning algorithm due to its ability to handle large used machine learning algorithms due to handle large datasets and its ability to achieve state-of-the-art performance in many machine learning tasks such as classification and regression. One of the key feature of XG Boost is its efficient handling of missing values, which allowsit to handle real-world data with missing values without requiring significant pre- processing. XGBoost has built-in-support for parallel processing, making it possible to train models on large datasets in a reasonable amount of time. To understand XGBoost classifier we first need to understand the following things:

  - Decision Tree
  - Bagging
  - Random Forest
  - Boosting
  - Gradient Boosting

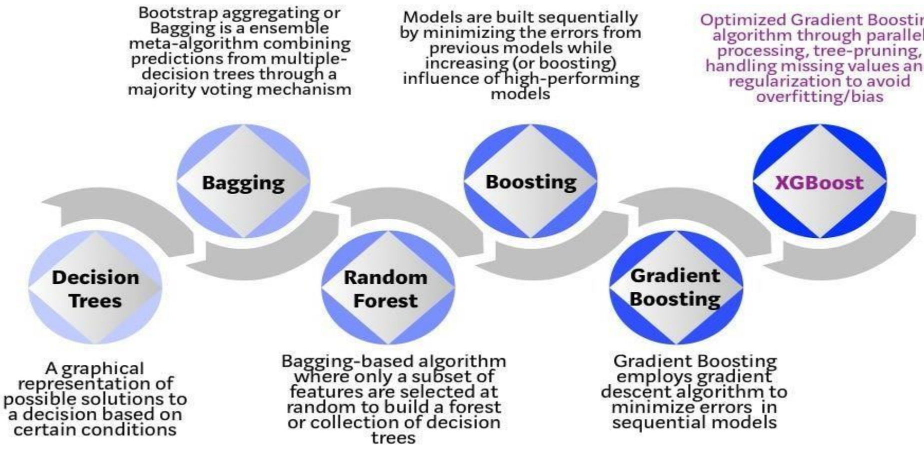
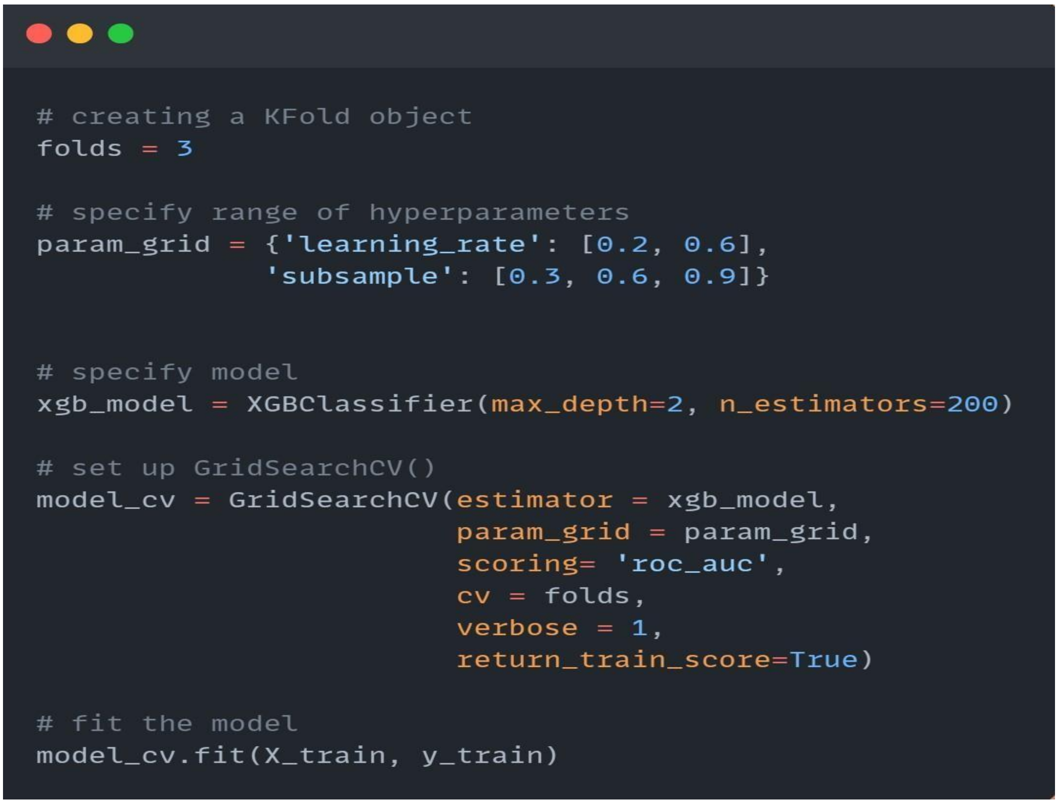

Bootstrap aggregating or Bagging is a ensemble meta-algorithm combining predictions from multiple-decision trees through a majority voting mechanism
Models are built sequentially by minimizing the errors from previous models while increasing (or boosting) influence of high-performing models
Optimized Gradient Boosting algorithm through parallel processing, tree-pruning, handling missing values and regularization to avoid overfitting/bias
Bagging
Boosting
XGBoost
Decision Trees
Random Forest
Gradient Boosting
A graphical representation of possible solutions to a decision based on certain conditions
Bagging-based algorithm where only a subset of features are selected at random to build a forest or collection of decision trees
Gradient Boosting employs gradient descent algorithm to minimize errors in sequential models

FIG.8 XG Boost diagram

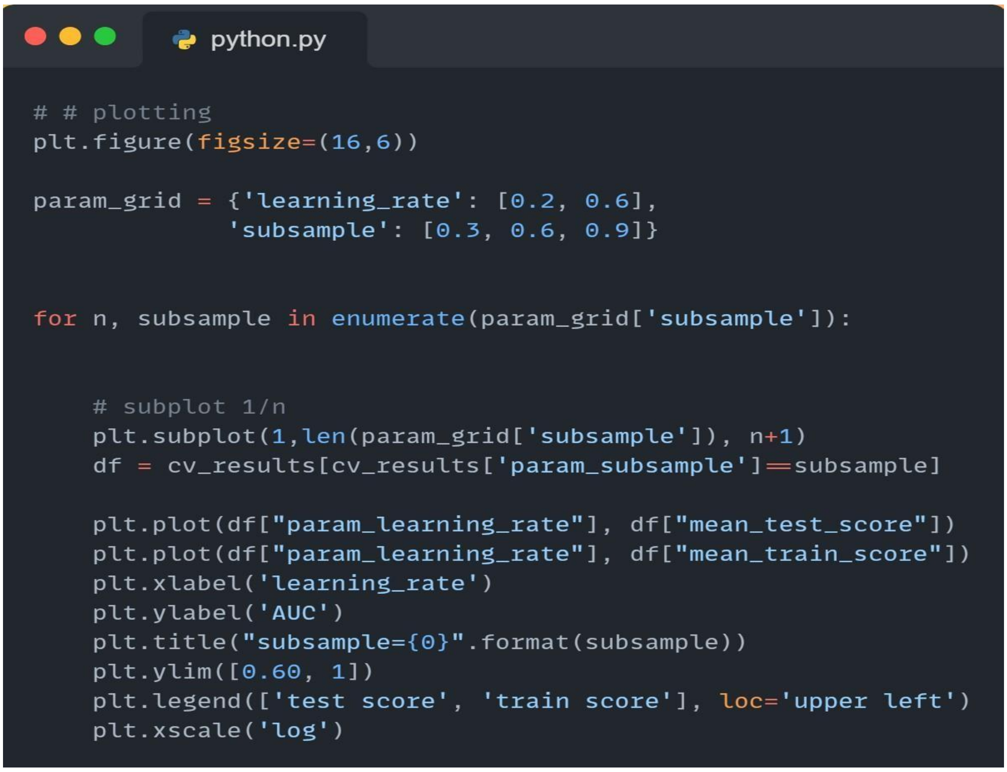
Code Snippets:



```
# creating a KFold object
folds = 3

# specify range of hyperparameters
param_grid = {'learning_rate': [0.2, 0.6],
             'subsample': [0.3, 0.6, 0.9]}


# specify model
xgb_model = XGBClassifier(max_depth=2, n_estimators=200)

# set up GridSearchCV()
model_cv = GridSearchCV(estimator = xgb_model,
                        param_grid = param_grid,
                        scoring= 'roc_auc',
                        cv = folds,
                        verbose = 1,
                        return_train_score=True)

# fit the model
model_cv.fit(X_train, y_train)
```

