```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix,classification_report

train_data = pd.read_csv("fraudtrain.csv")

train_data.head()
```

```
   Unnamed: 0 trans_date_trans_time            cc_num  \
0           0   2019-01-01 00:00:18  2703186189652095
1           1   2019-01-01 00:00:44      630423337322
2           2   2019-01-01 00:00:51    38859492057661
3           3   2019-01-01 00:01:16  3534093764340240
4           4   2019-01-01 00:03:06   375534208663984

                             merchant       category     amt
first  \
0          fraud_Rippin, Kub and Mann       misc_net    4.97
Jennifer
1     fraud_Heller, Gutmann and Zieme    grocery_pos  107.23
Stephanie
2                fraud_Lind-Buckridge  entertainment  220.11
Edward
3  fraud_Kutch, Hermiston and Farrell  gas_transport   45.00
Jeremy
4                 fraud_Keeling-Crist       misc_pos   41.96
Tyler

      last gender                         street  ...      lat
long  \
0    Banks      F                561 Perry Cove  ...  36.0788  -
81.1781
1     Gill      F  43039 Riley Greens Suite 393  ...  48.8878 -
118.2105
2  Sanchez      M      594 White Dale Suite 530  ...  42.1808 -
112.2620
3    White      M   9443 Cynthia Court Apt. 038  ...  46.2306 -
112.1138
4   Garcia      M              408 Bradley Rest  ...  38.4207  -
79.4629

   city_pop                                job         dob  \
0      3495          Psychologist, counselling  1988-03-09
1       149  Special educational needs teacher  1978-06-21
2      4154        Nature conservation officer  1962-01-19
3      1939                    Patent attorney  1967-01-12
4        99     Dance movement psychotherapist  1986-03-28
```

```
                          trans_num   unix_time  merch_lat  merch_long
\
0  0b242abb623afc578575680df30655b9  1325376018  36.011293  -82.048315

1  1f76529f8574734946361c461b024d99  1325376044  49.159047 -118.186462

2  a1a22d70485983eac12b5b88dad1cf95  1325376051  43.150704 -112.154481

3  6b849c168bdad6f867558c3793159a81  1325376076  47.034331 -112.561071

4  a41d7549acf90789359a9aa5346dcb46  1325376186  38.674999  -78.632459


   is_fraud
0         0
1         0
2         0
3         0
4         0

[5 rows x 23 columns]
```

```
train_data.isnull().sum()
```

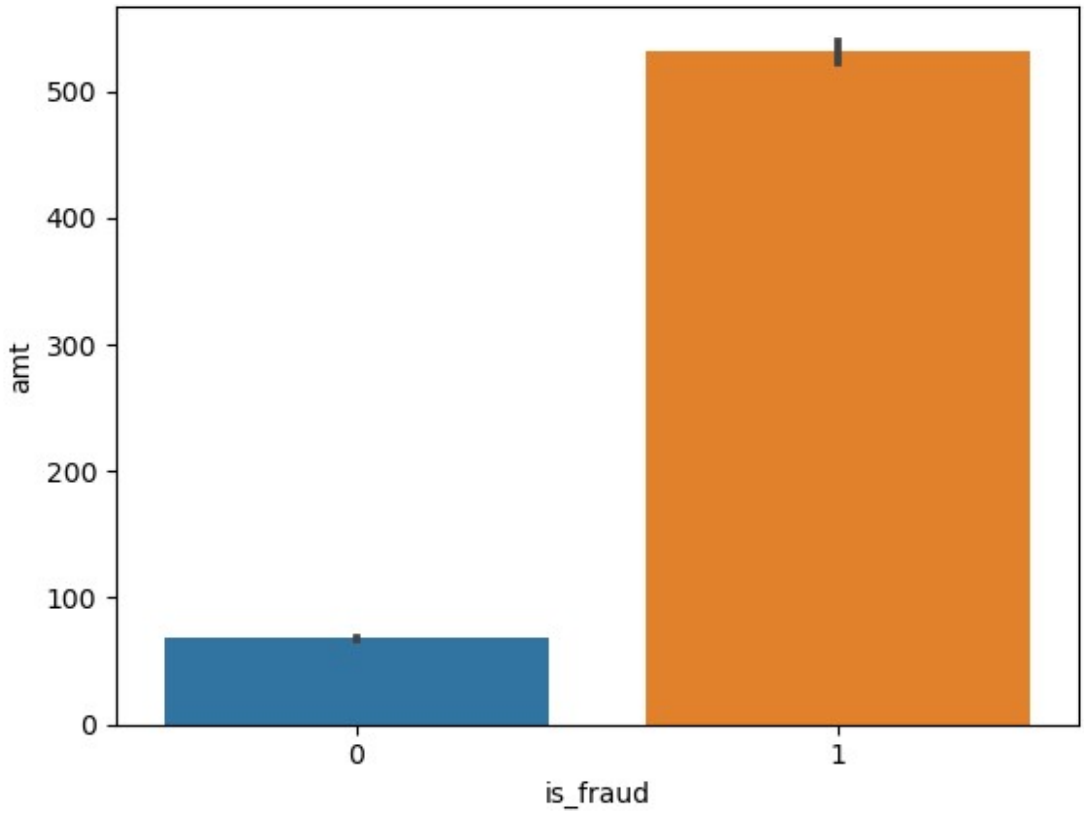
```
Unnamed: 0                0
trans_date_trans_time     0
cc_num                    0
merchant                  0
category                  0
amt                       0
first                     0
last                      0
gender                    0
street                    0
city                      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
```

```
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

train_data.describe()

         Unnamed: 0        cc_num           amt            zip  
lat  \
count  1.296675e+06  1.296675e+06  1.296675e+06  1.296675e+06  
1.296675e+06
mean   6.483370e+05  4.171920e+17  7.035104e+01  4.880067e+04  
3.853762e+01
std    3.743180e+05  1.308806e+18  1.603160e+02  2.689322e+04  
5.075808e+00
min    0.000000e+00  6.041621e+10  1.000000e+00  1.257000e+03  
2.002710e+01
25%    3.241685e+05  1.800429e+14  9.650000e+00  2.623700e+04  
3.462050e+01
50%    6.483370e+05  3.521417e+15  4.752000e+01  4.817400e+04  
3.935430e+01
75%    9.725055e+05  4.642255e+15  8.314000e+01  7.204200e+04  
4.194040e+01
max    1.296674e+06  4.992346e+18  2.894890e+04  9.978300e+04
```

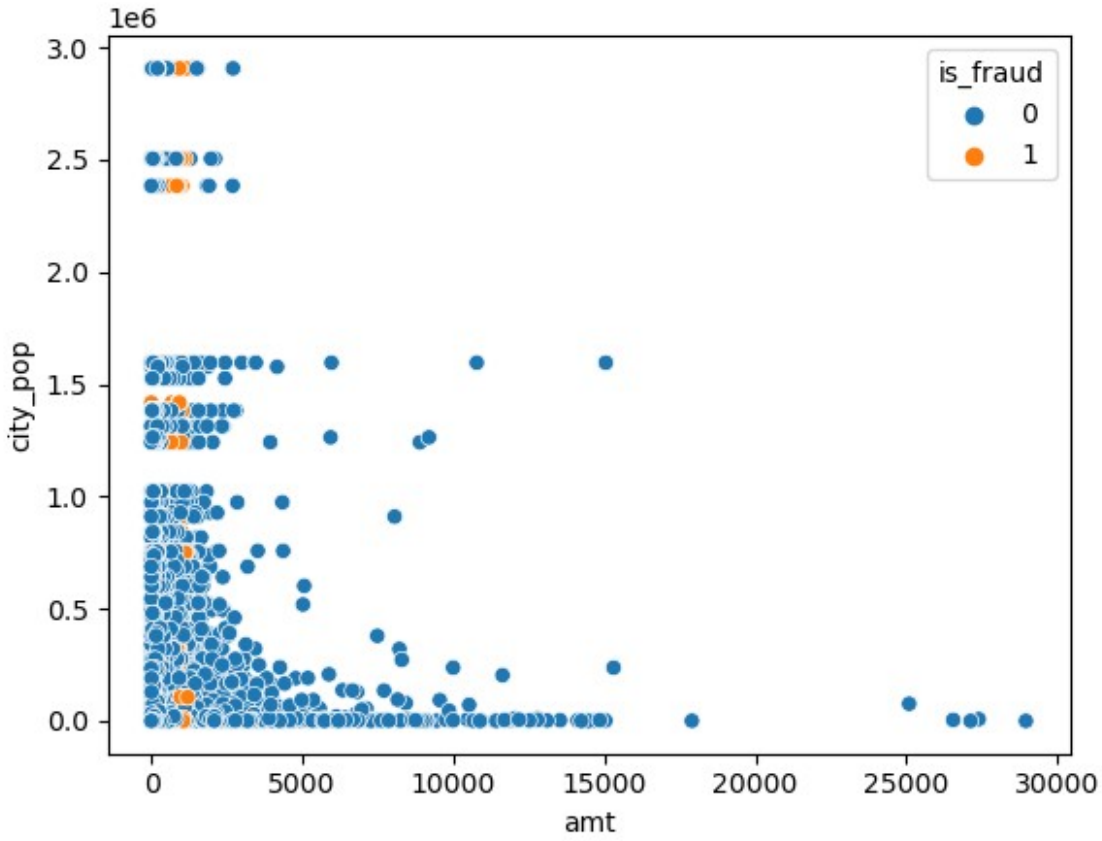
```
6.669330e+01

                long      city_pop     unix_time     merch_lat
merch_long  \
count  1.296675e+06  1.296675e+06  1.296675e+06  1.296675e+06
1.296675e+06
mean  -9.022634e+01  8.882444e+04  1.349244e+09  3.853734e+01 -
9.022646e+01
std    1.375908e+01  3.019564e+05  1.284128e+07  5.109788e+00
1.377109e+01
min   -1.656723e+02  2.300000e+01  1.325376e+09  1.902779e+01 -
1.666712e+02
25%   -9.679800e+01  7.430000e+02  1.338751e+09  3.473357e+01 -
9.689728e+01
50%   -8.747690e+01  2.456000e+03  1.349250e+09  3.936568e+01 -
8.743839e+01
75%   -8.015800e+01  2.032800e+04  1.359385e+09  4.195716e+01 -
8.023680e+01
max   -6.795030e+01  2.906700e+06  1.371817e+09  6.751027e+01 -
6.695090e+01

           is_fraud
count  1.296675e+06
mean   5.788652e-03
std    7.586269e-02
min    0.000000e+00
25%    0.000000e+00
50%    0.000000e+00
75%    0.000000e+00
max    1.000000e+00

 train_data.duplicated()

0          False
1          False
2          False
3          False
4          False
           ...
1296670    False
1296671    False
1296672    False
1296673    False
1296674    False
Length: 1296675, dtype: bool

train_data['is_fraud'].unique()

array([0, 1], dtype=int64)

sns.barplot(data=train_data,x='is_fraud',y='amt')
```

```
<Axes: xlabel='is_fraud', ylabel='amt'>
```



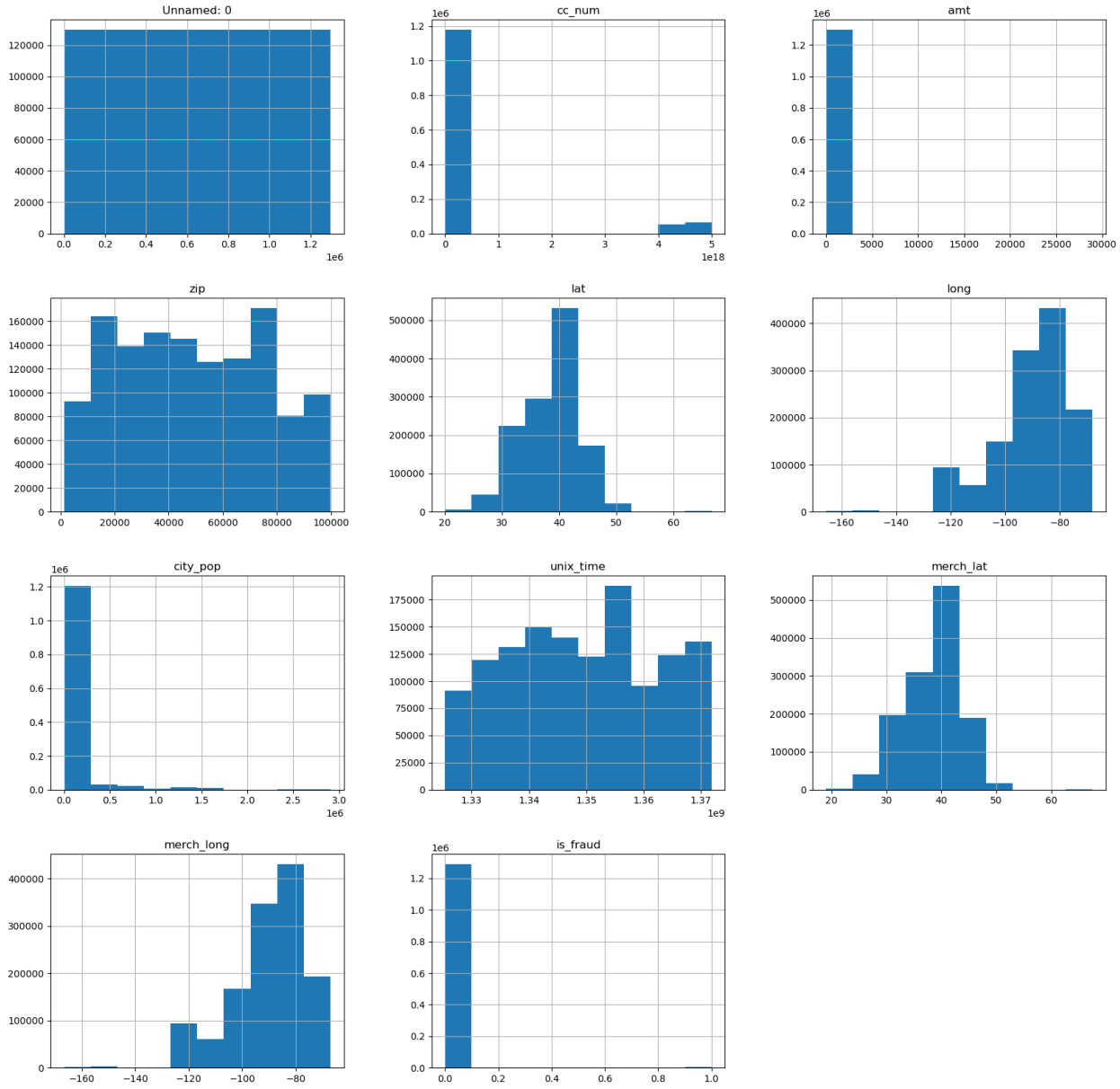

```
sns.scatterplot(data=train_data,x='amt',y='city_pop',hue='is_fraud')
```

```
<Axes: xlabel='amt', ylabel='city_pop'>
```




```
train_data.hist(figsize=(20,20))
plt.legend('is_fraud')
```

```
<matplotlib.legend.Legend at 0x2050fd16150>
```




```
Fraud=train_data[train_data['is_fraud']==1]
Valid=train_data[train_data['is_fraud']==0]

outlier_fraction=len(Fraud)/float(len(Valid))
print(outlier_fraction)

print('Fraud Cases: {}'.format(len(Fraud)))
print('Valid Cases: {}'.format(len(Valid)))
```

```
0.005822355331224998
Fraud Cases: 7506
Valid Cases: 1289169
```

```
X=train_data.drop(['is_fraud','trans_date_trans_time','first','last','
merchant','category','gender','street','city','state','job','dob','tra
ns_num'],axis=1)
y=train_data['is_fraud']

scaler = StandardScaler()
X  = scaler.fit_transform(X)

log_model = LogisticRegression()

log_model.fit(X,y)

LogisticRegression()

log_model.get_params()

{'C': 1.0,
 'class_weight': None,
 'dual': False,
 'fit_intercept': True,
 'intercept_scaling': 1,
 'l1_ratio': None,
 'max_iter': 100,
 'multi_class': 'deprecated',
 'n_jobs': None,
 'penalty': 'l2',
 'random_state': None,
 'solver': 'lbfgs',
 'tol': 0.0001,
 'verbose': 0,
 'warm_start': False}

log_model.coef_

array([[-0.02919181, -0.01279864,  0.44007827, -0.01175853,
0.02094865,
         0.0091826 ,  0.02054498, -0.04220605,  0.0143182 ,
0.00877577]])

test_data = pd.read_csv('fraudTest.csv')

X_test =
test_data.drop(['is_fraud','trans_date_trans_time','first','last','mer
chant','category','gender','street','city','state','job','dob','trans_
num'],axis=1)

y_test = test_data['is_fraud']

X_test = scaler.transform(X_test)

y_pred = log_model.predict(X_test)
```

```
confusion_matrix(y_test,y_pred)
```

```
array([[553222,    352],
       [  2145,      0]], dtype=int64)
```

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

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    555367
           1       0.00      0.00      0.00       352

    accuracy                           1.00    555719
   macro avg       0.50      0.50      0.50    555719
weighted avg       1.00      1.00      1.00    555719
```