jasikagupta04

Python Project

Fraud Transaction Detection Insights

By Jasika Gupta

Jasika Gupta

jasikagupta04@gmail.com

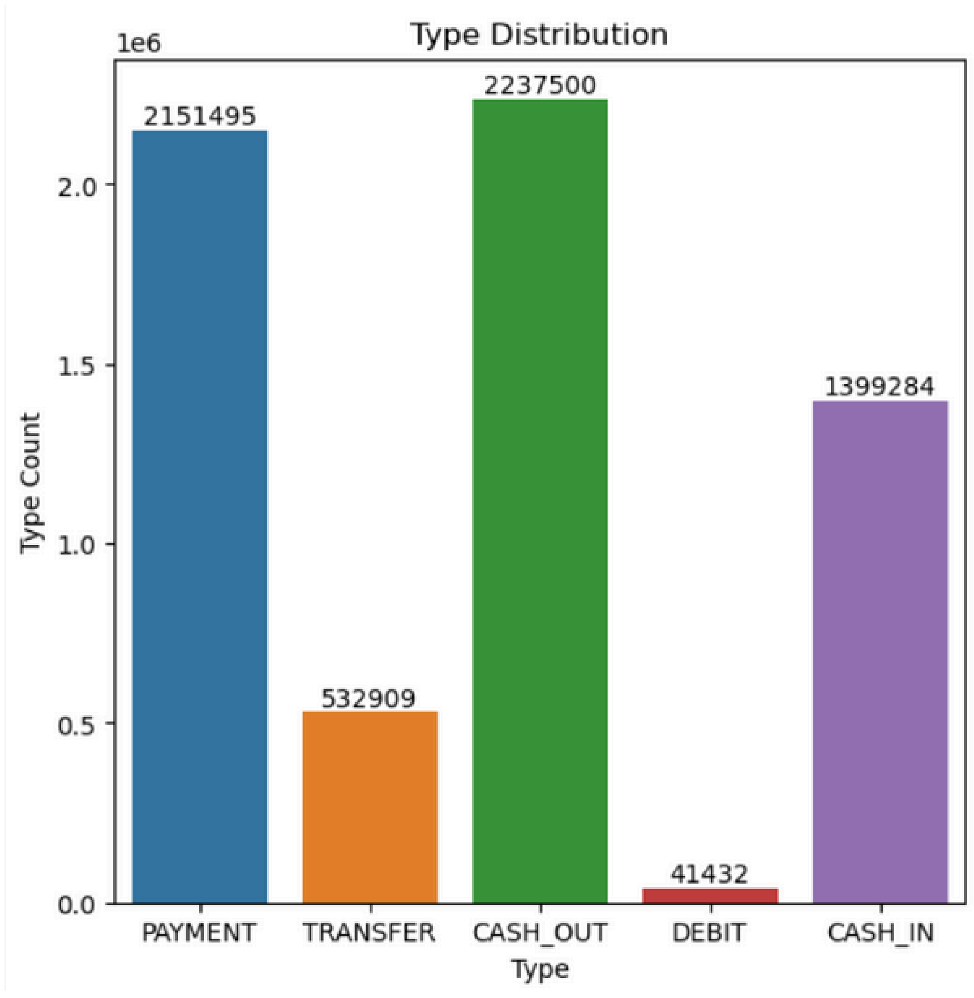
Introduction

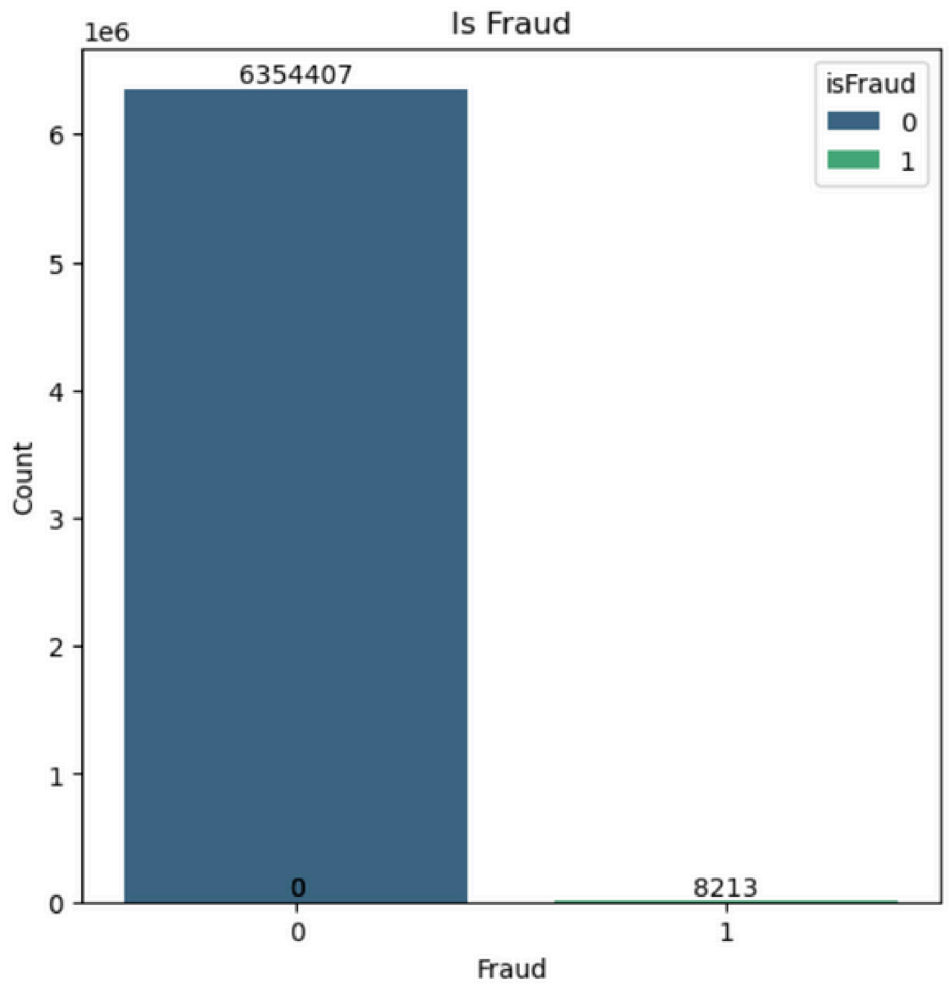
This notebook presents an **Exploratory Data Analysis and Visualization** along with a **Logistic Regression Model** for detecting fraud.

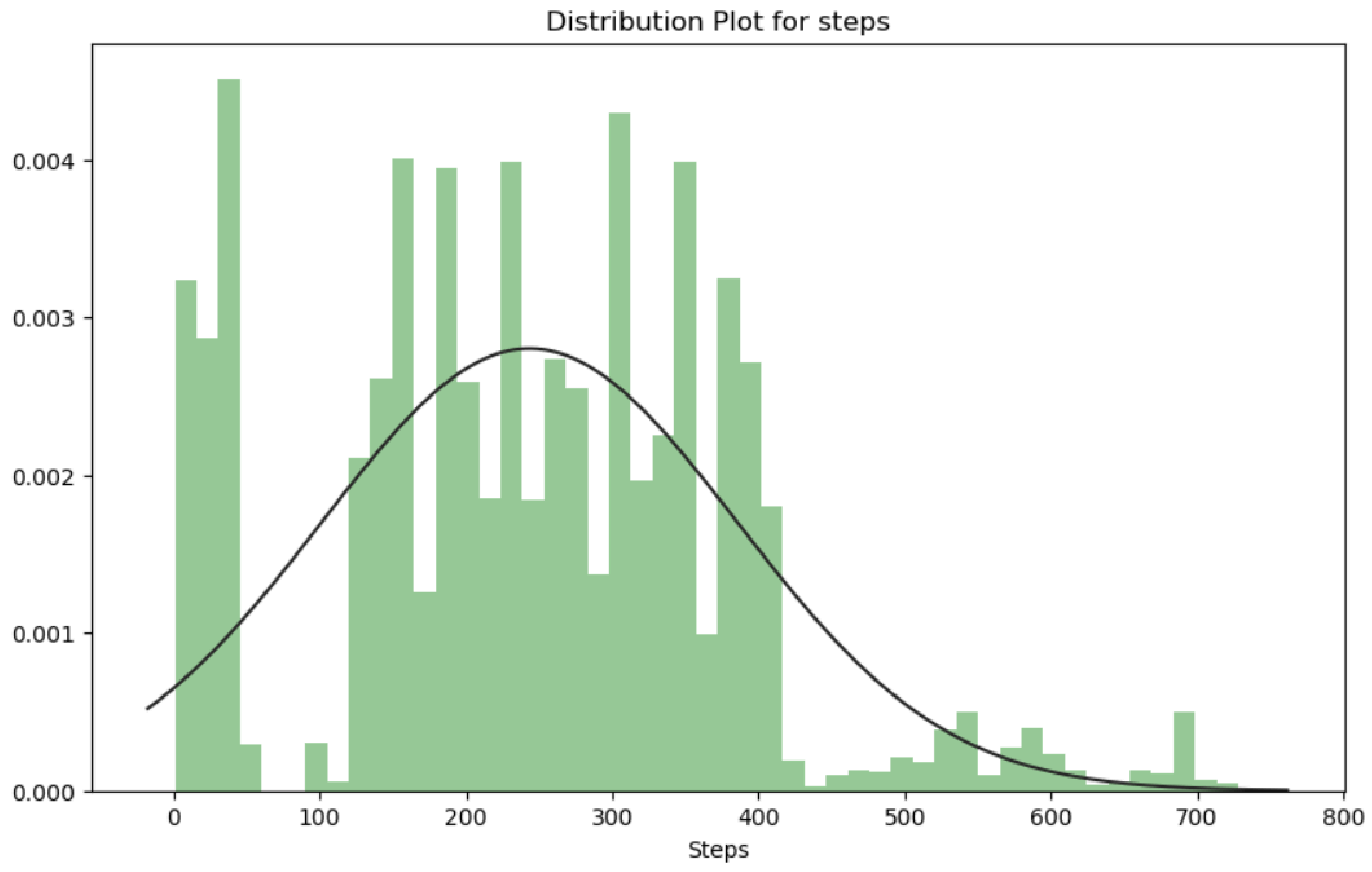
Fraud detection is a critical concern across various industries, including banking and finance, insurance, law enforcement, government agencies, and more.
In recent years, there has been a significant rise in fraudulent activities, making detection both essential and challenging.

Given the rarity of fraud cases within a vast dataset, identifying these anomalies is both crucial and complex.
EDA and data analysis play a vital role in uncovering patterns and trends within transactional data, enabling the identification of fraudulent activities. Using Python, vast numbers of transactions can be analyzed to highlight suspicious patterns, and logistic regression provides a straightforward and effective approach for modeling the likelihood of fraud.

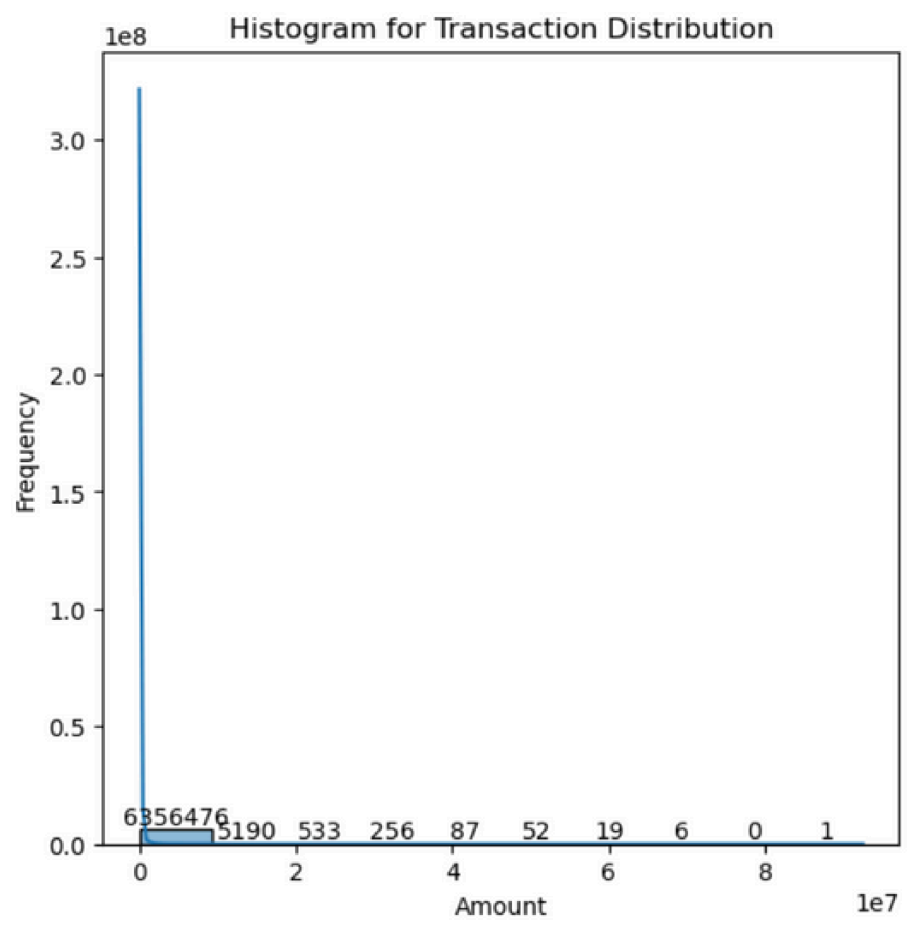
Data Visualization (1/2)

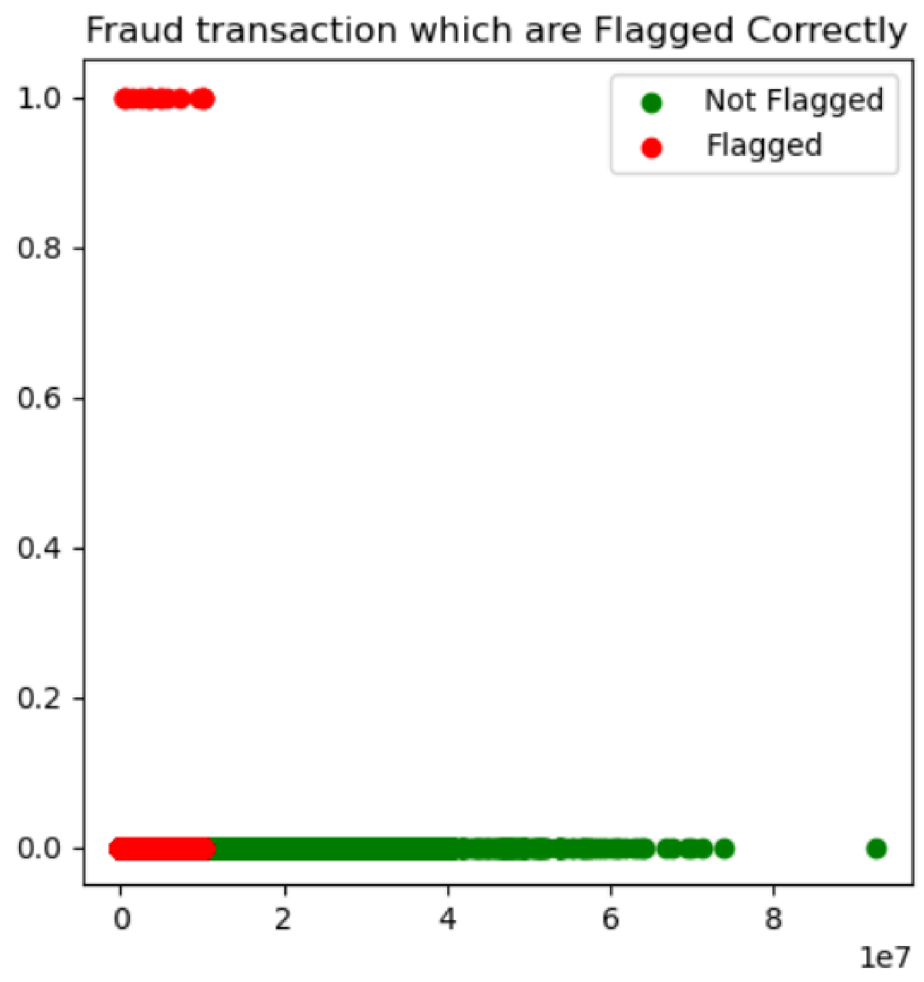


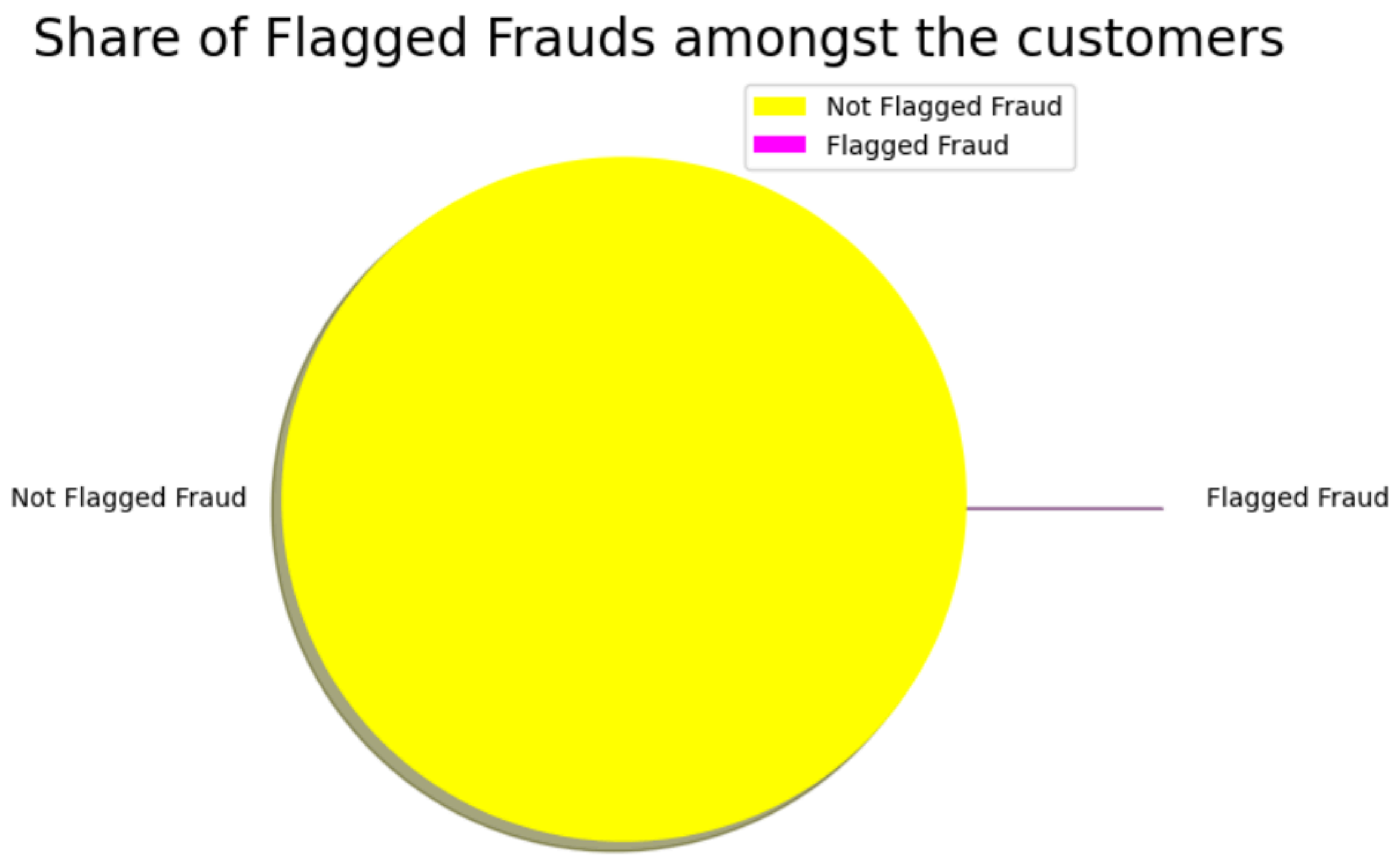




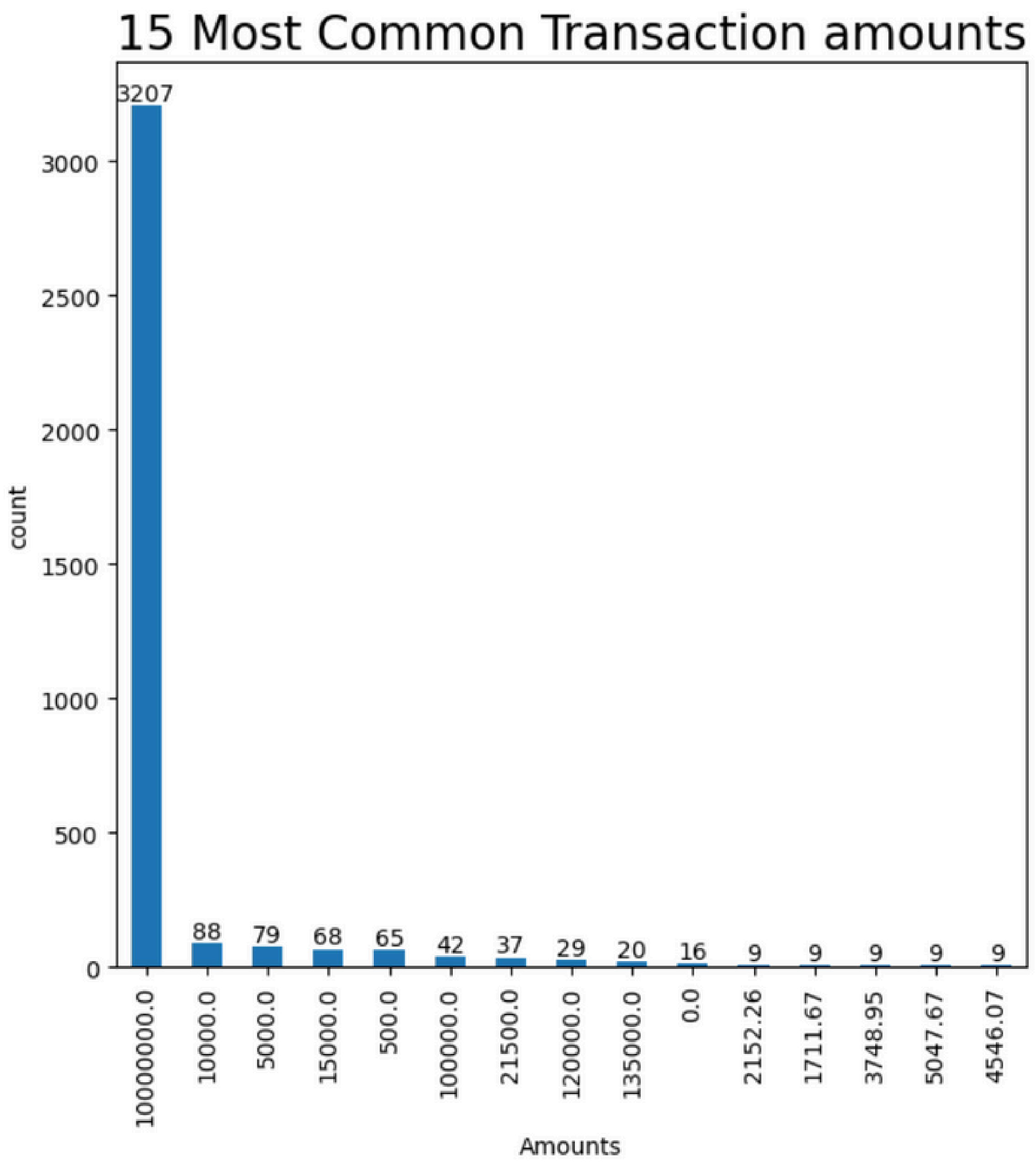
Data Visualization (2/2)

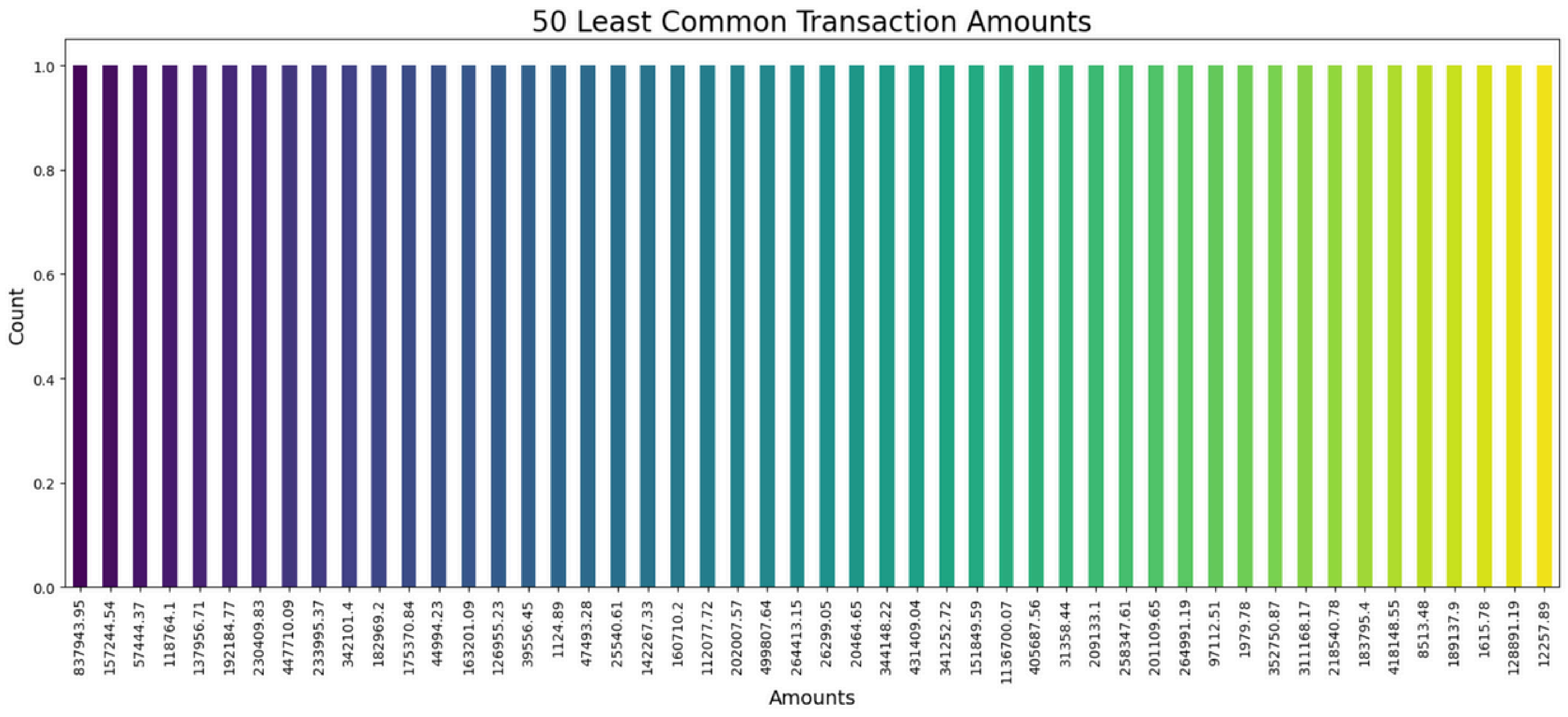




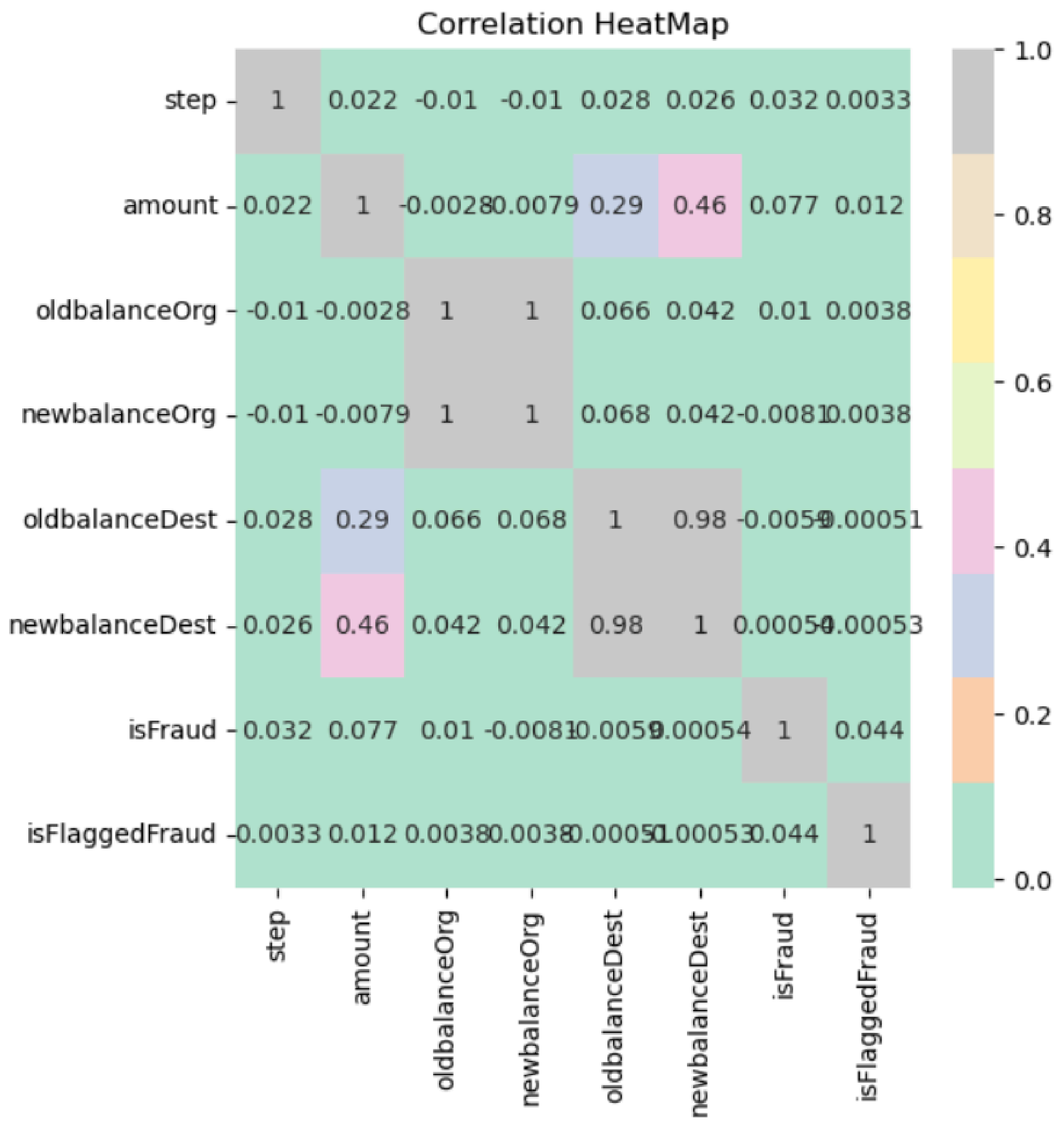


Transaction Amount





Correlation Between Different Variable



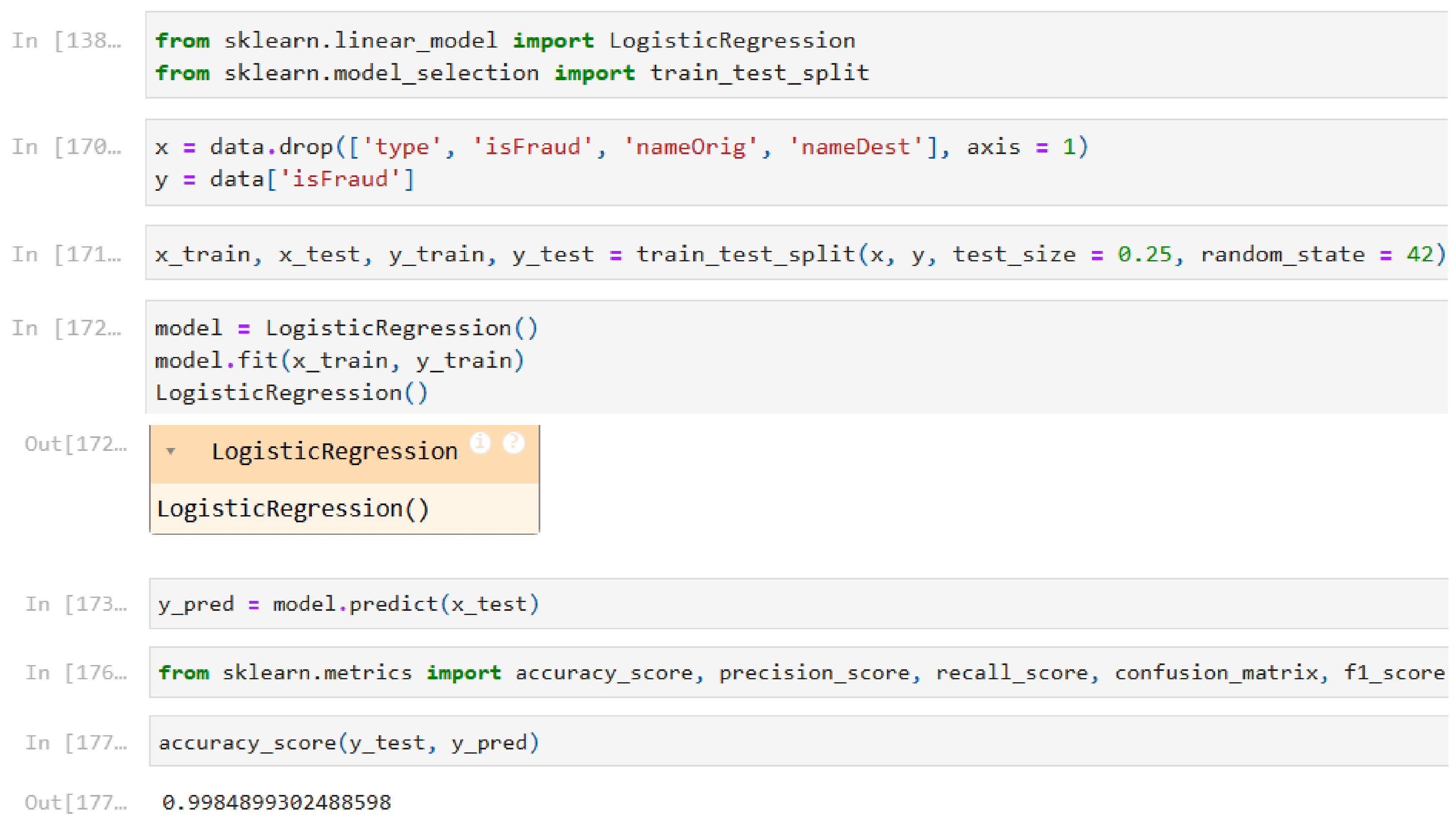
We can conclude from this heatmap:

- OldbalanceOrg and NewbalanceOrg are highly correlated.
- OldbalanceDest and NewbalanceDest are highly correlated.
- Amount is correlated with isFraud(Target Variable).

Regression Model for Fraud Detection

The **Logistic Regression** model was **trained** on **25% of the dataset** to predict the probability of a fraudulent transaction. With the **accuracy of 99.8%,** the model demonstrates the **ability to identify key fraudulent patterns while maintaining a balance between sensitivity and specificity.**

PYTHON CODES FPR LOGISTIC REGRESSION MODEL:



```
In [138…  from sklearn.linear_model import LogisticRegression
          from sklearn.model_selection import train_test_split

In [170…  x = data.drop(['type', 'isFraud', 'nameOrig', 'nameDest'], axis = 1)
          y = data['isFraud']

In [171…  x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25, random_state = 42)

In [172…  model = LogisticRegression()
          model.fit(x_train, y_train)
          LogisticRegression()

Out[172…  LogisticRegression
          LogisticRegression()

In [173…  y_pred = model.predict(x_test)

In [176…  from sklearn.metrics import accuracy_score, precision_score, recall_score, confusion_matrix, f1_score

In [177…  accuracy_score(y_test, y_pred)

Out[177…  0.9984899302488598
```

Performance Evaluation

- **Accuracy Score: 0.9985 (99.85%)**

The model achieved very high accuracy, indicating that the majority of predictions (both fraud and non-fraud) were correct.

- **Precision Score: 0.4132 (41.32%)**

A precision of 41.32% suggests that the model has a high rate of false positives (non-fraudulent transactions incorrectly classified as fraudulent).

- **Recall Score (Sensitivity): 0.4156 (41.56%)**

A recall of 41.56% implies that the model misses more than half of the actual fraud cases, which is a critical issue in fraud detection as it may allow fraudulent activities to go unnoticed.

- **Confusion Matrix:**
  - **True Positives (TP): 850**

Fraudulent transactions correctly identified.

  - **False Positives (FP): 1207**

Non-fraudulent transactions incorrectly classified as fraudulent.

  - **True Negatives (TN): 1,587,403**

Non-fraudulent transactions correctly identified.

  - **False Negatives (FN): 1,195**

Fraudulent transactions incorrectly classified as non-fraudulent.

- **Specificity: 0.9992 (99.92%)**

With a specificity of 99.92%, the model performs exceptionally well in identifying legitimate transactions, which is expected given the imbalanced dataset.

jasikagupta04

Thank You!

By Jasika Gupta

Jasika Gupta

jasikagupta04@gmail.com