In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import metrics
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
train_data = pd.read_csv('C:\\Users\\vaibhav vishal\\OneDrive\\Documents\\f
test_data = pd.read_csv('C:\\Users\\vaibhav vishal\\OneDrive\\Documents\\fr
```

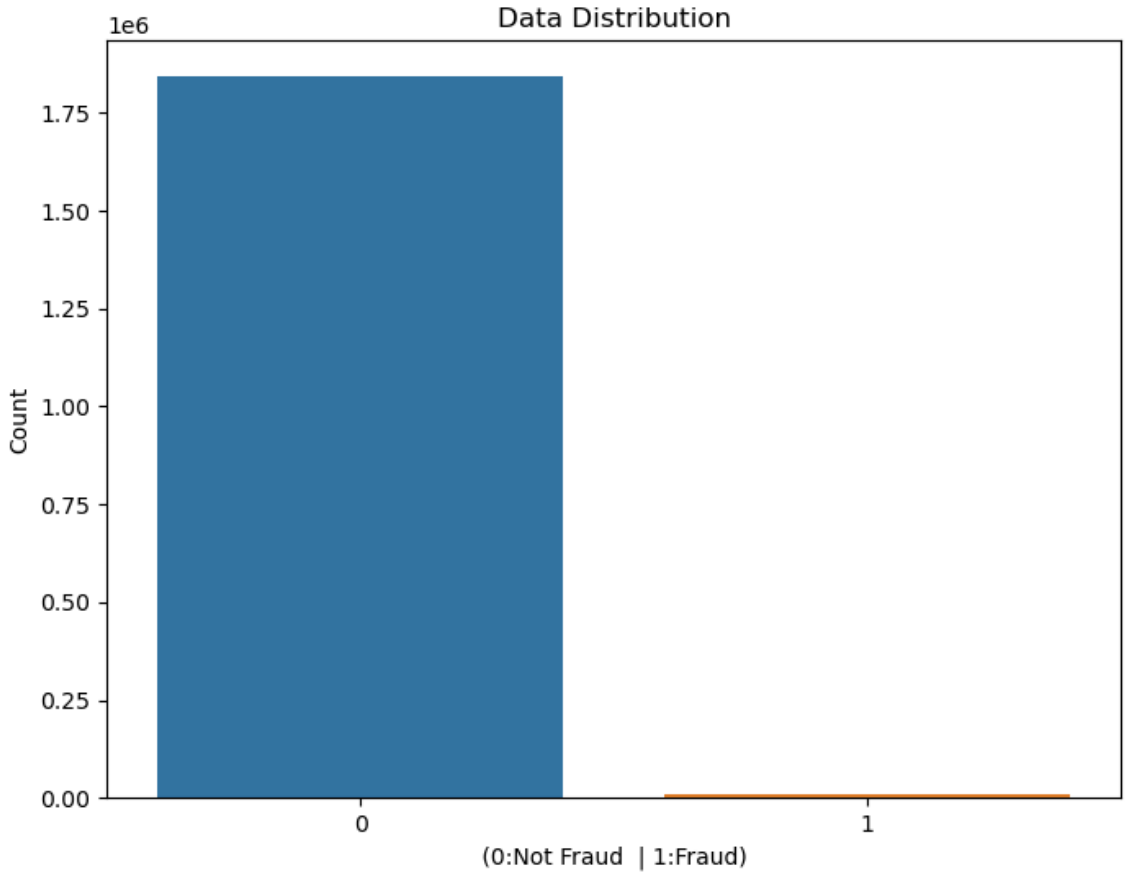
In [3]:
```
train_data.head()
```

Out[3]:

	Unnamed: 0	trans_date_trans_time	cc_num	merchant	category	amt
0	0	2019-01-01 00:00:18	2703186189652095	fraud_Rippin, Kub and Mann	misc_net	4.97
1	1	2019-01-01 00:00:44	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.23
2	2	2019-01-01 00:00:51	38859492057661	fraud_Lind-Buckridge	entertainment	220.11
3	3	2019-01-01 00:01:16	3534093764340240	fraud_Kutch, Hermiston and Farrell	gas_transport	45.00
4	4	2019-01-01 00:03:06	375534208663984	fraud_Keeling-Crist	misc_pos	41.96

5 rows × 23 columns

In [4]:
```
plt.figure(figsize=(8, 6))
sns.countplot(x='is_fraud', data=pd.concat([train_data, test_data], ignore_
plt.title('Data Distribution')
plt.xlabel(' (0:Not Fraud  | 1:Fraud) ')
plt.ylabel('Count')
plt.show()
```



In [5]:
```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1296675 entries, 0 to 1296674
Data columns (total 23 columns):
 #   Column                 Non-Null Count    Dtype  
---  ------                 --------------    -----  
 0   Unnamed: 0             1296675 non-null  int64  
 1   trans_date_trans_time  1296675 non-null  object 
 2   cc_num                 1296675 non-null  int64  
 3   merchant               1296675 non-null  object 
 4   category               1296675 non-null  object 
 5   amt                    1296675 non-null  float64
 6   first                  1296675 non-null  object 
 7   last                   1296675 non-null  object 
 8   gender                 1296675 non-null  object 
 9   street                 1296675 non-null  object 
 10  city                   1296675 non-null  object 
 11  state                  1296675 non-null  object 
 12  zip                    1296675 non-null  int64  
 13  lat                    1296675 non-null  float64
 14  long                   1296675 non-null  float64
 15  city_pop               1296675 non-null  int64  
 16  job                    1296675 non-null  object 
 17  dob                    1296675 non-null  object 
 18  trans_num              1296675 non-null  object 
 19  unix_time              1296675 non-null  int64  
 20  merch_lat              1296675 non-null  float64
 21  merch_long             1296675 non-null  float64
 22  is_fraud               1296675 non-null  int64  
dtypes: float64(5), int64(6), object(12)
memory usage: 227.5+ MB
```

In [6]:
```
test_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 555719 entries, 0 to 555718
Data columns (total 23 columns):
 #   Column                 Non-Null Count   Dtype  
---  ------                 --------------   -----  
 0   Unnamed: 0             555719 non-null  int64  
