In [3]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [4]:
```
test_data=pd.read_csv('fraudTest.csv')
train_data=pd.read_csv('fraudTrain.csv')
```

In [5]:
```
test_data.head()
```

Out[5]:

	Unnamed: 0	trans_date_trans_time	cc_num	merchant	category	amt	first	last	gender	street	...	la
0	0	2020-06-21 12:14:25	2291163933867244	fraud_Kirlin and Sons	personal_care	2.86	Jeff	Elliott	M	351 Darlene Green	...	33.965
1	1	2020-06-21 12:14:33	3573030041201292	fraud_Sporer-Keebler	personal_care	29.84	Joanne	Williams	F	3638 Marsh Union	...	40.320
2	2	2020-06-21 12:14:53	3598215285024754	fraud_Swaniawski, Nitzsche and Welch	health_fitness	41.28	Ashley	Lopez	F	9333 Valentine Point	...	40.672
3	3	2020-06-21 12:15:15	3591919803438423	fraud_Haley Group	misc_pos	60.05	Brian	Williams	M	32941 Krystal Mill Apt. 552	...	28.569
4	4	2020-06-21 12:15:17	3526826139003047	fraud_Johnston-Casper	travel	3.19	Nathan	Massey	M	5783 Evan Roads Apt. 465	...	44.252

5 rows × 23 columns

In [6]:
```
train_data.head()
```

Out[6]:

	Unnamed: 0	trans_date_trans_time	cc_num	merchant	category	amt	first	last	gender	street	...	lat
0	0	2019-01-01 00:00:18	2703186189652095	fraud_Rippin, Kub and Mann	misc_net	4.97	Jennifer	Banks	F	561 Perry Cove	...	36.0788
1	1	2019-01-01 00:00:44	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.23	Stephanie	Gill	F	43039 Riley Greens Suite 393	...	48.8878
2	2	2019-01-01 00:00:51	38859492057661	fraud_Lind-Buckridge	entertainment	220.11	Edward	Sanchez	M	594 White Dale Suite 530	...	42.1808
3	3	2019-01-01 00:01:16	3534093764340240	fraud_Kutch, Hermiston and Farrell	gas_transport	45.00	Jeremy	White	M	9443 Cynthia Court Apt. 038	...	46.2306
4	4	2019-01-01 00:03:06	375534208663984	fraud_Keeling-Crist	misc_pos	41.96	Tyler	Garcia	M	408 Bradley Rest	...	38.4207

5 rows × 23 columns

In [7]:
```
#train data nul
train_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
```

In [8]:
```
#train data nul
test_data.isnull().sum()
```

Out[8]:
```
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 [9]:
```
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 [10]:
```
train_data.describe()
```

Out[10]:

	Unnamed: 0	cc_num	amt	zip	lat	long	city_pop	unix_time	merch_lat	n
count	1.296675e+06	1.296675e+06	1.296675e+06	1.296675e+06	1.296675e+06	1.296675e+06	1.296675e+06	1.296675e+06	1.296675e+06	1.2
mean	6.483370e+05	4.171920e+17	7.035104e+01	4.880067e+04	3.853762e+01	-9.022634e+01	8.882444e+04	1.349244e+09	3.853734e+01	-9.0
std	3.743180e+05	1.308806e+18	1.603160e+02	2.689322e+04	5.075808e+00	1.375908e+01	3.019564e+05	1.284128e+07	5.109788e+00	1.3
min	0.000000e+00	6.041621e+10	1.000000e+00	1.257000e+03	2.002710e+01	-1.656723e+02	2.300000e+01	1.325376e+09	1.902779e+01	-1.6
25%	3.241685e+05	1.800429e+14	9.650000e+00	2.623700e+04	3.462050e+01	-9.679800e+01	7.430000e+02	1.338751e+09	3.473357e+01	-9.6
50%	6.483370e+05	3.521417e+15	4.752000e+01	4.817400e+04	3.935430e+01	-8.747690e+01	2.456000e+03	1.349250e+09	3.936568e+01	-8.7
75%	9.725055e+05	4.642255e+15	8.314000e+01	7.204200e+04	4.194040e+01	-8.015800e+01	2.032800e+04	1.359385e+09	4.195716e+01	-8.0
max	1.296674e+06	4.992346e+18	2.894890e+04	9.978300e+04	6.669330e+01	-6.795030e+01	2.906700e+06	1.371817e+09	6.751027e+01	-6.6

