In [90]:
```
import pandas as pd
import numpy as np
```

In [91]:
```
data=pd.read_csv("fraudTest.csv")
```

In [92]:
```
pd.set_option("display.max_columns",None)
```

In [93]:
```
data.head(1)
```

Out[93]:

| | Unnamed: 0 | trans_date_trans_time | cc_num | merchant | category | amt | first | last | ge |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 21-06-2020 12:14 | 2.291160e+15 | fraud_Kirlin and Sons | personal_care | 2.86 | Jeff | Elliott | |

In [94]:
```
data.columns
```

Out[94]:
```
Index(['Unnamed: 0', 'trans_date_trans_time', 'cc_num', 'merchant', 'category',
       'amt', 'first', 'last', 'gender', 'street', 'city', 'state', 'zip',
       'lat', 'long', 'city_pop', 'job', 'dob', 'trans_num', 'unix_time',
       'merch_lat', 'merch_long', 'is_fraud'],
      dtype='object')
```

In [204…
```
data["is_fraud"].value_counts()
```

Out[204]:
```
is_fraud
0    553574
1      2145
Name: count, dtype: int64
```

In [95]:
```
from sklearn.utils import resample

# Assuming 'data_set' is your DataFrame containing the dataset
# Assuming 'is_fraud' is the column containing the class labels

# Separate majority and minority classes
majority_class = data[data['is_fraud'] == 0]
minority_class = data[data['is_fraud'] == 1]

# Downsample majority class to match the count of the minority class
majority_downsampled = resample(majority_class,
                                replace=False,  # Sample without replacement
                                n_samples=2145,  # Match the desired count of minor
                                random_state=42)  # Reproducible results

# Combine downsampled majority class with minority class
balanced_data = pd.concat([majority_downsampled, minority_class])

# Display counts of each class
print(balanced_data['is_fraud'].value_counts())
```

```
is_fraud
0    2145
1    2145
Name: count, dtype: int64
```

In [96]:
```
balanced_data.head(1)
```

Out[96]:

| | Unnamed: 0 | trans_date_trans_time | cc_num | merchant | category | amt | firs |
|---|---|---|---|---|---|---|---|
| **547885** | 547885 | 29-12-2020 19:17 | 3.712260e+14 | fraud_Medhurst, Cartwright and Ebert | personal_care | 60.51 | Stac |

In [97]:
```
balanced_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 4290 entries, 547885 to 517571
Data columns (total 23 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Unnamed: 0             4290 non-null   int64
 1   trans_date_trans_time  4290 non-null   object
 2   cc_num                 4290 non-null   float64
 3   merchant               4290 non-null   object
 4   category               4290 non-null   object
 5   amt                    4290 non-null   float64
 6   first                  4290 non-null   object
 7   last                   4290 non-null   object
 8   gender                 4290 non-null   object
 9   street                 4290 non-null   object
 10  city                   4290 non-null   object
 11  state                  4290 non-null   object
 12  zip                    4290 non-null   int64
 13  lat                    4290 non-null   float64
 14  long                   4290 non-null   float64
 15  city_pop               4290 non-null   int64
 16  job                    4290 non-null   object
 17  dob                    4290 non-null   object
 18  trans_num              4290 non-null   object
 19  unix_time              4290 non-null   int64
 20  merch_lat              4290 non-null   float64
 21  merch_long             4290 non-null   float64
 22  is_fraud               4290 non-null   int64
dtypes: float64(6), int64(5), object(12)
memory usage: 804.4+ KB
```

In [ ]:

In [98]:
```
new_data.head(1)
```

Out[98]:

| | amt | lat | long | merch_lat | merch_long | is_fraud | cc_num |
|---|---|---|---|---|---|---|---|
| **547885** | 60.51 | 38.9311 | -89.2463 | 39.205918 | -88.295627 | 0 | 3.712260e+14 |

In [99]:
```
cato_count=balanced_data["category"].value_counts()
```

In [100…
```
cato_count
```

Out[100]:
```
category
grocery_pos       687
shopping_net      655
shopping_pos      415
misc_net          385
gas_transport     352
home              281
kids_pets         244
personal_care     221
entertainment     216
misc_pos          208
food_dining       203
health_fitness    188
grocery_net       122
travel            113
Name: count, dtype: int64
```

In [101…
```
balanced_data["category"]=balanced_data["category"].apply(lambda x:x if  cato_count
```

In [102…
```
catego_dummies=pd.get_dummies(balanced_data["category"])
```

In [103…
```
catego_dummiess=catego_dummies.astype(int)
```

In [ ]:
```
```

In [104…
```
balanced_data.head(1)
```

Out[104]:

| | Unnamed: 0 | trans_date_trans_time | cc_num | merchant | category | amt | first | |
|---|---|---|---|---|---|---|---|---|
| **547885** | 547885 | 29-12-2020 19:17 | 3.712260e+14 | fraud_Medhurst, Cartwright and Ebert | other catogory | 60.51 | Stacy | L |

In [105…
```
merchant_counts=data["merchant"].value_counts()
```

In [106…
```
merchant_counts.shape
```

Out[106]:
```
(693,)
```

In [107…
```
balanced_data["merchant"]=data["merchant"].apply(lambda x:x if  merchant_counts.get
```

In [108…
```
balanced_data["merchant"].value_counts().shape
```

Out[108]:
```
(7,)
```

In [109…
```
balanced_data["merchant"].value_counts()
```

Out[109]:
```
merchant
other catogory         4201
fraud_Kilback LLC        23
fraud_Boyer PLC          21
fraud_Kuhn LLC           14
fraud_Schumm PLC         14
fraud_Dickinson Ltd       9
fraud_Cormier LLC         8
Name: count, dtype: int64
```

In [110…
```
merchant_dummies=pd.get_dummies(balanced_data["merchant"])
```

In [111…
```
merchant_dummiess=merchant_dummies.astype(int)
```

In [112…
```
data["merchant"].value_counts()
```

Out[112]:
```
merchant
fraud_Kilback LLC                       1859
fraud_Cormier LLC                       1597
fraud_Schumm PLC                        1561
fraud_Kuhn LLC                          1521
fraud_Dickinson Ltd                     1519
                                        ...
fraud_Treutel-King                       323
fraud_Satterfield-Lowe                   319
fraud_Kessler Group                      318
fraud_Jerde-Hermann                      312
fraud_Ritchie, Bradtke and Stiedemann    304
Name: count, Length: 693, dtype: int64
```

In [113…
```
balanced_data.head(1)
```

Out[113]:

| | Unnamed: 0 | trans_date_trans_time | cc_num | merchant | category | amt | first | las |
|---|---|---|---|---|---|---|---|---|
| **547885** | 547885 | 29-12-2020 19:17 | 3.712260e+14 | other catogory | other catogory | 60.51 | Stacy | Lamber |

In [115…
```
new_data=balanced_data[["amt","lat","long","merch_lat","merch_long","is_fraud","cc_
```

In [118…
```
input_data=pd.concat([new_data,catego_dummiess,merchant_dummiess],axis=1)
```

In [119…
```
input_data.head(1)
```

Out[119]:

| | amt | lat | long | merch_lat | merch_long | is_fraud | cc_num | gas_transport | g |
|---|---|---|---|---|---|---|---|---|---|
| **547885** | 60.51 | 38.9311 | -89.2463 | 39.205918 | -88.295627 | 0 | 3.712260e+14 | 0 | |

In [120…
```
x=input_data.drop(["is_fraud"],axis=1)
```

In [121…
```
y=input_data["is_fraud"]
```

In [122…
```
from sklearn.model_selection import train_test_split as tts
```

In [123…
```
x_train,x_test,y_train,y_test=tts(x,y,test_size=0.2,random_state=42)
```

In [124…
```
x_train.shape
```

Out[124]:
```
(3432, 20)
```

In [125…
```
x_test.shape
```

```
Out[125]: (858, 20)
```

```
In [141…: from sklearn.linear_model import LogisticRegression
```

```
In [147…: lr_model=LogisticRegression()
```

```
In [148…: lr_model.fit(x_train,y_train)
```

```
Out[148]:
▾ LogisticRegression
LogisticRegression()
```

```
In [149…: lr_model_ypred=lr_model.predict(x_test)
```

```
In [ ]:
```

```
In [151…: from sklearn.metrics import accuracy_score
```

```
In [152…: y_test.head()
```

```
Out[152]: 301402    1
296263    0
221912    1
372012    1
330135    1
Name: is_fraud, dtype: int64
```

```
In [153…: lr_model_ypred_score=accuracy_score(lr_model_ypred,y_test)
```

```
In [154…: lr_model_ypred_score
```

```
Out[154]: 0.5268065268065268
```

```
In [156…: from sklearn.tree import DecisionTreeClassifier
```

```
In [157…: dtc_model=DecisionTreeClassifier()
```

```
In [165…: dtc_model.fit(x_train,y_train)
```

```
Out[165]:
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
In [166…: dtc_model_ypred=dtc_model.predict(x_test)
```

```
In [167…: dtc_model_ypred_score=accuracy_score(dtc_model_ypred,y_test)
```

```
In [168…: dtc_model_ypred_score
```

```
Out[168]: 0.9230769230769231
```

```
In [171…: from sklearn.ensemble import RandomForestClassifier
```

```
In [198…: rfc_model=RandomForestClassifier(n_estimators=100)
```

In [199…
```
rfc_model.fit(x_train,y_train)
```

Out[199]:
```
▾ RandomForestClassifier
RandomForestClassifier()
```

In [200…
```
rfc_model_ypred=rfc_model.predict(x_test)
```

In [201…
```
rfc_model_ypred_score=accuracy_score(rfc_model_ypred,y_test)
```

In [202…
```
rfc_model_ypred_score
```

Out[202]:
```
0.9405594405594405
```

In [203…
```
x_test.head(1)
```

Out[203]:

| | amt | lat | long | merch_lat | merch_long | cc_num | gas_transport | grocery_p |
|---|---|---|---|---|---|---|---|---|
| **301402** | 51.86 | 36.1486 | -105.6648 | 35.672329 | -106.219095 | 5.456710e+15 | 0 | |

In [226…
```
input_value=[]
#input for the prediction
features=["amt","lat","long","merch_lat","merch_long","cc_num"]
for feature in features:
    feature=float(input(f"enter the {feature}"))
    input_value.append(feature)

#input for the category details

feature_cat=["gas_transport","grocery_pos","home","misc_net","other catogory"]
value=input(f"enter the category select any one {feature_cat}")
a=[]
for feature_c in feature_cat:
    if feature_c==feature:
        a.append(1)
    else :
        a.append(0)

#input for the merchant details

features_merchant=["shopping_net","shopping_pos","fraud_Boyer PLC","fraud_Cormier L
feature_merch=input(f"enter the merchant deatails select any one  {features_merchar
b=[]
for feature_m in features_merchant:
    if feature_m==feature_merch:
        b.append(1)
    else :
        b.append(0)
```

```
enter the amt51.86
enter the lat36.1486
enter the long-105.66
enter the merch_lat35.6723
enter the merch_long-106.219095
enter the cc_num5.456710e+15
enter the category select any one ['gas_transport', 'grocery_pos', 'home', 'misc_n
et', 'other catogory']other catogory
enter the merchant deatails select any one  ['shopping_net', 'shopping_pos', 'frau
d_Boyer PLC', 'fraud_Cormier LLC', 'fraud_Dickinson Ltd', 'fraud_Kilback LLC', 'fr
aud_Kuhn LLC', 'fraud_Schumm PLC', 'other catogory']other catogory
```

In [227…

```
input_values=input_value+a+b
```

In [228…

```
input_array = np.array(input_values).reshape(1, -1)
```

In [229…

```
prediction=rfc_model.predict(input_array)
```

```
C:\Users\BANDI GANESH\anaconda6\Lib\site-packages\sklearn\base.py:493: UserWarnin
g: X does not have valid feature names, but RandomForestClassifier was fitted with
feature names
  warnings.warn(
```

In [230…

```
if prediction== 0:
    print("it is safe transcation")
else :
    print("it is fraud transcation")
```

```
it is safe transcation
```

In [231…

```
x_test.head(1)
```

Out[231]:

| | amt | lat | long | merch_lat | merch_long | cc_num | gas_transport | grocery_p |
|---|---|---|---|---|---|---|---|---|
| **301402** | 51.86 | 36.1486 | -105.6648 | 35.672329 | -106.219095 | 5.456710e+15 | 0 | |

In [224…

```
y_test.head(1)
```

Out[224]:

```
301402    1
Name: is_fraud, dtype: int64
```

In [ ]: