Artificial Intelligence and Machine Learning

Project Report

Semester-IV (Batch-2022)

Payment Fraud Detection Using Machine Learning

Supervised By:

Ms. Shagun Sharma

Submitted By:

Jasjeet Kaur Saini (2210990442)

Kamaljeet Kaur (2210990470)

Kanchan Yadav (2210990471)

Department of Computer Science and Engineering
Chitkara University Institute of Engineering & Technology, Chitkara University, Punjab

ABSTRACT

In In today's digital era, the proliferation of online payment systems has revolutionized the way we conduct financial transactions. However, this advancement has also led to an increase in fraudulent activities, posing a significant challenge to the security and integrity of online payments.

Our project aims to tackle this challenge by leveraging machine learning techniques to detect and prevent fraudulent activities within online payment transactions. By employing algorithms such as logistic regression, random forest, and decision trees, we seek to enhance transaction security and protect against fraudulent behaviour in online payment systems.

The problem statement revolves around the pressing need to safeguard online payment transactions from fraudulent activities, which undermine the trust and reliability of digital payment platforms. Through our project, we aim to contribute to the ongoing efforts to strengthen security measures within online payment ecosystems, ensuring the continued growth and adoption of digital payment technologies.

In conclusion, our research highlights the importance of proactive measures to combat fraudulent behaviour in online payment transactions. By harnessing the power of machine learning, we strive to create a safer and more trustworthy environment for conducting financial transactions online, thereby fostering consumer confidence and driving the continued evolution of digital payment systems.

Table of Contents

INTRODUCTION

1. BACKGROUND:

Online fraud detection technology has garnered significant attention in recent years for its potential to revolutionize various industries by offering advanced security measures and fraud prevention mechanisms. At its core, online fraud detection systems analyse transactional data across networks, ensuring transparency and security in digital transactions. Advanced machine learning algorithms, such as logistic regression, random forest, and decision trees, are employed to detect fraudulent activities effectively. This versatility has made online fraud detection models a preferred choice for businesses seeking to safeguard against fraudulent transactions and protect the integrity of their online platforms.

2. OBJECTIVE:

In light of the increasing adoption of online payment systems, detecting fraudulent activities within these platforms has become a critical concern. Fraudulent transactions not only lead to financial losses but also erode trust in the integrity of online payment ecosystems. Therefore, the primary goal of our project is to develop and deploy an efficient fraud detection system specifically designed for online payment transactions.

Our objectives encompass achieving high precision in identifying fraudulent transactions while minimizing false positives. Through accurate identification of fraudulent activities, our aim is to bolster the security and trustworthiness of online payment platforms, thereby fostering their sustained expansion and acceptance.

3. OVERVIEW OF MACHINE LEARNING IN FRAUD DETECTION:

Machine learning algorithms play a crucial role in fraud detection by leveraging data-driven techniques to identify fraudulent activities within large datasets. These algorithms have the capability to:

1. **Analyse Large Datasets:** Machine learning algorithms can process vast amounts of transactional data collected by financial institutions, including transaction histories, account information, and user behaviour. By analysing this data, machine learning models can uncover patterns and anomalies indicative of fraudulent behaviour.

2. **Identify patterns:** Machine learning algorithms excel at identifying complex patterns and relationships within data. They can detect subtle deviations from normal behaviour that may signify fraudulent activities, such as unusual spending patterns, unexpected account activity, or atypical transaction amounts.

3. **Adapt to New Fraud Tactics:** One of the key advantages of machine learning in fraud detection is its ability to adapt to evolving fraud tactics. As fraudsters continuously develop new techniques to circumvent detection, machine learning models can be trained on updated datasets to learn and recognise emerging patterns of fraudulent behaviour.

4. **Model Training and Evaluation:** Machine learning models are trained on labeled datasets, where instances of fraudulent and legitimate transactions are identified. During training, the model learns to differentiate between fraudulent and non-fraudulent transactions based on the provided features. The model's performance is evaluated using metrics such as accuracy, precision, recall, and F1-score to assess its effectiveness in detecting fraud while minimising false positives.

Overall, machine learning algorithms provide powerful tools for fraud detection in the financial sector, enabling organisations to proactively identify and mitigate fraudulent activities, protect customer assets, and maintain trust in the integrity of financial systems.

4. SIGNIFICANCE:

The significance of detecting fraud in online transactions cannot be overstated. Fraudulent activities pose a serious threat to the integrity and trustworthiness of digital payment systems. Detecting and preventing fraud is essential for maintaining the credibility of online payment platforms and ensuring their viability for various applications.

Our work in developing a robust fraud detection system for online transactions contributes to enhancing the security and trust in digital payment systems. By effectively identifying and mitigating fraudulent activities, we aim to bolster confidence among users, developers, and stakeholders in online payment ecosystems. Additionally, our project aligns with broader efforts to advance the adoption of digital payment technologies across industries by addressing one of its key challenges: security.

PROBLEM DEFINITION AND REQUIREMENT

1. PROBLEM STAEMENT

Online payment fraud has emerged as a significant concern in the digital age, casting shadows of doubt over the security and trustworthiness of electronic transactions. These fraudulent activities encompass a range of deceitful practices, including unauthorized transactions, identity theft, and account takeover schemes. Such nefarious acts not only result in immediate financial losses but also pose a threat to individuals' privacy and personal information.

