```
import pandas as pd
import numpy as np
```

```
data = pd.read_csv('/content/Dataset.csv')
print(data)
```

```
   step      type    amount     nameOrig  oldbalanceOrg  newbalanceOrig  \
0     1   PAYMENT   9839.64  C1231006815         170136       160296.36
1     1   PAYMENT   1864.28  C1666544295          21249        19384.72
2     1  TRANSFER    181.00  C1305486145            181            0.00
3     1  CASH_OUT    181.00   C840083671            181            0.00
4     1   PAYMENT  11668.14  C2048537720          41554        29885.86

      nameDest  oldbalanceDest  newbalanceDest  isFraud  isFlaggedFraud
0  M1979787155               0               0        0               0
1  M2044282225               0               0        0               0
2   C553264065               0               0        1               0
3    C38997010           21182               0        1               0
4  M1230701703               0               0        0               0
```

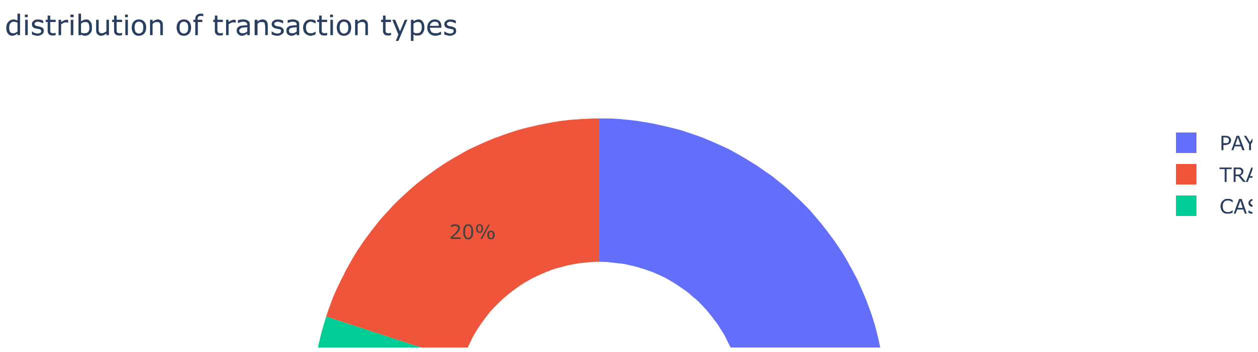
```
print(data.isnull().sum())
```

```
step              0
type              0
amount            0
nameOrig          0
oldbalanceOrg     0
newbalanceOrig    0
nameDest          0
oldbalanceDest    0
newbalanceDest    0
isFraud           0
isFlaggedFraud    0
dtype: int64
```

```
print(data.type.value_counts())
```

```
PAYMENT     3
TRANSFER    1
CASH_OUT    1
Name: type, dtype: int64
```

```
types = data["type"].value_counts()
transactions = types.index
quantity = types.values
import plotly.express as px
fig = px.pie(data,values = quantity,names = transactions,hole = 0.5,title = "distribution of transaction type
fig.show()
```



```
correlation = data.corr()
print(correlation["isFraud"].sort_values(ascending = False))
```

```
isFraud           1.000000
oldbalanceDest    0.612372
newbalanceOrig   -0.567535
oldbalanceOrg    -0.596495
amount           -0.749066
step                   NaN
newbalanceDest         NaN
isFlaggedFraud         NaN
Name: isFraud, dtype: float64
<ipython-input-16-cfeedf096772>:1: FutureWarning:

The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default
```

```
data["type"] = data["type"].map({"CASH_OUT": 1,"PAYMENT": 2,"CASH_OUT": 3,"TRANSFER": 4})
data["isFraud"] = data["isFraud"].map({0:"No Fraud",1:"Fraud"})
print(data)
```

```
   step  type    amount     nameOrig  oldbalanceOrg  newbalanceOrig  \
0     1     2   9839.64  C1231006815         170136       160296.36
1     1     2   1864.28  C1666544295          21249        19384.72
2     1     4    181.00  C1305486145            181            0.00
3     1     3    181.00   C840083671            181            0.00
4     1     2  11668.14  C2048537720          41554        29885.86

      nameDest  oldbalanceDest  newbalanceDest   isFraud  isFlaggedFraud
0  M1979787155               0               0  No Fraud               0
1  M2044282225               0               0  No Fraud               0
2   C553264065               0               0     Fraud               0
3    C38997010           21182               0     Fraud               0
4  M1230701703               0               0  No Fraud               0
```

```
from sklearn.model_selection import train_test_split
x = np.array(data[["type","amount","oldbalanceOrg","newbalanceOrig"]])
y = np.array(data[["isFraud"]])
```

```
from sklearn.tree import DecisionTreeClassifier
xtrain, xtest, ytrain, ytest = train_test_split(x,y,test_size=0.10,random_state=42)
model = DecisionTreeClassifier()
model.fit(xtrain, ytrain)
print(model.score(xtest,ytest))
```

```
1.0
```

```
features = np.array([[4,9000.60,9000.60,0.0]])
print(model.predict(features))
```

```
['Fraud']
```