Credit Card Transactions Fraud Detection Dataset

About Data Set:

```
This is a simulated credit card transaction dataset containing legitimate and fraud transactions from the duration 1st Jan 2019 - 31st Dec 2020. I
```

Import Libraries

```
import pandas as pd
import numpy as np

pd.options.display.max_columns = 100
pd.options.display.max_rows = 900
pd.set_option('float_format', '{:f}'.format)

import matplotlib
import matplotlib.pyplot as plt

%matplotlib inline
```

Load Dataset

```
df1_1 = pd.read_csv("fraudTrain.csv")
df1_1 = df1_1.drop(df1_1.columns[0], axis=1)

df1_2 = pd.read_csv("fraudTest.csv")
df1_2 = df1_2.drop(df1_2.columns[0], axis=1)

frames = [df1_1, df1_2]
df1 = pd.concat(frames)

df1.head()
```

	trans_date_trans_time	cc_num	merchant	category	amt
0	2019-01-01 00:00:18	2703186189652095	fraud_Rippin, Kub and Mann	misc_net	4.970000
1	2019-01-01 00:00:44	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.230000
2	2019-01-01 00:00:51	38859492057661	fraud_Lind-Buckridge	entertainment	220.110000
3	2019-01-01 00:01:16	3534093764340240	fraud_Kutch, Hermiston and Farrell	gas_transport	45.000000
4	2019-01-01 00:03:06	375534208663984	fraud_Keeling-Crist	misc_pos	41.960000

```
df1.shape
```

```
(1852394, 22)
```

```
df1.describe()
```

	cc_num	amt	zip	lat	long	city_pop	unix_time
count	1852394.000000	1852394.000000	1852394.000000	1852394.000000	1852394.000000	1852394.000000	1852394.000000
mean	417386038394006464.000000	70.063567	48813.258191	38.539311	-90.227832	88643.674509	1358674218.834364
std	1309115265318020352.000000	159.253975	26881.845966	5.071470	13.747895	301487.618344	18195081.387560
min	60416207185.000000	1.000000	1257.000000	20.027100	-165.672300	23.000000	1325376018.000000

```
df1.dtypes
```

```
trans_date_trans_time     object
cc_num                     int64
merchant                  object
category                  object
amt                      float64
first                     object
last                      object
gender                    object
street                    object
city                      object
state                     object
zip                        int64
lat                      float64
long                     float64
city_pop                   int64
job                       object
dob                       object
trans_num                 object
unix_time                  int64
merch_lat                float64
merch_long               float64
is_fraud                   int64
dtype: object
```

Distribution of Dependent Column (Fraud vs Non Fraud)

```
df_fraud = df1[['is_fraud','trans_num']].groupby(['is_fraud']).count().reset_index()
df_fraud.columns = ['is_fraud','count']

df_fraud['percent'] = (df_fraud['count']/df_fraud['count'].sum())*100

df_fraud
```

	is_fraud	count	percent
0	0	1842743	99.478999
1	1	9651	0.521001

Unique Values

```
df1.nunique()
```

```
trans_date_trans_time    1819551
cc_num                       999
merchant                     693
category                      14
amt                        60616
first                        355
last                         486
gender                         2
street                       999
city                         906
state                         51
zip                          985
lat                          983
long                         983
city_pop                     891
job                          497
dob                          984
trans_num                1852394
unix_time                1819583
merch_lat                1754157
merch_long               1809753
is_fraud                       2
dtype: int64
```

Unsupported Cell Type. Double-Click to inspect/edit the content.

Derive Variables

Convert "trans_date_trans_time" object to DataTime Type

```
df1['trans_date_trans_time'] = pd.to_datetime(df1['trans_date_trans_time'])
```

```
df1.dtypes['trans_date_trans_time']
```

```
dtype('<M8[ns]')
```

Derive 'Transaction Hour' Feature from 'Transaction Time' Feature

```
df1['trans_hour'] = df1['trans_date_trans_time'].dt.hour
df1['trans_hour']
```

```
0          0
1          0
2          0
3          0
4          0
          ..
555714    23
555715    23
555716    23
555717    23
555718    23
Name: trans_hour, Length: 1852394, dtype: int64
```

Derive 'Day of Week' Feature from 'Transaction Time' Feature

```
df1['day_of_week'] = df1['trans_date_trans_time'].dt.day_name()
df1['day_of_week']
```

```
0          Tuesday
1          Tuesday
2          Tuesday
3          Tuesday
4          Tuesday
            ...
555714    Thursday
555715    Thursday
555716    Thursday
555717    Thursday
555718    Thursday
Name: day_of_week, Length: 1852394, dtype: object
```

```
df1['day_of_week'].unique()
```

```
array(['Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday',
       'Monday'], dtype=object)
```

```
#days = {'Monday':0,'Tuesday':1,'Wednesday':2,'Thursday':3,'Friday':4,'Saturday':5,'Sunday':6}

#df1['day_of_week'] = df1['day_name'].apply(lambda x: days[x])
```

Derive 'Year Month' Feature from 'Transaction Time' Feature

```
df1['year_month'] = df1['trans_date_trans_time'].dt.to_period('M')
df1['year_month']
```

```
0         2019-01
1         2019-01
2         2019-01
3         2019-01
4         2019-01
           ...
555714    2020-12
555715    2020-12
555716    2020-12
555717    2020-12
555718    2020-12
Name: year_month, Length: 1852394, dtype: period[M]
```

```
df1.head()
```

	trans_date_trans_time	cc_num	merchant	category	amt	first	last	gender	street	city	state
0	2019-01-01 00:00:18	2703186189652095	fraud_Rippin, Kub and Mann	misc_net	4.970000	Jennifer	Banks	F	561 Perry Cove	Moravian Falls	NC
1	2019-01-01 00:00:44	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.230000	Stephanie	Gill	F	43039 Riley Greens Suite 393	Orient	WA
2	2019-01-01 00:00:51	38859492057661	fraud_Lind-Buckridge	entertainment	220.110000	Edward	Sanchez	M	594 White Dale Suite 530	Malad City	ID
3	2019-01-01 00:01:16	3534093764340240	fraud_Kutch, Hermiston and Farrell	gas_transport	45.000000	Jeremy	White	M	9443 Cynthia Court Apt. 038	Boulder	MT
4	2019-01-01 00:03:06	375534208663984	fraud_Keeling-Crist	misc_pos	41.960000	Tyler	Garcia	M	408 Bradley Rest	Doe Hill	VA

Cost Benefit Analysis

```
df1.describe()
```

	cc_num	amt	zip	lat	long	city_pop	unix_time
count	1852394.000000	1852394.000000	1852394.000000	1852394.000000	1852394.000000	1852394.000000	1852394.000000
mean	417386038394006464.000000	70.063567	48813.258191	38.539311	-90.227832	88643.674509	1358674218.834364
std	1309115265318020352.000000	159.253975	26881.845966	5.071470	13.747895	301487.618344	18195081.387560
min	60416207185.000000	1.000000	1257.000000	20.027100	-165.672300	23.000000	1325376018.000000
25%	180042946491150.000000	9.640000	26237.000000	34.668900	-96.798000	741.000000	1343016823.750000
50%	3521417320836166.000000	47.450000	48174.000000	39.354300	-87.476900	2443.000000	1357089331.000000
75%	4642255475285942.000000	83.100000	72042.000000	41.940400	-80.158000	20328.000000	1374581485.250000
max	4992346398065154048.000000	28948.900000	99921.000000	66.693300	-67.950300	2906700.000000	1388534374.000000

```
df1.pivot_table(index='year_month',values='is_fraud',aggfunc=['count','sum']).describe()
```

	count	sum
	is_fraud	is_fraud
count	24.000000	24.000000
mean	77183.083333	402.125000
std	22822.330801	84.175444
min	47791.000000	258.000000
25%	68588.000000	339.000000
50%	71735.500000	386.000000
75%	86197.000000	457.250000
max	141060.000000	592.000000

```
df1.pivot_table(index='year_month',values='is_fraud',aggfunc=['count','sum'])
```

	count	sum
	is_fraud	is_fraud
year_month		
2019-01	52525	506
2019-02	49866	517
2019-03	70939	494
2019-04	68078	376
2019-05	72532	408
2019-06	86064	354
2019-07	86596	331
2019-08	87359	382
2019-09	70652	418
2019-10	68758	454
2019-11	70421	388
2019-12	141060	592
2020-01	52202	343
2020-02	47791	336
2020-03	72850	444
2020-04	66892	302
2020-05	74343	527
2020-06	87805	467
2020-07	85848	321

```
round(df1.loc[df1.is_fraud==1].pivot_table(index='year_month',values='amt',aggfunc='sum').describe(),2)
```

	amt
count	24.000000
mean	213392.220000
std	47093.970000
min	141138.680000
25%	183611.420000
50%	203326.060000
75%	241604.380000
max	335157.540000

```
df1.loc[df1.is_fraud==1].pivot_table(index='year_month',values='amt',aggfunc='mean').describe()
```

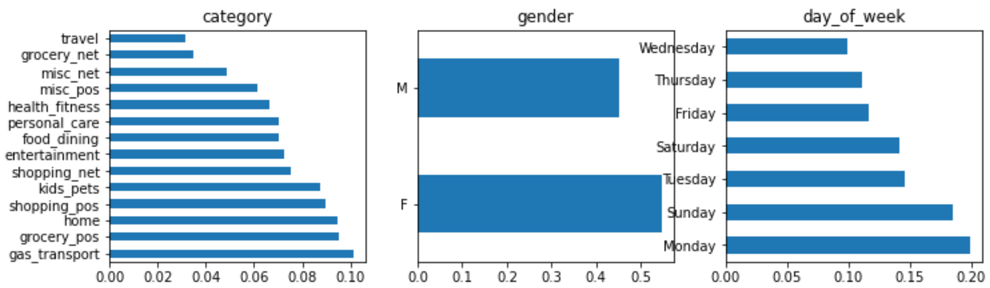
	amt
count	24.000000
mean	530.415010
std	26.948415
min	481.047753
25%	514.364957
50%	528.654707
75%	545.528482
max	596.179382

```
df1.dtypes
```

```
trans_date_trans_time    datetime64[ns]
cc_num                            int64
merchant                         object
category                         object
amt                             float64
first                            object
last                             object
```

```
gender                     object
street                     object
city                       object
state                      object
zip                         int64
lat                       float64
long                      float64
city_pop                    int64
job                        object
dob                        object
trans_num                  object
unix_time                   int64
merch_lat                 float64
merch_long                float64
is_fraud                    int64
trans_hour                  int64
day_of_week                object
year_month              period[M]
dtype: object
```

```
cat_cols=['category','gender','day_of_week']
plt.figure(figsize=[20,7])
for ind, col in enumerate(cat_cols):
    plt.subplot(2,5,ind+1)
    df1[col].value_counts(normalize=True).plot.barh()
    plt.title(col)
plt.show()
```




```
df1.state.value_counts(normalize=True)
```

```
TX    0.073024
NY    0.064467
PA    0.061635
CA    0.043455
OH    0.035968
MI    0.035535
IL    0.033585
FL    0.032809
AL    0.031592
MO    0.029639
MN    0.024527
AR    0.024083
NC    0.023286
VA    0.022542
WI    0.022532
SC    0.022528
KY    0.022123
IN    0.021345
IA    0.020948
OK    0.020541
MD    0.020160
GA    0.020158
WV    0.019720
NJ    0.018965
NE    0.018584
KS    0.017782
MS    0.016207
LA    0.016170
WY    0.014995
WA    0.014597
OR    0.014256
TN    0.013449
ME    0.012650
NM    0.012647
ND    0.011435
CO    0.010671
SD    0.009487
MA    0.009481
VT    0.009076
MT    0.009073
AZ    0.008293
UT    0.008290
NH    0.006331
CT    0.005927
```

```
NV   0.004350
ID   0.004338
DC   0.002769
HI   0.001970
AK   0.001600
RI   0.000402
DE   0.000005
Name: state, dtype: float64
```

```
df1.job.value_counts(normalize=True,ascending=False)
```

```
Film/video editor                                     0.007503
Exhibition designer                                   0.007108
Surveyor, land/geomatics                              0.006713
Naval architect                                       0.006712
Materials engineer                                    0.006322
Designer, ceramics/pottery                            0.006310
Environmental consultant                              0.005924
Financial adviser                                     0.005918
Systems developer                                     0.005918
IT trainer                                            0.005907
Copywriter, advertising                               0.005529
Scientist, audiological                               0.005525
Chartered public finance accountant                   0.005512
Chief Executive Officer                               0.005506
Podiatrist                                            0.005142
Comptroller                                           0.005137
Magazine features editor                              0.005132
Agricultural consultant                               0.005128
Paramedic                                             0.005125
Sub                                                   0.005122
Audiological scientist                                0.004751
Historic buildings inspector/conservation officer     0.004744
Building surveyor                                     0.004743
Librarian, public                                     0.004736
Musician                                              0.004735
Scientist, research (maths)                           0.004733
Barrister                                             0.004733
Clothing/textile technologist                         0.004732
Mining engineer                                       0.004730
Immunologist                                          0.004729
Water engineer                                        0.004718
Quantity surveyor                                     0.004362
Mechanical engineer                                   0.004352
Secondary school teacher                              0.004349
Financial trader                                      0.004348
Prison officer                                        0.004348
Sales professional, IT                                0.004347
Land/geomatics surveyor                               0.004347
Engineer, automotive                                  0.004346
Counsellor                                            0.004344
Petroleum engineer                                    0.004344
Psychologist, forensic                                0.004342
Claims inspector/assessor                             0.004341
Early years teacher                                   0.004341
Geoscientist                                          0.004341
Energy engineer                                       0.004339
Psychotherapist, child                                0.004338
Pensions consultant                                   0.004338
Make                                                  0.004334
Firefighter                                           0.004330
Chemical engineer                                     0.003959
Science writer                                        0.003958
Engineer, biomedical                                  0.003957
Drilling engineer                                     0.003952
Research scientist (physical sciences)                0.003951
Medical sales representative                          0.003946
Librarian, academic                                   0.003945
Scientist, marine                                     0.003944
```

```
#df1.year_month.value_counts()
```