FIG.9 XG Boost Model Bulding

The above part of the code include the model building of the algorithm XG Boost. The certain parameters are passed in the “param_grid” i.e. learning rate and the subsample. After that the “xgb_model” initializes an XG Boost Classifier with specified parameters: “max_depth=2” and “n_estimators=200”. The “model_cv.fit(X_train, y_train)” is used for training the data. It perform cross-validate grid-search over the parameter grid defined earlier.



```python
# # plotting
plt.figure(figsize=(16,6))

param_grid = {'learning_rate': [0.2, 0.6],
             'subsample': [0.3, 0.6, 0.9]}


for n, subsample in enumerate(param_grid['subsample']):


    # subplot 1/n
    plt.subplot(1,len(param_grid['subsample']), n+1)
    df = cv_results[cv_results['param_subsample']==subsample]

    plt.plot(df["param_learning_rate"], df["mean_test_score"])
    plt.plot(df["param_learning_rate"], df["mean_train_score"])
    plt.xlabel('learning_rate')
    plt.ylabel('AUC')
    plt.title("subsample={0}".format(subsample))
    plt.ylim([0.60, 1])
    plt.legend(['test score', 'train score'], loc='upper left')
    plt.xscale('log')
```

FIG.10 Graph plotting of XG Boost

This part of code is plotting the mean test and train scores for different combination of parameters(learning_rate , subsamples) from the cross-validation results obtained from previousgridsearch. The subplot(1,len(param_grid[‘subsample’]), n+1): This creates subplot where eachsubplot corresponds to different value of subsample. ‘1’ represent the number of rows of subplot, ‘len(param_grid[‘subsample’])’ represent the number of columns of subplot, and ‘n+1’ is the current subplot index. Plt.plot(df[“param_learning_rate”], df[‘mean_test_score’]) plots the mean test score against learning_rate for the current ‘subsample’ value.

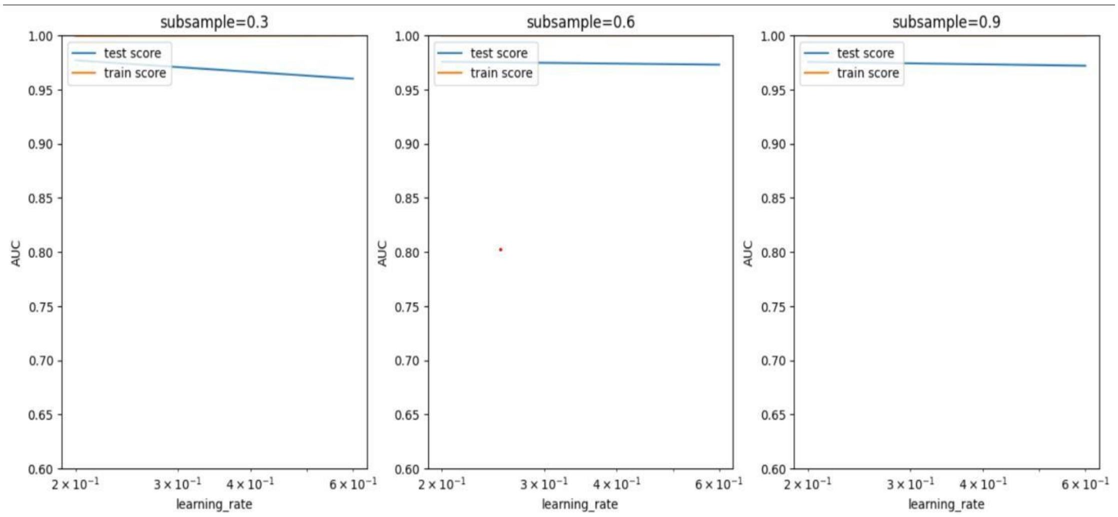
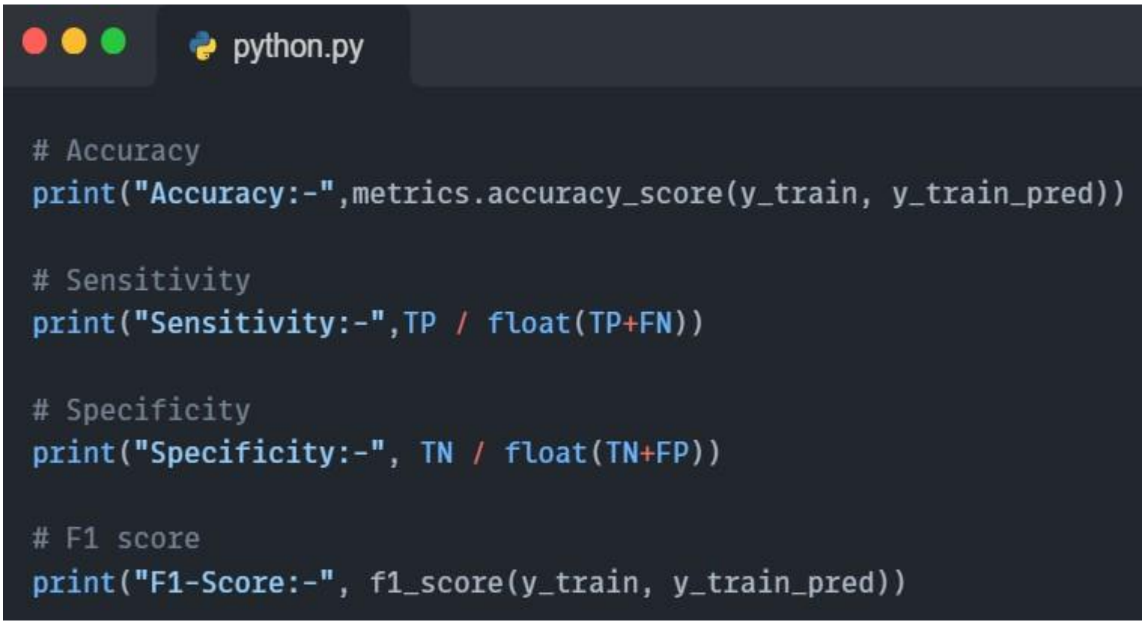


FIG.11 Plotting of the mean test and train score



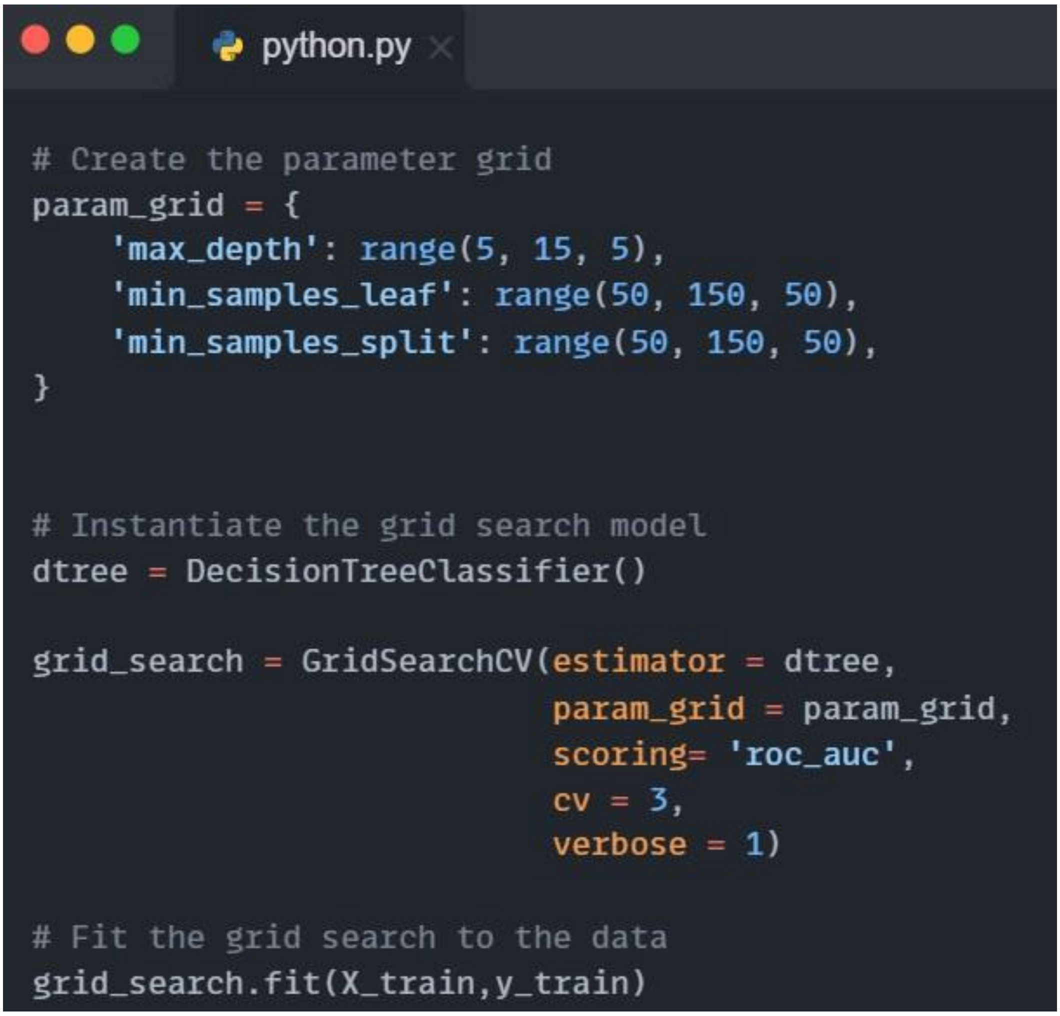
```
# Accuracy
print("Accuracy:-",metrics.accuracy_score(y_train, y_train_pred))

# Sensitivity
print("Sensitivity:-",TP / float(TP+FN))

# Specificity
print("Specificity:-", TN / float(TN+FP))

# F1 score
print("F1-Score:-", f1_score(y_train, y_train_pred))
```

FIG.12 Printing the Accuracy, Sensitivity, Specificity, F1-Score

This part of the code prints the Accuracy, Sensitivity, Specificity and F1-Score of the XGBoost.

- **Accuracy:-** Accuracy measures how often a machine learning model correctly predictsthe outcome of the model.

- **Sensitivity:-** Sensitivity measures how well a machine learning model detects positiveinstances.
- **Specificity:-** Ability to predict true negative of each category available.
- **F1-Score:-** F1-Score measures the model accuracy on the dataset .

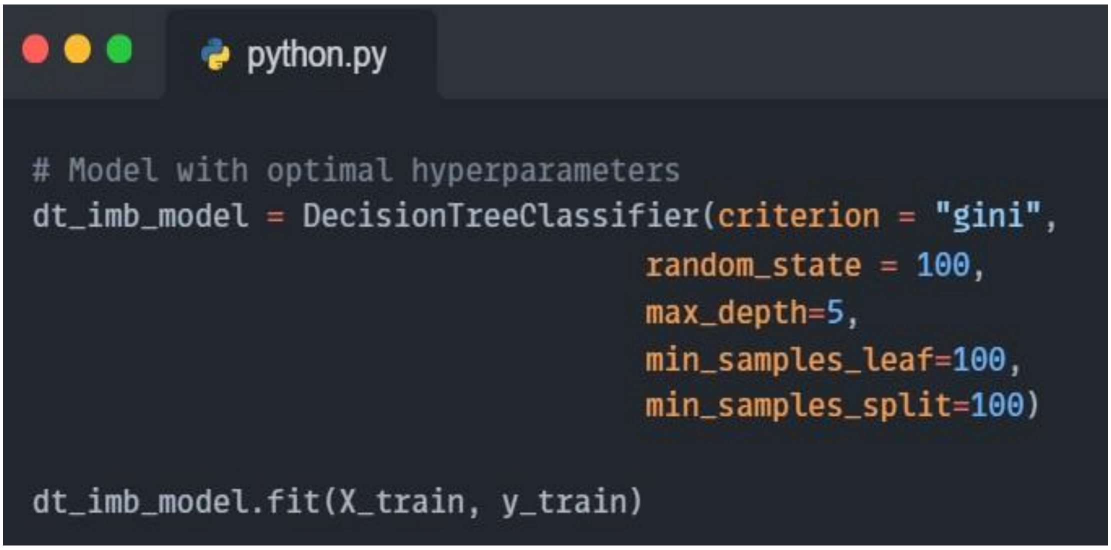


```
# Create the parameter grid
param_grid = {
    'max_depth': range(5, 15, 5),
    'min_samples_leaf': range(50, 150, 50),
    'min_samples_split': range(50, 150, 50),
}


# Instantiate the grid search model
dtree = DecisionTreeClassifier()

grid_search = GridSearchCV(estimator = dtree,
                           param_grid = param_grid,
                           scoring= 'roc_auc',
                           cv = 3,
                           verbose = 1)

# Fit the grid search to the data
grid_search.fit(X_train,y_train)
```

FIG. 13 Model building of Decision Tree

The above part of the code include the model building of the algorithm Decision Tree. The certain parameters are passed in the “param_grid” i.e. max_rate, min_sample_leaf and the min_samples_split. After that the “dtree” initializes an Decision Tree Classifier with specified parameters: “cv = 3” , “estimators=dtree”, scoring = ‘roc_auc’ and verbose = 1.

The “grid_search.fit(X_train, y_train)” is used for training the data. It perform cross-validate grid-search over the parameter grid defined earlier.

