1. **Import libraries and load data:** Let's start by importing the necessary libraries (for example, pandas, numpy, matplotlib) and loading your data set.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

2. Let's look at the data structure, the **types of variables**, the first few lines and the main statistical indicators.

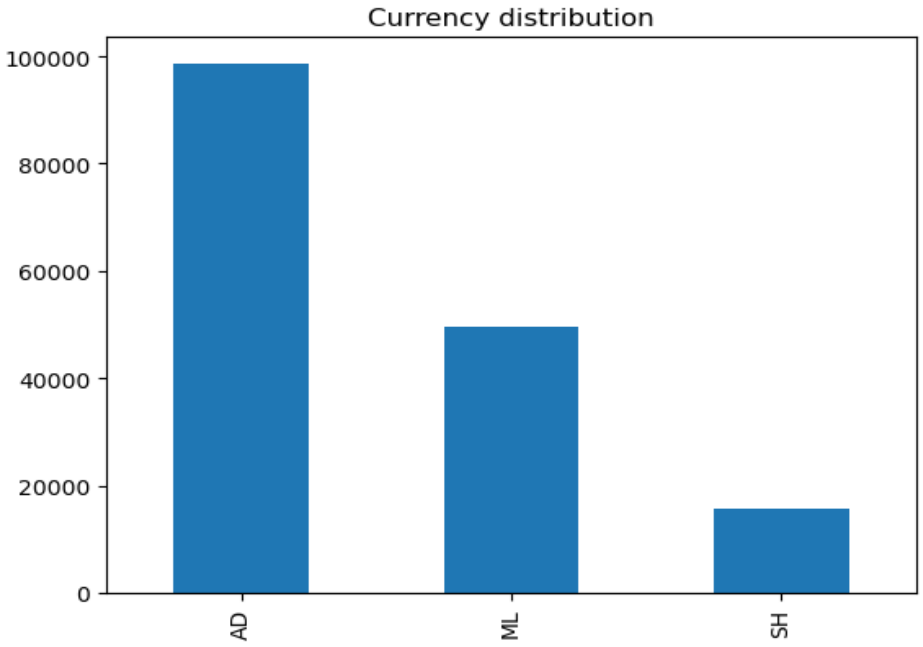
```
print(data.head())
```

```
    Time Country  Value Currency             Status
0  25324       D  18.61       ML  Accepted No Fraud
1  41036       B  18.79       SH  Accepted No Fraud
2  24310       B  27.63       AD  Accepted No Fraud
3    948       D  16.92       ML  Accepted No Fraud
4   5280       A  26.54       AD  Accepted No Fraud
```

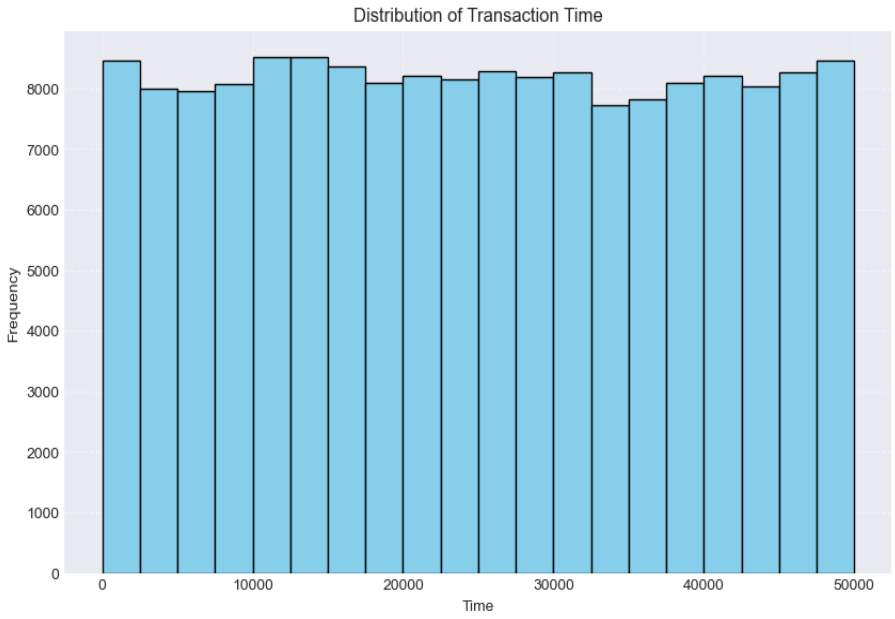
```
print(data.describe())
```

```
                Time          Value
count  163710.000000  163710.000000
mean    24949.465811      22.101443
std     14464.721250       7.416671
min         1.000000     -13.670000
25%     12471.250000      18.080000
50%     24826.500000      22.990000
75%     37547.750000      27.190000
max     49999.000000      45.470000
```