Derive Age of the Customer:

```
Age of Customer = Trasaction Date - DOB
```

```
df1['dob'] = pd.to_datetime(df1['dob'])
```

```
df1['age'] = np.round((df1['trans_date_trans_time'] - df1['dob'])/np.timedelta64(1,'Y'))
df1['age']
```

```
0         31.000000
1         41.000000
2         57.000000
3         52.000000
4         33.000000
             ...
555714    55.000000
555715    21.000000
555716    39.000000
555717    55.000000
555718    28.000000
Name: age, Length: 1852394, dtype: float64
```

```
df1['age'].describe()
```

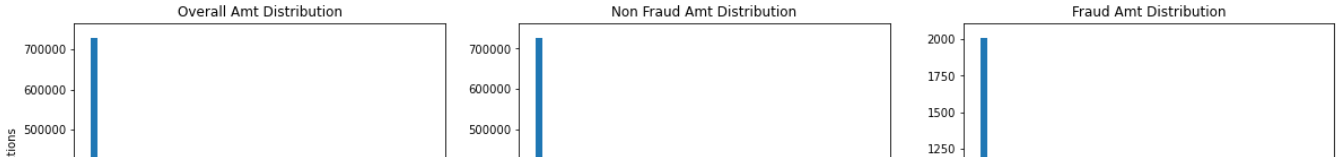
```
count    1852394.000000
mean          46.266173
std           17.412388
min           14.000000
25%           33.000000
50%           44.000000
75%           57.000000
max           96.000000
Name: age, dtype: float64
```

Distribution of the amt

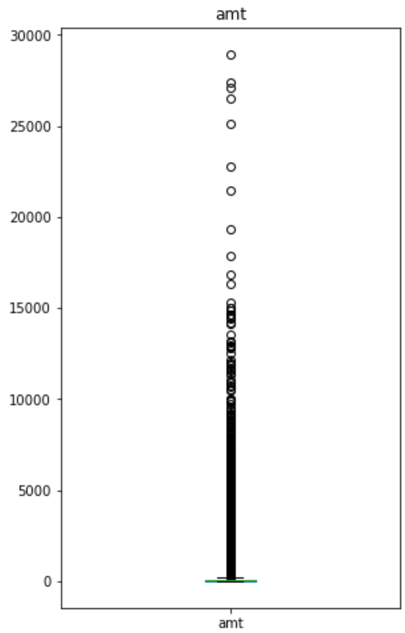
```
pd.concat(
[df1['amt'].describe(percentiles = [0.5,0.95,0.999])\
.reset_index().rename(columns={'index': 'Row Type', 'amt':'Overall Amt Distribution'}),
df1.loc[df1['is_fraud']==0,['amt']].describe(percentiles = [0.5,0.95,0.999])\
.reset_index(drop = 1).rename(columns={'amt':'Non Fraud Amt Distribution'}),
df1.loc[df1['is_fraud']==1,['amt']].describe(percentiles = [0.5,0.95,0.999])\
.reset_index(drop = 1).rename(columns={'amt':'Fraud Amt Distribution'})],
axis=1
)
```

	Row Type	Overall Amt Distribution	Non Fraud Amt Distribution	Fraud Amt Distribution
0	count	1852394.000000	1842743.000000	9651.000000
1	mean	70.063567	67.651278	530.661412
2	std	159.253975	153.548108	391.028873
3	min	1.000000	1.000000	1.060000
4	50%	47.450000	47.240000	390.000000
5	95%	195.340000	189.590000	1084.090000
6	99.9%	1517.241050	1519.622580	1293.127000
7	max	28948.900000	28948.900000	1376.040000

```
fig, ax = plt.subplots(1,3,figsize=(20,5))
ax[0].hist(df1[df1['amt']<=1500]['amt'], bins=50)
ax[1].hist(df1[(df1['is_fraud']==0) & (df1['amt']<=1500)]['amt'], bins=50)
ax[2].hist(df1[(df1['is_fraud']==1) & (df1['amt']<=1500)]['amt'], bins=50)

ax[0].set_title('Overall Amt Distribution')
ax[1].set_title('Non Fraud Amt Distribution')
ax[2].set_title('Fraud Amt Distribution')

ax[0].set_xlabel('Transaction Amount')
ax[0].set_ylabel('#.of Transactions')

ax[1].set_xlabel('Transaction Amount')
ax[2].set_xlabel('Transaction Amount')
plt.show()
```




```
num_cols=['amt']
plt.figure(figsize=[10,8])
for ind, col in enumerate(num_cols):
    plt.subplot(1,2,ind+1)
    df1[col].plot.box()
    plt.title(col)
plt.show()
```




```
Insight 1:

Distribution and Mean of Fraud Transaction's Amount is way different from the Non Fraud Transaction's Amount

Mean of Non Fraud Transactions: 67.6
Mean of Fraud Transactions: 531.3
```

Timeline Plots

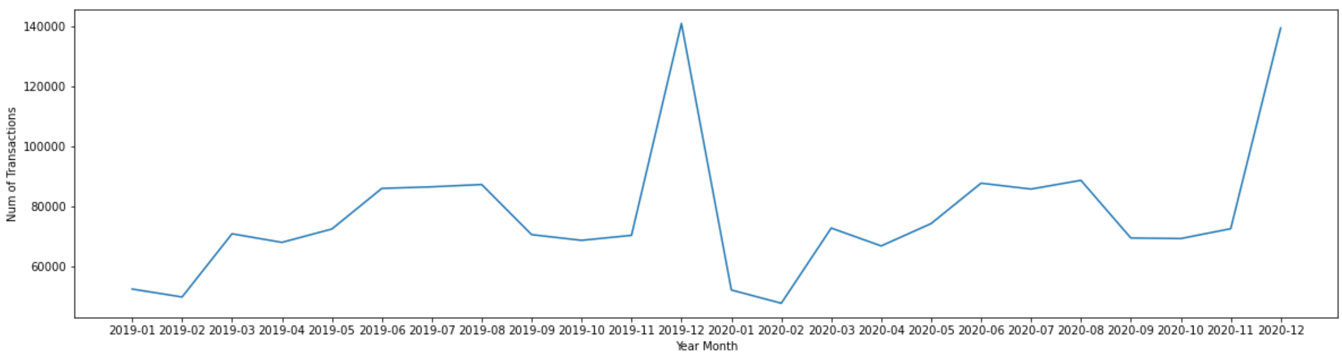
Year Month vs Number of Transactions

```
df1_timeline01 = df1.groupby(df1['year_month'])[['trans_num','cc_num']].nunique().reset_index()
df1_timeline01.columns = ['year_month','num_of_transactions','customers']
df1_timeline01.head()
```

	year_month	num_of_transactions	customers
0	2019-01	52525	913
1	2019-02	49866	918
2	2019-03	70939	916
3	2019-04	68078	913
4	2019-05	72532	910

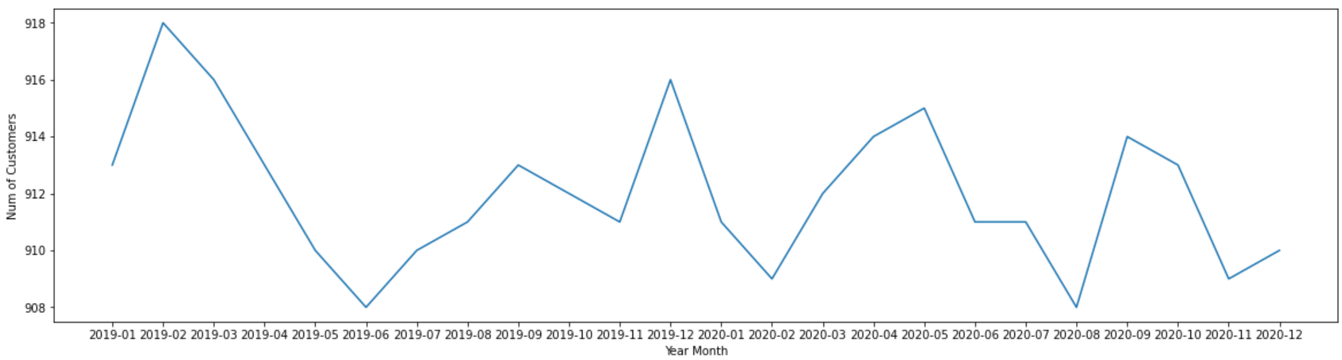
```
x = np.arange(0,len(df1_timeline01),1)

fig, ax = plt.subplots(1,1,figsize=(20,5))
ax.plot(x,df1_timeline01['num_of_transactions'])
ax.set_xticks(x)
ax.set_xticklabels(df1_timeline01['year_month'])
```

```
ax.set_xlabel('Year Month')
ax.set_ylabel('Num of Transactions')
plt.show()
```



Year Month vs Number of Customers Done the Transactions

```
x = np.arange(0,len(df1_timeline01),1)

fig, ax = plt.subplots(1,1,figsize=(20,5))
ax.plot(x,df1_timeline01['customers'])
ax.set_xticks(x)
ax.set_xticklabels(df1_timeline01['year_month'])

ax.set_xlabel('Year Month')
ax.set_ylabel('Num of Customers')
plt.show()
```



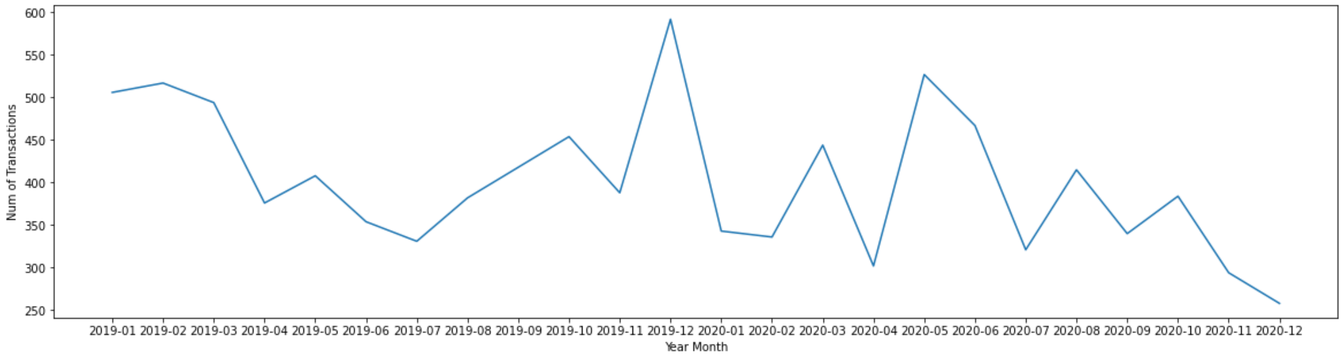
Fraud Transactions

```
df_fraud_transactions = df1[df1['is_fraud']==1]

df1_timeline02 = df_fraud_transactions.groupby(df_fraud_transactions['year_month'])[['trans_num','cc_num']].nunique().reset_index()
df1_timeline02.columns = ['year_month','num_of_fraud_transactions','fraud_customers']
df1_timeline02.head()
```