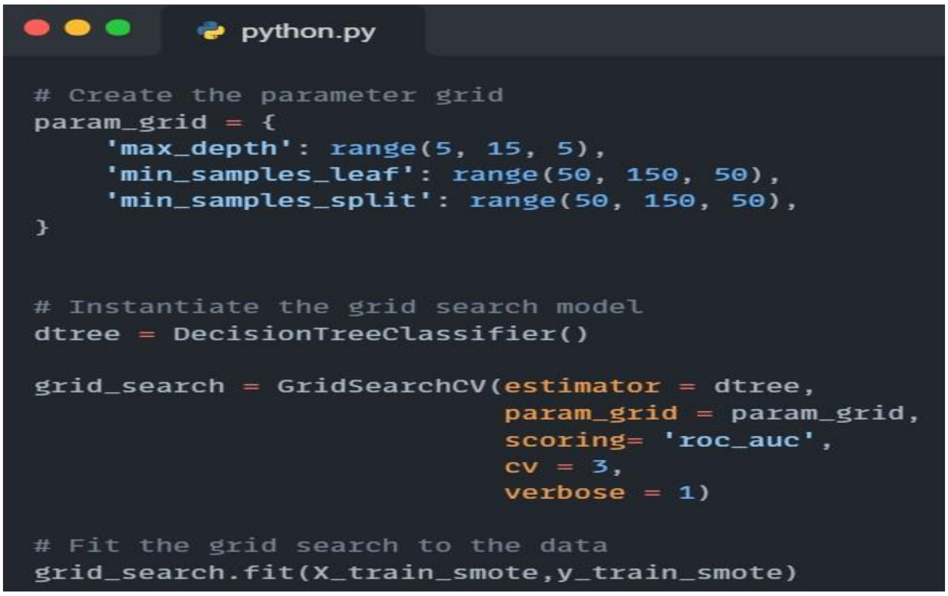


```python
# Model with optimal hyperparameters
dt_imb_model = DecisionTreeClassifier(criterion = "gini",
                                      random_state = 100,
                                      max_depth=5,
                                      min_samples_leaf=100,
                                      min_samples_split=100)

dt_imb_model.fit(X_train, y_train)
```

FIG.14 Creating a Decision Tree Classifier

This part of code is for creating a Decision Tree Classifier of Decision Tree model. The first line of code initialize the decision tree classifier object with different hyperparameters (criterion = “gini”, random_state = 100, max_depth = 5, min_sample_leaf = 100, min_sample_split = 100), these are the different hyperparameters used to specify splitting, sets of random seed for reproductibility, maximum depth, minimum number of samples for leaf node, minimum number of samples required to split an internal node.

After this code segment is executed, ‘dt_imb_model’ will be trained decision tree classifier model with the specified hyperparameter. It can be used to make predictions on new data or evaluate its performance on unseen data.

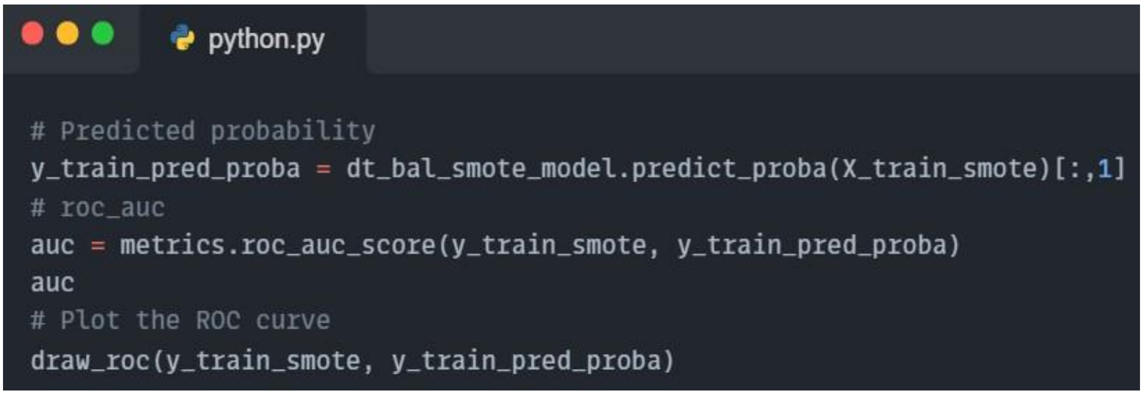


```python
# Create the parameter grid
param_grid = {
    'max_depth': range(5, 15, 5),
    'min_samples_leaf': range(50, 150, 50),
    'min_samples_split': range(50, 150, 50),
}


# Instantiate the grid search model
dtree = DecisionTreeClassifier()

grid_search = GridSearchCV(estimator = dtree,
                           param_grid = param_grid,
                           scoring= 'roc_auc',
                           cv = 3,
                           verbose = 1)

# Fit the grid search to the data
grid_search.fit(X_train_smote,y_train_smote)
```

Fig. 15 Model building of Decision Tree after balancing data

This part of code is for building the decision tree model after balancing the dataset using SMOTE technique. We need to balance the dataset because the dataset is so unbalanced the non-fraud transactions are too much as compared to fraud transactions.

In the above code the ‘param-grid’ defines the grid of hyperparameters to sraech. It includes ranges for ‘max_depth’, ‘min_samples_leaf’, and ‘min_samples_split’. The ‘dtree’ initializes a decision tree classifier object without specifying any hyperparameters. At the end of the code the ‘grid_search’ will continue the result of the hyperparameter tuning, including the best hyperparameters found and the performance metrics.



```python
# Predicted probability
y_train_pred_proba = dt_bal_smote_model.predict_proba(X_train_smote)[:,1]
# roc_auc
auc = metrics.roc_auc_score(y_train_smote, y_train_pred_proba)
auc
# Plot the ROC curve
draw_roc(y_train_smote, y_train_pred_proba)
```

Fig. 16 Roc curve plotting

This code deals with the computing of ROC_AUC(Receiver Operating Characteristics Area Under the Curve) score and plotting the ROC curve for a decision tree model.

The ‘y_train_pred_proba = dt_bal_smote_model.predict_proba(X_train_smote)[:,1]’ , this line predicts probability estimates for the positive class using the decision tree model on training data. ‘predict_proba’ returns an array where each row corresponds to a sample and each columncorresponds to a class. ‘[:,1]’ selects the probabilities of the positive class. The computing of ROC AUC score using the ground truth labels and the predicted probabilities of the positive class. ROC AUC is a performance metric the evaluates the model’s ability to distinguish between classes, with higher values indicating better performance.

Overall this code is evaluating the performance of the decision tree model on the training databy computing the ROC AUC score and visualizing the ROC curve.