 1   trans_date_trans_time  555719 non-null  object 
 2   cc_num                 555719 non-null  int64  
 3   merchant               555719 non-null  object 
 4   category               555719 non-null  object 
 5   amt                    555719 non-null  float64
 6   first                  555719 non-null  object 
 7   last                   555719 non-null  object 
 8   gender                 555719 non-null  object 
 9   street                 555719 non-null  object 
 10  city                   555719 non-null  object 
 11  state                  555719 non-null  object 
 12  zip                    555719 non-null  int64  
 13  lat                    555719 non-null  float64
 14  long                   555719 non-null  float64
 15  city_pop               555719 non-null  int64  
 16  job                    555719 non-null  object 
 17  dob                    555719 non-null  object 
 18  trans_num              555719 non-null  object 
 19  unix_time              555719 non-null  int64  
 20  merch_lat              555719 non-null  float64
 21  merch_long             555719 non-null  float64
 22  is_fraud               555719 non-null  int64  
dtypes: float64(5), int64(6), object(12)
memory usage: 97.5+ MB
```

In [7]:
```
train_data.isnull().sum(),test_data.isnull().sum()
```

Out[7]:
```
(Unnamed: 0              0
 trans_date_trans_time    0
 cc_num                   0
 merchant                 0
 category                 0
 amt                      0
 first                    0
 last                     0
 gender                   0
 street                   0
 city                     0
 state                    0
 zip                      0
 lat                      0
 long                     0
 city_pop                 0
 job                      0
 dob                      0
 trans_num                0
 unix_time                0
 merch_lat                0
 merch_long               0
 is_fraud                 0
 dtype: int64,
 Unnamed: 0              0
 trans_date_trans_time    0
 cc_num                   0
 merchant                 0
 category                 0
 amt                      0
 first                    0
 last                     0
 gender                   0
 street                   0
 city                     0
 state                    0
 zip                      0
 lat                      0
 long                     0
 city_pop                 0
 job                      0
 dob                      0
 trans_num                0
 unix_time                0
 merch_lat                0
 merch_long               0
 is_fraud                 0
 dtype: int64)
```

In [8]:
```
cols_to_drop = ['Unnamed: 0','cc_num','merchant','first','last','trans_num'
train_data.drop(columns=cols_to_drop,inplace = True)
test_data.drop(columns=cols_to_drop,inplace = True)
```

In [9]:
```
print(train_data.shape)
print(test_data.shape)
```