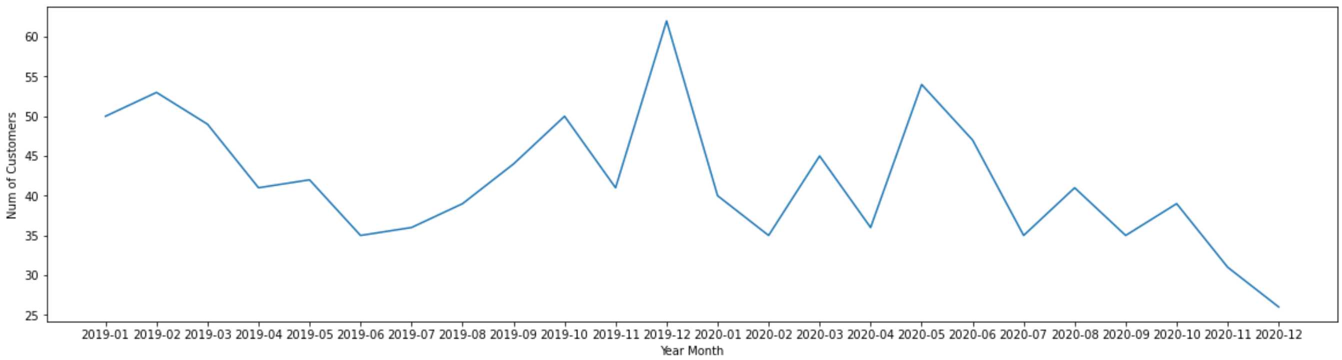
	year_month	num_of_fraud_transactions	fraud_customers
0	2019-01	506	50
1	2019-02	517	53
2	2019-03	494	49
3	2019-04	376	41
4	2019-05	408	42

```
x = np.arange(0,len(df1_timeline02),1)

fig, ax = plt.subplots(1,1,figsize=(20,5))
ax.plot(x,df1_timeline02['num_of_fraud_transactions'])
```

```
ax.set_xticks(x)
ax.set_xticklabels(df1_timeline02['year_month'])

ax.set_xlabel('Year Month')
ax.set_ylabel('Num of Transactions')
plt.show()
```




```
x = np.arange(0,len(df1_timeline02),1)

fig, ax = plt.subplots(1,1,figsize=(20,5))
ax.plot(x,df1_timeline02['fraud_customers'])
ax.set_xticks(x)
ax.set_xticklabels(df1_timeline02['year_month'])

ax.set_xlabel('Year Month')
ax.set_ylabel('Num of Customers')
plt.show()
```

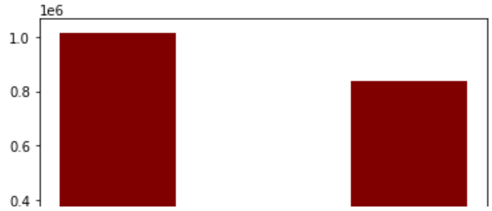


Gender

```
df_gender = df1[['gender','trans_num']].groupby(['gender']).count().reset_index()
df_gender.columns = ['Gender','gender_count']

df_gender['percent'] = (df_gender['gender_count']/df_gender['gender_count'].sum())*100

df_gender
```

	Gender	gender_count	percent
0	F	1014749	54.780408
1	M	837645	45.219592

```
plt.bar(df_gender['Gender'], df_gender['gender_count'], color ='maroon',
        width = 0.4)

plt.show()
```




```
df_fraud_gender = df1[['gender','is_fraud','trans_num']].groupby(['gender','is_fraud']).count().reset_index()
df_fraud_gender.columns = ['Gender','is_fraud','count']

df_fraud_gender = df_fraud_gender.merge(df_gender[['Gender','gender_count']],how='inner',\
                                        left_on='Gender',right_on='Gender')


df_fraud_gender['percent_grp'] = (df_fraud_gender['count']/df_fraud_gender['gender_count'])*100
```

```
df_fraud_gender
```

	Gender	is_fraud	count	gender_count	percent_grp
0	F	0	1009850	1014749	99.517221
1	F	1	4899	1014749	0.482779
2	M	0	832893	837645	99.432695
3	M	1	4752	837645	0.567305

Category

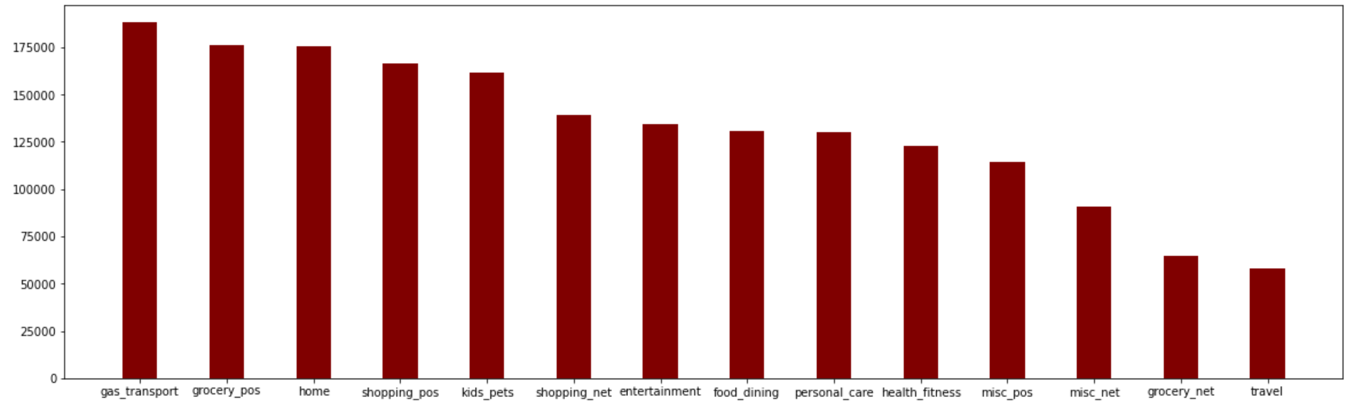
```
df_category = df1[['category','trans_num']].groupby(['category']).count().reset_index()
df_category.columns = ['Category','category_count']

df_category['percent'] = (df_category['category_count']/df_category['category_count'].sum())*100

df_category.sort_values(by = ['percent'], ascending=False)
```

	Category	category_count	percent
2	gas_transport	188029	10.150594
4	grocery_pos	176191	9.511529
6	home	175460	9.472067
12	shopping_pos	166463	8.986371
7	kids_pets	161727	8.730702
11	shopping_net	139322	7.521186
0	entertainment	134118	7.240252
1	food_dining	130729	7.057300
10	personal_care	130085	7.022534
5	health_fitness	122553	6.615925
9	misc_pos	114229	6.166561
8	misc_net	90654	4.893883
3	grocery_net	64878	3.502387
13	travel	57956	3.128708

```
df_category = df_category.sort_values(by = ['percent'], ascending=False).reset_index()
df_category
```

	index	Category	category_count	percent
0	2	gas_transport	188029	10.150594
1	4	grocery_pos	176191	9.511529
2	6	home	175460	9.472067
3	12	shopping_pos	166463	8.986371
4	7	kids_pets	161727	8.730702
5	11	shopping_net	139322	7.521186
6	0	entertainment	134118	7.240252

```
fig = plt.figure(figsize = (20, 6))


plt.bar(df_category['Category'], df_category['category_count'], color ='maroon',
        width = 0.4)

plt.show()
```



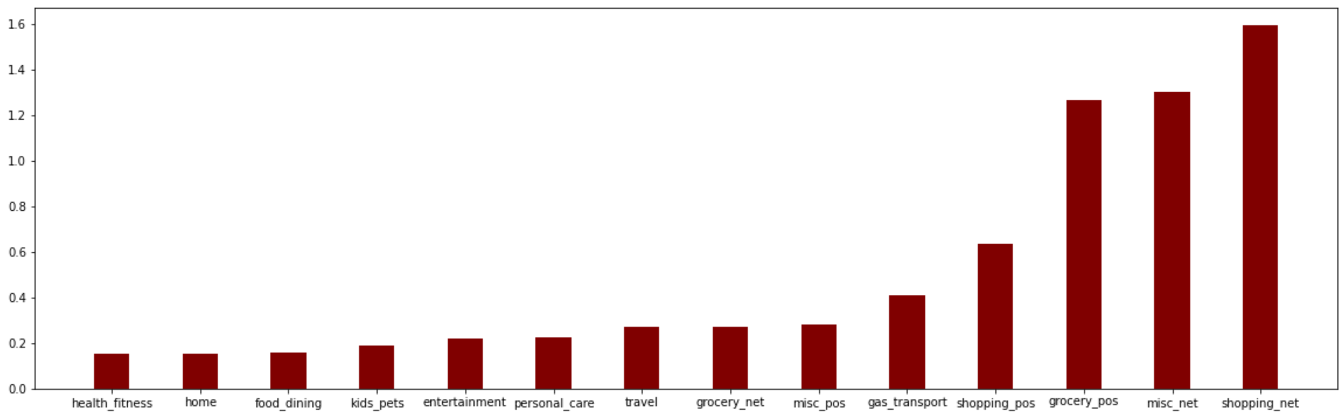

```
df_fraud_category = df1[['category','is_fraud','trans_num']].groupby(['category','is_fraud']).count().reset_index()
df_fraud_category.columns = ['Category','is_fraud','count']

df_fraud_category = df_fraud_category.merge(df_category[['Category','category_count','percent']],how='inner',\
                                  left_on='Category',right_on='Category')


df_fraud_category['percent_grp'] = (df_fraud_category['count']/df_fraud_category['category_count'])*100
#df_fraud_category.head()


df_fraud_category.sort_values(by = ['category_count'], ascending=False)
```

	Category	is_fraud	count	category_count	percent	percent_grp
4	gas_transport	0	187257	188029	10.150594	99.589425
5	gas_transport	1	772	188029	10.150594	0.410575
8	grocery_pos	0	173963	176191	9.511529	98.735463
9	grocery_pos	1	2228	176191	9.511529	1.264537
13	home	1	265	175460	9.472067	0.151032
12	home	0	175195	175460	9.472067	99.848968
25	shopping_pos	1	1056	166463	8.986371	0.634375
24	shopping_pos	0	165407	166463	8.986371	99.365625
14	kids_pets	0	161423	161727	8.730702	99.812029
15	kids_pets	1	304	161727	8.730702	0.187971
23	shopping_net	1	2219	139322	7.521186	1.592713
22	shopping_net	0	137103	139322	7.521186	98.407287

```
df_fraud=df_fraud_category[df_fraud_category['is_fraud'] == 1].sort_values(by = ['percent_grp'])
df_fraud
```

	Category	is_fraud	count	category_count	percent	percent_grp
11	health_fitness	1	185	122553	6.615925	0.150955
13	home	1	265	175460	9.472067	0.151032
3	food_dining	1	205	130729	7.057300	0.156813
15	kids_pets	1	304	161727	8.730702	0.187971
1	entertainment	1	292	134118	7.240252	0.217719
21	personal_care	1	290	130085	7.022534	0.222931
27	travel	1	156	57956	3.128708	0.269170
7	grocery_net	1	175	64878	3.502387	0.269737
19	misc_pos	1	322	114229	6.166561	0.281890
5	gas_transport	1	772	188029	10.150594	0.410575
25	shopping_pos	1	1056	166463	8.986371	0.634375
9	grocery_pos	1	2228	176191	9.511529	1.264537
17	misc_net	1	1182	90654	4.893883	1.303859
23	shopping_net	1	2219	139322	7.521186	1.592713

```
fig = plt.figure(figsize = (20, 6))


plt.bar(df_fraud['Category'] , df_fraud['percent_grp'], color ='maroon',
        width = 0.4)

plt.show()
```



Merchant

```
df1.merchant.value_counts(normalize=True, ascending=False)
```

```
fraud_Kilback LLC                          0.003380
fraud_Cormier LLC                          0.002832
fraud_Schumm PLC                           0.002804
fraud_Kuhn LLC                             0.002716
fraud_Boyer PLC                            0.002699
fraud_Dickinson Ltd                        0.002674
fraud_Emard Inc                            0.002088
fraud_Cummerata-Jones                      0.002084
fraud_Corwin-Collins                       0.002080
fraud_Rodriguez Group                      0.002075
fraud_Kling Inc                            0.002074
fraud_Erdman-Kertzmann                     0.002072
fraud_Parisian and Sons                    0.002072
fraud_Huels-Hahn                           0.002070
fraud_Stroman, Hudson and Erdman           0.002067
fraud_Kutch LLC                            0.002067
fraud_Jenkins, Hauck and Friesen           0.002061
fraud_Prohaska-Murray                      0.002056
fraud_Olson, Becker and Koch               0.002055
fraud_Eichmann, Bogan and Rodriguez        0.002050
fraud_Greenholt, Jacobi and Gleason        0.002048
fraud_Christiansen, Goyette and Schamberger   0.002048
fraud_Bartoletti-Wunsch                    0.002048
fraud_Mraz-Herzog                          0.002045
fraud_Connelly, Reichert and Fritsch       0.002045
fraud_Berge LLC                            0.002044
fraud_Streich, Hansen and Veum             0.002043
fraud_Bins-Rice                            0.002043
fraud_Brekke and Sons                      0.002041
fraud_Friesen-Stamm                        0.002037
fraud_Torp-Labadie                         0.002035
fraud_Ledner-Pfannerstill                  0.002032
fraud_Raynor, Reinger and Hagenes          0.002031
fraud_Koss and Sons                        0.002029
fraud_Schmitt Inc                          0.002023
fraud_Tillman, Dickinson and Labadie       0.002022
fraud_Schaefer, McGlynn and Bosco          0.002020
fraud_Bernhard Inc                         0.002020
fraud_Kutch, Hermiston and Farrell         0.002011
fraud_Conroy-Cruickshank                   0.002009
fraud_Cummings LLC                         0.002009
fraud_Zieme, Bode and Dooley               0.002008
fraud_Luettgen PLC                         0.002008
fraud_Sporer Inc                           0.002008
fraud_Huels-Nolan                          0.002005
fraud_Lind, Huel and McClure               0.002005
fraud_Robel, Cummerata and Prosacco        0.001998
fraud_Harris Inc                           0.001997
fraud_Kuvalis Ltd                          0.001997
fraud_Reilly, Heaney and Cole              0.001996
fraud_Raynor, Feest and Miller             0.001983
fraud_Schaefer, Maggio and Daugherty       0.001982
fraud_Pacocha-O'Reilly                     0.001970
fraud_Heller-Langosh                       0.001969
fraud_Marks Inc                            0.001967
fraud_Friesen-D'Amore                      0.001965
fraud_Harber Inc                           0.001965
fraud_Hackett-Lueilwitz                    0.001957
```

```
df_merchant = df1[['merchant','trans_num']].groupby(['merchant']).count().reset_index()
df_merchant.columns = ['Merchant','merchant_count']

df_merchant['percent'] = (df_merchant['merchant_count']/df_merchant['merchant_count'].sum())*100

df_merchant.sort_values(by = ['percent'], ascending=False)
```

	Merchant	merchant_count	percent
316	fraud_Kilback LLC	6262	0.338049
105	fraud_Cormier LLC	5246	0.283201
571	fraud_Schumm PLC	5195	0.280448
349	fraud_Kuhn LLC	5031	0.271594
70	fraud_Boyer PLC	4999	0.269867
136	fraud_Dickinson Ltd	4953	0.267384
157	fraud_Emard Inc	3867	0.208757
117	fraud_Cummerata-Jones	3860	0.208379
107	fraud_Corwin-Collins	3853	0.208001
522	fraud_Rodriguez Group	3843	0.207461
321	fraud_Kling Inc	3841	0.207353
474	fraud_Parisian and Sons	3839	0.207245
162	fraud_Erdman-Kertzmann	3839	0.207245
272	fraud_Huels-Hahn	3835	0.207029
607	fraud_Stroman, Hudson and Erdman	3829	0.206705
358	fraud_Kutch LLC	3828	0.206652
285	fraud_Jenkins, Hauck and Friesen	3817	0.206058
488	fraud_Prohaska-Murray	3809	0.205626
463	fraud_Olson, Becker and Koch	3806	0.205464
153	fraud_Eichmann, Bogan and Rodriguez	3798	0.205032
208	fraud_Greenholt, Jacobi and Gleason	3794	0.204816
92	fraud_Christiansen, Goyette and Schamberger	3794	0.204816
24	fraud_Bartoletti-Wunsch	3793	0.204762
99	fraud_Connelly, Reichert and Fritsch	3788	0.204492
444	fraud_Mraz-Herzog	3788	0.204492
43	fraud_Berge LLC	3786	0.204384
605	fraud_Streich, Hansen and Veum	3785	0.204330
55	fraud_Bins-Rice	3784	0.204276
77	fraud_Brekke and Sons	3781	0.204114
183	fraud_Friesen-Stamm	3774	0.203736
625	fraud_Torp-Labadie	3769	0.203466
381	fraud_Ledner-Pfannerstill	3764	0.203197
498	fraud_Raynor, Reinger and Hagenes	3763	0.203143
332	fraud_Koss and Sons	3758	0.202873
559	fraud_Schmitt Inc	3747	0.202279
622	fraud_Tillman, Dickinson and Labadie	3746	0.202225
549	fraud_Schaefer, McGlynn and Bosco	3742	0.202009
47	fraud_Bernhard Inc	3741	0.201955
360	fraud_Kutch, Hermiston and Farrell	3725	0.201091
103	fraud_Conroy-Cruickshank	3722	0.200929
119	fraud_Cummings LLC	3721	0.200875
691	fraud_Zieme, Bode and Dooley	3720	0.200821
590	fraud_Sporer Inc	3719	0.200767
399	fraud_Luettgen PLC	3719	0.200767
273	fraud_Huels-Nolan	3714	0.200497
389	fraud_Lind, Huel and McClure	3714	0.200497
518	fraud_Robel, Cummerata and Prosacco	3701	0.199796
231	fraud_Harris Inc	3700	0.199742
365	fraud_Kuvalis Ltd	3700	0.199742

508	fraud_Reilly, Heaney and Cole	3698	0.199634
497	fraud_Raynor, Feest and Miller	3673	0.198284
548	fraud_Schaefer, Maggio and Daugherty	3671	0.198176
468	fraud_Pacocha-O'Reilly	3650	0.197042
243	fraud_Heller-Langosh	3648	0.196934
407	fraud_Marks Inc	3643	0.196664
181	fraud_Friesen-D'Amore	3640	0.196502
229	fraud_Harber Inc	3640	0.196502
217	fraud_Hackett-Lueilwitz	3626	0.195747
155	fraud_Eichmann-Kilback	3616	0.195207
132	fraud_Denesik, Powlowski and Pouros	3611	0.194937
395	fraud_Lockman, West and Runte	3607	0.194721
461	fraud_O'Reilly, Mohr and Purdy	3605	0.194613
447	fraud_Murray-Smitham	3603	0.194505
420	fraud_Medhurst Inc	3600	0.194343
196	fraud_Goodwin-Nitzsche	3598	0.194235
29	fraud_Bauch-Raynor	3597	0.194181
7	fraud_Altenwerth-Kilback	3594	0.194019
552	fraud_Schiller, Blanda and Johnson	3585	0.193533
211	fraud_Gulgowski LLC	3584	0.193479
614	fraud_Terry Ltd	3583	0.193425
563	fraud_Schoen, Kuphal and Nitzsche	3581	0.193317
394	fraud_Lockman Ltd	3580	0.193263
194	fraud_Goldner, Kovacek and Abbott	3580	0.193263
456	fraud_O'Connell, Botsford and Hand	3578	0.193155
69	fraud_Botsford and Sons	3576	0.193047
309	fraud_Kiehn-Emmerich	3574	0.192940
663	fraud_White and Sons	3570	0.192724
512	fraud_Renner Ltd	3570	0.192724
492	fraud_Quitzon-Goyette	3562	0.192292
364	fraud_Kutch-Wilderman	3562	0.192292
94	fraud_Cole PLC	3562	0.192292
466	fraud_Osinski, Ledner and Leuschke	3559	0.192130
572	fraud_Schumm, Bauch and Ondricka	3559	0.192130
130	fraud_Deckow-O'Conner	3558	0.192076
481	fraud_Pollich LLC	3558	0.192076
191	fraud_Gislason Group	3556	0.191968
268	fraud_Hudson-Ratke	3555	0.191914
100	fraud_Connelly-Carter	3555	0.191914
87	fraud_Casper, Hand and Zulauf	3553	0.191806
270	fraud_Huel, Hammes and Witting	3553	0.191806
15	fraud_Bahringer, Bergnaum and Quitzon	3552	0.191752
73	fraud_Bradtke PLC	3551	0.191698
402	fraud_Lynch-Wisozk	3550	0.191644
359	fraud_Kutch and Sons	3547	0.191482
494	fraud_Rau and Sons	3546	0.191428
352	fraud_Kunze Inc	3535	0.190834
550	fraud_Schamberger-O'Keefe	3535	0.190834
187	fraud_Gaylord-Powlowski	3534	0.190780
427	fraud_Miller-Hauck	3533	0.190726

596	fraud_Stark-Koss	3533	0.190726
213	fraud_Gutmann, McLaughlin and Wiza	3531	0.190618
459	fraud_O'Keefe-Hudson	3531	0.190618
336	fraud_Kovacek, Dibbert and Ondricka	3531	0.190618
111	fraud_Crist, Jakubowski and Littel	3529	0.190510
241	fraud_Heller, Gutmann and Zieme	3528	0.190456
413	fraud_McDermott, Osinski and Morar	3527	0.190402
645	fraud_Vandervort-Funk	3519	0.189970
279	fraud_Jakubowski Inc	3517	0.189862
164	fraud_Ernser-Feest	3516	0.189808
602	fraud_Stracke-Lemke	3514	0.189700
415	fraud_McDermott-Weimann	3513	0.189646
543	fraud_Rutherford-Mertz	3508	0.189377
95	fraud_Cole, Hills and Jewess	3508	0.189377
672	fraud_Windler, Goodwin and Kovacek	3507	0.189323
91	fraud_Champlin-Casper	3505	0.189215
146	fraud_Doyle Ltd	3502	0.189053
678	fraud_Wolf Inc	3499	0.188891
147	fraud_DuBuque LLC	3497	0.188783
25	fraud_Barton Inc	3497	0.188783
322	fraud_Kling, Howe and Schneider	3495	0.188675
679	fraud_Wuckert, Wintheiser and Friesen	3494	0.188621
64	fraud_Bogisich Inc	3494	0.188621
41	fraud_Beier and Sons	3492	0.188513
97	fraud_Collier LLC	3489	0.188351
435	fraud_Moore, Williamson and Emmerich	3488	0.188297
20	fraud_Bailey-Morar	3488	0.188297
12	fraud_Auer-Mosciski	3487	0.188243
533	fraud_Rowe, Batz and Goodwin	3483	0.188027
442	fraud_Mosciski, Gislason and Mertz	3482	0.187973
328	fraud_Koepp-Parker	3481	0.187919
345	fraud_Kuhic Inc	3475	0.187595
251	fraud_Hettinger, McCullough and Fay	3471	0.187379
68	fraud_Botsford PLC	3470	0.187325
675	fraud_Witting, Beer and Ernser	3468	0.187217
603	fraud_Streich Ltd	3468	0.187217
308	fraud_Kiehn Inc	3465	0.187055
46	fraud_Berge-Ullrich	3465	0.187055
457	fraud_O'Connell-Ullrich	3460	0.186785
581	fraud_Skiles LLC	3458	0.186677
608	fraud_Strosin-Cruickshank	3457	0.186623
490	fraud_Prosacco, Kreiger and Kovacek	3454	0.186461
86	fraud_Cartwright-Harris	3445	0.185976
472	fraud_Padberg-Welch	3443	0.185868
188	fraud_Gerhold LLC	3441	0.185760
471	fraud_Padberg-Sauer	3432	0.185274
362	fraud_Kutch-Ferry	3427	0.185004
42	fraud_Beier-Hyatt	3426	0.184950
680	fraud_Wuckert-Goldner	3425	0.184896
558	fraud_Schmidt-Larkin	3423	0.184788

601	fraud_Stoltenberg-Beatty	3421	0.184680
506	fraud_Reilly LLC	3420	0.184626
257	fraud_Hilpert-Conroy	3416	0.184410
670	fraud_Willms, Kris and Bergnaum	3408	0.183978
239	fraud_Heidenreich PLC	3408	0.183978
329	fraud_Koepp-Witting	3405	0.183816
575	fraud_Schuppe, Nolan and Hoeger	3401	0.183600
428	fraud_Moen, Reinger and Murphy	3393	0.183168
438	fraud_Morissette PLC	3391	0.183060
570	fraud_Schultz, Simonis and Little	3388	0.182898
610	fraud_Swaniawski, Lowe and Robel	3383	0.182629
333	fraud_Koss, Hansen and Lueilwitz	3383	0.182629
32	fraud_Baumbach, Hodkiewicz and Walsh	3381	0.182521
545	fraud_Sawayn PLC	3377	0.182305
148	fraud_Durgan, Gislason and Spencer	3375	0.182197
483	fraud_Pouros-Conroy	3375	0.182197
310	fraud_Kihn Inc	3373	0.182089
252	fraud_Hickle Group	3366	0.181711
246	fraud_Hermann and Sons	3363	0.181549
409	fraud_Mayert Group	3362	0.181495
287	fraud_Jewess LLC	3360	0.181387
491	fraud_Quitzon, Green and Bashirian	3359	0.181333
484	fraud_Pouros-Haag	3357	0.181225
277	fraud_Jacobi and Sons	3349	0.180793
467	fraud_Pacocha-Bauch	3346	0.180631
89	fraud_Champlin and Sons	3344	0.180523
429	fraud_Mohr Inc	3343	0.180469
659	fraud_Weimann, Kuhic and Beahan	3340	0.180307
127	fraud_Daugherty, Pouros and Beahan	3337	0.180145
656	fraud_Watsica, Haag and Considine	3337	0.180145
531	fraud_Roob, Conn and Tremblay	3335	0.180037
640	fraud_Turner and Sons	3330	0.179767
67	fraud_Botsford Ltd	3329	0.179713
341	fraud_Kris-Padberg	3327	0.179605
425	fraud_Metz-Boehm	3323	0.179389
53	fraud_Bins, Balistreri and Beatty	3321	0.179282
224	fraud_Haley, Jewess and Bechtelar	3321	0.179282
60	fraud_Bode-Rempel	3319	0.179174
45	fraud_Berge-Hills	3317	0.179066
387	fraud_Lesch Ltd	3314	0.178904
17	fraud_Bahringer, Schoen and Corkery	3313	0.178850
473	fraud_Pagac LLC	3313	0.178850
683	fraud_Yost, Schamberger and Windler	3313	0.178850
391	fraud_Little Ltd	3312	0.178796
50	fraud_Bernier and Sons	3303	0.178310
562	fraud_Schoen Ltd	3302	0.178256
529	fraud_Romaguera, Wehner and Tromp	3302	0.178256
101	fraud_Conroy Ltd	3301	0.178202
660	fraud_Weimann-Lockman	3300	0.178148
480	fraud_Pfeffer and Sons	3300	0.178148

202	fraud_Goyette-Gerhold	3300	0.178148
462	fraud_Okuneva, Schneider and Rau	3300	0.178148
454	fraud_Nitzsche, Kessler and Wolff	3299	0.178094
400	fraud_Lynch Ltd	3296	0.177932
131	fraud_Denesik and Sons	3296	0.177932
496	fraud_Rau-Robel	3292	0.177716
546	fraud_Schaefer Ltd	3290	0.177608
264	fraud_Hoppe-Parisian	3288	0.177500
403	fraud_Macejkovic-Lesch	3286	0.177392
48	fraud_Bernhard, Grant and Langworth	3285	0.177338
627	fraud_Torphy-Goyette	3285	0.177338
179	fraud_Friesen Inc	3281	0.177122
33	fraud_Baumbach, Strosin and Nicolas	3281	0.177122
201	fraud_Goyette, Howell and Collier	3280	0.177068
39	fraud_Beer-Jast	3279	0.177014
482	fraud_Pouros, Walker and Spencer	3279	0.177014
267	fraud_Hudson-Grady	3273	0.176690
537	fraud_Ruecker-Mayert	3273	0.176690
22	fraud_Barrows PLC	3271	0.176582
469	fraud_Pacocha-Weissnat	3271	0.176582
141	fraud_Dooley Inc	3268	0.176420
144	fraud_Douglas, Schneider and Turner	3263	0.176150
651	fraud_Waelchi Inc	3262	0.176096
26	fraud_Barton LLC	3261	0.176042
66	fraud_Bogisich-Weimann	3260	0.175988
85	fraud_Cartwright PLC	3258	0.175881
61	fraud_Bode-Schuster	3257	0.175827
495	fraud_Rau-Grant	3249	0.175395
612	fraud_Swift PLC	3246	0.175233
197	fraud_Gottlieb Group	3245	0.175179
234	fraud_Hauck, Dietrich and Funk	3230	0.174369
112	fraud_Crona and Sons	3227	0.174207
369	fraud_Lang, Towne and Schuppe	3226	0.174153
396	fraud_Lowe, Dietrich and Erdman	3222	0.173937
284	fraud_Jast-McDermott	3222	0.173937
192	fraud_Gleason and Sons	3216	0.173613
398	fraud_Lubowitz-Walter	3215	0.173559
606	fraud_Streich, Rolfson and Wilderman	3213	0.173451
643	fraud_Ullrich Ltd	3202	0.172857
126	fraud_Daugherty LLC	3201	0.172803
565	fraud_Schoen-Quigley	3201	0.172803
658	fraud_Weber, Thiel and Hammes	3189	0.172156
78	fraud_Brown Inc	3174	0.171346
169	fraud_Fahey Inc	3171	0.171184
373	fraud_Larkin Ltd	3171	0.171184
671	fraud_Windler LLC	3170	0.171130
122	fraud_Dare, Casper and Bartoletti	3169	0.171076
76	fraud_Breitenberg-Hermiston	3168	0.171022
519	fraud_Roberts, Daniel and Macejkovic	3166	0.170914
424	fraud_Metz, Russel and Metz	3163	0.170752

455	fraud_Nolan-Williamson	3160	0.170590
553	fraud_Schimmel-Olson	3159	0.170536
228	fraud_Hammes-Beatty	3150	0.170050
38	fraud_Bednar PLC	3143	0.169672
652	fraud_Waelchi-Wolf	3117	0.168269
667	fraud_Wilkinson PLC	3110	0.167891
193	fraud_Gleason-Macejkovic	2894	0.156230
687	fraud_Zboncak, Rowe and Murazik	2886	0.155798
256	fraud_Hills-Witting	2866	0.154719
616	fraud_Terry-Huel	2864	0.154611
504	fraud_Reichert, Shanahan and Hayes	2861	0.154449
594	fraud_Stanton, Jakubowski and Baumbach	2859	0.154341
174	fraud_Fisher Inc	2849	0.153801
346	fraud_Kuhic LLC	2842	0.153423
175	fraud_Fisher-Schowalter	2839	0.153261
557	fraud_Schmidt and Sons	2833	0.152937
486	fraud_Predovic Inc	2833	0.152937
487	fraud_Price Inc	2825	0.152505
347	fraud_Kuhic, Bins and Pfeffer	2825	0.152505
443	fraud_Mosciski, Ziemann and Farrell	2821	0.152289
392	fraud_Little, Gutmann and Lynch	2818	0.152127
372	fraud_Langworth, Boehm and Gulgowski	2817	0.152073
430	fraud_Mohr-Bayer	2807	0.151534
450	fraud_Nicolas, Hills and McGlynn	2806	0.151480
476	fraud_Parker, Nolan and Trantow	2795	0.150886
198	fraud_Gottlieb, Considine and Schultz	2794	0.150832
13	fraud_Auer-West	2793	0.150778
386	fraud_Lemke-Gutmann	2790	0.150616
555	fraud_Schmeler, Bashirian and Price	2788	0.150508
238	fraud_Heathcote, Yost and Kertzmann	2786	0.150400
630	fraud_Towne, Greenholt and Koepp	2783	0.150238
8	fraud_Ankunding LLC	2782	0.150184
72	fraud_Boyer-Reichert	2779	0.150022
237	fraud_Heathcote LLC	2778	0.149968
200	fraud_Goyette Inc	2773	0.149698
649	fraud_Volkman-Predovic	2771	0.149590
604	fraud_Streich, Dietrich and Barton	2769	0.149482
367	fraud_Labadie, Treutel and Bode	2767	0.149374
528	fraud_Romaguera, Cruickshank and Greenholt	2767	0.149374
295	fraud_Kassulke PLC	2766	0.149320
31	fraud_Baumbach, Feeney and Morar	2766	0.149320
249	fraud_Hermiston, Russel and Price	2763	0.149158
337	fraud_Kozey-Boehm	2758	0.148888
502	fraud_Reichert, Huels and Hoppe	2758	0.148888
133	fraud_Dibbert and Sons	2758	0.148888
282	fraud_Jast Ltd	2757	0.148834
6	fraud_Altenwerth, Cartwright and Koss	2755	0.148726
421	fraud_Medhurst PLC	2746	0.148241
311	fraud_Kihn, Abernathy and Douglas	2745	0.148187
209	fraud_Greenholt, O'Hara and Balistreri	2743	0.148079

81	fraud_Brown-Greenholt	2742	0.148025
189	fraud_Gerlach Inc	2740	0.147917
35	fraud_Becker, Harris and Harvey	2737	0.147755
3	fraud_Abshire PLC	2733	0.147539
536	fraud_Ruecker, Beer and Collier	2732	0.147485
56	fraud_Bins-Tillman	2729	0.147323
451	fraud_Nienow PLC	2728	0.147269
538	fraud_Runolfsdottir, Mueller and Hand	2727	0.147215
499	fraud_Reichel Inc	2726	0.147161
574	fraud_Schuppe LLC	2722	0.146945
510	fraud_Rempel Inc	2721	0.146891
27	fraud_Bashirian Group	2720	0.146837
178	fraud_Frami Group	2714	0.146513
593	fraud_Stamm-Witting	2714	0.146513
619	fraud_Thiel-Thiel	2712	0.146405
323	fraud_Kling-Ernser	2709	0.146243
210	fraud_Grimes LLC	2707	0.146135
145	fraud_Douglas-White	2703	0.145919
503	fraud_Reichert, Rowe and Mraz	2703	0.145919
595	fraud_Stark-Batz	2702	0.145865
313	fraud_Kihn-Fritsch	2697	0.145595
470	fraud_Padberg-Rogahn	2696	0.145541
167	fraud_Fadel, Mertz and Rippin	2693	0.145379
115	fraud_Cruickshank-Mills	2692	0.145325
637	fraud_Turcotte, McKenzie and Koss	2689	0.145164
666	fraud_Wilkinson Ltd	2688	0.145110
28	fraud_Bauch-Blanda	2686	0.145002
541	fraud_Runte-Mohr	2684	0.144894
453	fraud_Nienow, Barrows and Romaguera	2684	0.144894
648	fraud_Volkman PLC	2684	0.144894
662	fraud_Welch, Rath and Koepp	2684	0.144894
509	fraud_Reinger, Weissnat and Strosin	2683	0.144840
289	fraud_Johns-Hoeger	2681	0.144732
236	fraud_Heaney-Marquardt	2681	0.144732
263	fraud_Hoppe, Harris and Bednar	2681	0.144732
296	fraud_Kautzer and Sons	2680	0.144678
686	fraud_Zboncak Ltd	2679	0.144624
288	fraud_Johns Inc	2676	0.144462
40	fraud_Beier LLC	2676	0.144462
11	fraud_Auer LLC	2674	0.144354
205	fraud_Graham, Hegmann and Hammes	2673	0.144300
586	fraud_Spencer PLC	2672	0.144246
684	fraud_Yost-Rogahn	2668	0.144030
195	fraud_Goldner-Lemke	2665	0.143868
547	fraud_Schaefer, Fay and Hilll	2664	0.143814
653	fraud_Walter, Hettinger and Kessler	2664	0.143814
320	fraud_Klein Group	2664	0.143814
254	fraud_Hills-Boyer	2663	0.143760
376	fraud_Larson-Moen	2662	0.143706
134	fraud_Dibbert-Green	2661	0.143652