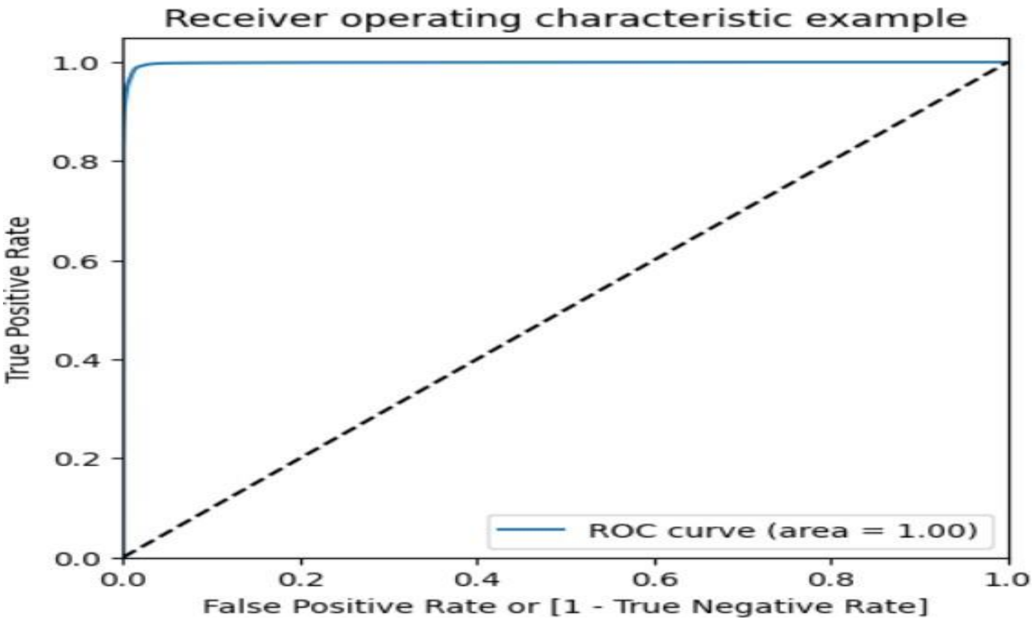


FIG. 17 ROC curve of decision tree

KEY CHALLENGES

There are many challenges or areas that need to be improved or considered in terms of creditcard fraud detection using machine learning models:

- **Imbalanced Datasets:** Handling unbalanced information is a major problem in fraud detection. The number of fraud-free transactions often far exceeds the number of fraudulent transactions. When learning about inconsistent data, some models may underperform, leading to biased predictions that favor most classes.

- **Measurements:** Although accuracy is often used as a measure of accuracy, it may notbe best for non-equivalent data. Precision, recall, and F1 score are important metrics as they provide a better understanding of the model's performance, especially when dealingwith heterogeneous classes.

- **Model selection and tuning:** This code uses a large amount of machine learning without the need for detailed hyperparameter tuning or optimization. Tuning hyperparameters can affect model performance. Additionally, the model selection may not be complete and other models or combinations may be better.

- **Feature Engineering:** Feature selection and engineering plays an important role in the performance of machine learning models. There may be room in the code to search for and select more important features or create new features that can increase the predictivepower of the model.

- **Cross validation:** Code does not compete, an important step to ensure the strength of the model. The model is reliable and does not have too much or too little information. Cross-validation helps predict how well the model fits new, unseen products.

- **Balancing Dataset:** The Dataset is highly unbalanced, so it is important to balance the dataset using SMOTE technique so that the data can be balanced.

- **Computational efficiency:** Some models, especially complex models such as clustering or XGBoost, can be computationally demanding or time consuming, especially for large data sets. Optimizing code for computer efficiency can be difficult.

Solving these problems requires a deep understanding of the data, careful selection of samples, sound evaluation strategies, and sample refinement. It improves their performance and general abilities.

CHAPTER-04

TESTING

TESTING STRATEGY

Credit card fraud project using machine learning (ML) consists of several important steps. First, given the disparity of fraud detection data (which is mostly in the legitimate market), it is important to classify data using methods such as stratified sampling to ensure that both fraud and scams are included in training and testing systems. Cross-validation techniques such as Logistic Regression can be used during model training to ensure robustness and evaluate how well the model fits unseen things. Because of inequality in the classroom, assessment should not focus solely on accuracy, Precision, recall, F1 score, and area under the ROC curve (AUC-ROC) are important parameters to evaluate the performance of the model. Additionally, techniques such as oversampling (SMOTE) or more advanced methods (such as Generative Adversarial Networks - GANs) can be used to solve the problem of under classification in data structuring, thus improving the model's ability to accurately identify fraudulent transactions. In addition to constantly updating and renewing models based on new information, it is also useful to constantly monitor and improve based on changes in fraud patterns and changes in fraudsters' application methods, and to test andapply techniques for using machine learning for credit card fraud.

The testing strategy used in the project involves several steps assess the performance of various machine learning models for credit card fraud detection:

- **Data Loading and Exploration:** The code starts by loading the dataset (creditcard.csv) using Pandas and performing initial exploratory data analysis. It checks the class distribution between fraud and non-fraud cases.

- **Data Preparation:** The data is divided into features(X) and target variable(y) columns. Then, it splits the dataset into training and testing sets using the train_test_split functionfrom sklearn.model_selection.

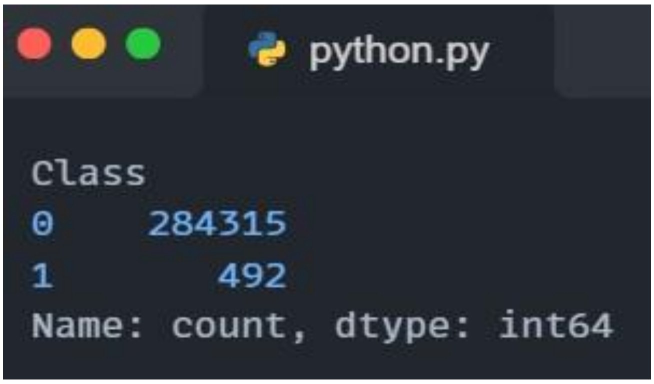
- **Model Building and Evaluation:** Several machine learning models are trained and tested on the data. There are various classifiers or algorithms used in the model like Logistic Regression (LR), Decision and XG Boost Classifier. For each model, it performs the following:
    1. Fits the model on the training data.
    2. Makes predictions using the trained model on the test data.
    3. Evaluate model performance using accuracy and precision scores and creates a confusion matrix to visualize true positives, true negatives, false positives, and false negatives.

- **Accuracy Comparison:** Finally, the code generates a comparison plot using Seaborn, displaying the accuracy of different models (Logistic Regression, KNN, Random Forest, Decision Tree, SVC, XGBoost) to visualize and compare their performances.

- **Data Balancing:** Since, the data is so unbalanced with a high rate of non-fraud transactions so we need to balance the data using SMOTE data balancing technique.

Overall, this strategy involves training multiple classification models and evaluating their performances using accuracy and precision scores along with confusion matrices to determine the effectiveness of each model in detecting credit card fraud. There all comparison story provides an overview of the comparison between these models.

TEST CASES AND OUTCOMES



```
Class
0    284315
1       492
Name: count, dtype: int64
```

FIG.18 Fraud and Not-Fraud Transactions

The above figure shows that the number of Not-Fraud transactions are 284315 and the number of Fraud transactions are 492. ‘Class 0’ shows the number of Not-Fraud transactions and ‘Class 1’ shows number of fraud transactions.

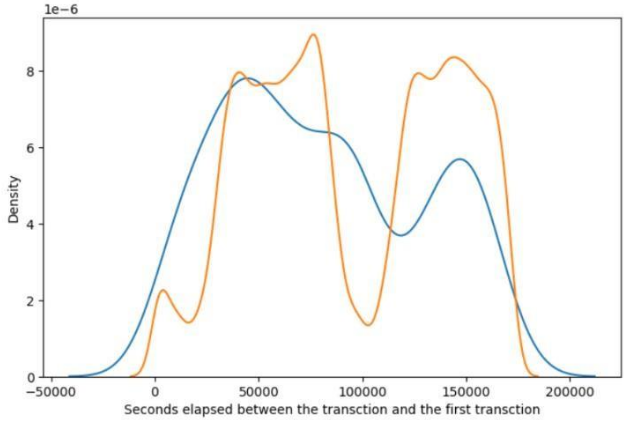


FIG.19 Density Time graph

The above graph is the distribution of class with time. Here the blue line shows that the transaction is fraudulent and the orange shows that the transactions are not-fraudulent. From the graph we can see that we do not get any specific pattern for the fraudulent and not-fraudulent with respect Time.