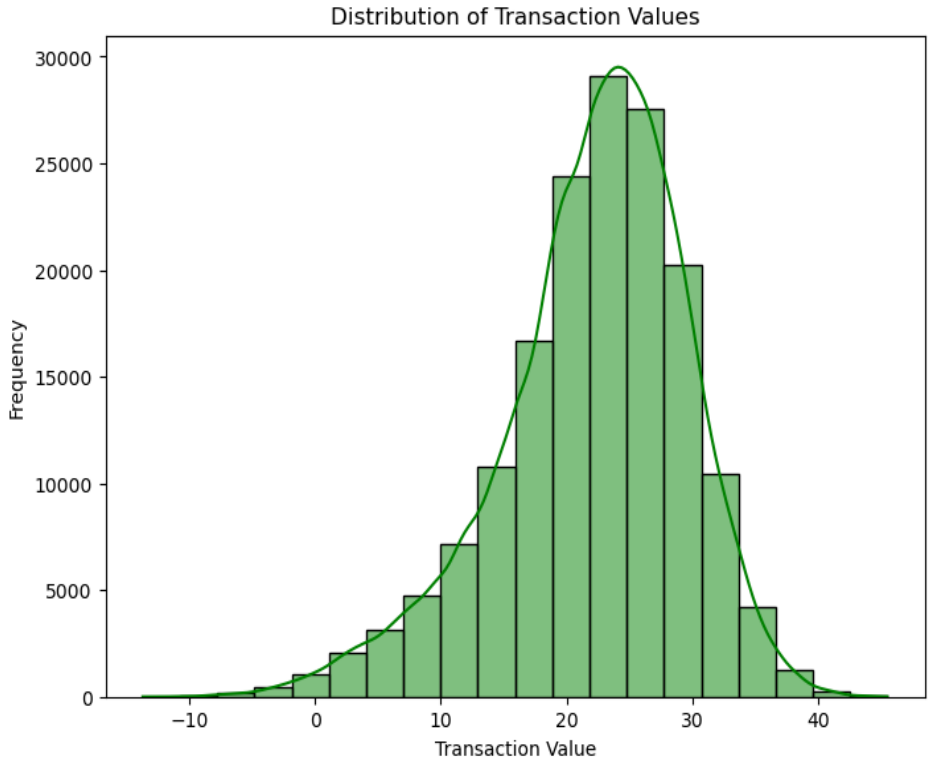
3. **Variable Analysis**: Let's look at the distribution of each variable and their relationship with the target variable (“Status”).



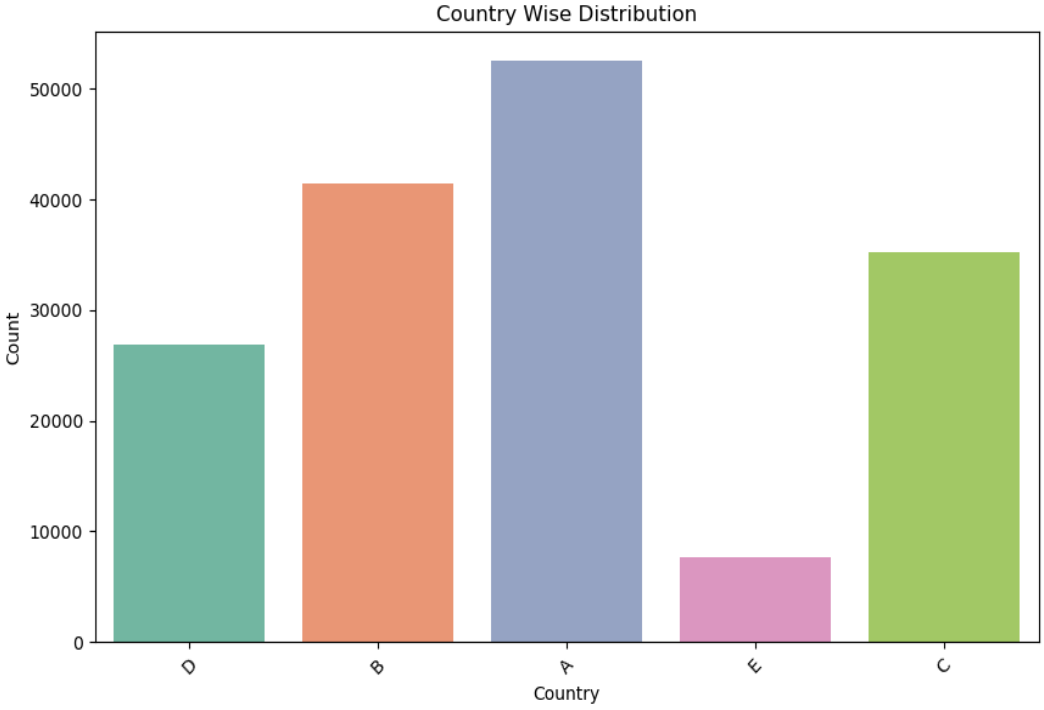
The image depicts a bar graph titled “Currency distribution.” It illustrates the unequal allocation of resources among three categories: AD, ML, and HS, with AD having the highest distribution and ML and HS significantly lower. The resulting plot shows the distribution of different currencies in the dataset, with each bar representing a specific currency type and its corresponding frequency.



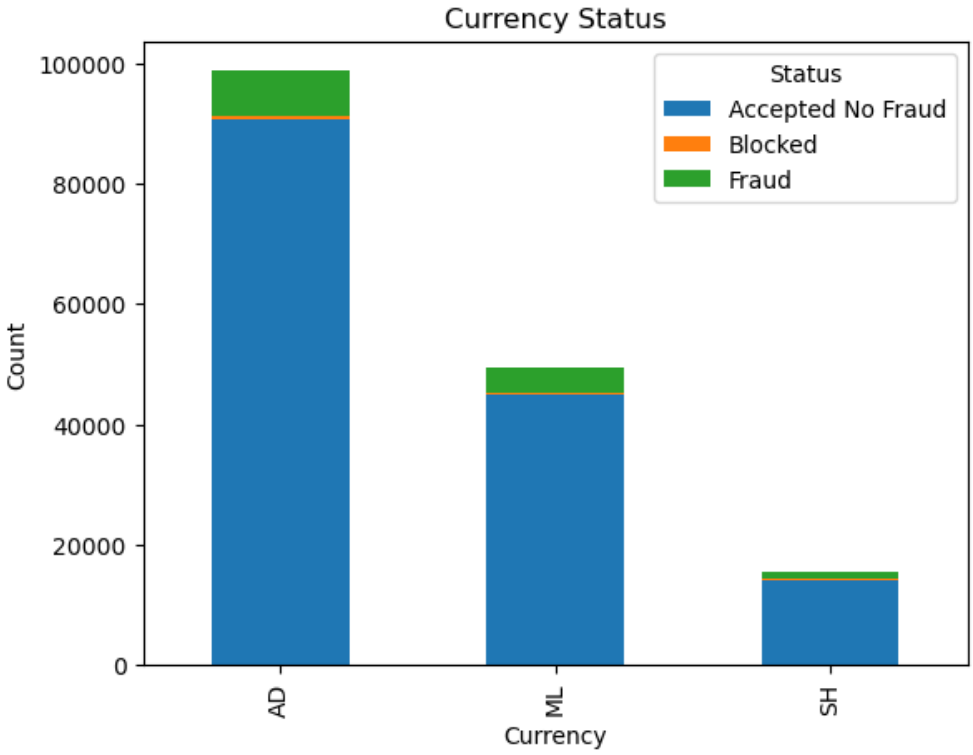
This graph illustrates the distribution of transaction “Time” among three categories: AD, ML, and HS. AD has the highest distribution, followed by ML and HS with significantly less. Each bar represents the frequency of transactions within a specific time interval.



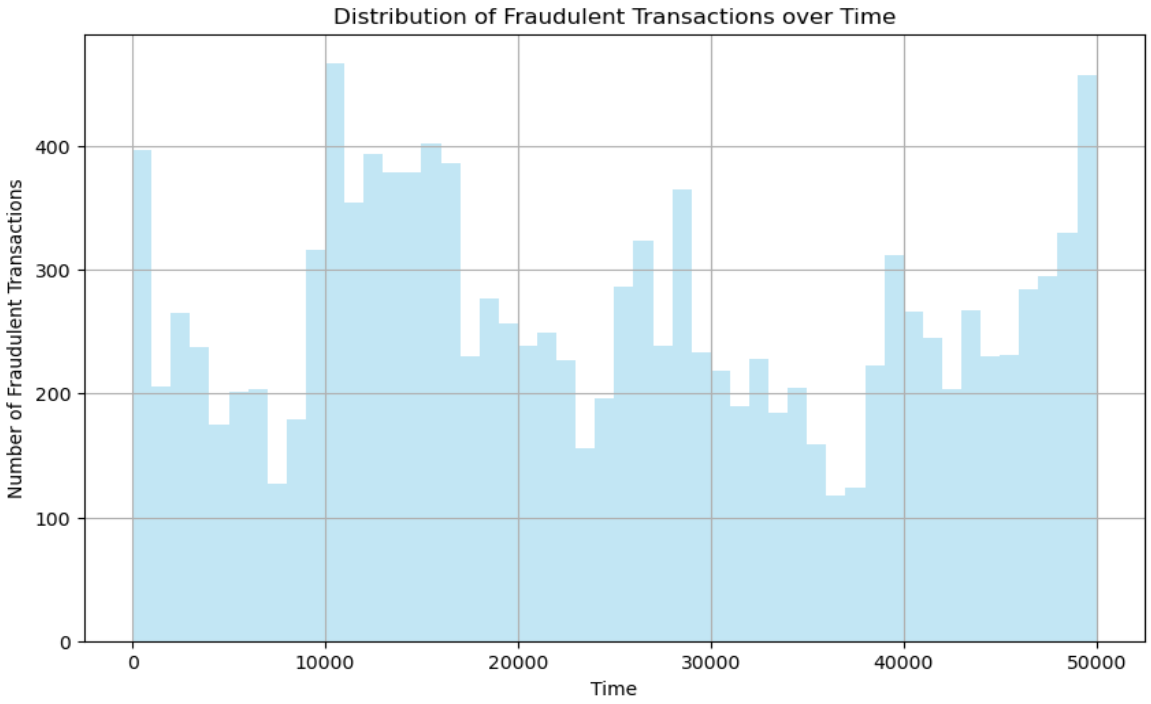
The histogram in the represents the distribution of transaction “Values”. The majority of transactions seem to occur around a positive value, suggesting common transaction amounts.The distribution appears to be right-skewed, with fewer occurrences of very high transaction values.The peak frequency indicates a specific value that occurs frequently in the dataset.Overall, this histogram provides insights into the transaction patterns and the prevalence of different transaction amounts.



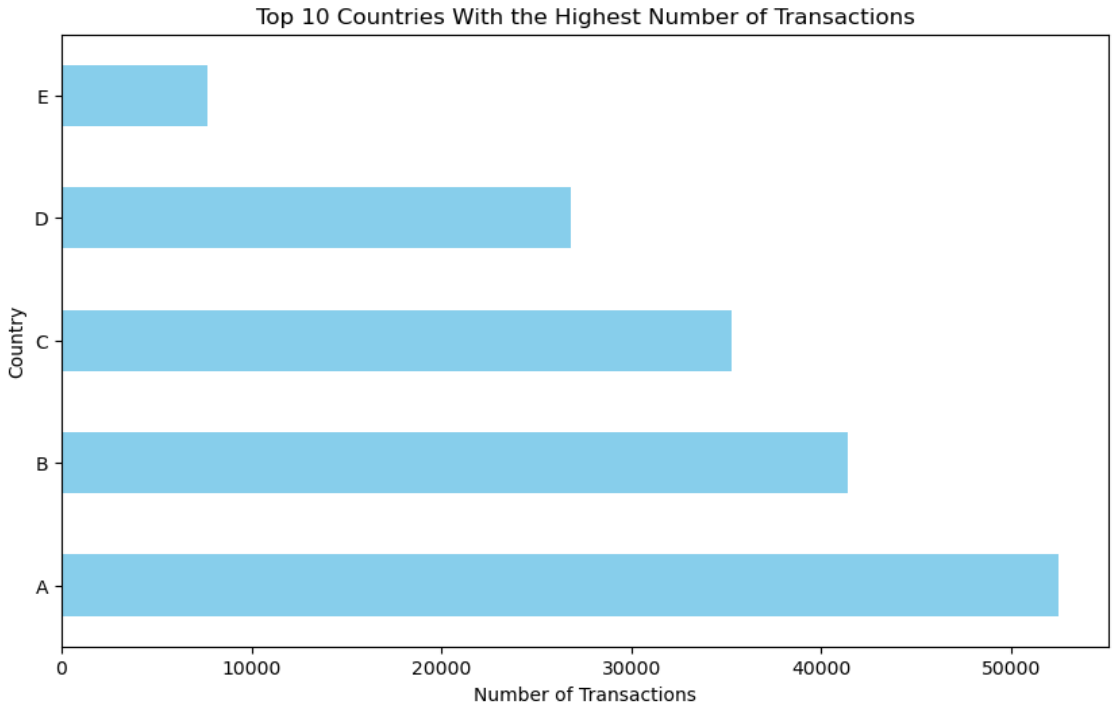
The bar chart shows the country-wise distribution of the data. The countries are represented by single-letter labels: O, S, R, L, and C.Notably, country O has the highest count, followed by S, while C has the lowest count.This visualization shows the distribution across these five countries, highlighting variations in the counts.



The bar chart shows the “Currency” status. Each currency type (AD, M, SH) is represented on the x-axis.The y-axis represents the count of different statuses (Accepted No Fraud, Blocked, Fraud).The bars are stacked, showing the distribution of each status within each currency type.The “Accepted No Fraud” status dominates across all currencies.



The histogram about the distribution of fraudulent transactions over time.The bars in sky blue colour indicate the frequency of fraudulent transactions at different time points. This visualization helps identify patterns or anomalies in fraudulent activity over time.

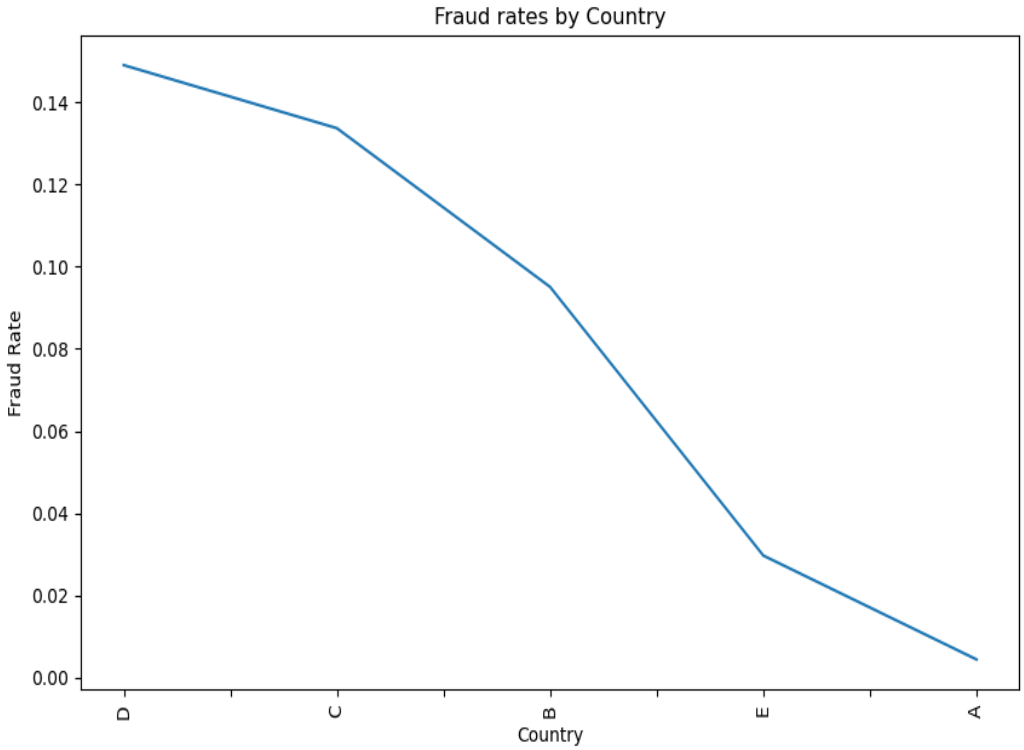


A bar chart representing the top 10 countries with the highest number of transactions.

4. Let's calculate and plot the **country with highest and lowest rate**:

```
print('Country with the highest fraud rate:', fraud_rates.idxmax(), 'Fraud Rate:', fraud_rates.max())
print('Country with the lowest fraud rate:', fraud_rates.idxmin(), 'Fraud Rate:', fraud_rates.min())
```

```
Country with the highest fraud rate: D Fraud Rate: 0.1490979573579842
Country with the lowest fraud rate: A Fraud Rate: 0.004361987847387569
```



5. Now about calculation of **average transaction** value for fraudulent transactions :

   The average of fraudulent transactions is – **20,2831$**

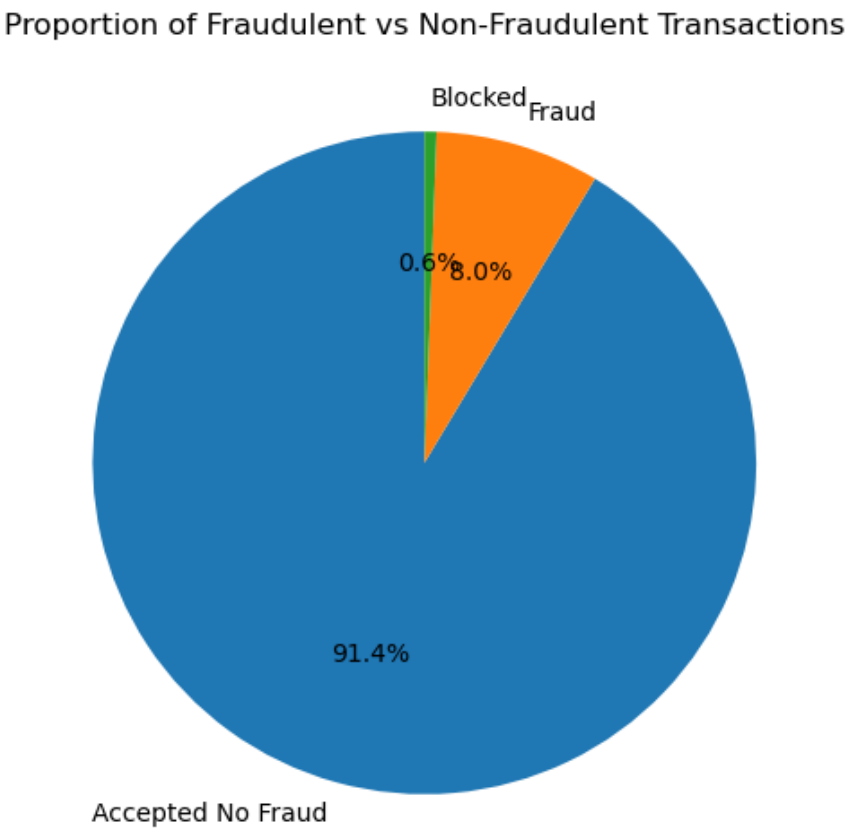
   The average of fraudulent transactions is – **22,2667$**

```
avg_value_fraud = data[data['Status'] == 'Fraud']['Value'].mean()
print(avg_value_fraud)
```

```
20.28309768931587
```

```
avg_value_nonfraud = data[data['Status'] == 'Accepted No Fraud']['Value'].mean()
print(avg_value_nonfraud)
```

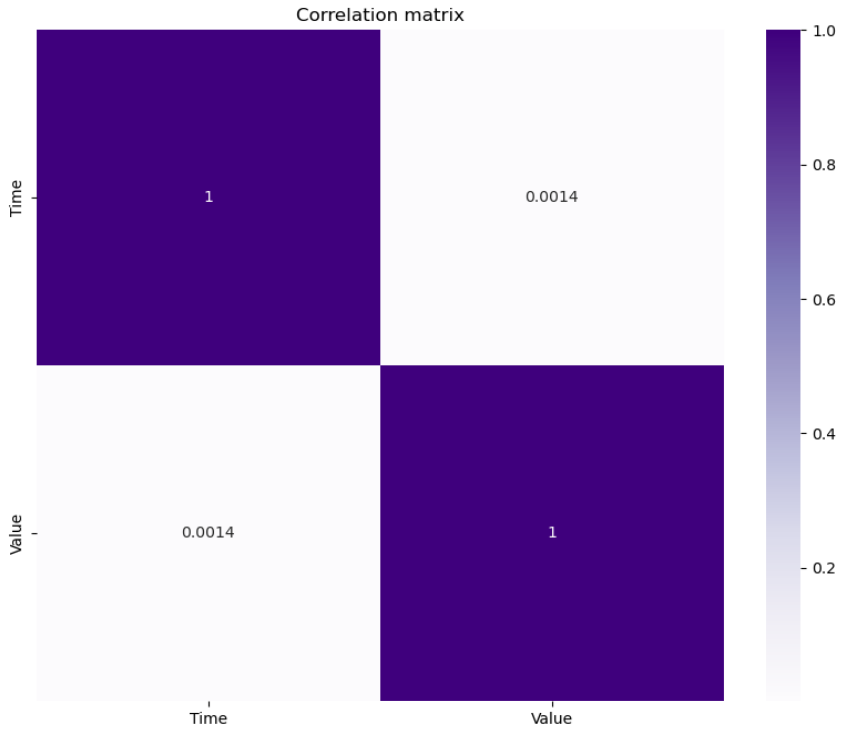
```
22.266816201333825
```



6. **Correlation Matrix.** The correlation matrix shows the pairwise correlations between the "Time"' and "Value" columns in dataset. The correlation coefficient between "Time" and "Value" is approximately **0.001378.** Since the value is close to zero, there is almost no linear correlation between these two variables.

```
corr_matrix = data.corr()
print(corr_matrix)
```

```
           Time     Value
Time   1.000000  0.001378
Value  0.001378  1.000000
```



7. **The fraud rates** for different currencies have been calculated:

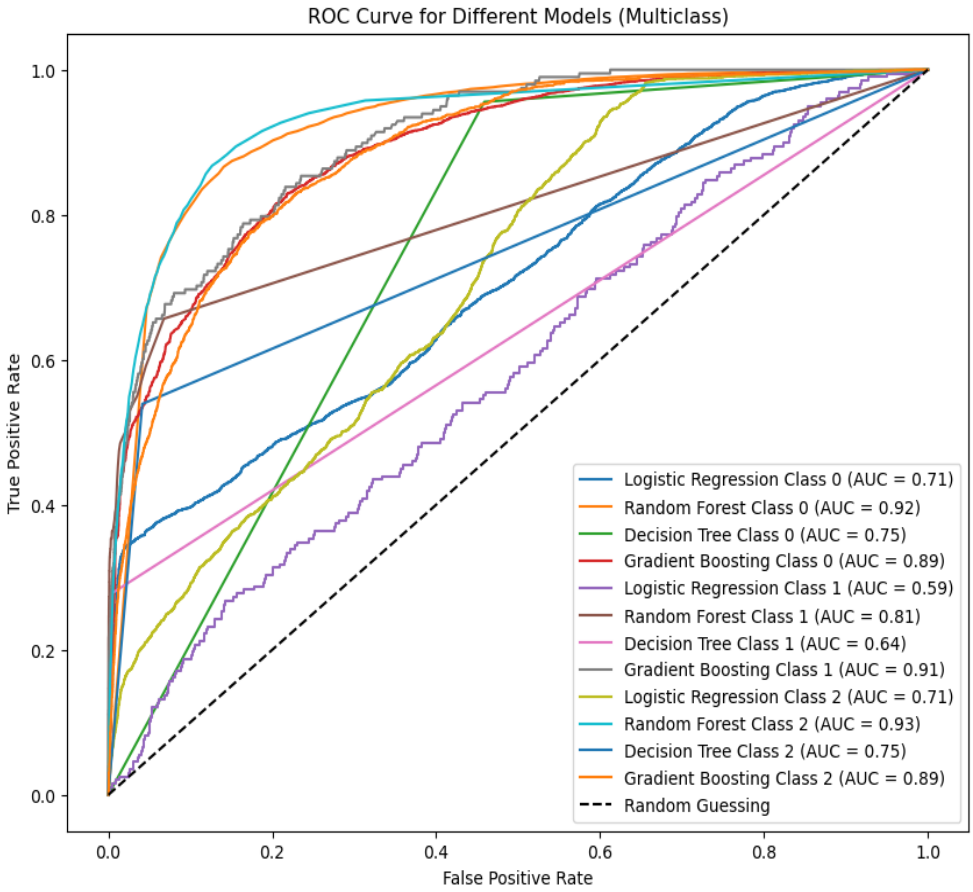
```
fraud_rates = data[data['Status'] == 'Fraud'].groupby('Currency').size()/ data.groupby('Currency').size()

print('Currency with the highest fraud rate:', fraud_rates.idxmax(), 'Fraud Rate:', fraud_rates.max())
print('Currency with the lowest fraud rate:', fraud_rates.idxmin(), 'Fraud Rate:', fraud_rates.min())
```

Currency with the highest fraud rate: ML Fraud Rate: 0.08821331178668822
Currency with the lowest fraud rate: AD Fraud Rate: 0.07649563982964916

8. **Lets set some models:**

Model name	**Model Accuracy**
Logistic Regression	0.9148494288681205
Random Forest Classifier	0.9366257406389347
DecisionTreeClassifier	0.918606071712174
GradientBoostingClassifier	0.9274021134933724

In conclusion we can say that , the best model fit is **RandomForest Classifier : 0.94**



The image displays a ROC (Receiver Operating Characteristic) curve for different machine learning models, including Logistic Regression, Random Forest, Decision Tree, and Gradient Boosting. Each model is evaluated for three different classes (Class 0, Class 1, Class 2), and their performance is represented by the AUC (Area Under the Curve) score. The curves plot the True Positive Rate against the False Positive Rate at various threshold settings. The dashed line represents random guessing.The graph shows a comparison of the effectiveness of various machine learning models in classifying data into multiple classes; it's interesting because it visually represents their performance and accuracy through ROC curves and AUC scores.

9. **Preventing Fraud Detection**:
    1. Transaction Monitoring: Implementing a monitoring system to detect unusual or suspicious transaction patterns.
    2. Biometric authentication: Considering the possibility of using biometric authentication to verify the identity of clients.
    3. Staff training and security updates: Recognizing the importance of staff training and regular software and security updates to minimize vulnerabilities.

General conclusions:

1. Identifing key factors that influence fraud in our dataset.

2. Developing and evaluating the models to predict the likelihood of fraud.
3. Fraud prevention methods based on data analysis and machine learning introducing.
4. Developing recomendations for more effective risk management and fraud protection.

Variable Analysis: Identifying the most important variables that affect the likelihood of fraud, such as transaction value, time of transaction, country, customer status, currency.

Correlations: Examining relationships between variables and identifiyng which factors correlated with fraudulent activity.

Selecting Models: Applying various machine learning models such as logistic regression, random forest, decision tree and gradient boosting to predict the likelihood of fraud.

Model evaluation: Evaluating the performance of each model using metrics such as accuracy, recall, F1-measure, ROC curve, etc.

Selecting the Best Model: Determine the most appropriate model for dataset and business goals.