437	fraud_Morissette LLC	2660	0.143598
388	fraud_Lesch, D'Amore and Brown	2659	0.143544
116	fraud_Cummerata-Hilpert	2658	0.143490
356	fraud_Kuphal-Toy	2658	0.143490
317	fraud_Kilback and Sons	2657	0.143436
378	fraud_Leannon-Ward	2656	0.143382
10	fraud_Armstrong, Walter and Gottlieb	2649	0.143004
114	fraud_Crooks and Sons	2649	0.143004
44	fraud_Berge, Kautzer and Harris	2648	0.142950
106	fraud_Cormier, Stracke and Thiel	2648	0.142950
0	fraud_Abbott-Rogahn	2647	0.142896
190	fraud_Gibson-Deckow	2647	0.142896
139	fraud_Donnelly LLC	2647	0.142896
49	fraud_Bernhard-Lesch	2645	0.142788
230	fraud_Harris Group	2644	0.142734
644	fraud_Upton PLC	2644	0.142734
390	fraud_Lind-Buckridge	2642	0.142626
125	fraud_Dare-Marvin	2641	0.142572
569	fraud_Schulist Ltd	2640	0.142518
361	fraud_Kutch, Steuber and Gerhold	2639	0.142464
152	fraud_Effertz, Welch and Schowalter	2639	0.142464
668	fraud_Will Ltd	2636	0.142302
304	fraud_Kerluke-Abshire	2635	0.142248
98	fraud_Connelly PLC	2633	0.142140
567	fraud_Schroeder, Hauck and Treutel	2632	0.142086
199	fraud_Gottlieb-Hansen	2631	0.142032
23	fraud_Bartoletti and Sons	2630	0.141978
368	fraud_Lakin, Ferry and Beatty	2630	0.141978
266	fraud_Howe PLC	2629	0.141924
465	fraud_Osinski Inc	2628	0.141870
382	fraud_Leffler-Goldner	2626	0.141762
551	fraud_Schiller Ltd	2625	0.141709
371	fraud_Langworth LLC	2624	0.141655
215	fraud_Haag-Blanda	2624	0.141655
170	fraud_Feil, Hilpert and Koss	2624	0.141655
418	fraud_McKenzie-Huels	2623	0.141601
74	fraud_Bradtke, Torp and Bahringer	2623	0.141601
315	fraud_Kilback Group	2620	0.141439
458	fraud_O'Hara-Wilderman	2618	0.141331
312	fraud_Kihn, Brakus and Goyette	2615	0.141169
440	fraud_Morissette-Schaefer	2614	0.141115
591	fraud_Sporer-Keebler	2610	0.140899
561	fraud_Schneider, Hayes and Nikolaus	2610	0.140899
128	fraud_Daugherty-Thompson	2609	0.140845
657	fraud_Weber and Sons	2608	0.140791
334	fraud_Koss, McLaughlin and Mayer	2608	0.140791
30	fraud_Baumbach Ltd	2607	0.140737
338	fraud_Kozey-Kuhlman	2606	0.140683
163	fraud_Erdman-Schaden	2605	0.140629
692	fraud_Zulauf LLC	2605	0.140629

477	fraud_Parker-Kunde	2603	0.140521
374	fraud_Larkin, Stracke and Greenfelder	2602	0.140467
294	fraud_Kassulke Inc	2601	0.140413
165	fraud_Ernser-Lynch	2600	0.140359
151	fraud_Effertz LLC	2600	0.140359
278	fraud_Jakubowski Group	2599	0.140305
161	fraud_Erdman-Ebert	2599	0.140305
576	fraud_Schuppe-Schuppe	2593	0.139981
599	fraud_Stiedemann Ltd	2592	0.139927
219	fraud_Hagenes, Kohler and Hoppe	2590	0.139819
489	fraud_Prosacco LLC	2589	0.139765
665	fraud_Wilkinson LLC	2589	0.139765
129	fraud_Deckow-Dare	2588	0.139711
366	fraud_Labadie LLC	2586	0.139603
685	fraud_Zboncak LLC	2586	0.139603
445	fraud_Mueller, Gerhold and Mueller	2583	0.139441
343	fraud_Kub PLC	2581	0.139333
485	fraud_Powlowski-Weimann	2581	0.139333
262	fraud_Homenick LLC	2578	0.139171
525	fraud_Rolfson-Kunde	2578	0.139171
408	fraud_Marvin-Lind	2575	0.139009
556	fraud_Schmeler-Howe	2575	0.139009
520	fraud_Roberts, Ryan and Smith	2575	0.139009
314	fraud_Kihn-Schuster	2574	0.138955
283	fraud_Jast and Sons	2574	0.138955
54	fraud_Bins-Howell	2573	0.138901
182	fraud_Friesen-Ortiz	2571	0.138793
564	fraud_Schoen, Nienow and Bauch	2571	0.138793
220	fraud_Hahn, Bahringer and McLaughlin	2570	0.138739
460	fraud_O'Keefe-Wisoky	2570	0.138739
16	fraud_Bahringer, Osinski and Block	2569	0.138685
370	fraud_Langosh, Wintheiser and Hyatt	2569	0.138685
412	fraud_McCullough, Hudson and Schuster	2567	0.138577
573	fraud_Schumm, McLaughlin and Carter	2566	0.138523
34	fraud_Bechtelar-Rippin	2566	0.138523
348	fraud_Kuhn Group	2563	0.138361
577	fraud_Shanahan-Lehner	2563	0.138361
500	fraud_Reichel LLC	2560	0.138200
104	fraud_Conroy-Emard	2559	0.138146
19	fraud_Bahringer-Streich	2558	0.138092
331	fraud_Konopelski, Schneider and Hartmann	2556	0.137984
642	fraud_Turner, Ziemann and Lehner	2554	0.137876
319	fraud_Kirlin and Sons	2552	0.137768
655	fraud_Watsica LLC	2549	0.137606
422	fraud_Medhurst, Cartwright and Ebert	2547	0.137498
433	fraud_Monahan-Morar	2542	0.137228
397	fraud_Lubowitz, Terry and Stracke	2542	0.137228
260	fraud_Hirthe-Beier	2541	0.137174
449	fraud_Nader-Maggio	2540	0.137120
689	fraud_Zemlak, Tillman and Cremin	2538	0.137012

5	fraud_Adams-Barrows	2535	0.136850
223	fraud_Haley, Batz and Auer	2534	0.136796
274	fraud_Hyatt, Russel and Gleichner	2531	0.136634
1	fraud_Abbott-Steuber	2529	0.136526
654	fraud_Waters-Cruickshank	2524	0.136256
166	fraud_Fadel Inc	2523	0.136202
2	fraud_Abernathy and Sons	2513	0.135662
579	fraud_Shields-Wunsch	2512	0.135608
159	fraud_Emmerich-Rau	2510	0.135500
344	fraud_Kub-Heaney	2491	0.134475
180	fraud_Friesen Ltd	2489	0.134367
138	fraud_Dietrich-Fadel	2487	0.134259
80	fraud_Brown, Homenick and Lesch	2483	0.134043
171	fraud_Feil-Morar	2482	0.133989
253	fraud_Hills, Hegmann and Schaefer	2482	0.133989
515	fraud_Rippin-VonRueden	2478	0.133773
406	fraud_Mante, Luettgen and Hackett	2472	0.133449
102	fraud_Conroy, Balistreri and Gorczany	2472	0.133449
611	fraud_Swaniawski, Nitzsche and Welch	2471	0.133395
350	fraud_Kulas Group	2466	0.133125
226	fraud_Hamill-D'Amore	2462	0.132909
624	fraud_Torp, Muller and Borer	2452	0.132369
84	fraud_Carroll PLC	2449	0.132207
113	fraud_Cronin, Kshlerin and Weber	2446	0.132045
532	fraud_Roob-Okuneva	2442	0.131829
507	fraud_Reilly and Sons	2439	0.131667
620	fraud_Thompson-Gleason	2439	0.131667
690	fraud_Ziemann-Waters	2438	0.131613
380	fraud_Ledner, Hartmann and Feest	2436	0.131506
185	fraud_Fritsch and Sons	2436	0.131506
204	fraud_Graham and Sons	2435	0.131452
14	fraud_Bahringer Group	2435	0.131452
493	fraud_Ratke and Sons	2433	0.131344
526	fraud_Romaguera Ltd	2433	0.131344
258	fraud_Hintz, Bauch and Smith	2427	0.131020
123	fraud_Dare, Fritsch and Zboncak	2423	0.130804
566	fraud_Schroeder Group	2420	0.130642
207	fraud_Greenholt Ltd	2419	0.130588
326	fraud_Klocko, Runolfsdottir and Breitenberg	2417	0.130480
540	fraud_Runte, Green and Emard	2410	0.130102
240	fraud_Heller PLC	2408	0.129994
276	fraud_Jacobi Inc	2405	0.129832
168	fraud_Fadel-Hilpert	2402	0.129670
305	fraud_Kertzmann LLC	2395	0.129292
160	fraud_Erdman-Durgan	2391	0.129076
628	fraud_Torphy-Kertzmann	2377	0.128320
124	fraud_Dare-Gibson	2373	0.128104
176	fraud_Flatley Group	2366	0.127727
592	fraud_Stamm-Rodriguez	2364	0.127619
682	fraud_Yost, Block and Koepp	2355	0.127133

248	fraud_Hermiston, Pacocha and Smith	2352	0.126971
414	fraud_McDermott-Rice	2349	0.126809
636	fraud_Turcotte, Batz and Buckridge	2349	0.126809
265	fraud_Howe Ltd	2345	0.126593
222	fraud_Haley Group	2340	0.126323
247	fraud_Hermann-Gaylord	2338	0.126215
93	fraud_Christiansen-Gusikowski	2330	0.125783
411	fraud_McCullough LLC	2328	0.125675
673	fraud_Wintheiser, Dietrich and Schimmel	2319	0.125189
677	fraud_Wiza, Schaden and Stark	2317	0.125081
404	fraud_Maggio-Fahey	2314	0.124919
225	fraud_Halvorson Group	2312	0.124811
384	fraud_Lehner, Reichert and Mills	2311	0.124757
669	fraud_Williamson LLC	2309	0.124650
583	fraud_Smith-Stokes	2306	0.124488
676	fraud_Wiza LLC	2305	0.124434
250	fraud_Herzog Ltd	2305	0.124434
626	fraud_Torp-Lemke	2302	0.124272
554	fraud_Schmeler Inc	2300	0.124164
57	fraud_Block Group	2297	0.124002
615	fraud_Terry, Johns and Bins	2295	0.123894
297	fraud_Keeling-Crist	2290	0.123624
298	fraud_Kemmer-Buckridge	2286	0.123408
587	fraud_Spencer-Runolfsson	2285	0.123354
83	fraud_Buckridge PLC	2278	0.122976
674	fraud_Wisozk and Sons	2275	0.122814
638	fraud_Turcotte-Halvorson	2271	0.122598
21	fraud_Balistreri-Nader	2270	0.122544
452	fraud_Nienow, Ankunding and Collier	2268	0.122436
82	fraud_Bruen-Yost	2266	0.122328
598	fraud_Stiedemann Inc	2265	0.122274
137	fraud_Dickinson-Rempel	2255	0.121734
417	fraud_McGlynn-Jaskolski	2250	0.121464
641	fraud_Turner, Ruecker and Parisian	2250	0.121464
521	fraud_Roberts-Beahan	2249	0.121410
513	fraud_Reynolds-Schinner	2240	0.120925
244	fraud_Herman Inc	2238	0.120817
618	fraud_Thiel PLC	2232	0.120493
431	fraud_Monahan, Bogisich and Ledner	2223	0.120007
580	fraud_Simonis-Prohaska	2220	0.119845
63	fraud_Boehm, Predovic and Reinger	2219	0.119791
214	fraud_Gutmann-Upton	2218	0.119737
629	fraud_Towne LLC	2218	0.119737
62	fraud_Boehm, Block and Jakubowski	2214	0.119521
110	fraud_Cremin, Hamill and Reichel	2203	0.118927
156	fraud_Eichmann-Russel	2196	0.118549
269	fraud_Huel Ltd	1919	0.103596
688	fraud_Zemlak Group	1888	0.101922
271	fraud_Huel-Langworth	1887	0.101868
379	fraud_Lebsack and Sons	1872	0.101058
245	fraud_Herman, Treutel and Dickens	1870	0.100950