In [11]:
```
train_data=train_data.drop(['Unnamed: 0'],axis=1)
```

In [12]:
```
train_data.head(2)
```

Out[12]:

	trans_date_trans_time	cc_num	merchant	category	amt	first	last	gender	street	city	...	lat	
0	2019-01-01 00:00:18	2703186189652095	fraud_Rippin, Kub and Mann	misc_net	4.97	Jennifer	Banks	F	561 Perry Cove	Moravian Falls	...	36.0788	-81.
1	2019-01-01 00:00:44	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.23	Stephanie	Gill	F	43039 Riley Greens Suite 393	Orient	...	48.8878	-118.

2 rows × 22 columns

In [13]:
```
test_data=test_data.drop(['Unnamed: 0'],axis=1)
test_data.head(2)
```

Out[13]:

	trans_date_trans_time	cc_num	merchant	category	amt	first	last	gender	street	city	...	lat	
0	2020-06-21 12:14:25	2291163933867244	fraud_Kirlin and Sons	personal_care	2.86	Jeff	Elliott	M	351 Darlene Green	Columbia	...	33.9659	-80
1	2020-06-21 12:14:33	3573030041201292	fraud_Sporer-Keebler	personal_care	29.84	Joanne	Williams	F	3638 Marsh Union	Altonah	...	40.3207	-110

2 rows × 22 columns

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

```
C:\Users\User\AppData\Local\Temp\ipykernel_13508\1402113604.py:1: FutureWarning: The default value of numeric_o
nly in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns o
r specify the value of numeric_only to silence this warning.
  train_data.corr()
```

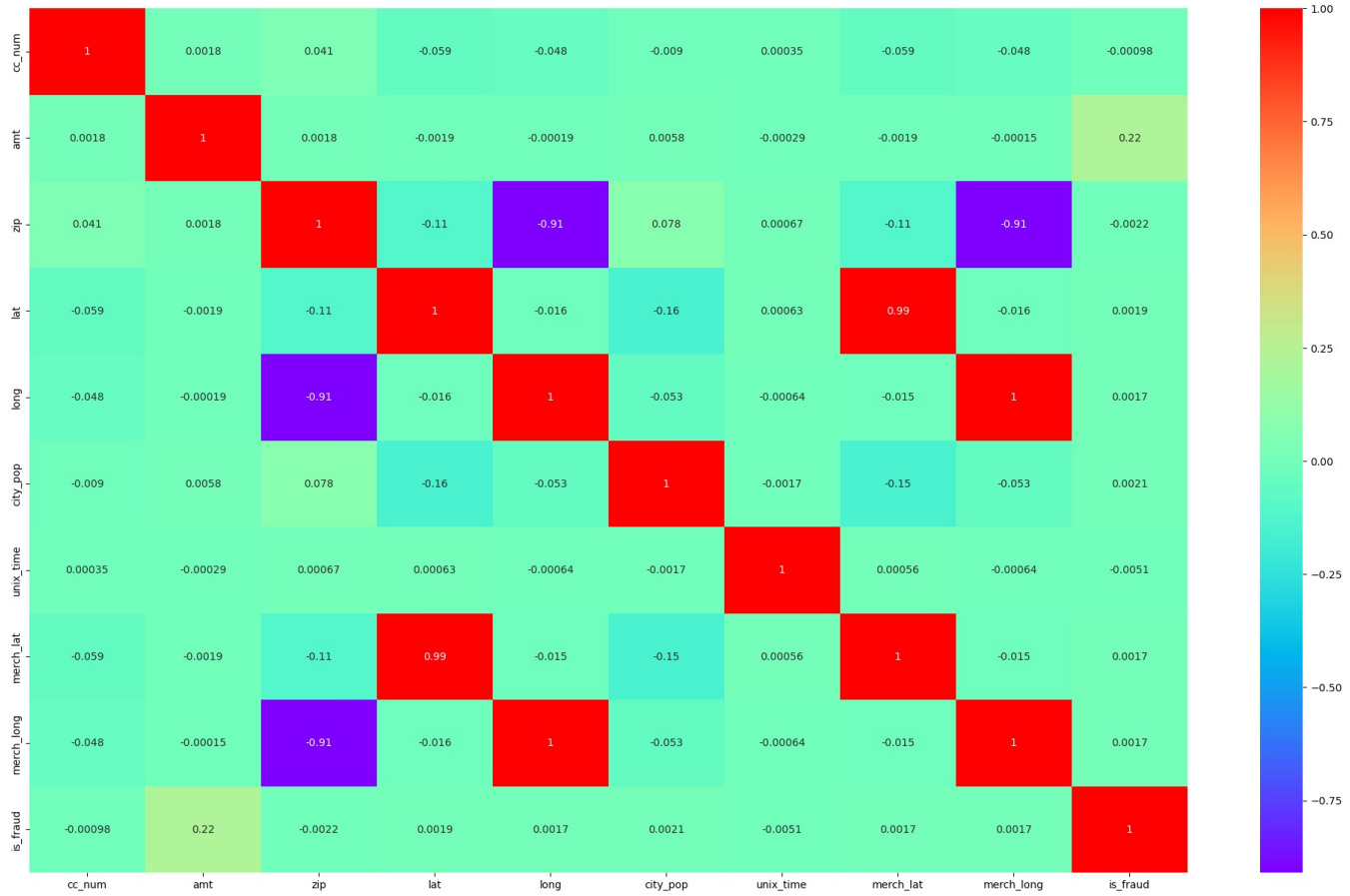
Out[14]:

	cc_num	amt	zip	lat	long	city_pop	unix_time	merch_lat	merch_long	is_fraud
cc_num	1.000000	0.001769	0.041459	-0.059271	-0.048278	-0.008991	0.000354	-0.058942	-0.048252	-0.000981
amt	0.001769	1.000000	0.001843	-0.001926	-0.000187	0.005818	-0.000293	-0.001873	-0.000151	0.219404
zip	0.041459	0.001843	1.000000	-0.114290	-0.909732	0.078467	0.000670	-0.113561	-0.908924	-0.002162
lat	-0.059271	-0.001926	-0.114290	1.000000	-0.015533	-0.155730	0.000632	0.993592	-0.015509	0.001894
long	-0.048278	-0.000187	-0.909732	-0.015533	1.000000	-0.052715	-0.000642	-0.015452	0.999120	0.001721
city_pop	-0.008991	0.005818	0.078467	-0.155730	-0.052715	1.000000	-0.001714	-0.154781	-0.052687	0.002136
unix_time	0.000354	-0.000293	0.000670	0.000632	-0.000642	-0.001714	1.000000	0.000561	-0.000635	-0.005078
merch_lat	-0.058942	-0.001873	-0.113561	0.993592	-0.015452	-0.154781	0.000561	1.000000	-0.015431	0.001741
merch_long	-0.048252	-0.000151	-0.908924	-0.015509	0.999120	-0.052687	-0.000635	-0.015431	1.000000	0.001721
is_fraud	-0.000981	0.219404	-0.002162	0.001894	0.001721	0.002136	-0.005078	0.001741	0.001721	1.000000

In [15]:
```
plt.figure(figsize=(25,15))
sns.heatmap(train_data.corr(),annot=True,cmap='rainbow')
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_13508\3809277712.py:2: FutureWarning: The default value of numeric_o
nly in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns o
r specify the value of numeric_only to silence this warning.
  sns.heatmap(train_data.corr(),annot=True,cmap='rainbow')
```

