ONLINE TRANSICTION FRAUD DETECTION SYSTEM USING PYTHON AND MACHINE LEARNING MODEL

SANKALP
Ministry of Skill Development
& Entrepreneurship

FITT

MPSSDEGB
कौशल से रोजगार, समृद्धि अपार

Group members :- Atul Kawre , Himanshu Kharole, Pragya Rahangdale, Sanjana Kushwaha, Yash Verma

ABSTRACT

When it comes to the simplicity of making a payment while sitting anywhere in the world, online payments have been a source of attractiveness. Over the past few decades, there has been an increase in online payments. E-payments enable businesses earn a lot of money in addition to consumers. However, because electronic payments are so simple, there is also a risk of fraud associated with them. A consumer must ensure that the payment he is paying is going exclusively to the appropriate service provider. Online fraud exposes users to the possibility of their data being compromised, as well as the inconvenience of having to report the fraud, block their payment method, and other things. When businesses are involved, it causes some issues; occasionally, they must issue refunds in order to keep customers. Therefore, it is crucial that both consumers and businesses are aware of these internet scams. A model to determine if an online payment is fraudulent or not is put forth in this study. To determine if a certain Online payment is fraudulent or not, some features like the type of payment, the recipient's identity, etc. would be taken into account.

CONTENTS

CHAPTER 1:
INTRODUCTION

1.1 Introduction

For organizations, financial institutions, and people everywhere, detecting and preventing fraud in financial transactions is a top priority. The need to investigate more sophisticated techniques has arisen as sophisticated fraud has made clear the limitations of conventional rule-based systems. This study explores how real-time monitoring systems and machine learning algorithms can be used to improve financial transaction fraud detection and prevention capabilities.

In the literature, the importance of fraud prevention and detection in financial transactions has been extensively discussed. In addition to causing significant financial losses, financial fraud also erodes public faith in the financial system (Association of Certified Fraud Examiners, 2020). Traditional rule-based systems look for suspected fraudulent actions using predetermined rules and patterns. But these systems struggle to adjust to new and developing fraud strategies, which results in many false negatives and potential financial losses (Kumar et al., 2020). The use of machine learning algorithms has drawn a lot of interest as a solution to these restrictions.

Large volumes of transactional data can be automatically mined for patterns and abnormalities using machine learning algorithms, leading to more precise and adaptable fraud detection. Financial institutions can examine past transactional data to find trends linked to fraudulent actions by utilizing machine learning techniques like supervised learning, unsupervised learning, and deep learning (Dal Pozzolo et al., 2015). Additionally, by continuously monitoring transactions in real-time and sending out notifications for suspected fraud, the integration of real-time monitoring systems improves fraud detection (Bolton et al., 2011). With timely action made possible by this proactive strategy, potential losses and damages are reduced.

The necessity for a more effective and efficient strategy to counteract changing fraud strategies is what motivates the use of machine learning algorithms and real-time monitoring systems. Financial fraud is dynamic, necessitating the use of adaptable systems that can recognize emerging trends and abnormalities. Detecting complex and changing fraud patterns is made possible by machine learning algorithms, allowing for early identification and prevention (Phua et al., 2010). In addition to machine learning, real-time monitoring systems offer fast response capabilities, enabling prompt intervention to stop fraudulent transactions (Kou et al., 2020).

1.2 Research Objectives

1. Investigate the use of machine learning algorithms for fraud detection in financial transactions.

2. Design and develop a real-time monitoring system for continuous fraud detection and prevention.
3. Assessing the performance of the suggested approach in comparison to conventional rule-based systems.
4. Exploring proactive measures for fraud prevention, such as dynamic risk scoring and adaptive thresholds.
5. Analyse scalability and deployment considerations for implementing the proposed system in real – world financial institutions.

1.3 Proposed system

In proposed system we use RFA for classification and regression of dataset. First, we will collect the dataset and analysis will be done on the collected dataset. After the analysis of dataset then cleaning of dataset is required. Generally, in any dataset there will be many duplicate and null values will be present, so to remove all those duplicate and null values cleaning process is required. Then we must split the dataset into two categories as trained dataset and testing dataset for comparing and analyzing the dataset. After dividing the dataset, we must apply the RFA where this algorithm will give us the better accuracy about the credit card fraud transactions. By applying the RFA, the dataset will be classified into four categories which will be obtained in the form of confusion matrix. Based on the above classification of data performance analysis will be done. In this analysis the accuracy of credit card fraud transactions can be obtained which will be finally represented in the form of graphical representation.

a) RFA

Random forest is also called as random decision forest which is used for classification, regression and other tasks that are performed by constructing multiple decision trees. This RFA is based on supervised learning and the major advantage of this algorithm is that it can be used for both classification and regression. RFA gives you better accuracy when compared with all other existing systems and this is most used algorithm. In this paper the use of RFA in fraud detection can give you accuracy of about 90 to 95%.

b) RFA implementation in online fraud detection

In fraud detection, the RFA gives better accuracy in results. First all the dataset will be collected and analyzed. During analysis process all the duplicate values and also the null values will be removed from the dataset. Now the dataset will be preprocessed based on the amount and transaction time for finding the accuracy of the resultant dataset. After the preprocessing of dataset into amount and transaction time now the dataset will be divided into two categories. The dataset is classified in two categories as trained data and test dataset. Here for dataset classification we use a software called „Scikit-learn‟. Scikit-learn is a free software for machine learning library in python where it contains features like classification, regression, clustering algorithms and various algorithms to interoperate with Python. After the preprocessing of the dataset now we apply the RFA. By applying RFA the preprocessed dataset will be analyzed again and then a confusion matrix will be obtained. In confusion matrix the dataset will be partitioned into four blocks as True Positive(TP), True Negative(TN), False Positive(FP) and False Negative(FN). Now the dataset will be partitioned continuously until all the data is validated. Now all these partitioned data will be evaluated and finally it will be represented as separate graphs. These separate graphs will give only less accuracy about the resultant dataset. So, in order to obtain better accuracy, we use RFA where it takes all the graph values and give us only necessary values with better accuracy when compared with all other algorithms.

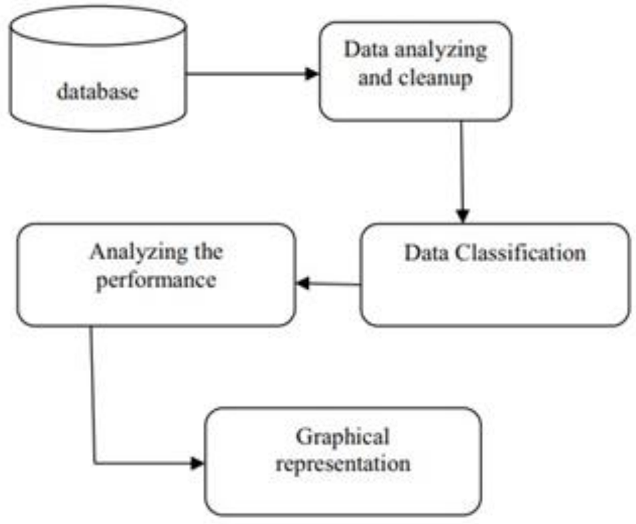


Fig. 1: System architecture.

In our architecture first we have a dataset where this contains all the details But here we take only Amount and Transaction time for analysis and preprocessing of dataset. The next step is the process of data cleaning where the dataset will be analyzed, and all the duplicate and null values will be eliminated from the dataset taken. The next step is the data partition where the dataset will be partitioned into two partitions as trained dataset and testing dataset. After that RFA will be applied and a confusion matrix will be obtained. Now the performance analysis will be done on the obtained confusion matrix. This Performance analysis will give the accuracy of about 90% in this online fraud detection system

1.4 Software Requirements Specification

1 Python

Python brings an exceptional amount of power and versatility to machine learning environments. The language's simple syntax simplifies data validation and streamlines the scraping, processing, refining, cleaning, arranging and analyzing processes,

2. NumPy

NumPy is a popular Python library for multi-dimensional array and matrix processing because it can be used to perform a great variety of mathematical operations. Its capability to handle linear algebra, Fourier transform, and more, makes NumPy ideal for machine learning and artificial intelligence (AI) projects, allowing users to manipulate the matrix to easily improve machine learning performance. NumPy is faster and easier to use than most other Python libraries.

3. Scikit-learn

Scikit-learn is a very popular machine learning library that is built on NumPy and SciPy. It supports most of the classic supervised and unsupervised learning algorithms, and it can also be used for data mining, modeling, and analysis. Scikit-learn's simple design offers a user-friendly library for those new to machine learning.

4. Pandas

Pandas is another Python library that is built on top of NumPy, responsible for preparing high-level data sets for machine learning and training. It relies on two types of data structures, one-dimensional (series) and two-dimensional (DataFrame). This allows Pandas to be applicable in a variety of industries including finance, engineering, and statistics. Unlike the slow-moving animals themselves, the Pandas library is quick, compliant, and flexible.

CHAPTER2:
Literature Review

In recent years, there has been a lot of study on applying machine learning algorithms to detect fraud in financial transactions. Various strategies and algorithms have been examined in several research to increase the precision and effectiveness of fraud detection systems. This section reviews earlier studies and research articles in the field, addressing the benefits and drawbacks of various strategies while identifying the gaps in the body of knowledge that the current study seeks to fill.

2.1 Supervised Learning Approaches

A fraud detection system based on logistic regression was proposed by Buczak & Guven

2016. The study proved that logistic regression is useful for spotting fraudulent transactions. A popular classification approach called logistic regression predicts the association between input features and the likelihood that a transaction is fraudulent. It is a desirable option for fraud detection systems because of its readability and simplicity.

Another well-liked supervised learning strategy for fraud detection is decision trees. To categorize occurrences as fraudulent or authentic, decision tree algorithms, such the C4.5 algorithm, build a tree-like model that divides the dataset depending on feature values. Because they can manage non-linear correlations between features and the target variable, decision trees have the advantage of being ideal for identifying intricate fraud patterns.

The ability of Support Vector Machines (SVMs) to handle high-dimensional data and nonlinear relationships has led to their use in fraud detection as well. SVMs look for an ideal hyperplane that can distinguish between fraudulent and legal transactions with the greatest margin. at dealing with unbalanced datasets, SVMs have shown to perform well at classifying fraudulent transactions.

Although these supervised learning algorithms are easy to use and interpret, they could have trouble spotting fraud. The complexity of fraud patterns is one of the biggest problems. The techniques used by fraudsters are constantly changing, creating complex and dynamic fraud patterns that these algorithms would find challenging to successfully detect.

The unbalanced character of fraud datasets—where the proportion of legal transactions is noticeably higher than that of fraudulent transactions—presents another difficulty. The model may be biased toward the majority class (legal transactions) because of unbalanced datasets, which will lead to decreased performance in identifying the minority class (fraudulent transactions).

Techniques such using the Synthetic Minority Over-sampling Technique (SMOTE), which oversamples the minority class, or under-sampling the majority class have been suggested as solutions to the problem of unbalanced data. These methods seek to improve the identification of fraudulent transactions while balancing the distribution of classes.

2.2 Unsupervised Learning Approaches

For spotting fraud in numerous domains, unsupervised learning techniques like clustering and anomaly detection have been investigated. The goal of these strategies, which do not require labelled data, is to find patterns and anomalies in the data that may point to fraudulent activity.

Clustering algorithms were used in a study by Ranshous et al. (2015) to identify fraud. To find clusters of connected fraudulent transactions, the authors used clustering techniques, which made it possible to spot trends and similarities in fraudulent behaviour. This method is especially beneficial for identifying innovative or previously unidentified fraud patterns that may not be picked up by predetermined rules or labelled data.

Unsupervised learning techniques have the advantage of being able to adapt to new fraud methods without relying on labels that have been predetermined. They can find irregularities and patterns in the data that may be signs of fraud. Unsupervised learning techniques face considerable difficulties due to their increased false positive rate when compared to supervised methods. Unsupervised models have a high rate of false positives because they can classify genuine transactions as anomalies or find clusters that include both valid and fraudulent transactions.

Another drawback is the challenge of identifying specific fraud incidents. While unsupervised learning techniques offer a more comprehensive perspective of fraud tendencies, they could fall short in terms of the level of detail needed to pinpoint fraudulent transactions or the participants. To recognize and authenticate specific fraud cases, more research and analysis are frequently required.

Hybrid methods that blend supervised and unsupervised techniques have been developed to solve the issues of false positives and the difficulty in identifying specific fraud instances.

2.3 Hybrid Approaches

In fraud detection research, hybrid systems that blend supervised and unsupervised techniques have gained popularity. These solutions try to take use of the advantages of both tactics while addressing the weaknesses of each, such as high false positive rates or the inability to manage intricate fraud patterns.

A hybrid fraud detection system with integrated clustering and classification algorithms was proposed by Bhattacharyya et al. (2018). The classification technique was used to separate between fraudulent and valid transactions inside each cluster once the clustering algorithm had identified groups of similar transactions. When compared to employing either strategy alone, our hybrid model showed enhanced fraud detection performance.

The benefit of hybrid techniques is their capacity for both supervised learning to capture well-known fraud patterns and unsupervised learning to detect new fraud patterns. Hybrid models seek to increase fraud detection accuracy while lowering false positives by incorporating the best features of both approaches.

However, using hybrid models in practical settings is not without its difficulties. When compared to individual approaches, these models are typically more intricate and computationally intensive. Large-scale implementation may be more difficult because to the need for additional resources and knowledge for the integration and coordination of multiple algorithms.

2.4 Feature extraction

The process of building new features out of already existing ones to collect more data. The following are some methods frequently employed for feature extraction in financial transaction data:

- **Aggregation:** The summarization of transaction data over predetermined time periods (e.g., daily, weekly) in order to extract characteristics like the total number of transactions, the average frequency of transactions, or the maximum amount of transactions.
- **Time-Based Features:** Extraction of temporal data, such as the day of the week, the hour of the day, or the amount of time since the last transaction, using transaction timestamps.
- **Statistical Features:** Calculating statistical measures of transaction amounts or other pertinent variables, such as mean, standard deviation, and skewness.
- **Text mining:** The process of extracting terms or patterns from text-based fields, such as transaction descriptions, that may be indicators of fraud.

CHAPTER 3
METHODOLOGY

3.1 Dataset Description

The dataset used for the research is a synthetic dataset generated for the purpose of this study, appendix 1. It contains information about financial transactions, including transaction IDs, customer IDs, transaction amounts, transaction timestamps, regions, states, customer categories, and account balances. The dataset consists of 10000 records and includes characteristics such as geographical information, customer profiles, and transaction details.

3.2 Preprocessing Steps

Before applying machine learning algorithms for fraud detection, several preprocessing steps were employed to clean and transform the data. These steps are as follow:

- **Handling missing values:** Identify and handle any missing values in the dataset, either by imputing them or removing the corresponding records.
- **Data normalization:** Scale numerical features such as transaction amounts and account balances to a common range to ensure they have a similar impact during model training.
- **Encoding categorical variables:** Convert categorical variables like regions, states, and customer categories into numerical representations using techniques like onehot encoding or label encoding.
- **Feature selection:** Identify and select the most relevant features that contribute significantly to fraud detection, considering their impact and reducing computational complexity.

3.3 Exploratory Data Analysis

Data visualization can be a valuable step to gain insights into the dataset and understand its characteristics. Visualization techniques applied were:

- Histograms: Plotting histograms can provide an overview of the distribution of numerical features such as transaction amounts and account balances.
- Bar plots: Visualizing categorical variables like regions, states, and customer categories using bar plots can help understand their frequency distribution.
- Scatter plots: Plotting transaction amounts against account balances can reveal potential patterns or outliers.
- Heatmaps: Using a heatmap, correlations between different features can be explored, which can help identify relationships and potential predictors of fraud.

By visualizing the data, it becomes easier to identify any anomalies, outliers, or patterns that may require further investigation or preprocessing before training the machine learning models.

3.4 Feature Engineering and Dimensionality Reduction

The specific properties of the financial transaction data and the goals of fraud detection should be aligned with the chosen feature engineering approaches and dimensionality reduction techniques. The following methods were adopted:

- **Feature Selection:** By focusing on the most crucial elements that helped with fraud detection, we scanned through the data to identify noise. This lessened the possibility of overfitting while also enhancing the model's accuracy and interpretability.
- **Feature Extraction:** Transaction data frequently contains important information that may not be readily captured by the raw features. This is known as feature extraction. Meaningful representations and identify significant fraud-related patterns or trends were created.
- **Dimensionality reduction:** Datasets related to financial transactions may be highly dimensional, which increases computing complexity and raises the possibility of overfitting. Methods for dimensionality reduction reduced the number of features while retaining the most important data, which helped to solve these problems.

The trade-off between model performance and interpretability were considered while choosing certain strategies. Higher predicted accuracy may be obtained using more sophisticated approaches like deep learning or ensemble methods, but they may also be more difficult to comprehend. To balance model complexity, interpretability, and computing efficiency, one must consider both the resources at hand as well as the needs of the fraud detection system.

3.5 Machine Learning Algorithms

The selection and implementation of machine learning algorithms for fraud detection depend on the specific requirements of the problem and the characteristics of the dataset. In this research, the following algorithms were applied:

- **Logistic Regression:** This algorithm is suitable for binary classification tasks and can provide interpretable results.
- **Decision Trees:** Decision trees can capture non-linear relationships and are effective in handling categorical features.
- **Random Forest:** This ensemble method combines multiple decision trees to improve accuracy and handle complex fraud patterns.

The Three algorithms were used to be able to establish the best possible result, and the associated algorithm as well as the applicable hyperparameters.

CHAPTER 4
SYSTEM DESIGN

4.1 PROPOSED WORK

Decision tree technique is statistical data mining technique in which independent and dependent properties are logically expressed in a structure in the form of a tree illustrated in Fig. 1. The categorization rules derived from the decision tree are if then expressions, and to generate each rule, all tests must pass. Decision trees usually split a complex problem into many simple ones, and use iteration to solve sub- problems. The tree is a predictive decision support tool that creates mappings of possible outcomes from different observations. There are numerous prominent classifiers for generating class models from decision trees. To improve precision and avoid overfitting, During the pruning step, such classifiers create a decision tree and afterwards clean up subtrees from the decision tree. This tree can be created by applying machine learning algorithms to the credit card database, such as ID3, C4.5, and multi-layer pruned classifier (MLPC). The aim of the Decision Tree model is to build a small decision tree with high precision. Based on credit card fraud detection, the decision tree has two stages. The initial step is to build a decision tree using the training data provided, and the later step is to use decision rules to classify incoming transactions. The decision tree's input data is labelled with class labels, such as legitimate or fraudulent. The

system monitors each account individually using appropriate descriptors to identify transactions and flags as legitimate or legitimate. In the course of Decision Tree depicted in Fig. 2, all training examples start with one node representing the tree data set at the root node. Each node is split into child nodes in a method-specific binary or multipartition fashion. The decision rules are read one by one from the decision table for each transaction that you classify as Match the transaction fields to each decision rule. It first finds an exact match and indicates the matched rule and transaction class of that class. If no match is found, the highest risk among matching rules is selected and the transaction class is populated with the matched rules of that class. This indicates if a new transaction is a fraud of the same form, The node has been renamed the leaf and is flagged as fraudulent. This model is both quick and adaptable. The MLPC approach is utilized as pre-pruning, which stops the tree's growth at the pruning level specified before construction. It consists of a tree-top-down recursive partitioning and conquest method. Initially all training examples are maintained on the route. The sample is then recursively split based on the chosen attributes. As the entropy metric, choose the split attribute.

Repeat the necessary stages until any of the four conditions is met: 1. All samples from a given node pertain to the same class. 2. There are also no other properties for partitioning. 3. There are no remaining sample. 4. Prune level is achieved as set.

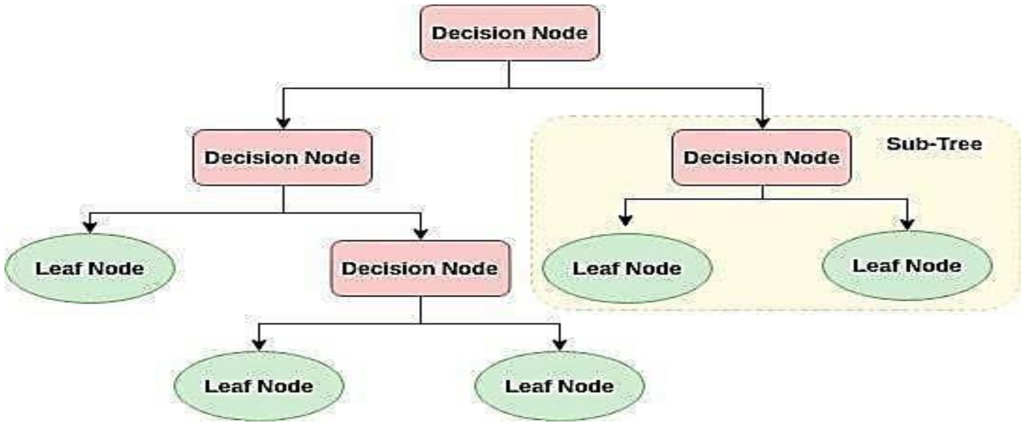


Fig 5.1 Decision Tree Architecture

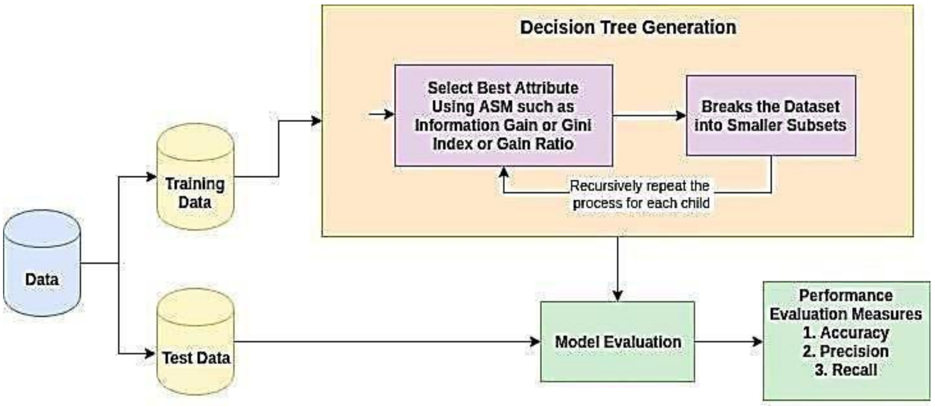

Decision Tree Generation
Select Best Attribute Using ASM such as Information Gain or Gini Index or Gain Ratio
Breaks the Dataset into Smaller Subsets
Recursively repeat the process for each child
Training Data
Data
Test Data
Model Evaluation
Performance Evaluation Measures
1. Accuracy
2. Precision
3. Recall

Fig 5.2 Decision Tree Flow Diagram

CHAPTER 5
RESULTS AND ANALYSIS

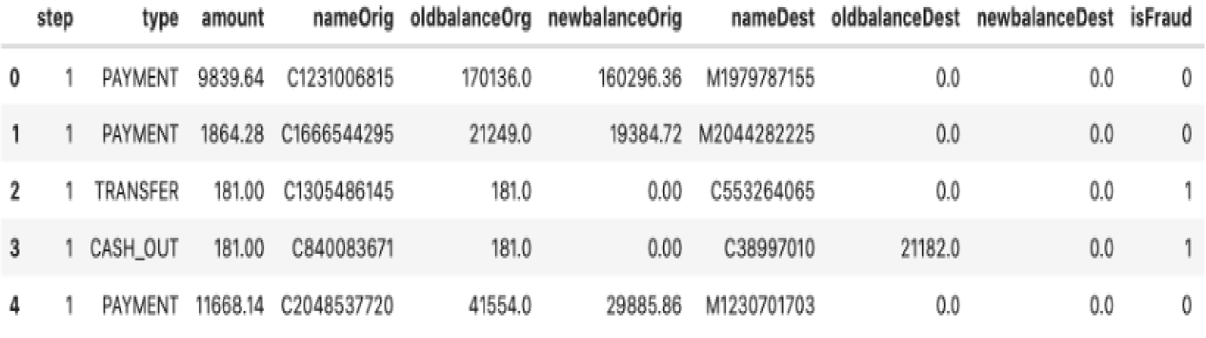
Visualizations are performed in each step, in order to highlight new insights about the underlying patterns and relationships contained within the data. The data analysis process for the deployment of classification models is based on the following steps.

5.1 Data Acquisition

- Download data
- Upload data in Python environment

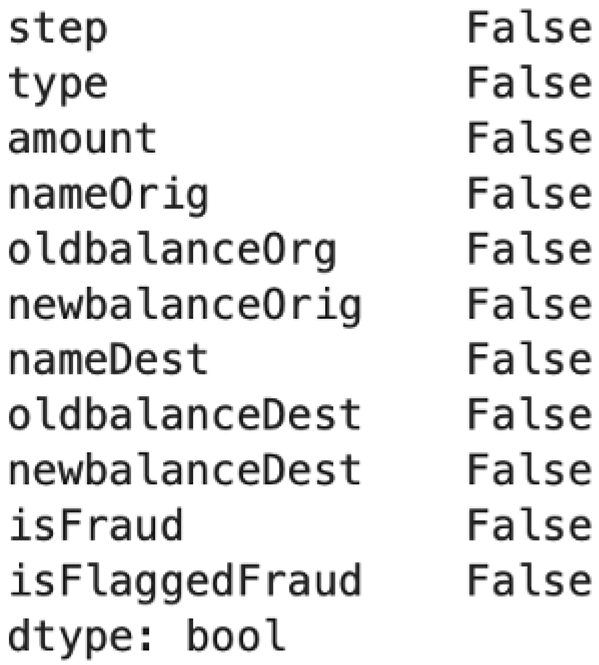
5.2 Data Exploration

Checking data head, info, summary statistics and null values



	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	C553264065	0.0	0.0	1
3	1	CASH_OUT	181.00	C840083671	181.0	0.00	C38997010	21182.0	0.0	1
4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703	0.0	0.0	0

Fig 7.1



```
step              False
type              False
amount            False
nameOrig          False
oldbalanceOrg     False
newbalanceOrig    False
nameDest          False
oldbalanceDest    False
newbalanceDest    False
isFraud           False
isFlaggedFraud    False
dtype: bool
```

Fig 7.2

The dataset consists of several predictor variables and one target variable, isFraud. The dataset consists of 6362620 entries with non-null values.

5.3 Feature Engineering

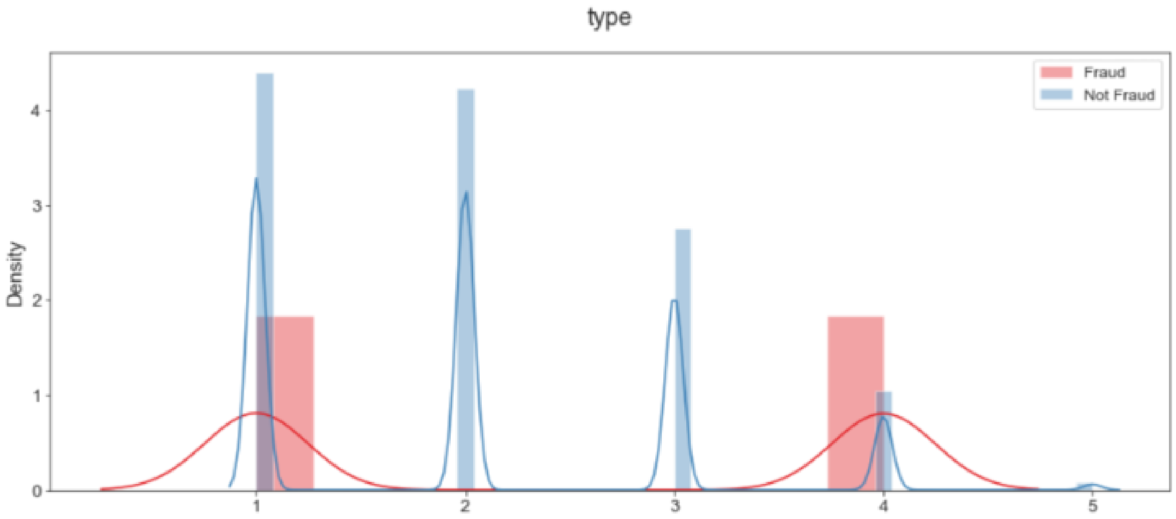


Fig 7.3

We can see from the above data that only two *type* of transactions are classified as fraud so we will drop the remaining types to generalize the data and we will only keep *Cash_out* and Transfer type.

The Type feature in our data is categorical so we will map it to convert it to numerical data 6,3544,407 transactions were **Not Fraud** transactions with 2762196 Not Fraud transactions after considering only two types which are relevant with only 0.3% Fraud transactions. This shows us that we have a very imbalanced data.

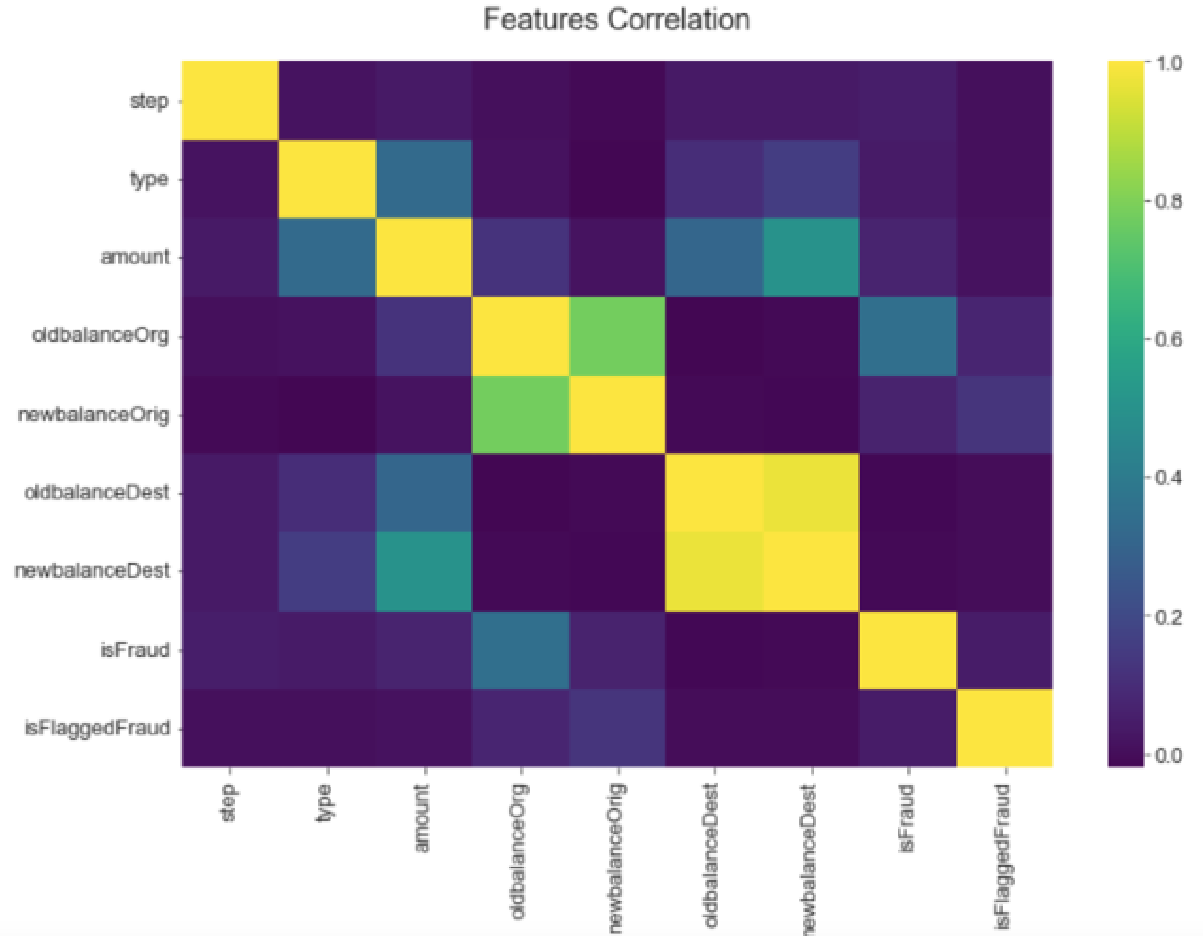


Fig 7.4

From the above correlation graph, we can see that highly correlated features are newbalanceOrig vs oldbalanceOrg, newbalanceDest vs oldbalanceDest and newbalanceDest vs amount

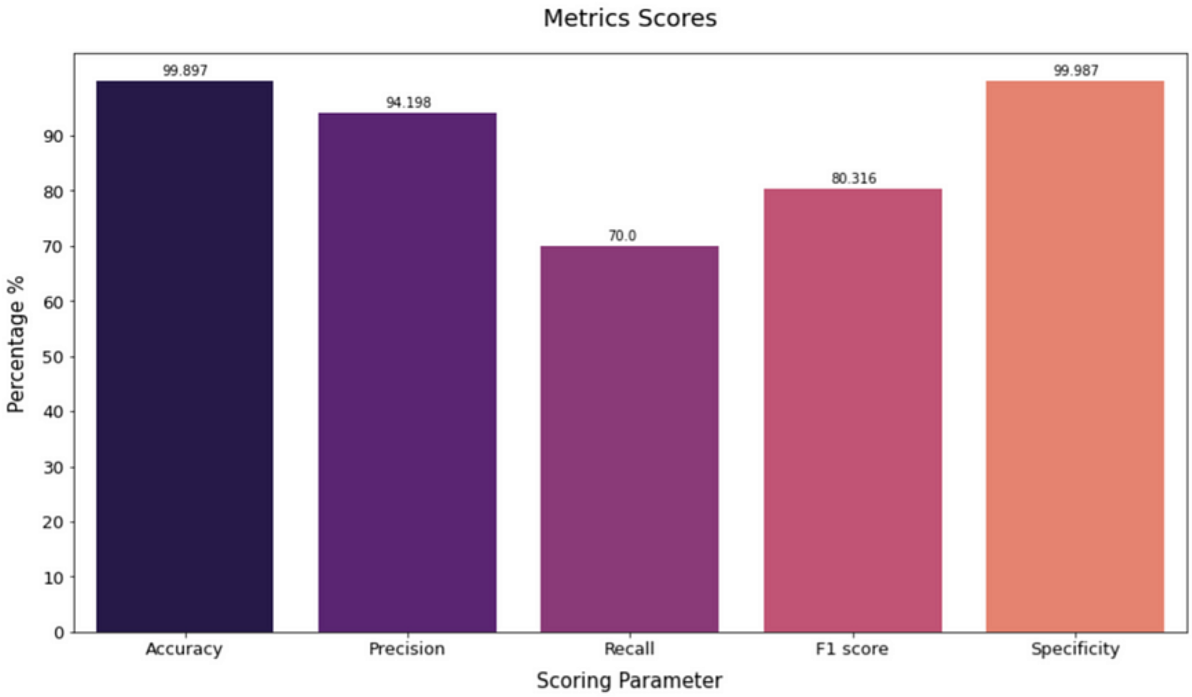
5.4 Results analysis

1. **Deployment of the models**

2. **Comparing prediction accuracy of ML models**

3. **Visualization of the results**

.



CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENTS

6.1 Conclusion

In conclusion, the development and implementation of an online fraud detection system using machine learning algorithms present a crucial solution for safeguarding digital transactions and protecting user data. The success of such a project relies on meticulous planning, strategic decision-making, and continuous adaptation to emerging threats. By leveraging advanced machine learning techniques, organizations can enhance their ability to detect and mitigate various forms of online fraud.

The goal was to predict whether a transaction is a legal transaction or a fraudulent transaction, this falls under the scope of a classification problem. We intend to deploy Supervised Machine Learning models in order to achieve the highest prediction accuracy.

6.2 FUTURE ENHANCEMENTS

a) Despite the advancements gained in this research, there are still a number of opportunities for system improvements and exploration in the future.

b) Examine the usage of ensemble models, like Random Forest or Gradient Boosting Machines, to combine the advantages of many methods and raise the accuracy of fraud detection.

c) Focus on creating more explainable AI models to offer insights into how fraud detection judgments are made, improving system transparency and trust.

REFERENCES

1 https://www.geeksforgeeks.org/online-payment-fraud-detection-using-machine-learning-in-python/

2 https://nevonprojects.com/online-transaction-fraud-detection-using-python-backlogging-on-e-commerce/

3 https://www.slideshare.net/irjetjournal/online-transaction-fraud-detection-system-based-on-machine-learning