```
(1296675, 14)
(555719, 14)
```

In [10]:
```
train_data['lat_dist'] = abs(round(train_data['merch_lat']-train_data['lat'
train_data['long_dist'] = abs(round(train_data['merch_long']-train_data['lo

test_data['lat_dist'] = abs(round(test_data['merch_lat']-test_data['lat'],2
test_data['long_dist'] = abs(round(test_data['merch_long']-test_data['long'
```

In [11]:
```
cols_to_drop = ['trans_date_trans_time','city','lat','long','job','dob','me
train_data.drop(columns=cols_to_drop,inplace = True)
test_data.drop(columns=cols_to_drop,inplace = True)
```

In [12]:
```
train_data.head()
```

Out[12]:

	amt	gender	zip	city_pop	is_fraud	lat_dist	long_dist
0	4.97	F	28654	3495	0	0.07	0.87
1	107.23	F	99160	149	0	0.27	0.02
2	220.11	M	83252	4154	0	0.97	0.11
3	45.00	M	59632	1939	0	0.80	0.45
4	41.96	M	24433	99	0	0.25	0.83

In [13]:
```
train_data.gender =[ 1 if value == "M" else 0 for value in train_data.gende
test_data.gender =[ 1 if value == "M" else 0 for value in test_data.gender]
```

In [14]:
```
train_data.head()
```

Out[14]:

	amt	gender	zip	city_pop	is_fraud	lat_dist	long_dist
0	4.97	0	28654	3495	0	0.07	0.87
1	107.23	0	99160	149	0	0.27	0.02
2	220.11	1	83252	4154	0	0.97	0.11
3	45.00	1	59632	1939	0	0.80	0.45
4	41.96	1	24433	99	0	0.25	0.83

In [15]:
```
#splitting data
X_train = train_data.drop('is_fraud',axis=1)
X_test = test_data.drop('is_fraud',axis=1)
y_train = train_data['is_fraud']
y_test = test_data['is_fraud']
```

In [16]:
```
print(X_train)
print(X_test)
```

```
             amt  gender    zip  city_pop  lat_dist  long_dist
0           4.97       0  28654      3495      0.07       0.87
1         107.23       0  99160       149      0.27       0.02
2         220.11       1  83252      4154      0.97       0.11
3          45.00       1  59632      1939      0.80       0.45
4          41.96       1  24433        99      0.25       0.83
...          ...     ...    ...       ...       ...        ...
1296670    15.56       1  84735       258      0.88       0.79
1296671    51.70       1  21790       100      0.36       0.74
1296672   105.93       1  88325       899      0.68       0.69
1296673    74.90       1  57756      1126      0.56       0.70
1296674     4.30       1  59871       218      0.72       0.31

[1296675 rows x 6 columns]
             amt  gender    zip  city_pop  lat_dist  long_dist
0           2.86       1  29209    333497      0.02       0.27
1          29.84       0  84002       302      0.87       0.48
2          41.28       0  11710     34496      0.18       0.66
3          60.05       1  32780     54767      0.24       0.06
4           3.19       1  49632      1126      0.71       0.87
...          ...     ...    ...       ...       ...        ...
555714     43.77       1  63453       519      0.55       0.56
555715    111.84       1  77566     28739      0.62       0.75
555716     86.88       0  99323      3684      0.46       0.81
555717      7.99       1  83643       129      0.15       0.63
555718     38.13       1  73034    116001      0.54       0.44

[555719 rows x 6 columns]
```

In [17]:
```
#LOGISTIC REGRESSION
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(X_train,y_train)
y_pred = lr.predict(X_test)
```

In [18]:
```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
```

```
Accuracy: 1.00
```

In [19]:
```
#DECISION TREE
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier(random_state = 45)
dtc.fit(X_train,y_train)
y_pred2 = dtc.predict(X_test)
```

In [20]:
```
accuracy = accuracy_score(y_test, y_pred2)
print(f'Accuracy: {accuracy:.2f}')
```

```
Accuracy: 0.99
```

In [ ]:

In [ ]:

In [ ]: