```
!pip install --upgrade scikit-learn
import numpy as np
import pandas as pd
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
# from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from datetime import datetime

# from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.impute import SimpleImputer
```

```
Requirement already satisfied: scikit-learn in c:\users\hp\appdata\
local\programs\python\python311\lib\site-packages (1.5.2)
Requirement already satisfied: numpy>=1.19.5 in c:\users\hp\appdata\
local\programs\python\python311\lib\site-packages (from scikit-learn)
(1.23.5)
Requirement already satisfied: scipy>=1.6.0 in c:\users\hp\appdata\
local\programs\python\python311\lib\site-packages (from scikit-learn)
(1.10.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\hp\appdata\
local\programs\python\python311\lib\site-packages (from scikit-learn)
(1.4.0)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\hp\
appdata\local\programs\python\python311\lib\site-packages (from
scikit-learn) (3.4.0)


[notice] A new release of pip is available: 24.1.2 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
credit_card_data =
pd.read_csv('C:/Users/HP/Downloads/sample/fraudTrain.csv')

credit_card_data.head()
```

```
   Unnamed: 0        cc_num     amt    zip      lat      long
city_pop  \
0           0  2.700000e+15    4.97  28654  36.0788  -81.1781
3495
1           1  6.300000e+11  107.23  99160  48.8878 -118.2105
149
2           2  3.890000e+13  220.11  83252  42.1808 -112.2620
4154
3           3  3.530000e+15   45.00  59632  46.2306 -112.1138
1939
4           4  3.760000e+14   41.96  24433  38.4207  -79.4629
99
```

```
    unix_time  merch_lat  merch_long  is_fraud
0  1325376018  36.011293  -82.048315         0
1  1325376044  49.159047 -118.186462         0
2  1325376051  43.150704 -112.154481         0
3  1325376076  47.034331 -112.561071         0
4  1325376186  38.674999  -78.632459         0
```

```
credit_card_data.tail()
```

```
         Unnamed: 0        cc_num     amt    zip      lat     long
city_pop  \
1048570     1048570  6.010000e+15   77.00  21405  39.0305 -76.5515
92106
1048571     1048571  4.840000e+15  116.94  52563  41.1826 -92.3097
1583
1048572     1048572  5.720000e+11   21.27  40202  38.2507 -85.7476
736284
1048573     1048573  4.650000e+18    9.52  11796  40.7320 -73.1000
4056
1048574     1048574  2.280000e+15    6.81  30009  34.0770 -84.3033
165556

          unix_time  merch_lat  merch_long  is_fraud
1048570  1362931649  38.779464  -76.317042         0
1048571  1362931670  41.400318  -92.726724         0
1048572  1362931711  37.293339  -84.798122         0
1048573  1362931718  39.773077  -72.213209         0
1048574  1362931730  33.601468  -83.891921         0
```

```
credit_card_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 11 columns):
 #   Column      Non-Null Count    Dtype
---  ------      --------------    -----
 0   Unnamed: 0  1048575 non-null  int64
 1   cc_num      1048575 non-null  float64
 2   amt         1048575 non-null  float64
 3   zip         1048575 non-null  int64
 4   lat         1048575 non-null  float64
 5   long        1048575 non-null  float64
 6   city_pop    1048575 non-null  int64
 7   unix_time   1048575 non-null  int64
 8   merch_lat   1048575 non-null  float64
 9   merch_long  1048575 non-null  float64
 10  is_fraud    1048575 non-null  int64
dtypes: float64(6), int64(5)
memory usage: 88.0 MB
```

```
credit_card_data.isnull().sum()
```

```
Unnamed: 0    0
cc_num        0
amt           0
zip           0
lat           0
long          0
city_pop      0
unix_time     0
merch_lat     0
merch_long    0
is_fraud      0
dtype: int64
```

```
credit_card_data['is_fraud'].value_counts()
```

```
0    1042569
1       6006
Name: is_fraud, dtype: int64
```

```
legit = credit_card_data[credit_card_data.is_fraud == 0]
fraud = credit_card_data[credit_card_data.is_fraud == 1]

print(legit.shape)
print(fraud.shape)
```

```
(1042569, 11)
(6006, 11)
```

```
legit.amt.describe()
```

```
count    1.042569e+06
mean     6.762744e+01
std      1.536956e+02
min      1.000000e+00
25%      9.600000e+00
50%      4.722000e+01
75%      8.247000e+01
max      2.894890e+04
Name: amt, dtype: float64
```

```
fraud.amt.describe()
```

```
count    6006.000000
mean      530.573492
std       391.333069
min         1.180000
25%       241.577500
50%       391.165000
75%       901.950000
```

```
max      1371.810000
Name: amt, dtype: float64
```

```
credit_card_data.groupby('is_fraud').mean()
```

```
              Unnamed: 0        cc_num         amt            zip
lat  \
is_fraud

0         524494.707643  4.174085e+17   67.627445  48805.355338
38.532842
1         488231.463869  3.775093e+17  530.573492  48148.078422
38.623988

                long      city_pop     unix_time  merch_lat  merch_long

is_fraud

0        -90.228376  89015.900200  1.344913e+09  38.532993  -90.228625

1        -89.858250  96323.951715  1.343602e+09  38.615091  -89.853555
```

```
legit_sample = legit.sample(n=7506)
```

```
new_dataset = pd.concat([legit_sample, fraud], axis=0)
```

```
new_dataset.head()
```

```
        Unnamed: 0        cc_num     amt    zip      lat      long
city_pop  \
170743      170743  6.010000e+15    2.30   7640  40.9918  -73.9800
4664
175676      175676  4.560000e+12   16.60  46143  39.5960  -86.1309
78968
401030      401030  3.560000e+15  109.66  13367  43.7893  -75.4156
8830
954930      954930  5.360000e+15   13.18  59714  45.7801 -111.1439
18182
221508      221508  3.420000e+14  149.04  31046  33.1194  -83.8235
3343

         unix_time  merch_lat  merch_long  is_fraud
170743  1333173061  40.314668  -73.228635         0
175676  1333308064  39.862638  -87.051289         0
401030  1341130941  44.317929  -74.886163         0
954930  1358545416  45.486894 -110.613095         0
221508  1335076694  33.678792  -83.865876         0
```

```
new_dataset.tail()
```

```
         Unnamed: 0        cc_num     amt    zip      lat     long
city_pop  \
1047089     1047089  3.590000e+15  690.49  57374  43.7557 -97.5936
343
1047157     1047157  3.550000e+15  324.74  76008  32.7004 -97.6039
13602
1047208     1047208  3.590000e+15  331.33  57374  43.7557 -97.5936
343
1047521     1047521  3.590000e+15  356.20  57374  43.7557 -97.5936
343
1047918     1047918  3.590000e+15  249.56  57374  43.7557 -97.5936
343

          unix_time  merch_lat  merch_long  is_fraud
1047089  1362887989  43.254214  -98.267759         1
1047157  1362889904  33.607221  -97.996506         1
1047208  1362891561  44.228731  -98.330520         1
1047521  1362903771  43.988931  -97.989985         1
1047918  1362917373  42.868322  -98.537668         1
```

```
new_dataset['is_fraud'].value_counts()
```

```
0    7506
1    6006
Name: is_fraud, dtype: int64
```

```
new_dataset.groupby('is_fraud').mean()
```

```
             Unnamed: 0        cc_num         amt           zip
lat  \
is_fraud

0         521477.470157  4.236582e+17   67.587298  48996.770317
38.454139
1         488231.463869  3.775093e+17  530.573492  48148.078422
38.623988

                long      city_pop     unix_time  merch_lat  merch_long

is_fraud

0         -90.297414  89116.262590  1.344810e+09  38.456461  -90.295584

1         -89.858250  96323.951715  1.343602e+09  38.615091  -89.853555
```

```
X = new_dataset.drop(columns='is_fraud', axis=1)
Y = new_dataset['is_fraud']

print(X)
```

```
         Unnamed: 0        cc_num     amt    zip      lat      long
city_pop  \
170743       170743  6.010000e+15    2.30   7640  40.9918  -73.9800
4664
175676       175676  4.560000e+12   16.60  46143  39.5960  -86.1309
78968
401030       401030  3.560000e+15  109.66  13367  43.7893  -75.4156
8830
954930       954930  5.360000e+15   13.18  59714  45.7801 -111.1439
18182
221508       221508  3.420000e+14  149.04  31046  33.1194  -83.8235
3343
...             ...           ...     ...    ...      ...       ...
...
1047089     1047089  3.590000e+15  690.49  57374  43.7557  -97.5936
343
1047157     1047157  3.550000e+15  324.74  76008  32.7004  -97.6039
13602
1047208     1047208  3.590000e+15  331.33  57374  43.7557  -97.5936
343
1047521     1047521  3.590000e+15  356.20  57374  43.7557  -97.5936
343
1047918     1047918  3.590000e+15  249.56  57374  43.7557  -97.5936
343

          unix_time  merch_lat  merch_long
170743   1333173061  40.314668  -73.228635
175676   1333308064  39.862638  -87.051289
401030   1341130941  44.317929  -74.886163
954930   1358545416  45.486894 -110.613095
221508   1335076694  33.678792  -83.865876
...             ...        ...         ...
1047089  1362887989  43.254214  -98.267759
1047157  1362889904  33.607221  -97.996506
1047208  1362891561  44.228731  -98.330520
1047521  1362903771  43.988931  -97.989985
1047918  1362917373  42.868322  -98.537668

[13512 rows x 10 columns]

print(Y)

170743     0
175676     0
401030     0
954930     0
221508     0
          ..
1047089    1
1047157    1
```

```
1047208    1
1047521    1
1047918    1
Name: is_fraud, Length: 13512, dtype: int64
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, stratify=Y, random_state=2)
```

```
print(X.shape, X_train.shape, X_test.shape)
```

```
(13512, 10) (10809, 10) (2703, 10)
```

```
model = LogisticRegression()
```

```
model.fit(X_train, Y_train)
```

```
LogisticRegression()
```

```
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

```
print('Accuracy on Training data : ', training_data_accuracy)
```

```
Accuracy on Training data :  0.5554630400592099
```

```
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

```
print('Accuracy score on Test Data : ', test_data_accuracy)
```

```
Accuracy score on Test Data :  0.5556788753237144
```