Out[15]:
```
<Axes: >
```



so,zip-long 0.9 zip-merchantlong 0.9 lat-merchantchat 0.9 long-merchantlong 0.9

so,can use zip and lat instead of other columns

In [16]:
```
new_train_data=train_data.drop(['merchant','merch_long','long'],axis=1)
new_train_data.head(2)
```

Out[16]:

	trans_date_trans_time	cc_num	category	amt	first	last	gender	street	city	state	zip	lat	city_pop
0	2019-01-01 00:00:18	2703186189652095	misc_net	4.97	Jennifer	Banks	F	561 Perry Cove	Moravian Falls	NC	28654	36.0788	3495
1	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	Stephanie	Gill	F	43039 Riley Greens Suite 393	Orient	WA	99160	48.8878	149

In [17]:
```
new_test_data=test_data.drop(['merchant','merch_long','long'],axis=1)
new_test_data
```

Out[17]:

	trans_date_trans_time	cc_num	category	amt	first	last	gender	street	city	state	zip	lat
0	2020-06-21 12:14:25	2291163933867244	personal_care	2.86	Jeff	Elliott	M	351 Darlene Green	Columbia	SC	29209	33.9659
1	2020-06-21 12:14:33	3573030041201292	personal_care	29.84	Joanne	Williams	F	3638 Marsh Union	Altonah	UT	84002	40.3207
2	2020-06-21 12:14:53	3598215285024754	health_fitness	41.28	Ashley	Lopez	F	9333 Valentine Point	Bellmore	NY	11710	40.6729
3	2020-06-21 12:15:15	3591919803438423	misc_pos	60.05	Brian	Williams	M	32941 Krystal Mill Apt. 552	Titusville	FL	32780	28.5697
4	2020-06-21 12:15:17	3526826139003047	travel	3.19	Nathan	Massey	M	5783 Evan Roads Apt. 465	Falmouth	MI	49632	44.2529
...	...	...	...	...	...	...	...	...	...	...	...	...
555714	2020-12-31 23:59:07	30560609640617	health_fitness	43.77	Michael	Olson	M	558 Michael Estates	Luray	MO	63453	40.4931
555715	2020-12-31 23:59:09	3556613125071656	kids_pets	111.84	Jose	Vasquez	M	572 Davis Mountains	Lake Jackson	TX	77566	29.0393
555716	2020-12-31 23:59:15	6011724471098086	kids_pets	86.88	Ann	Lawson	F	144 Evans Islands Apt. 683	Burbank	WA	99323	46.1966
555717	2020-12-31 23:59:24	4079773899158	travel	7.99	Eric	Preston	M	7020 Doyle Stream Apt. 951	Mesa	ID	83643	44.6255
555718	2020-12-31 23:59:34	4170689372027579	entertainment	38.13	Samuel	Frey	M	830 Myers Plaza Apt. 384	Edmond	OK	73034	35.6665

555719 rows × 19 columns

In [18]:
```
import pandas as pd

category = []
for column in new_train_data:
    if new_train_data[column].dtype == 'object':
        category.append(column)
category
```

Out[18]:
```
['trans_date_trans_time',
 'category',
 'first',
 'last',
 'gender',
 'street',
 'city',
 'state',
 'job',
 'dob',
 'trans_num']
```

In [19]:
```
cat_Data=new_train_data[['trans_date_trans_time',
 'category',
 'first',
 'last',
 'gender',
 'street',
 'city',
 'state',
 'job',
 'dob',
 'trans_num']]
```

In [20]:
```
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()

catogary_Data=pd.DataFrame(cat_Data)

final_Categary=catogary_Data.apply(labelencoder.fit_transform)
```

In [21]:
```
final_Categary
```

Out[21]:

	trans_date_trans_time	category	first	last	gender	street	city	state	job	dob	trans_num
0	0	8	162	18	0	568	526	27	370	779	56438
1	1	4	309	157	0	435	612	47	428	607	159395
2	2	0	115	381	1	602	468	13	307	302	818703
3	3	2	163	463	1	930	84	26	328	397	544575
4	4	9	336	149	1	418	216	45	116	734	831111
...	...	...	...	...	...	...	...	...	...	...	...
1296670	1274786	0	121	332	1	154	330	44	215	298	344658
1296671	1274787	1	160	463	1	856	813	20	360	630	199896
1296672	1274788	1	74	67	1	158	346	32	308	412	366013
1296673	1274789	1	179	304	1	433	471	41	485	639	1086299
1296674	1274790	1	160	404	1	127	782	26	467	895	726622

1296675 rows × 11 columns

In [22]:
```
#test data encoding
cat_Data_test=new_test_data[['trans_date_trans_time',
 'category',
 'first',
 'last',
 'gender',
 'street',
 'city',
 'state',
 'job',
 'dob',
 'trans_num']]

from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()

catogary_Data=pd.DataFrame(cat_Data_test)

test_cat_Data=catogary_Data.apply(labelencoder.fit_transform)
test_data_final=test_cat_Data
test_data_final.head(10)
```

Out[22]:

	trans_date_trans_time	category	first	last	gender	street	city	state	job	dob	trans_num
0	0	10	151	115	1	341	157	39	275	376	98699
1	1	10	163	457	0	354	16	43	392	760	108785
2	2	5	24	249	0	865	61	33	259	421	433979
3	3	9	42	457	1	320	764	8	407	718	71993
4	4	13	247	261	1	548	247	21	196	177	190585
5	5	7	85	120	0	727	90	33	361	796	263939
6	6	5	189	409	0	9	117	4	455	130	49712
7	7	10	256	119	0	340	725	40	124	446	303363
8	8	12	86	121	1	400	503	37	13	475	246521
9	9	1	189	313	0	751	624	42	41	189	363381

In [23]:
```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1296675 entries, 0 to 1296674
Data columns (total 22 columns):
 #   Column                 Non-Null Count    Dtype  
---  ------                 --------------    -----  
 0   trans_date_trans_time  1296675 non-null  object 
 1   cc_num                 1296675 non-null  int64  
 2   merchant               1296675 non-null  object 
 3   category               1296675 non-null  object 
 4   amt                    1296675 non-null  float64
 5   first                  1296675 non-null  object 
 6   last                   1296675 non-null  object 
 7   gender                 1296675 non-null  object 
 8   street                 1296675 non-null  object 
 9   city                   1296675 non-null  object 
 10  state                  1296675 non-null  object 
 11  zip                    1296675 non-null  int64  
 12  lat                    1296675 non-null  float64
 13  long                   1296675 non-null  float64
 14  city_pop               1296675 non-null  int64  
 15  job                    1296675 non-null  object 
 16  dob                    1296675 non-null  object 
 17  trans_num              1296675 non-null  object 
 18  unix_time              1296675 non-null  int64  
 19  merch_lat              1296675 non-null  float64
 20  merch_long             1296675 non-null  float64
 21  is_fraud               1296675 non-null  int64  
dtypes: float64(5), int64(5), object(12)
memory usage: 217.6+ MB
```

In [24]:
```
float_data= []
for column in train_data:
    if train_data[column].dtype == 'float64':
        float_data.append(column)
float_data
```

Out[24]:
```
['amt', 'lat', 'long', 'merch_lat', 'merch_long']
```

In [25]:
```
int_data= []
for column in train_data:
    if train_data[column].dtype == 'int64':
        int_data.append(column)
int_data
```

Out[25]:
```
['cc_num', 'zip', 'city_pop', 'unix_time', 'is_fraud']
```

In [26]:
```
Num_Data=pd.DataFrame(train_data[['amt', 'lat', 'long', 'merch_lat', 'merch_long','cc_num', 'zip', 'city_pop',
```

In [27]:
```
Num_Data
```

Out[27]:

	amt	lat	long	merch_lat	merch_long	cc_num	zip	city_pop	unix_time
0	4.97	36.0788	-81.1781	36.011293	-82.048315	2703186189652095	28654	3495	1325376018
1	107.23	48.8878	-118.2105	49.159047	-118.186462	630423337322	99160	149	1325376044
2	220.11	42.1808	-112.2620	43.150704	-112.154481	38859492057661	83252	4154	1325376051
3	45.00	46.2306	-112.1138	47.034331	-112.561071	3534093764340240	59632	1939	1325376076
4	41.96	38.4207	-79.4629	38.674999	-78.632459	375534208663984	24433	99	1325376186
...	...	...	...	...	...	...	...	...	...
1296670	15.56	37.7175	-112.4777	36.841266	-111.690765	30263540414123	84735	258	1371816728
1296671	51.70	39.2667	-77.5101	38.906881	-78.246528	6011149206456997	21790	100	1371816739
1296672	105.93	32.9396	-105.8189	33.619513	-105.130529	3514865930894695	88325	899	1371816752
1296673	74.90	43.3526	-102.5411	42.788940	-103.241160	2720012583106919	57756	1126	1371816816
1296674	4.30	45.8433	-113.8748	46.565983	-114.186110	4292902571056973207	59871	218	1371816817

1296675 rows × 9 columns

In [28]:
```
from sklearn import preprocessing
normalized_data = preprocessing.normalize(Num_Data)
```

In [29]:
```
normalized_data
```

Out[29]:
```
array([[ 1.83857110e-15,  1.33467684e-14, -3.00305248e-14, ...,
         1.06000837e-11,  1.29291871e-12,  4.90301417e-07],
       [ 1.70091678e-10,  7.75474020e-11, -1.87509300e-10, ...,
         1.57290784e-07,  2.36348596e-10,  2.10235414e-03],
       [ 5.66425314e-12,  1.08546967e-12, -2.88892093e-12, ...,
         2.14238518e-09,  1.06897949e-10,  3.41068805e-05],
       ...,
       [ 3.01377071e-14,  9.37150965e-15, -3.01060985e-14, ...,
         2.51289812e-11,  2.55770780e-13,  3.90289923e-07],
       [ 2.75366373e-14,  1.59383821e-14, -3.76987594e-14, ...,
         2.12337253e-11,  4.13968673e-13,  5.04342084e-07],
       [ 1.00165329e-18,  1.06788587e-17, -2.65262950e-17, ...,
         1.39465080e-14,  5.07814926e-17,  3.19554612e-10]])
```

In [30]:
```
final_int=pd.DataFrame(normalized_data,columns=Num_Data.columns)
```

In [31]:
```
final_int
```

Out[31]:

	amt	lat	long	merch_lat	merch_long	cc_num	zip	city_pop	unix_time
0	1.838571e-15	1.334677e-14	-3.003052e-14	1.332180e-14	-3.035245e-14	1.000000	1.060008e-11	1.292919e-12	4.903014e-07
1	1.700917e-10	7.754740e-11	-1.875093e-10	7.797766e-11	-1.874712e-10	0.999998	1.572908e-07	2.363486e-10	2.102354e-03
2	5.664253e-12	1.085470e-12	-2.888921e-12	1.110429e-12	-2.886154e-12	1.000000	2.142385e-09	1.068979e-10	3.410688e-05
3	1.273311e-14	1.308132e-14	-3.172349e-14	1.330874e-14	-3.185005e-14	1.000000	1.687335e-11	5.486555e-13	3.750257e-07
4	1.117342e-13	1.023095e-13	-2.115996e-13	1.029866e-13	-2.093883e-13	1.000000	6.506198e-11	2.636245e-13	3.529309e-06
...	...	...	...	...	...	...	...	...	...
1296670	5.141500e-13	1.246302e-12	-3.716607e-12	1.217348e-12	-3.690605e-12	1.000000	2.799904e-09	8.525110e-12	4.532902e-05
1296671	8.600685e-15	6.532312e-15	-1.289439e-14	6.472453e-15	-1.301690e-14	1.000000	3.624931e-12	1.663575e-14	2.282121e-07
1296672	3.013771e-14	9.371510e-15	-3.010610e-14	9.564949e-15	-2.991025e-14	1.000000	2.512898e-11	2.557708e-13	3.902899e-07
1296673	2.753664e-14	1.593838e-14	-3.769876e-14	1.573116e-14	-3.795613e-14	1.000000	2.123373e-11	4.139687e-13	5.043421e-07
1296674	1.001653e-18	1.067886e-17	-2.652629e-17	1.084720e-17	-2.659881e-17	1.000000	1.394651e-14	5.078149e-17	3.195546e-10

1296675 rows × 9 columns

In [32]:
```
final_Categary
```

Out[32]:

	trans_date_trans_time	category	first	last	gender	street	city	state	job	dob	trans_num
0	0	8	162	18	0	568	526	27	370	779	56438
1	1	4	309	157	0	435	612	47	428	607	159395
2	2	0	115	381	1	602	468	13	307	302	818703
3	3	2	163	463	1	930	84	26	328	397	544575
4	4	9	336	149	1	418	216	45	116	734	831111
...	...	...	...	...	...	...	...	...	...	...	...
1296670	1274786	0	121	332	1	154	330	44	215	298	344658
1296671	1274787	1	160	463	1	856	813	20	360	630	199896
1296672	1274788	1	74	67	1	158	346	32	308	412	366013
1296673	1274789	1	179	304	1	433	471	41	485	639	1086299
1296674	1274790	1	160	404	1	127	782	26	467	895	726622

1296675 rows × 11 columns

In [33]:
```
new_train_data['is_fraud']
```

Out[33]:
```
0          0
1          0
2          0
3          0
4          0
          ..
1296670    0
1296671    0
1296672    0
1296673    0
1296674    0
Name: is_fraud, Length: 1296675, dtype: int64
```

In [34]:
```
final_train_data = pd.concat([final_int, final_Categary, new_train_data['is_fraud']], axis=1)
```

In [35]:
```
final_train_data
```

Out[35]:

	amt	lat	long	merch_lat	merch_long	cc_num	zip	city_pop	unix_time	trans_date_trans_time	...
0	1.838571e-15	1.334677e-14	-3.003052e-14	1.332180e-14	-3.035245e-14	1.000000	1.060008e-11	1.292919e-12	4.903014e-07	0	...
1	1.700917e-10	7.754740e-11	-1.875093e-10	7.797766e-11	-1.874712e-10	0.999998	1.572908e-07	2.363486e-10	2.102354e-03	1	...
2	5.664253e-12	1.085470e-12	-2.888921e-12	1.110429e-12	-2.886154e-12	1.000000	2.142385e-09	1.068979e-10	3.410688e-05	2	...
3	1.273311e-14	1.308132e-14	-3.172349e-14	1.330874e-14	-3.185005e-14	1.000000	1.687335e-11	5.486555e-13	3.750257e-07	3	...
4	1.117342e-13	1.023095e-13	-2.115996e-13	1.029866e-13	-2.093883e-13	1.000000	6.506198e-11	2.636245e-13	3.529309e-06	4	...
...	...	...	...	...	...	...	...	...	...	...	...
1296670	5.141500e-13	1.246302e-12	-3.716607e-12	1.217348e-12	-3.690605e-12	1.000000	2.799904e-09	8.525110e-12	4.532902e-05	1274786	...
1296671	8.600685e-15	6.532312e-15	-1.289439e-14	6.472453e-15	-1.301690e-14	1.000000	3.624931e-12	1.663575e-14	2.282121e-07	1274787	...
1296672	3.013771e-14	9.371510e-15	-3.010610e-14	9.564949e-15	-2.991025e-14	1.000000	2.512898e-11	2.557708e-13	3.902899e-07	1274788	...
1296673	2.753664e-14	1.593838e-14	-3.769876e-14	1.573116e-14	-3.795613e-14	1.000000	2.123373e-11	4.139687e-13	5.043421e-07	1274789	...
1296674	1.001653e-18	1.067886e-17	-2.652629e-17	1.084720e-17	-2.659881e-17	1.000000	1.394651e-14	5.078149e-17	3.195546e-10	1274790	...

1296675 rows × 21 columns

In [36]:
```
x_data=pd.concat([Num_Data,final_Categary,new_train_data['is_fraud']],axis=1)
x_data
```

Out[36]:

	amt	lat	long	merch_lat	merch_long	cc_num	zip	city_pop	unix_time	trans_date_trans_time	...	fi
0	4.97	36.0788	-81.1781	36.011293	-82.048315	2703186189652095	28654	3495	1325376018	0	...	1
1	107.23	48.8878	-118.2105	49.159047	-118.186462	630423337322	99160	149	1325376044	1	...	3
2	220.11	42.1808	-112.2620	43.150704	-112.154481	38859492057661	83252	4154	1325376051	2	...	1
3	45.00	46.2306	-112.1138	47.034331	-112.561071	3534093764340240	59632	1939	1325376076	3	...	1
4	41.96	38.4207	-79.4629	38.674999	-78.632459	375534208663984	24433	99	1325376186	4	...	3
...	...	...	...	...	...	...	...	...	...	...	...	
1296670	15.56	37.7175	-112.4777	36.841266	-111.690765	30263540414123	84735	258	1371816728	1274786	...	1
1296671	51.70	39.2667	-77.5101	38.906881	-78.246528	6011149206456997	21790	100	1371816739	1274787	...	1
1296672	105.93	32.9396	-105.8189	33.619513	-105.130529	3514865930894695	88325	899	1371816752	1274788	...	
1296673	74.90	43.3526	-102.5411	42.788940	-103.241160	2720012583106919	57756	1126	1371816816	1274789	...	1
1296674	4.30	45.8433	-113.8748	46.565983	-114.186110	4292902571056973207	59871	218	1371816817	1274790	...	1

1296675 rows × 21 columns

In [37]:
```
x=final_train_data.drop('is_fraud',axis=1)
y=final_train_data['is_fraud']
```

In [38]:
```
test_x=x_data.drop('is_fraud',axis=1)
test_y=x_data['is_fraud']
```

In [39]:
```
from sklearn.linear_model import LogisticRegression
reg=LogisticRegression()
```

In [40]:
```
x.columns = x.columns.astype(str)
reg.fit(x,y)
```

Out[40]:
```
▾ LogisticRegression
LogisticRegression()
```

In [41]:
```
reg.score(x,y)
```

Out[41]:
```
0.9942113482561166
```

In [42]:
```
reg.score(test_x,test_y)
```

Out[42]:
```
0.9942113482561166
```

In [44]:
```
y_pred=reg.predict(test_x)
```

In [45]:
```
y_pred
```

Out[45]:
```
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

In [46]:
```
df1=pd.DataFrame({'Actual': y,'Predicted': y_pred})
```

In [47]:
```
df1
```

Out[47]:

	Actual	Predicted
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
...	...	...
1296670	0	0
1296671	0	0
1296672	0	0
1296673	0	0
1296674	0	0

1296675 rows × 2 columns

In [48]:
```
from sklearn.metrics import mean_squared_error

# Assuming reg is your regression model
y_pred = reg.predict(test_x)

# Calculate mean squared error
mse = mean_squared_error(test_y, y_pred)

mse
```

Out[48]:
```
0.005788651743883394
```

In [51]:
```
from sklearn.metrics import confusion_matrix
cnf_matrix = confusion_matrix(test_y, y_pred)
```

In [52]:
```
cnf_matrix
```

Out[52]:
```
array([[1289169,       0],
       [   7506,       0]], dtype=int64)
```

In [ ]: