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

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/CSVs/Fraud.csv')
df.head()
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDe
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M19797871
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M20442822
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	C5532640
3	1	CASH_OUT	181.00	C840083671	181.0	0.00	C389970
4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M12307017

```
df.shape
```

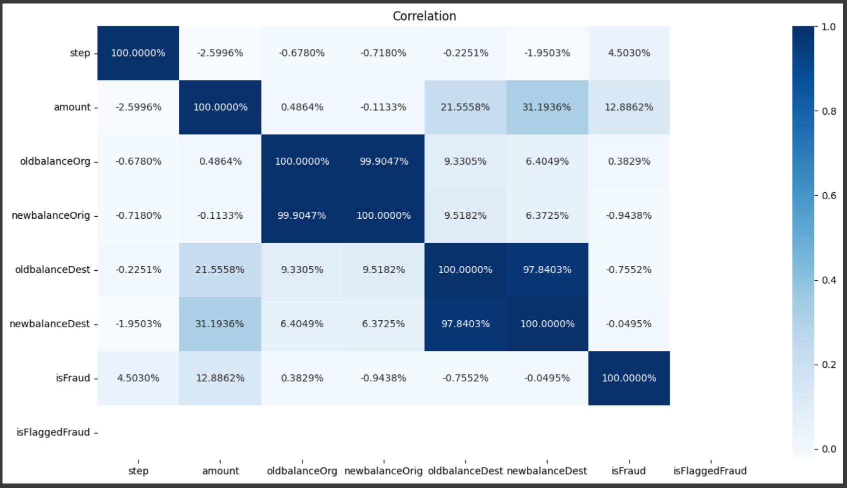
```
(1048575, 11)
```

```
df.describe()
```

	step	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	r
count	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06	
mean	2.696617e+01	1.586670e+05	8.740095e+05	8.938089e+05	9.781600e+05	
std	1.562325e+01	2.649409e+05	2.971751e+06	3.008271e+06	2.296780e+06	
min	1.000000e+00	1.000000e-01	0.000000e+00	0.000000e+00	0.000000e+00	
25%	1.500000e+01	1.214907e+04	0.000000e+00	0.000000e+00	0.000000e+00	
50%	2.000000e+01	7.634333e+04	1.600200e+04	0.000000e+00	1.263772e+05	
75%	3.900000e+01	2.137619e+05	1.366420e+05	1.746000e+05	9.159235e+05	
max	9.500000e+01	1.000000e+07	3.890000e+07	3.890000e+07	4.210000e+07	

CHECK FOR CORRELATION

```
plt.figure(figsize=(15,8))
sns.heatmap(df.corr(numeric_only= True), cmap='Blues',fmt='.4%', annot=True)
plt.title('Correlation')
plt.show()
```



```
from sklearn.preprocessing import LabelEncoder
encoder = {}
for i in df.select_dtypes('object').columns:
    encoder[i] = LabelEncoder()
    df[i] = encoder[i].fit_transform(df[i])
```

```
df.head()
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbala
0	1	3	9839.64	125164	170136.0	160296.36	274917	
1	1	3	1864.28	361319	21249.0	19384.72	286716	
2	1	4	181.00	165236	181.0	0.00	73550	
3	1	1	181.00	961662	181.0	0.00	65464	
4	1	3	11668.14	567915	41554.0	29885.86	138001	

```
x = df.drop(columns=['isFraud'])
y = df['isFraud']
```

```
y.value_counts()
```

```
0    1047433
1       1142
Name: isFraud, dtype: int64
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.30,random_state=0)
```

```
from sklearn.linear_model import LogisticRegression
```

```
log_reg = LogisticRegression()
log_reg.fit(x_train,y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
y_pred = log_reg.predict(x_test)
```

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
confusion_matrix(y_test,y_pred)
```

```
array([[314210,     30],
       [   206,    127]])
```

```
accuracy_score(y_test, y_pred)
```

```
0.9992497766814062
```

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    314240
           1       0.81      0.38      0.52       333

    accuracy                           1.00    314573
   macro avg       0.90      0.69      0.76    314573
weighted avg       1.00      1.00      1.00    314573
```

```
from sklearn.metrics import roc_curve, roc_auc_score
```

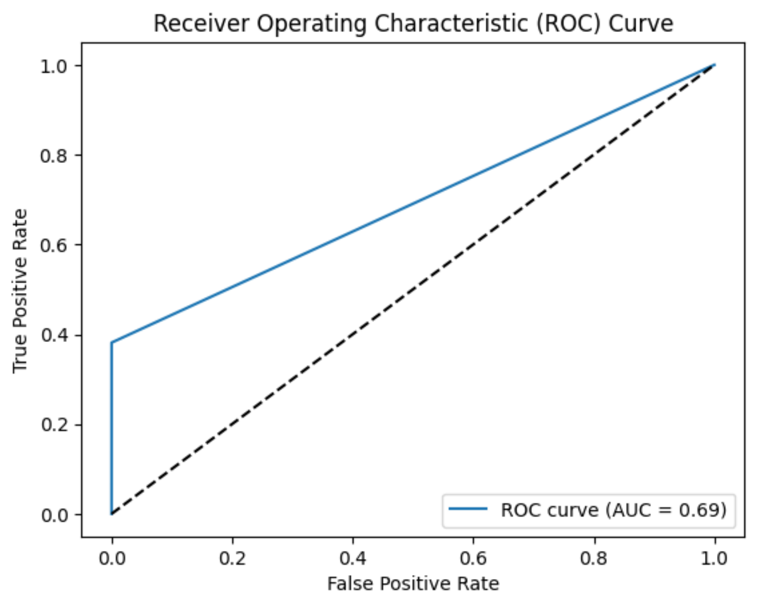
```
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, roc_auc_score

# Calculate the false positive rate, true positive rate, and threshold
fpr, tpr, thresholds = roc_curve(y_test, y_pred)

# Calculate the AUC score
auc_score = roc_auc_score(y_test, y_pred)
```

```

# Plot the ROC curve
plt.plot(fpr, tpr, label='ROC curve (AUC = {:.2f})'.format(auc_score))
plt.plot([0, 1], [0, 1], 'k--')  # Random guessing line
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc='lower right')
```


Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
ROC curve (AUC = 0.69)