```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

train = pd.read_csv('/content/fraudTrain.csv')
test = pd.read_csv('/content/fraudTest.csv')
%matplotlib inline


from scipy.stats import skew
from scipy.stats import norm
from scipy.special import boxcox1p, inv_boxcox
from datetime import date, datetime
import time
print(train.isnull().sum().sum())
print(test.isnull().sum().sum())
print(train.isna().sum().sum())
print(test.isna().sum().sum())
```

```
2
13
2
13
```

```
test.isnull().values.any()
test.head(10)
```

	Unnamed: 0	trans_date_trans_time	cc_num	merchant	category
0	0	2020-06-21 12:14:25	2291163933867244	fraud_Kirlin and Sons	personal_care
1	1	2020-06-21 12:14:33	3573030041201292	fraud_Sporer-Keebler	personal_care
2	2	2020-06-21 12:14:53	3598215285024754	fraud_Swaniawski, Nitzsche and Welch	health_fitness
3	3	2020-06-21 12:15:15	3591919803438423	fraud_Haley Group	misc_pos
4	4	2020-06-21 12:15:17	3526826139003047	fraud_Johnston-Casper	travel
5	5	2020-06-21 12:15:37	30407675418785	fraud_Daugherty LLC	kids_pets
6	6	2020-06-21 12:15:44	213180742685905	fraud_Romaguera Ltd	health_fitness
7	7	2020-06-21 12:15:50	3589289942931264	fraud_Reichel LLC	personal_care
8	8	2020-06-21 12:16:10	3596357274378601	fraud_Goyette, Howell and Collier	shopping_pos
9	9	2020-06-21 12:16:11	3546897637165774	fraud_Kilback Group	food_dining

10 rows × 23 columns

```
sns.countplot(train['cc_num'])
```

```
<Axes: ylabel='count'>
```


```
train.groupby('cc_num').size()
```

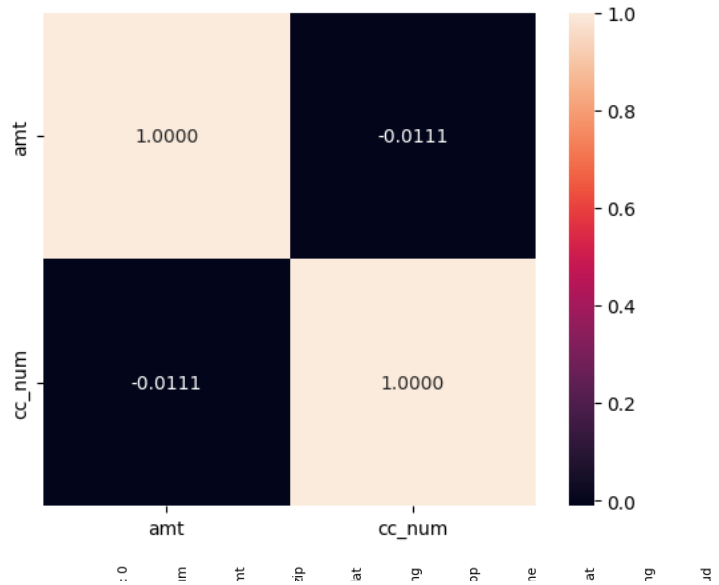
```
cc_num
60416207185            18
60422928733             9
60423098130             9
60427851591             6
60487002085             4
                       ..
4958589671582726883    19
4973530368125489546    12
4980323467523543940     3
4989847570577635369    13
4992346398065154184    23
Length: 908, dtype: int64
```

```
plt.subplots(figsize=(12, 9))
sns.heatmap(train.corr(), square=True, annot=True, fmt='.2f')
```

```
<ipython-input-7-fc8450d4d3f5>:2: FutureWarning: The default value of numeric_only in
  sns.heatmap(train.corr(), square=True, annot=True, fmt='.2f')
<Axes: >
```

```
sns.heatmap(train[['amt', 'cc_num']].corr(), fmt='.4f', annot=True, square=True)
```

```
<Axes: >
```




```
train[train['city_pop'] > 1000]['cc_num'].describe()
```

```
count    1.083900e+04
mean     4.203040e+17
std      1.318053e+18
min      6.041621e+10
25%      2.131141e+14
50%      3.527537e+15
75%      4.671727e+15
max      4.989848e+18
Name: cc_num, dtype: float64
```

```
train['lat'].describe()
```

```
count    15593.000000
mean        38.537959
std          5.164709
min         20.027100
25%         34.690200
50%         39.342600
75%         41.811400
max         65.689900
Name: lat, dtype: float64
```

```
sns.distplot(train['zip'])
```

```
<ipython-input-11-3601ca12c53b>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```

```
sns.distplot(train['lat'])
```

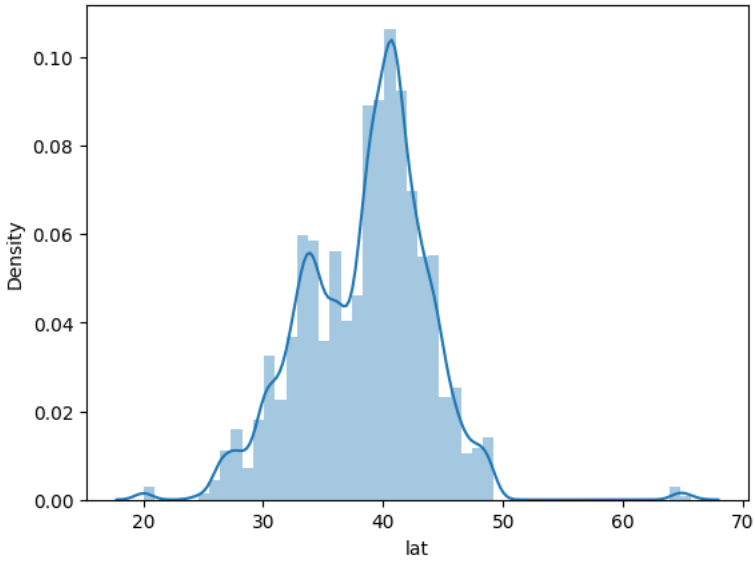
```
<ipython-input-12-74ca85e6132f>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(train['lat'])
<Axes: xlabel='lat', ylabel='Density'>
```




```
c = ['unix_time','merch_lat','merch_long']
from sklearn import preprocessing
scaler = preprocessing.RobustScaler()
train_X = scaler.fit_transform(train[c])


train_X = pd.DataFrame(train_X, columns=c)


sns.distplot(train_X['merch_lat'])
```

```
<ipython-input-15-800b8910e484>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(train_X['merch_lat'])
<Axes: xlabel='merch_lat', ylabel='Density'>
```

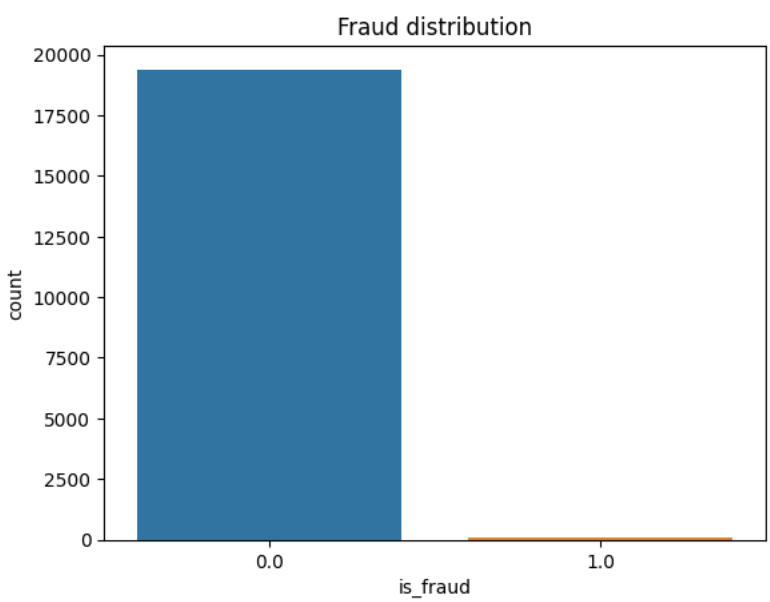
```
train_X.columns
```

```
Index(['unix_time', 'merch_lat', 'merch_long'], dtype='object')
```

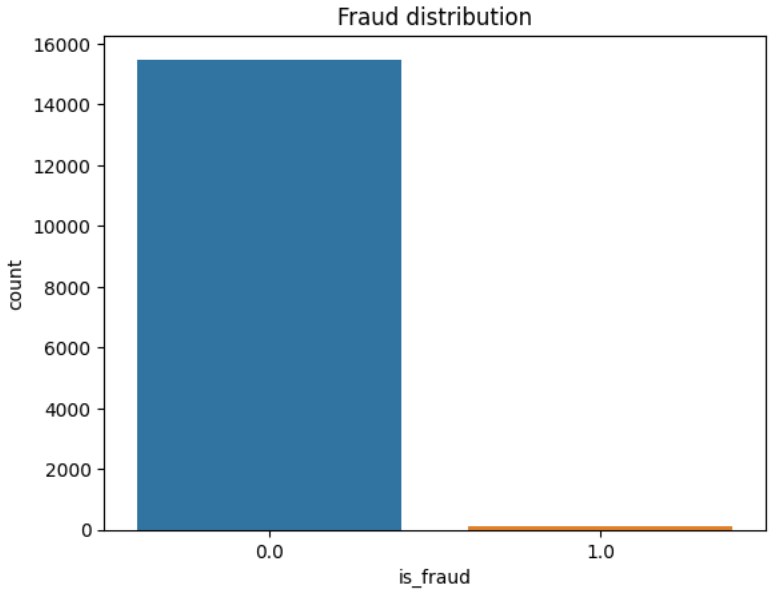
```
training_features = train_X
```

```
training_target = train['cc_num']
```

```
sns.countplot(x='is_fraud', data=test)
plt.title('Fraud distribution')
plt.show()
```




```
sns.countplot(x='is_fraud', data=train)
plt.title('Fraud distribution')
plt.show()
```



```
processed_df = pd.get_dummies(
    data=test,
    columns=['state', 'gender'],
    drop_first=True
)
```

```
processed_df.head()
```

	Unnamed: 0	trans_date_trans_time	cc_num	merchant	category	
0	0	2020-06-21 12:14:25	2291163933867244	fraud_Kirlin and Sons	personal_care	
1	1	2020-06-21 12:14:33	3573030041201292	fraud_Sporer-Keebler	personal_care	1
2	2	2020-06-21 12:14:53	3598215285024754	fraud_Swaniawski, Nitzsche and Welch	health_fitness	4
3	3	2020-06-21 12:15:15	3591919803438423	fraud_Haley Group	misc_pos	0
4	4	2020-06-21 12:15:17	3526826139003047	fraud_Johnston-Casper	travel	

5 rows × 71 columns

```
correlation_matrix = test[['is_fraud','amt']].corr()
plt.figure(figsize=(15, 10))
sns.heatmap(data=correlation_matrix, annot=True)
```

```
<Axes: >
```


```
processed_df = pd.get_dummies(
    data=train,
    columns=['state', 'gender'],
    drop_first=True
)
```

```
processed_df.head()
```

	Unnamed: 0	trans_date_trans_time	cc_num	merchant	category	a
0	0	2019-01-01 00:00:18	2703186189652095	fraud_Rippin, Kub and Mann	misc_net	4.
1	1	2019-01-01 00:00:44	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.
2	2	2019-01-01 00:00:51	38859492057661	fraud_Lind-Buckridge	entertainment	220.
3	3	2019-01-01 00:01:16	3534093764340240	fraud_Kutch, Hermiston and Farrell	gas_transport	45.
4	4	2019-01-01 00:03:06	375534208663984	fraud_Keeling-Crist	misc_pos	41.

5 rows × 71 columns

```
correlation_matrix = train[['is_fraud','amt']].corr()
plt.figure(figsize=(15, 10))
sns.heatmap(data=correlation_matrix, annot=True)
```

<Axes: >