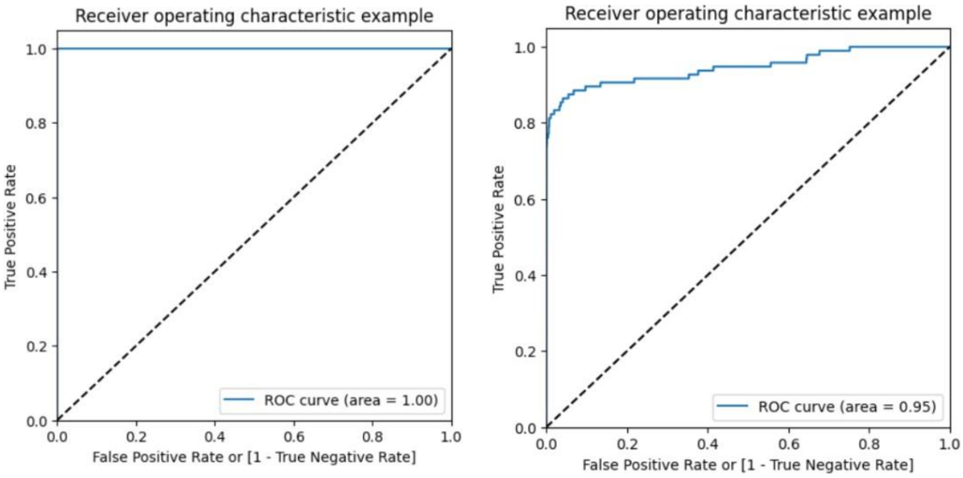


FIG.20 Train case of XG Boost **FIG.21 Test case of XG Boost**

The above figures are the test case and train case ROC curve of the XG Boost Classifiers for balanced dataset. In train case the ROC curve is straight which shows that the model is perfect and suits best for the fraud detection. On the other hand, the test case ROC curve goes nearly straight up the left side of the graph and then levels out at the top. This indicates a good model with area of 0.95.

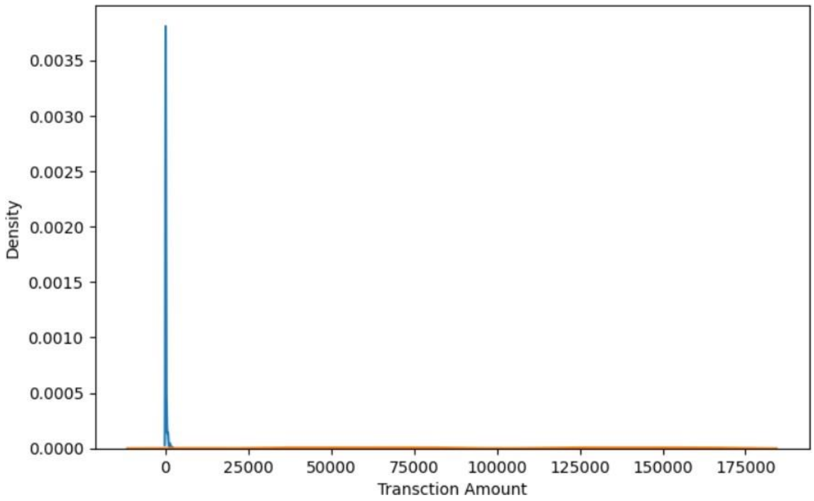


FIG.22 Density Time graph

The above graph is the distribution of class with respect to the amount. We can see that the fraudulent transaction are mostly densed in the lower range of amount, whereas the not-fraudulent transactions are spreaded throughout low to high range of amount.

CHAPTER-05

RESULTS AND EVALUATION

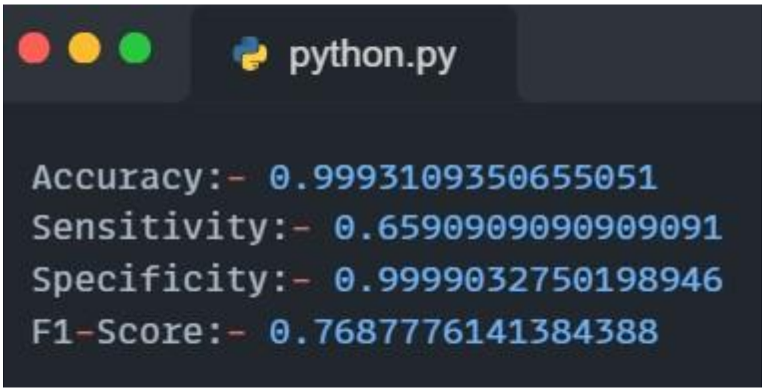
RESULTS

The project is provided using various machine learning models for analysis to verify credit card transactions. It trains various classifiers such as logistic regression, decision tree, XG Boost and compares their accuracy. After running the code, it will display the accuracy scoresand ROC curve of different learning models used in the fraud detection task, along with a comparison table showing their performance.

Results for Balanced and Unbalanced data of different

classifiers:For Unbalanced Datasets:

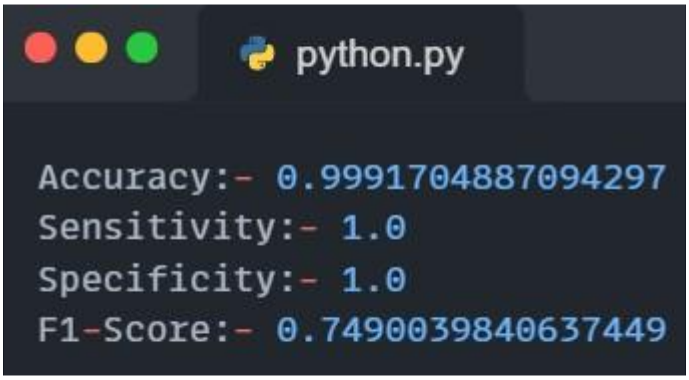
- **Logistic Regression:** The accuracy using model in logistic regression is 0.99931, sensitivity is 0.65909, specificity is 0.999903 and F1- Score is 0.76877 .



```
Accuracy:- 0.9993109350655051
Sensitivity:- 0.6590909090909091
Specificity:- 0.9999032750198946
F1-Score:- 0.7687776141384388
```

FIG.23 Result of LR

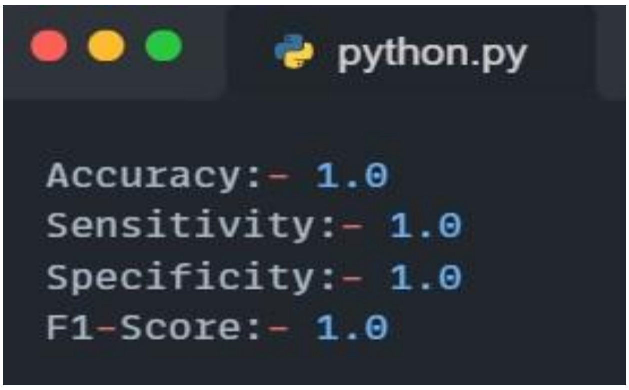
- **Decision Tree Classifiers:** The accuracy using model in Decision Tree Classifiers is0.99917, Sensitivity is 1.0, Specificity is 1.0, F1-Score is 0.749003.



```
Accuracy:- 0.9991704887094297
Sensitivity:- 1.0
Specificity:- 1.0
F1-Score:- 0.7490039840637449
```

FIG.24 Result of Decision Tree

- **XG Boost Classifier:** The accuracy using model in XG Boost Classifier is 1.0, Sensitivity is 1.0, Specificity is 1.0 and F1-Score is 1.0.



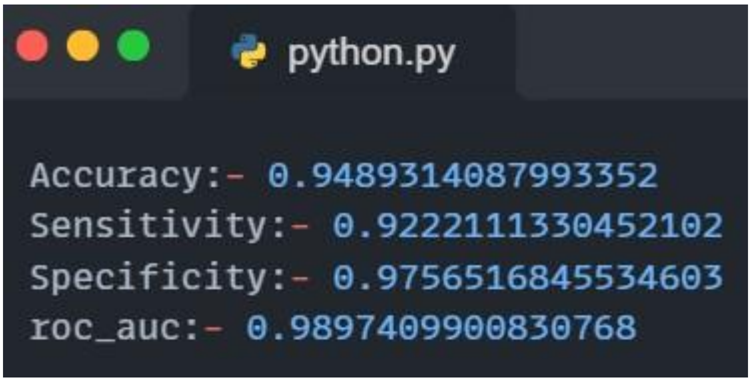
```
Accuracy:- 1.0
Sensitivity:- 1.0
Specificity:- 1.0
F1-Score:- 1.0
```

FIG.25 XG Boost Result

After performing all the model on different classifiers and matching the accuracy and other parameters of all the classifiers we came to know that the XG Boost Classifier seems to perform best as compared to all the other models, with a accuracy score of 1.0**,** Sensitivity is 1.0, Specificity is 1.0 and F1-Score is 1.0 **.** It means that the XG Boost handles imbalance dataset more accurate then other classifiers. XGBoost uses regularization technique to avoid overfitting, making it more robust and less prone to overfitting the training data compared to other models. XG Boost uses gradient boosting, which minimizes errors by optimizing the lossfunction, leading to improved predictive performance.

For Balanced Dataset:

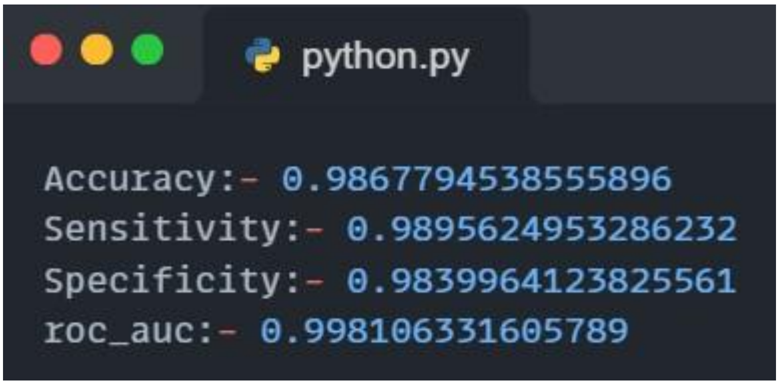
- **Logestic Regression:-** After balancing the dataset the accuracy score is 0.948931,Sensitivity is 0.92221, Specificity is 0.97565 and ROC score is 0.9897.



```
Accuracy:- 0.9489314087993352
Sensitivity:- 0.9222111330452102
Specificity:- 0.9756516845534603
roc_auc:- 0.9897409900830768
```

FIG.26 Result of LR after balancing

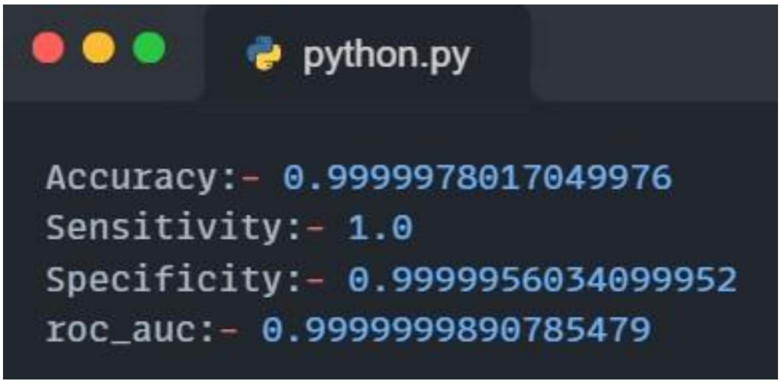
- **Decision Tree Classifier:-** After balancing the dataset the accuracy score is 0.98677,Sensitivity is 0.98956, Specificity is 0.9839964, ROC score is 0.99810.



```
Accuracy:- 0.9867794538555896
Sensitivity:- 0.9895624953286232
Specificity:- 0.9839964123825561
roc_auc:- 0.998106331605789
```

FIG.27 Result of Decision Tree after balancing

- **XG Boost Classifier:-** After balancing the dataset the accuracy is 0.9999 , Sensitivityis 1.0, Specificity is 0.9999 , ROC score is 0.999999.



```
Accuracy:- 0.9999978017049976
Sensitivity:- 1.0
Specificity:- 0.9999956034099952
roc_auc:- 0.9999999890785479
```

FIG.28 Result of XG Boost after balancing

After performing all the model on different classifiers and matching the accuracy and other parameters of all the classifiers on balanced dataset using SMOTE we came to know that the XG Boost Classifier seems to perform best as compared to all the other models, with a accuracy score of 0.99999**,** Sensitivity is 1.0, Specificity is 0.99999 and ROC score is 0.9999**.** It means that the XG Boost handles balanced dataset more accurate then other classifiers.

EVALUATION

From the evaluation of the model we came to know that for unbalanced dataset 99.82 are not fraud and 0.172749 are fraud. Out of all the cases run during the model execution the not fraudcases are 284315 and the fraud cases are 492.

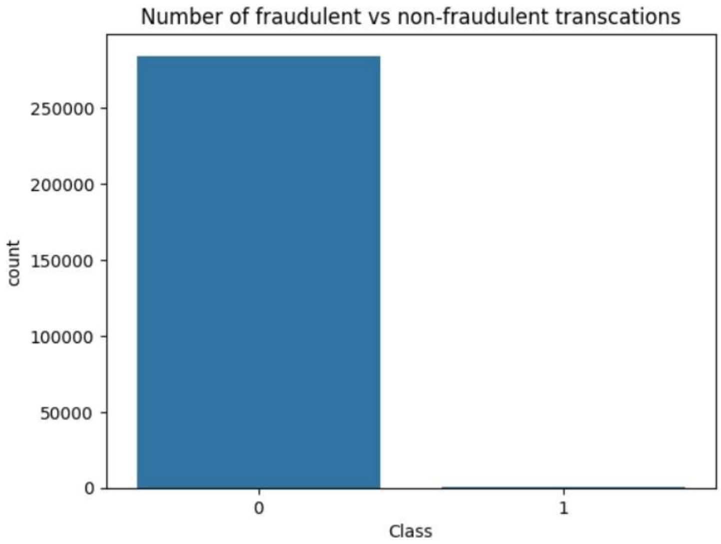


FIG.29 Graph of Fraud vs Not fraud

The above figure shows the graph between Fraud and Not Fraud transactions. After examining the graph we came to know that the number of not fraud transactions as too much as compared to fraud transactions.

After examining the model we came to know that the XG Boost Classifier works more relevantly are compared to other classifiers.

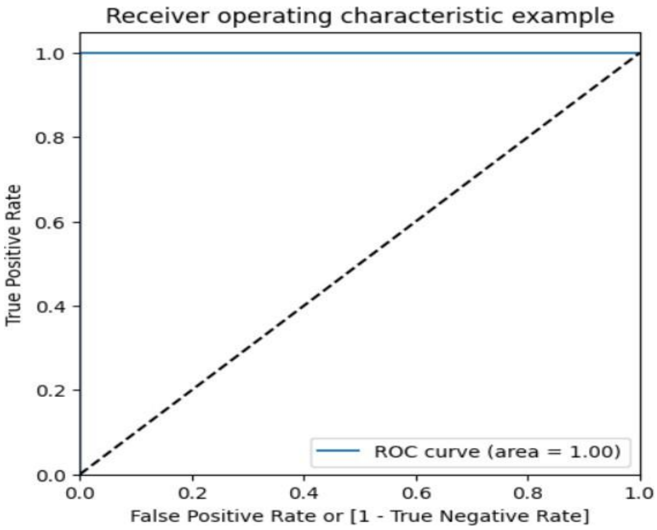


FIG.30 ROC curve of XG Boost classifier

In the ROC curve, shows a perfect classification scenario. This curve goes straight up the graph and then levels out at the top. This indicates very good model, with an AUC of 1.0. An AUC of signifies a perfect model that can flawlessly distinguish between fraud and not fraud transations.

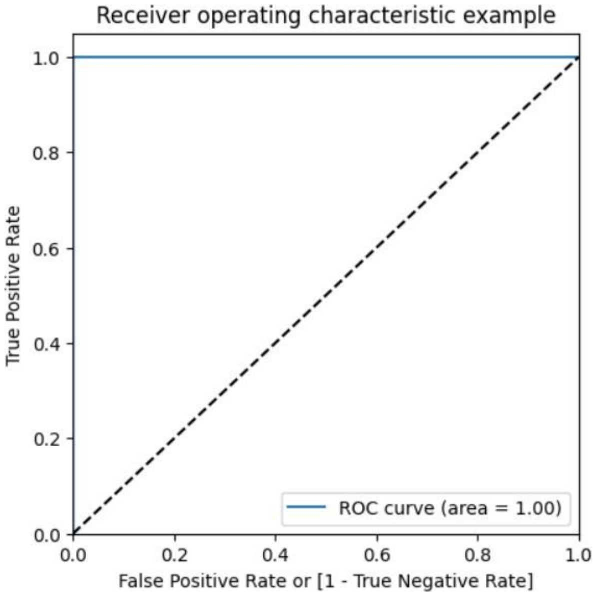


FIG. 31 ROC Curve for XG Boost balanced dataset

About balancing the dataset, the area under ROC curve indicates the overall performance of the model. The curve shows that the area under the curve is almost 0.999 which is almost equals 1.0which is considered to be a very good score.

CHAPTER-06

CONCLUSIONS AND FUTURE SCOPE

CONCLUSION

The conclusion is that the XG Boost classifier performs best in terms of accuracy and other parameters including ROC score on credit card identification test text.

- XG Boost has the highest ROC score of 1.0, indicating that it accurately identifies themajority of fraudulent transactions while maintaining low cost.
- After balancing the dataset the ROC score of 0.9999, which shows that the modelperform quite perfect as compared to the other classifiers.
- After the XG Boost, Decision Tree Classifier work will on balanced dataset with a rocscore of 0.998.
- If the priority is to reduce negativity (misclassification is not fraud), a more accuratemodel such as XG Boost and Decision Tree will be preferred.

Consequently, considering the balance of accuracy and other parameters, it is recommendedto use the XG Boost classifier as it performs best in testing card detection withdrawalpatterns. In summary, although XG Boost showed the best performance in the test model, the most suitable model should be selected according to the specific needs, calculation needs and multi-purpose credit card fraud. Additional fine-tuning and rigorous testing is recommended before deploying the prototype in a production environment.

REFERANCE

[1] Altman, E.I., Marco, G., & Varetto, F.Corporate distress diagnosis comparisons using linear discriminant analysis and neural networks. Journal of Banking and Finance, 18(3), 505–529, 1994.

[2] Chen J. Development and Application of Intelligent Transaction Scoring Model of the fraud risk. Credit card in China. 2006.

[3] Dorronsoro, Ginel, Sgnchez and Cruz. Neural fraud detection in credit card operations.Neural Networks, IEEE Transactions. Volume: 8, Issue: 4: 827-834, 1997.

[4] Flitman A.M. Towards analysing student failures: neural networks compared with regression analysis and multiple discriminant analysis. Computers & Operations Research,Volume 24, Issue 4, 367-377, 1997.

[5] Ghosh, S.Reilly, D.L. Credit card fraud detection with a neural-network. Decision Support and Knowledge-Based Systems, Proceedings of the Twenty-Seventh Hawaii International Conference.Volume 3: 621-630, 1994.

[6] Hanagandi, V.Dhar, A.Buescher, K.Density-based clustering and radial basis function modeling to generate credit card fraud scores. Computational Intelligence for Financial Engineering, 1996.

[7] Hansen, J.V., McDonald, J.B., Messier, W.F., & Bell, T.B.A generalized qualitative- response model and the analysis of management fraud. Management Science, 42(7),
1022-1032, 1996.

[8] Martin, D.Early warning of bank failure: A logestic regression approach. Journal ofBanking and Finance, 1, 249–276, 1997.

[9] Ohlson, J.A. Financial ratios and probabilistic prediction of bank ruptcy. Journalof Accounting Research, 18(1), 109–131, 1980.

[10] QuinlanJ. R.C 4.5 Programs for Machine Learning, Morgan Kaufmann, SanMateo, CA.1993.

[11] Quinlan J.R. Introduction to decision trees. Mach Learning;1(1):81–106, 1986.

[12] Rumel hart D.E. McClell and J.L. Parallel Distributed Processing. Experiments inthe Microstructure of Cognition, MIT Press, Cambridge, MA, 1986.

[13] Raj, S.B.E., & Portia, A.A. (2011,March). Analysis on credit card fraud detectionmethods.

[14] In 2011 International Conference on Computer, Communication andElectrical Technology (ICCCET) (pp. 152-156). IEEE.

[15] Shen, A., Tong, R., & Deng, Y. (2007, June). Application of classification models on credit card fraud detection. In 2007 International conference on service systems and servicemanagement (pp. 1-4). IEEE.

[16] Awoyemi,J.O.,Adetunmbi,A.O.,&Oluwadare,S.A.(2017,October).Creditcard frauddetection using machine learning techniques: A comparative analysis. In 2017 International Conference on Computing Networking and Informatics (ICCNI) (pp. 1-9). IEEE.

[17] Xuan, S., Liu, G., Li, Z., Zheng, L., Wang, S., &Jiang, C.(2018,March). Random forest for credit card fraud detection. In 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC) (pp. 1-6). IEEE.

[18] John, H., &Naaz, S.(2019). Credit card fraud detection using local outlier factor andisolation forest. Int. J. Comput. Sci. Eng, 7(4), 1060-1064

[19] Iwasokun GB, Omomule TG, Akinyede RO. Encryption and tokenization-based system for credit card information security. Int J Cyber Sec Digital Forensics. 2018;7(3):283–93.