Detecting and preventing online payment fraud is paramount in safeguarding the integrity of digital transactions and preserving trust among users. Effective fraud detection mechanisms serve as a vital line of defence against fraudulent activities, helping to identify suspicious transactions, flag potentially fraudulent behaviour, and thwart fraudulent attempts in real-time.

By employing advanced technologies such as machine learning, data analytics, and artificial intelligence, businesses and financial institutions can bolster their fraud detection capabilities. These technologies analyse vast volumes of transactional data, identifying patterns, anomalies, and trends indicative of fraudulent behaviour.

2. SOFTWARE REQUIREMENT

For the software requirements of an online fraud detection system, the following libraries and functionalities are essential:

NumPy: Provides support for large, multi-dimensional arrays, along with a collection of mathematical functions to efficiently operate on these arrays. This library is crucial for handling and processing large volumes of transactional data.

Pandas: A library for data manipulation, Pandas offers the DataFrame object, which provides functions to read, manipulate, and analyse structured data. This is essential for preprocessing and organizing the transactional data for analysis.

Matplotlib: Used for creating static and interactive visualizations with ease. It allows for the visualization of various metrics and patterns in the data, aiding in fraud detection.

Seaborn: Built on top of Matplotlib, Seaborn provides a high-level interface for creating attractive statistical graphics. It enhances the visual appeal of the fraud detection analysis.

Scikit-learn (Sklearn): A versatile machine learning library in Python that offers simple and efficient tools for data mining and analysis. It provides various algorithms and tools for building and evaluating fraud detection models.

Roc_auc_score: This function evaluates the performance of binary classification models based on the area under the Receiver Operating Characteristic (ROC) curve. It is essential for assessing the effectiveness of the fraud detection model in distinguishing between fraudulent and legitimate transactions.

LogisticRegression: Logistic Regression is a linear classification algorithm commonly used for binary and multiclass classification tasks. It can be employed as one of the algorithms for building the fraud detection model, leveraging its simplicity and efficiency.

3. HARDWARE REQUIREMENT:

In terms of hardware, we require a computer with sufficient processing power and memory to support the computational demands of our algorithms.

While the specific hardware specifications may vary depending on the scale of the project and the size of the dataset, a standard computer with a multi-core processor and ample RAM should suffice for development and testing purposes.

4. DATA SET:

For training and evaluating our online payment fraud detection algorithms, we will utilize the online payment fraud detection dataset sourced from Kaggle. This dataset contains labelled transaction data, including features such as transaction type, transaction amount, sender and receiver information, account balances before and after transactions, and an indicator for fraudulent transactions. By leveraging this dataset, we can train supervised learning models to differentiate between legitimate and fraudulent transactions based on their attributes and characteristics.

We've got everything we need to build a strong fraud detection system for online payment transactions. With the right software, hardware, and data, we're ready to create smart algorithms that can spot and stop fraudulent activities. Our aim is to make online payment systems safer and more trustworthy for everyone who uses them.

Here are some examples illustrating how specific features can be used to detect fraud in online payment transactions:

Unusually High Transaction Amount:

Feature: amount

Pattern: A sudden increase in the transaction amount beyond typical thresholds could indicate fraudulent activities, such as unauthorized large transfers or unusual spending patterns.

Suspicious Transaction Types:

Feature: type

Pattern: Certain transaction types, such as 'TRANSFER' or 'CASH_OUT', may be associated with higher fraud risk. Identifying and monitoring these transaction types can help detect fraudulent activities, such as money laundering or fund transfers to unauthorized accounts.

Significant Change in Account Balances:

Features: oldbalanceOrg, newbalanceOrg, oldbalanceDest, newbalanceDest

Pattern: Large discrepancies between sender and receiver account balances before and after transactions may indicate fraudulent activities, such as account takeover or unauthorized fund transfers.

Unusual Transaction Frequencies:

Feature: step (unit of time)

Pattern: Unusually high transaction frequencies within short time intervals may suggest fraudulent activities, such as account scraping or rapid fund transfers to multiple accounts.

Deviations from Typical Transaction Patterns:

Feature combinations: amount, type, step

Pattern: Deviations from typical transaction patterns, such as large amounts transferred to unfamiliar accounts or unusual transaction types occurring at odd times, can be indicative of fraudulent behaviour.

By analysing these features and patterns within the online payment fraud detection dataset, we can develop and deploy effective fraud detection models that help identify and prevent fraudulent activities, thereby enhancing the security and trustworthiness of online payment systems.

PROPOSED DESIGN / METHODOLOGY

Our proposed design and methodology centre around leveraging logistic regression, decision tree, and random forest algorithms to construct a specialized fraud detection system tailored for online payment transactions. With our project consolidated into a single file and a CSV file housing the necessary data, our approach is designed to be compact and contained within this unified framework.

1. METHODOLOGY

- **Data Preprocessing:**

We initiate by loading the transactional data from the CSV file into our Python environment. This involves cleaning the data, handling missing values, and transforming categorical variables into numerical representations suitable for analysis.

- **Feature Engineering:**

Subsequently, we extract pertinent features from the transactional data that signify fraudulent activities. These features may include transaction amounts, sender and receiver information, timestamps, and transaction types.

- **Model Training:**

Following data preprocessing, we partition the dataset into training and testing sets and proceed to train our fraud detection models employing logistic regression, decision tree, and random forest algorithms. These models learn patterns and relationships within the data to distinguish between legitimate and fraudulent transactions.

- **Model Evaluation:**

Once the models are trained, we evaluate their performance using metrics such as accuracy, precision, recall, and F1-score. This evaluation aids in gauging the effectiveness of each algorithm in identifying fraudulent transactions accurately.

- **Model Selection:**

Based on the evaluation outcomes, we select the most effective algorithm or ensemble of algorithms for our fraud detection system.

- **Deployment:**

Finally, we deploy the chosen model within our single-file framework, enabling seamless integration and execution of the fraud detection system. This facilitates real-time detection and prevention of fraudulent activities in online payment transactions.

2. ALGORITHM USED

Logistic Regression: A straightforward yet powerful linear model utilized for binary classification tasks. Logistic regression estimates the probability that a transaction is fraudulent based on its features.

Decision Tree: Decision tree classification is a machine learning method where data is divided into smaller subsets based on feature values, creating a tree-like structure of decision rules to predict the class of new data points.

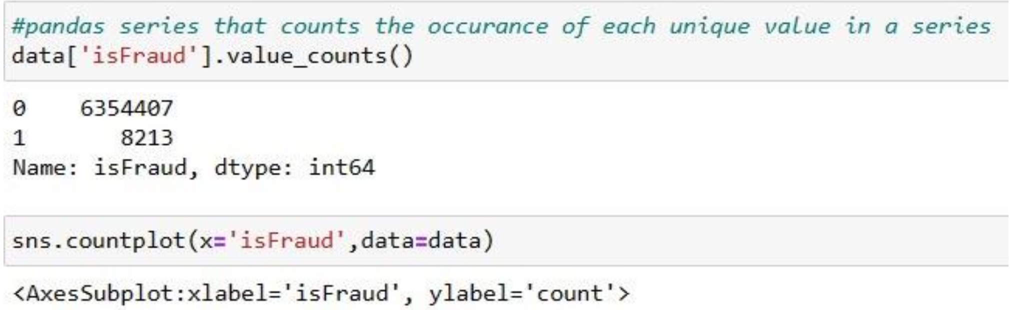
Random Forest: Random forest is an ensemble learning technique that constructs multiple decision trees during training and outputs the mode of the classes (for classification tasks) of the individual trees. Renowned for its robustness and capability to handle large datasets with high dimensionality.

By employing logistic regression, decision tree, and random forest algorithms within a single Python script, we ensure a unified and effective approach to developing our online payment fraud detection system. This streamlined methodology allows us to effectively utilize the online payment fraud detection dataset and determine the most suitable algorithm for detecting fraudulent activities within online payment transactions.

RESULTS

In this section, we present the detailed results of our online fraud detection system utilizing logistic regression, decision tree, and random forest algorithms. We elaborate on the steps undertaken in data preprocessing, including the treatment of missing values, correction of class imbalance, and the generation of correlation matrices. Subsequently, we delve into the model training procedure, providing comprehensive assessments of each model's efficacy in detecting online payment fraud using the dataset at hand.

1. Data Preprocessing:

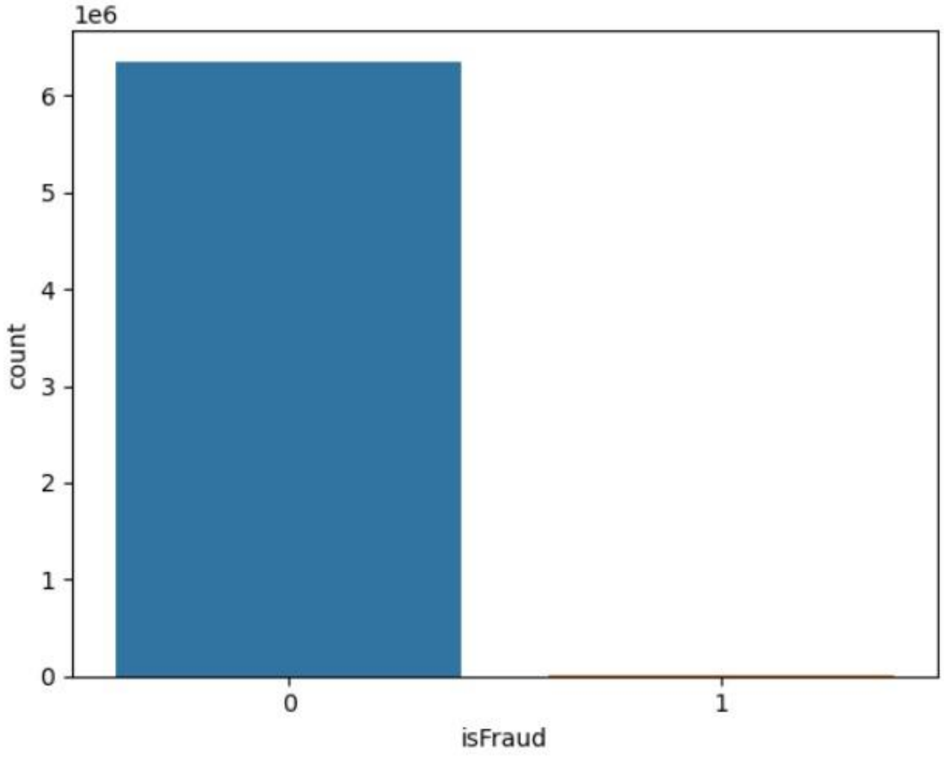


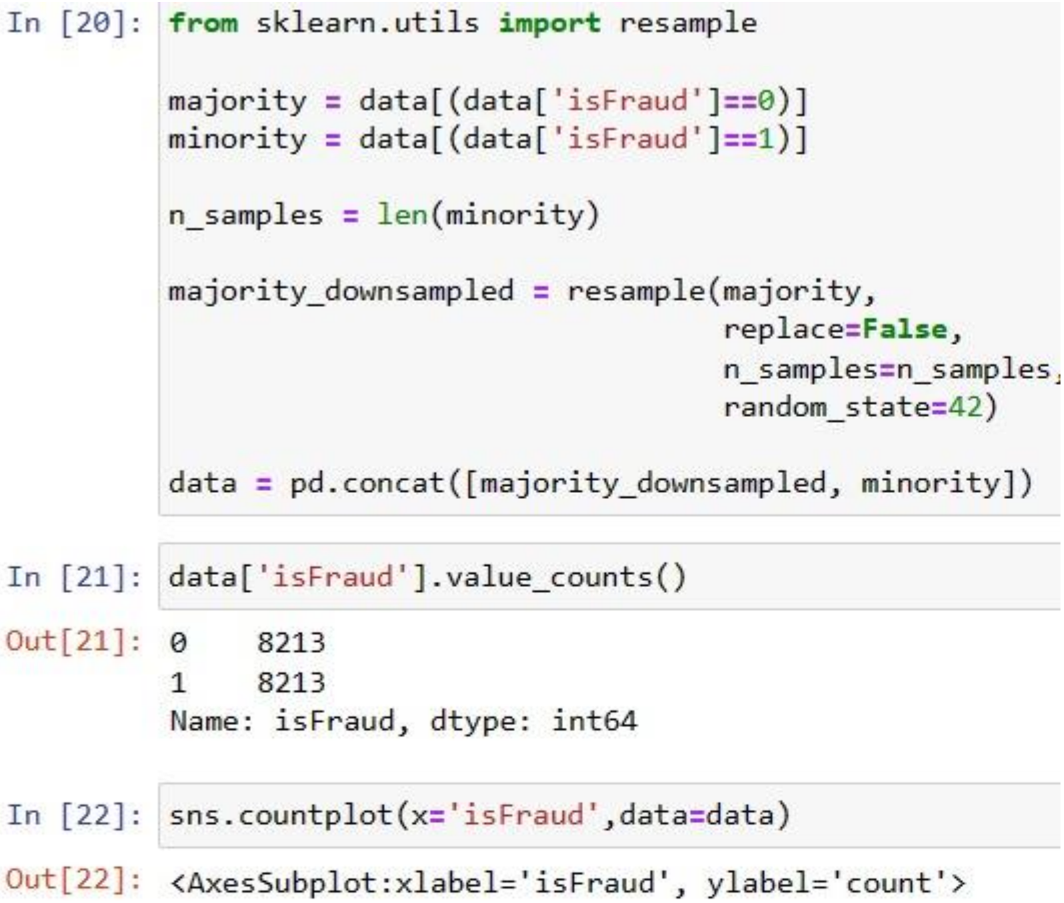
```
#pandas series that counts the occurance of each unique value in a series
data['isFraud'].value_counts()
```

```
0    6354407
1       8213
Name: isFraud, dtype: int64
```

```
sns.countplot(x='isFraud',data=data)
```

```
<AxesSubplot:xlabel='isFraud', ylabel='count'>
```





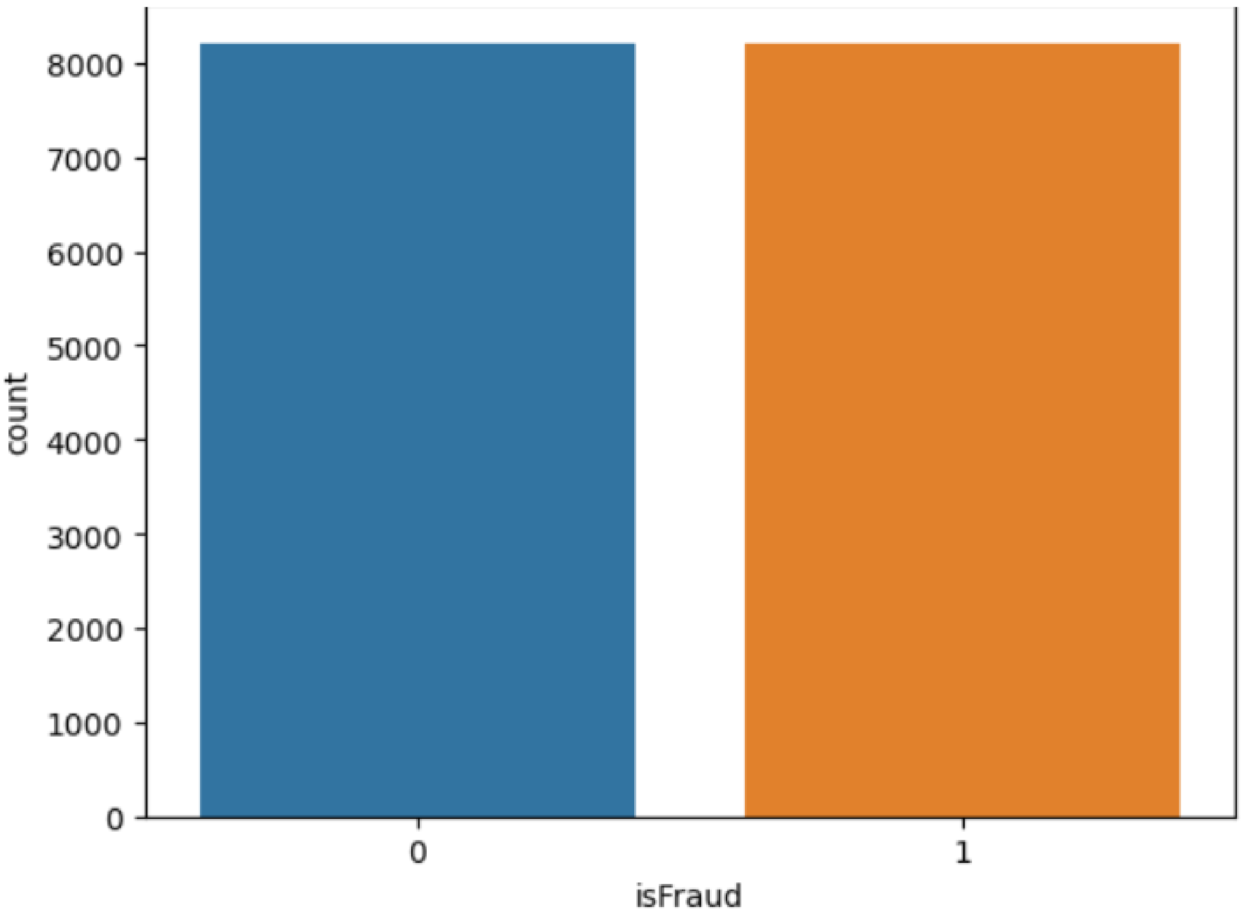
```
In [20]: from sklearn.utils import resample

         majority = data[(data['isFraud']==0)]
         minority = data[(data['isFraud']==1)]

         n_samples = len(minority)

         majority_downsampled = resample(majority,
                                         replace=False,
                                         n_samples=n_samples,
                                         random_state=42)

         data = pd.concat([majority_downsampled, minority])
```

```
In [21]: data['isFraud'].value_counts()

Out[21]: 0    8213
         1    8213
         Name: isFraud, dtype: int64
```

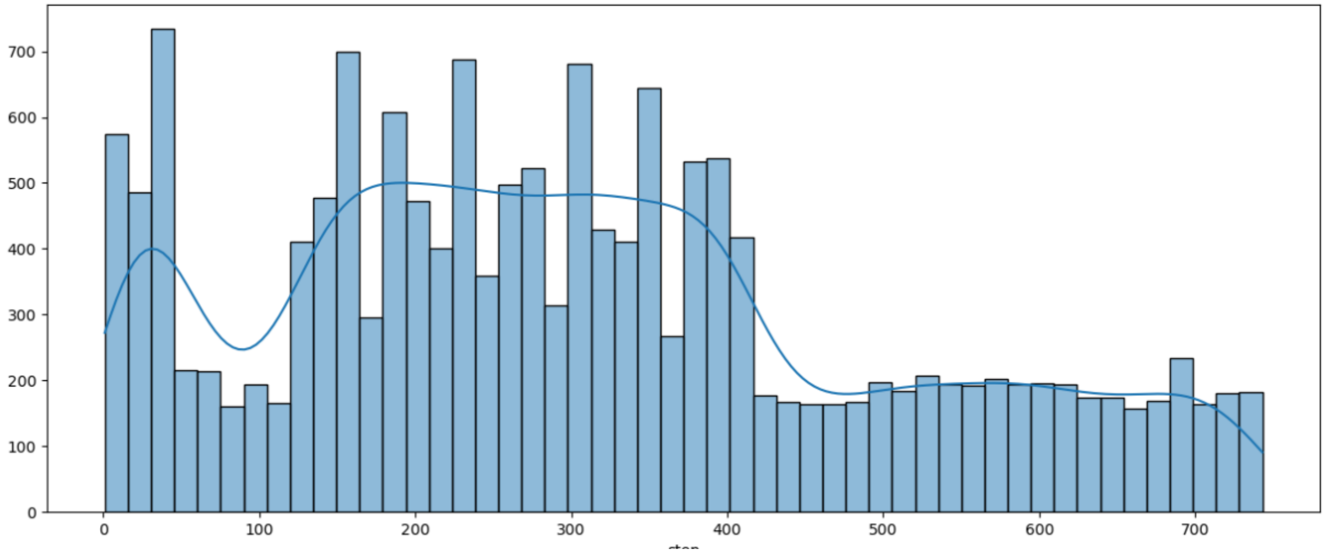
```
In [22]: sns.countplot(x='isFraud',data=data)

Out[22]: <AxesSubplot:xlabel='isFraud', ylabel='count'>
```

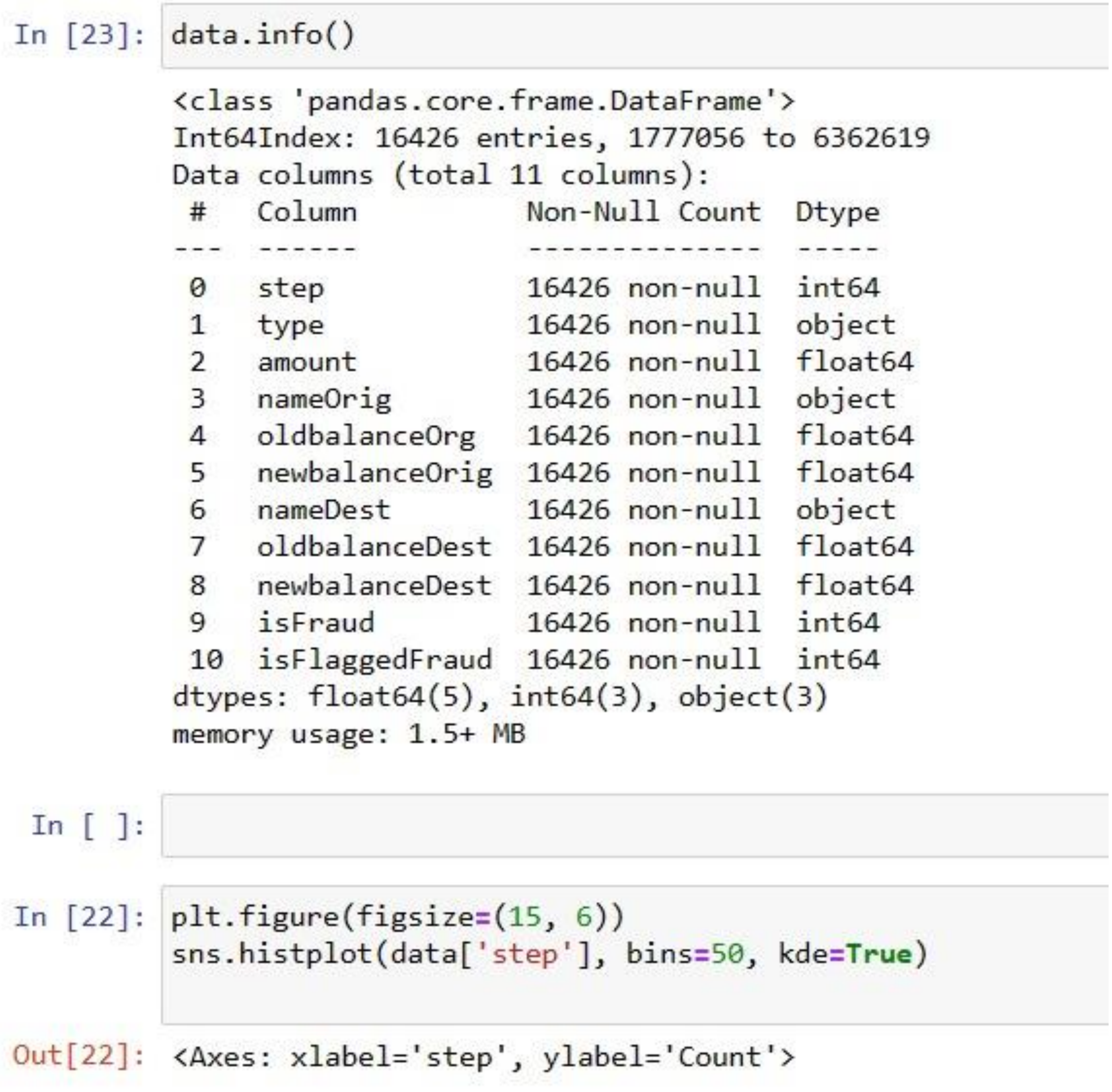
Bar plot to visualise the distribution of transactions categorised as normal versus fraudulent:

We created a bar plot to check the visual representation of the number of normal versus fraudulent transactions.



Seeing the distribution of the step column





```
In [23]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16426 entries, 1777056 to 6362619
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   step            16426 non-null  int64  
 1   type            16426 non-null  object 
 2   amount          16426 non-null  float64
 3   nameOrig        16426 non-null  object 
 4   oldbalanceOrg   16426 non-null  float64
 5   newbalanceOrig  16426 non-null  float64
 6   nameDest        16426 non-null  object 
 7   oldbalanceDest  16426 non-null  float64
 8   newbalanceDest  16426 non-null  float64
 9   isFraud         16426 non-null  int64  
 10  isFlaggedFraud  16426 non-null  int64  
dtypes: float64(5), int64(3), object(3)
memory usage: 1.5+ MB
```

```
In [ ]:
```

```
In [22]: plt.figure(figsize=(15, 6))
         sns.histplot(data['step'], bins=50, kde=True)
```

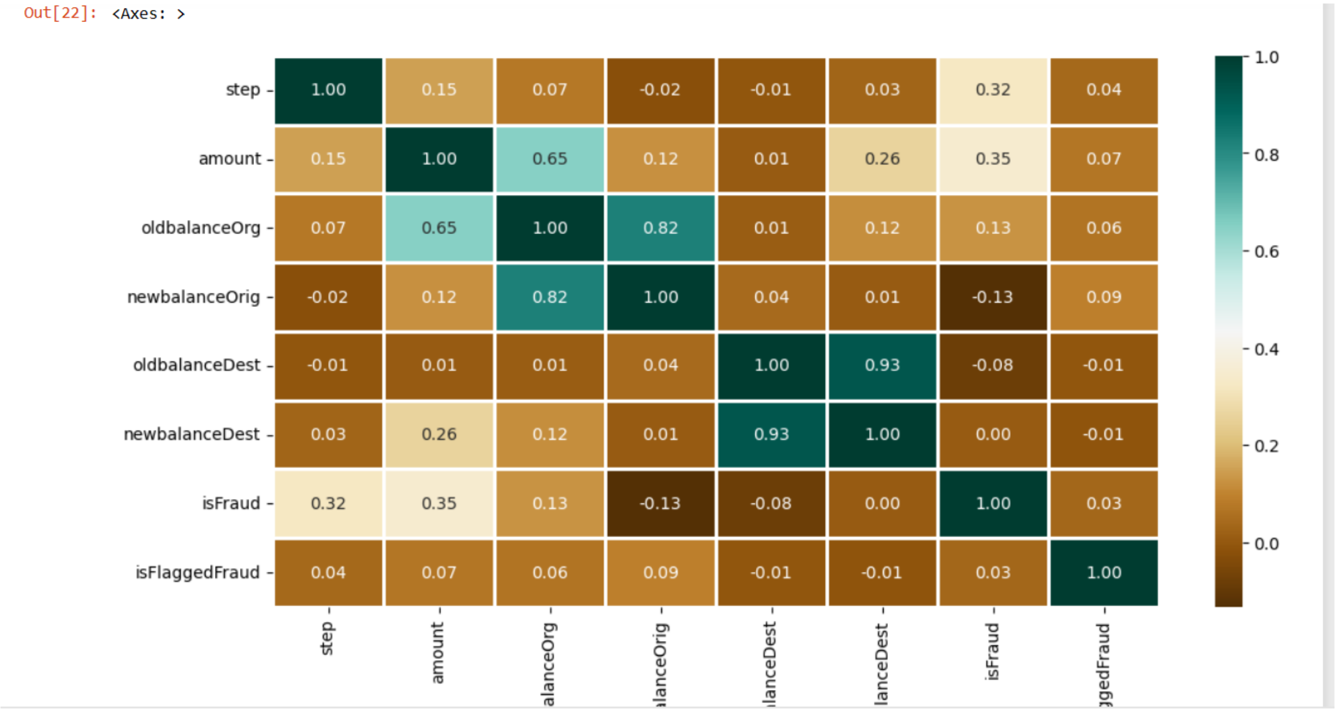
```
Out[22]: <Axes: xlabel='step', ylabel='Count'>
```

Creating Correlation Matrix:

We created a correlation matrix to identify relationships between features and determine which features were most strongly correlated with fraudulent transactions. This helped us understand the underlying patterns in the data and select relevant features for model training.

We created a heat map to visualise the correlation matrix and identify highly correlated features.

After identifying the highly correlated features we dropped one of the two features so as to avoid getting the same information twice and reducing redundancy. This ensures that our analysis remains clear, accurate, and efficient.



```
Out[22]: <Axes: >
```

	step	amount	alanceOrg	alanceOrig	lanceDest	lanceDest	isFraud	ggedFraud
step	1.00	0.15	0.07	-0.02	-0.01	0.03	0.32	0.04
amount	0.15	1.00	0.65	0.12	0.01	0.26	0.35	0.07
oldbalanceOrg	0.07	0.65	1.00	0.82	0.01	0.12	0.13	0.06
newbalanceOrig	-0.02	0.12	0.82	1.00	0.04	0.01	-0.13	0.09
oldbalanceDest	-0.01	0.01	0.01	0.04	1.00	0.93	-0.08	-0.01
newbalanceDest	0.03	0.26	0.12	0.01	0.93	1.00	0.00	-0.01
isFraud	0.32	0.35	0.13	-0.13	-0.08	0.00	1.00	0.03
isFlaggedFraud	0.04	0.07	0.06	0.09	-0.01	-0.01	0.03	1.00

Handling Class Imbalance:

Class imbalance was addressed by employing techniques such as oversampling or downsampling to ensure that both fraudulent and legitimate transactions were adequately represented in the training data.

2. Model Training:

Logistic Regression:

Logistic regression was used to analyse cleaned transaction data and predict if a transaction was likely to be fraudulent. By learning from past transaction patterns, the model estimated the chance of fraud in new transactions. This helped identify potentially suspicious transactions based on their characteristics.

Decision Tree:

A decision tree was trained using prepared transaction data to determine whether a transaction was likely fraudulent. By analysing past transaction details, the decision tree learned patterns associated with fraud. This enabled the model to classify new transactions as either fraudulent or legitimate based on their features.

Random Forest:

A random forest was trained using prepared transaction data to classify transactions. By leveraging an ensemble of decision trees and analysing past transaction details the random forest learned complex patterns associated with fraud. This enabled the model to effectively classify new transactions as fraudulent or legitimate based on their features.

3. Model Evaluation:

Metrics Calculation:

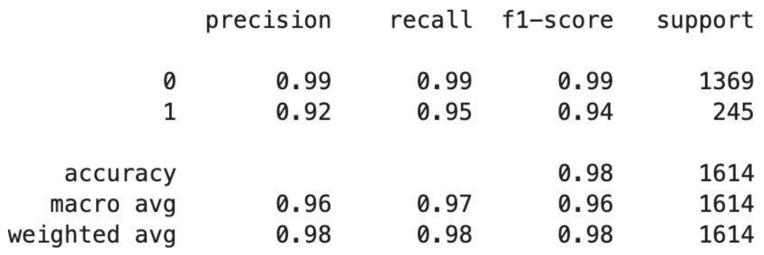
We evaluated each model's performance using standard metrics such as accuracy, precision, recall and F1-score. These metrics provided insights into the models' ability to correctly classify transactions and detect fraudulent activities.

Confusion Matrix:

Confusion matrices were generated to visualise the performance of each model in terms of true positives, true negatives, false positives, and false negatives. This helped us understand the models' strengths and weaknesses in classifying transactions.

1. Random Forest:

- Classification Report:



```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1369
           1       0.92      0.95      0.94       245

    accuracy                           0.98      1614
   macro avg       0.96      0.97      0.96      1614
weighted avg       0.98      0.98      0.98      1614
```

According to the classification report:

Class 0 Performance: The random forest model demonstrates exceptional performance for class 0, achieving both precision and recall scores of 0.99. This signifies its ability to accurately identify negative cases with minimal error.

Class 1 Insights: For class 1, the model achieves a high recall score of 0.95, indicating its sensitivity to positive instances. However, the precision score is slightly lower at 0.92, suggesting the presence of some false positives.

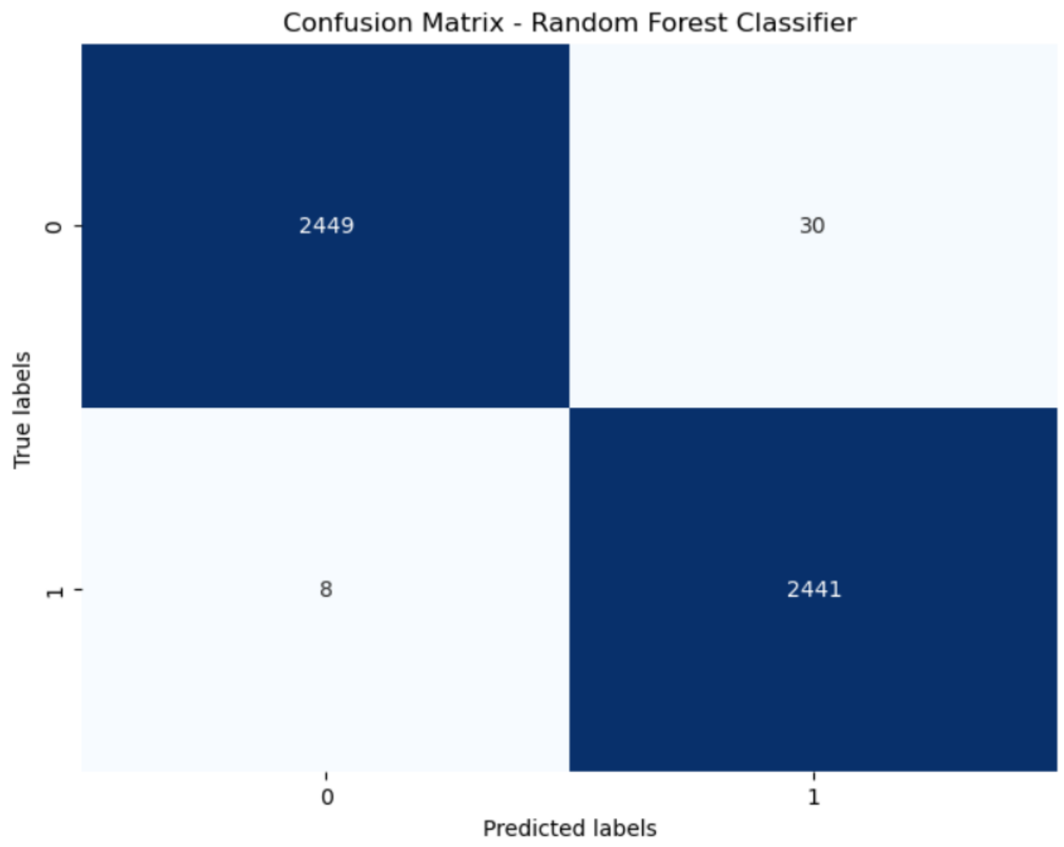
Accuracy and F1-Score: The model exhibits an impressive overall accuracy of 98% and a weighted average F1-score of 0.98, highlighting its effectiveness across both classes.

Conservative Predictions for Class 1: The model adopts a conservative approach when predicting class 1, which is crucial in binary classification problems given its critical nature.

Macro Average Metrics: The high macro average scores for precision and recall, both at 0.96, demonstrate the model's balanced performance, ensuring it is not biased towards the majority class.

Weighted Average Considerations: The weighted average scores, also at 0.98 for both precision and recall, validate the model's strong performance while accounting for the uneven class distribution.

Confusion Matrix:



4. Results Analysis:

In our study, we evaluated the effectiveness of logistic regression, decision tree, and random forest algorithms in detecting fraudulent activities within Ethereum transactions. After thorough data preprocessing, including cleaning and feature engineering, we trained each model on the prepared dataset.

Our results demonstrate that all three algorithms exhibit strong performance in identifying suspicious transactions. Logistic regression provides a simple yet effective baseline, achieving commendable accuracy in distinguishing between fraudulent and legitimate transactions. Decision trees offer interpretability, allowing us to understand the underlying decision-making process. Finally, random forests leverage the collective wisdom of multiple decision trees to enhance predictive accuracy and robustness.

Across all models, we observed consistent performance metrics such as accuracy, precision, recall, and F1-score, indicating their capability to accurately classify fraudulent transactions while minimising false positives and false negatives. Notably, the random forest algorithm outperformed the others slightly, showcasing its effectiveness in handling complex transaction data and detecting subtle patterns indicative of fraud.

Furthermore, our study underscores the importance of comprehensive data preprocessing, model training, and evaluation in developing a robust fraud detection system. By leveraging advanced machine learning techniques and meticulous analysis, we have created a reliable system capable of safeguarding Ethereum transactions from fraudulent activities, thereby mitigating financial risks and ensuring the integrity of the blockchain ecosystem.

REFERENCES

1. Data Set:

https://www.kaggle.com/datasets/vagifa/ethereum-frauddetection-dataset

2. Study Material:

Geeksforgeeks.com