245	fraud_Herman, Treutel and Dickens	1870	0.100950
59	fraud_Block-Parisian	1867	0.100788
383	fraud_Lehner, Mosciski and King	1866	0.100735
108	fraud_Corwin-Gorczany	1864	0.100627
36	fraud_Bednar Group	1860	0.100411
623	fraud_Tillman, Fritsch and Schmitt	1852	0.099979
434	fraud_Moore, Dibbert and Koepp	1850	0.099871
325	fraud_Klocko LLC	1848	0.099763
149	fraud_Durgan-Auer	1846	0.099655
661	fraud_Welch Inc	1844	0.099547
354	fraud_Kuphal-Bartoletti	1843	0.099493
140	fraud_Donnelly PLC	1843	0.099493
327	fraud_Koelpin and Sons	1837	0.099169
172	fraud_Ferry, Lynch and Kautzer	1836	0.099115
206	fraud_Greenfelder, Bartoletti and Davis	1831	0.098845
300	fraud_Kerluke Inc	1827	0.098629
65	fraud_Bogisich-Homenick	1826	0.098575
560	fraud_Schmitt Ltd	1822	0.098359
416	fraud_McGlynn-Heathcote	1815	0.097981
475	fraud_Parisian, Schiller and Altenwerth	1815	0.097981
355	fraud_Kuphal-Predovic	1814	0.097927
290	fraud_Johnson, Runolfsdottir and Mayer	1813	0.097873
351	fraud_Kunde-Sanford	1813	0.097873
235	fraud_Hayes, Marquardt and Dibbert	1812	0.097819
426	fraud_Miller-Harris	1811	0.097765
523	fraud_Rodriguez, Yost and Jenkins	1805	0.097441
524	fraud_Rohan, White and Aufderhar	1801	0.097226
293	fraud_Jones, Sawayn and Romaguera	1799	0.097118
597	fraud_Stehr, Jewess and Schimmel	1788	0.096524
535	fraud_Ruecker Group	1784	0.096308
647	fraud_Volkman Ltd	1783	0.096254
227	fraud_Hamill-Daugherty	1783	0.096254
121	fraud_Dach-Nader	1782	0.096200
142	fraud_Dooley-Thompson	1776	0.095876
301	fraud_Kerluke PLC	1774	0.095768
514	fraud_Rippin, Kub and Mann	1768	0.095444
52	fraud_Bernier, Volkman and Hoeger	1763	0.095174
177	fraud_Flatley-Durgan	1763	0.095174
303	fraud_Kerluke, Kertzmann and Wiza	1759	0.094958
405	fraud_Mante Group	1758	0.094904
342	fraud_Kris-Weimann	1756	0.094796
302	fraud_Kerluke, Considine and Macejkovic	1753	0.094634
281	fraud_Jaskolski-Vandervort	1751	0.094526
448	fraud_Nader-Heller	1744	0.094148
79	fraud_Brown PLC	1737	0.093771
173	fraud_Ferry, Reichel and DuBuque	1354	0.073095
4	fraud_Adams, Kovacek and Kuhlman	1354	0.073095
51	fraud_Bernier, Streich and Jewess	1353	0.073041
363	fraud_Kutch-Hegmann	1340	0.072339
324	fraud_Kling-Grant	1336	0.072123
530	fraud_Roob LLC	1336	0.072123

631	fraud_Towne, Walker and Borer	1332	0.071907
600	fraud_Stokes, Christiansen and Sipes	1329	0.071745
588	fraud_Spinka Inc	1328	0.071691
439	fraud_Morissette, Weber and Wiegand	1327	0.071637
299	fraud_Kemmer-Reinger	1327	0.071637
542	fraud_Rutherford, Homenick and Bergstrom	1320	0.071259
635	fraud_Tromp, Kerluke and Glover	1318	0.071151
446	fraud_Murray Ltd	1315	0.070989
539	fraud_Runolfsson and Sons	1315	0.070989
534	fraud_Rowe-Vandervort	1313	0.070881
135	fraud_Dicki Ltd	1313	0.070881
478	fraud_Paucek-Wiza	1307	0.070557
589	fraud_Spinka-Welch	1305	0.070449
609	fraud_Swaniawski, Bahringer and Ledner	1304	0.070395
120	fraud_Dach-Borer	1303	0.070341
585	fraud_Smitham-Schiller	1301	0.070233
212	fraud_Gutmann Ltd	1300	0.070179
291	fraud_Johnston, Nikolaus and Maggio	1299	0.070125
582	fraud_Skiles-Ankunding	1298	0.070071
88	fraud_Cassin-Harvey	1296	0.069964
232	fraud_Harris, Gusikowski and Heaney	1294	0.069856
259	fraud_Hintz-Bruen	1293	0.069802
436	fraud_Morar Inc	1292	0.069748
578	fraud_Shields Inc	1292	0.069748
280	fraud_Jaskolski-Dibbert	1292	0.069748
464	fraud_Ortiz Group	1291	0.069694
96	fraud_Collier Inc	1290	0.069640
357	fraud_Kutch Group	1289	0.069586
664	fraud_Wiegand-Lowe	1289	0.069586
681	fraud_Wuckert-Walter	1286	0.069424
410	fraud_McCullough Group	1285	0.069370
650	fraud_VonRueden Group	1282	0.069208
511	fraud_Rempel PLC	1281	0.069154
632	fraud_Trantow PLC	1281	0.069154
613	fraud_Swift, Bradtke and Marquardt	1278	0.068992
158	fraud_Emmerich-Luettgen	1274	0.068776
186	fraud_Funk Group	1271	0.068614
584	fraud_Smitham-Boehm	1265	0.068290
479	fraud_Pfeffer LLC	1260	0.068020
255	fraud_Hills-Olson	1256	0.067804
242	fraud_Heller-Abshire	1255	0.067750
118	fraud_Cummings Group	1250	0.067480
18	fraud_Bahringer-Larson	1240	0.066940
109	fraud_Corwin-Romaguera	1231	0.066455
58	fraud_Block-Hauck	1229	0.066347
275	fraud_Hyatt-Blick	1219	0.065807
233	fraud_Hartmann, Rowe and Hermann	1215	0.065591
150	fraud_Ebert-Daugherty	1203	0.064943
501	fraud_Reichel, Bradtke and Blanda	1201	0.064835
292	fraud_Johnston-Casper	1201	0.064835

218	fraud_Hagenes, Hermann and Stroman	1199	0.064727
184	fraud_Fritsch LLC	1195	0.064511
37	fraud_Bednar Inc	1190	0.064241
621	fraud_Tillman LLC	1188	0.064133
617	fraud_Thiel Ltd	1186	0.064025
216	fraud_Hackett Group	1185	0.063971
419	fraud_McLaughlin, Armstrong and Koepp	1183	0.063863
154	fraud_Eichmann, Hayes and Treutel	1180	0.063701
401	fraud_Lynch-Mohr	1180	0.063701
517	fraud_Ritchie, Oberbrunner and Cremin	1178	0.063593
527	fraud_Romaguera and Sons	1172	0.063269
335	fraud_Kovacek Ltd	1170	0.063162
330	fraud_Kohler, Lindgren and Koelpin	1169	0.063108
646	fraud_Veum-Koelpin	1169	0.063108
441	fraud_Mosciski Group	1167	0.063000
568	fraud_Schroeder, Wolff and Hermiston	1167	0.063000
432	fraud_Monahan, Hermann and Johns	1162	0.062730
353	fraud_Kunze, Larkin and Mayert	1158	0.062514
71	fraud_Boyer-Haley	1156	0.062406
318	fraud_Kilback, Nitzsche and Leffler	1155	0.062352
375	fraud_Larson, Quitzon and Spencer	1155	0.062352
9	fraud_Ankunding-Carroll	1155	0.062352
639	fraud_Turner LLC	1154	0.062298
203	fraud_Goyette-Herzog	1152	0.062190
339	fraud_Kozey-McDermott	1152	0.062190
385	fraud_Lemke and Sons	1146	0.061866
75	fraud_Breitenberg LLC	1146	0.061866
377	fraud_Leannon-Nikolaus	1146	0.061866
90	fraud_Champlin, Rolfson and Connelly	1143	0.061704
340	fraud_Kris-Kertzmann	1143	0.061704
261	fraud_Hodkiewicz, Prohaska and Paucek	1132	0.061110
393	fraud_Little-Gleichner	1131	0.061056
307	fraud_Kessler Inc	1130	0.061002
634	fraud_Tromp Group	1128	0.060894

```
df_fraud_merchant = df1[['merchant','is_fraud','trans_num']].groupby(['merchant','is_fraud']).count().reset_index()
df_fraud_merchant.columns = ['Merchant','is_fraud','count']

df_fraud_merchant = df_fraud_merchant.merge(df_merchant[['Merchant','merchant_count','percent']],how='inner',\
                                left_on='Merchant',right_on='Merchant')


df_fraud_merchant['percent_grp'] = (df_fraud_merchant['count']/df_fraud_merchant['merchant_count'])*100
```

633	fraud_Treutel-King	1098	0.059275

```
df_fraud_merchant[df_fraud_merchant['is_fraud'] == 1].sort_values(by = ['percent_grp'],ascending=False)
```

	Merchant	is_fraud	count	merchant_count	percent	percent_grp
673	fraud_Kozey-Boehm	1	60	2758	0.148888	2.175489
490	fraud_Herman, Treutel and Dickens	1	38	1870	0.100950	2.032086
1226	fraud_Terry-Huel	1	56	2864	0.154611	1.955307
607	fraud_Kerluke-Abshire	1	50	2635	0.142248	1.897533
884	fraud_Mosciski, Ziemann and Farrell	1	53	2821	0.152289	1.878766
1105	fraud_Schmeler, Bashirian and Price	1	52	2788	0.150508	1.865136
691	fraud_Kuhic LLC	1	53	2842	0.153423	1.864884
563	fraud_Jast Ltd	1	51	2757	0.148834	1.849837
743	fraud_Langworth, Boehm and Gulgowski	1	52	2817	0.152073	1.845935
1051	fraud_Romaguera, Cruickshank and Greenholt	1	51	2767	0.149374	1.843151
145	fraud_Boyer-Reichert	1	51	2779	0.150022	1.835193
476	fraud_Heathcote, Yost and Kertzmann	1	51	2786	0.150400	1.830581
401	fraud_Goyette Inc	1	50	2773	0.149698	1.803101
771	fraud_Lemke-Gutmann	1	50	2790	0.150616	1.792115
840	fraud_Medhurst PLC	1	48	2746	0.148241	1.747997
985	fraud_Rau and Sons	1	60	3546	0.191428	1.692047
474	fraud_Heathcote LLC	1	47	2778	0.149968	1.691865
299	fraud_Durgan-Auer	1	31	1846	0.099655	1.679307
419	fraud_Greenholt, O'Hara and Balistreri	1	46	2743	0.148079	1.676996
866	fraud_Moore, Dibbert and Koepp	1	31	1850	0.099871	1.675676
73	fraud_Bednar Group	1	31	1860	0.100411	1.666667
693	fraud_Kuhic, Bins and Pfeffer	1	47	2825	0.152505	1.663717
971	fraud_Price Inc	1	47	2825	0.152505	1.663717
387	fraud_Gleason-Macejkovic	1	48	2894	0.156230	1.658604
605	fraud_Kerluke, Kertzmann and Wiza	1	29	1759	0.094958	1.648664
1182	fraud_Stanton, Jakubowski and Baumbach	1	47	2859	0.154341	1.643931
293	fraud_Doyle Ltd	1	57	3502	0.189053	1.627641
159	fraud_Brown PLC	1	28	1737	0.093771	1.611975
850	fraud_Miller-Harris	1	29	1811	0.097765	1.601325
683	fraud_Kris-Weimann	1	28	1756	0.094796	1.594533
733	fraud_Labadie, Treutel and Bode	1	44	2767	0.149374	1.590170
1109	fraud_Schmidt and Sons	1	45	2833	0.152937	1.588422
1180	fraud_Stamm-Witting	1	43	2714	0.146513	1.584377
55	fraud_Bashirian Group	1	43	2720	0.146837	1.580882
1016	fraud_Rempel Inc	1	43	2721	0.146891	1.580301
397	fraud_Gottlieb, Considine and Schultz	1	44	2794	0.150832	1.574803
854	fraud_Moen, Reinger and Murphy	1	53	3393	0.183168	1.562039
351	fraud_Fisher-Schowalter	1	44	2839	0.153261	1.549841
1254	fraud_Towne, Greenholt and Koepp	1	43	2783	0.150238	1.545095
831	fraud_McGlynn-Heathcote	1	28	1815	0.097981	1.542700
809	fraud_Mante Group	1	27	1758	0.094904	1.535836
1134	fraud_Schultz, Simonis and Little	1	52	3388	0.182898	1.534829
63	fraud_Baumbach, Feeney and Morar	1	42	2766	0.149320	1.518438
1202	fraud_Streich, Dietrich and Barton	1	42	2769	0.149482	1.516793
1288	fraud_Volkman Ltd	1	27	1783	0.096254	1.514302
1240	fraud_Tillman, Fritsch and Schmitt	1	28	1852	0.099979	1.511879
942	fraud_Padberg-Welch	1	52	3443	0.185868	1.510311
349	fraud_Fisher Inc	1	43	2849	0.153801	1.509302
621	fraud_Kihn, Abernathy and Douglas	1	41	2745	0.148187	1.493625

894	fraud_Nader-Heller	1	26	1744	0.094148	1.490826
948	fraud_Parisian, Schiller and Altenwerth	1	27	1815	0.097981	1.487603
617	fraud_Kiehn-Emmerich	1	53	3574	0.192940	1.482932
131	fraud_Bogisich-Homenick	1	27	1826	0.098575	1.478642
653	fraud_Koelpin and Sons	1	27	1837	0.099169	1.469788
27	fraud_Auer-West	1	41	2793	0.150778	1.467956
707	fraud_Kuphal-Bartoletti	1	27	1843	0.099493	1.465003
13	fraud_Altenwerth, Cartwright and Koss	1	40	2755	0.148726	1.451906
829	fraud_McDermott-Weimann	1	51	3513	0.189646	1.451751
1284	fraud_Vandervort-Funk	1	51	3519	0.189970	1.449275
589	fraud_Kassulke PLC	1	40	2766	0.149320	1.446132
1043	fraud_Rohan, White and Aufderhar	1	26	1801	0.097226	1.443642
1041	fraud_Rodriguez, Yost and Jenkins	1	26	1805	0.097441	1.440443
657	fraud_Koepp-Witting	1	49	3405	0.183816	1.439060
561	fraud_Jaskolski-Vandervort	1	25	1751	0.094526	1.427756
858	fraud_Mohr-Bayer	1	40	2807	0.151534	1.425009
892	fraud_Murray-Smitham	1	51	3603	0.194505	1.415487
1001	fraud_Reichert, Huels and Hoppe	1	39	2758	0.148888	1.414068
281	fraud_Donnelly PLC	1	26	1843	0.099493	1.410743
1003	fraud_Reichert, Rowe and Mraz	1	38	2703	0.145919	1.405845
1198	fraud_Stracke-Lemke	1	49	3514	0.189700	1.394422
379	fraud_Gerlach Inc	1	38	2740	0.147917	1.386861
615	fraud_Kiehn Inc	1	48	3465	0.187055	1.385281
603	fraud_Kerluke, Considine and Macejkovic	1	24	1753	0.094634	1.369082
355	fraud_Flatley-Durgan	1	24	1763	0.095174	1.361316
1210	fraud_Strosin-Cruickshank	1	47	3457	0.186623	1.359560
703	fraud_Kunze Inc	1	48	3535	0.190834	1.357850
1024	fraud_Rippin, Kub and Mann	1	24	1768	0.095444	1.357466
1316	fraud_Welch Inc	1	25	1844	0.099547	1.355748
535	fraud_Hudson-Ratke	1	48	3555	0.191914	1.350211
41	fraud_Bailey-Morar	1	47	3488	0.188297	1.347477
789	fraud_Lockman, West and Runte	1	48	3607	0.194721	1.330746
852	fraud_Miller-Hauck	1	47	3533	0.190726	1.330314
539	fraud_Huel, Hammes and Witting	1	47	3553	0.191806	1.322826
1115	fraud_Schmitt Ltd	1	24	1822	0.098359	1.317234
1367	fraud_Zboncak, Rowe and Murazik	1	38	2886	0.155798	1.316701
599	fraud_Kerluke Inc	1	24	1827	0.098629	1.313629
783	fraud_Little, Gutmann and Lynch	1	37	2818	0.152127	1.312988
787	fraud_Lockman Ltd	1	47	3580	0.193263	1.312849
930	fraud_Osinski, Ledner and Leuschke	1	46	3559	0.192130	1.292498
512	fraud_Hills-Witting	1	37	2866	0.154719	1.290998
1349	fraud_Wolf Inc	1	45	3499	0.188891	1.286082
916	fraud_O'Keefe-Hudson	1	45	3531	0.190618	1.274427
1369	fraud_Zemlak Group	1	24	1888	0.101922	1.271186
1138	fraud_Schumm, Bauch and Ondricka	1	45	3559	0.192130	1.264400
882	fraud_Mosciski, Gislason and Mertz	1	44	3482	0.187973	1.263642
1292	fraud_Volkman-Predovic	1	35	2771	0.149590	1.263082
17	fraud_Ankunding LLC	1	35	2782	0.150184	1.258088
105	fraud_Bernier, Volkman and Hoeger	1	22	1763	0.095174	1.247873
649	fraud_Klocko LLC	1	23	1848	0.099763	1.244589

147	fraud_Bradtke PLC	1	44	3551	0.191698	1.239088
655	fraud_Koepp-Parker	1	43	3481	0.187919	1.235277
189	fraud_Cole PLC	1	44	3562	0.192292	1.235261
389	fraud_Goldner, Kovacek and Abbott	1	44	3580	0.193263	1.229050
757	fraud_Lebsack and Sons	1	23	1872	0.101058	1.228632
1005	fraud_Reichert, Shanahan and Hayes	1	35	2861	0.154449	1.223348
934	fraud_Pacocha-O'Reilly	1	44	3650	0.197042	1.205479
25	fraud_Auer-Mosciski	1	42	3487	0.188243	1.204474
119	fraud_Block-Parisian	1	22	1867	0.100788	1.178361
393	fraud_Goodwin-Nitzsche	1	42	3598	0.194235	1.167315
470	fraud_Hayes, Marquardt and Dibbert	1	21	1812	0.097819	1.158940
51	fraud_Barton Inc	1	40	3497	0.188783	1.143838
295	fraud_DuBuque LLC	1	40	3497	0.188783	1.143838
345	fraud_Ferry, Lynch and Kautzer	1	21	1836	0.099115	1.143791
59	fraud_Bauch-Raynor	1	41	3597	0.194181	1.139839
173	fraud_Cartwright-Harris	1	39	3445	0.185976	1.132075
969	fraud_Predovic Inc	1	32	2833	0.152937	1.129545
261	fraud_Deckow-O'Conner	1	40	3558	0.192076	1.124227
689	fraud_Kuhic Inc	1	39	3475	0.187595	1.122302
1065	fraud_Ruecker Group	1	20	1784	0.096308	1.121076
1121	fraud_Schoen, Kuphal and Nitzsche	1	40	3581	0.193317	1.117006
541	fraud_Huel-Langworth	1	21	1887	0.101868	1.112878
1095	fraud_Schamberger-O'Keefe	1	39	3535	0.190834	1.103253
701	fraud_Kunde-Sanford	1	20	1813	0.097873	1.103144
709	fraud_Kuphal-Predovic	1	20	1814	0.097927	1.102536
141	fraud_Boyer PLC	1	55	4999	0.269867	1.100220
697	fraud_Kuhn LLC	1	55	5031	0.271594	1.093222
217	fraud_Corwin-Gorczany	1	20	1864	0.100627	1.072961
175	fraud_Casper, Hand and Zulauf	1	38	3553	0.191806	1.069519
1188	fraud_Stehr, Jewess and Schimmel	1	19	1788	0.096524	1.062640
478	fraud_Heidenreich PLC	1	36	3408	0.183978	1.056338
434	fraud_Hackett-Lueilwitz	1	38	3626	0.195747	1.047987
1061	fraud_Rowe, Batz and Goodwin	1	36	3483	0.188027	1.033592
1081	fraud_Rutherford-Mertz	1	36	3508	0.189377	1.026226
669	fraud_Kovacek Ltd	1	12	1170	0.063162	1.025641
765	fraud_Lehner, Mosciski and King	1	19	1866	0.100735	1.018221
717	fraud_Kutch and Sons	1	36	3547	0.191482	1.014942
601	fraud_Kerluke PLC	1	18	1774	0.095768	1.014656
211	fraud_Cormier LLC	1	53	5246	0.283201	1.010294
129	fraud_Bogisich Inc	1	35	3494	0.188621	1.001717
482	fraud_Heller, Gutmann and Zieme	1	35	3528	0.190456	0.992063
631	fraud_Kilback LLC	1	62	6262	0.338049	0.990099
1067	fraud_Ruecker, Beer and Collier	1	27	2732	0.147485	0.988287
671	fraud_Kovacek, Dibbert and Ondricka	1	34	3531	0.190618	0.962900
285	fraud_Dooley-Thompson	1	17	1776	0.095876	0.957207
537	fraud_Huel Ltd	1	18	1919	0.103596	0.937989
1196	fraud_Stoltenberg-Beatty	1	32	3421	0.184680	0.935399
979	fraud_Quitzon, Green and Bashirian	1	31	3359	0.181333	0.922894
85	fraud_Beier-Hyatt	1	31	3426	0.184950	0.904845
579	fraud_Johnson, Runolfsdottir and Mayer	1	16	1813	0.097873	0.882515

1085	fraud_Sawayn PLC	1	27	3377	0.182305	0.799526
805	fraud_Macejkovic-Lesch	1	26	3286	0.177392	0.791236
1136	fraud_Schumm PLC	1	41	5195	0.280448	0.789220
454	fraud_Hamill-Daugherty	1	14	1783	0.096254	0.785193
1220	fraud_Swift, Bradtke and Marquardt	1	10	1278	0.068992	0.782473
585	fraud_Jones, Sawayn and Romaguera	1	14	1799	0.097118	0.778210
65	fraud_Baumbach, Hodkiewicz and Walsh	1	26	3381	0.182521	0.769003
283	fraud_Dooley Inc	1	25	3268	0.176420	0.764994
45	fraud_Barrows PLC	1	25	3271	0.176582	0.764292
203	fraud_Conroy Ltd	1	25	3301	0.178202	0.757346
1274	fraud_Turner and Sons	1	25	3330	0.179767	0.750751
243	fraud_Dach-Nader	1	13	1782	0.096200	0.729517
906	fraud_Nitzsche, Kessler and Wolff	1	24	3299	0.178094	0.727493
965	fraud_Pouros-Haag	1	24	3357	0.181225	0.714924
413	fraud_Greenfelder, Bartoletti and Davis	1	13	1831	0.098845	0.709995
289	fraud_Douglas, Schneider and Turner	1	23	3263	0.176150	0.704873
448	fraud_Haley, Jewess and Bechtelar	1	23	3321	0.179282	0.692562
1306	fraud_Watsica, Haag and Considine	1	23	3337	0.180145	0.689242
932	fraud_Pacocha-Bauch	1	23	3346	0.180631	0.687388
553	fraud_Jacobi and Sons	1	23	3349	0.180793	0.686772
817	fraud_Mayert Group	1	23	3362	0.181495	0.684117
492	fraud_Hermann and Sons	1	23	3363	0.181549	0.683913
946	fraud_Parisian and Sons	1	26	3839	0.207245	0.677260
403	fraud_Goyette, Howell and Collier	1	22	3280	0.177068	0.670732
1248	fraud_Torphy-Goyette	1	22	3285	0.177338	0.669711
263	fraud_Denesik and Sons	1	22	3296	0.177932	0.667476
799	fraud_Lynch Ltd	1	22	3296	0.177932	0.667476
957	fraud_Pfeffer and Sons	1	22	3300	0.178148	0.666667
35	fraud_Bahringer, Schoen and Corkery	1	22	3313	0.178850	0.664051
135	fraud_Botsford Ltd	1	22	3329	0.179713	0.660859
856	fraud_Mohr Inc	1	22	3343	0.180469	0.658092
973	fraud_Prohaska-Murray	1	25	3809	0.205626	0.656340
573	fraud_Jewess LLC	1	22	3360	0.181387	0.654762
1144	fraud_Schuppe, Nolan and Hoeger	1	22	3401	0.183600	0.646869
1262	fraud_Tromp Group	1	7	1128	0.060894	0.620567
207	fraud_Conroy-Cruickshank	1	23	3722	0.200929	0.617947
1214	fraud_Swaniawski, Lowe and Robel	1	20	3383	0.182629	0.591191
663	fraud_Koss and Sons	1	22	3758	0.202873	0.585418
944	fraud_Pagac LLC	1	19	3313	0.178850	0.573498
1312	fraud_Weimann, Kuhic and Beahan	1	19	3340	0.180307	0.568862
504	fraud_Hickle Group	1	19	3366	0.181711	0.564468
963	fraud_Pouros-Conroy	1	19	3375	0.182197	0.562963
67	fraud_Baumbach, Strosin and Nicolas	1	18	3281	0.177122	0.548613
773	fraud_Lesch Ltd	1	18	3314	0.178904	0.543150
681	fraud_Kris-Padberg	1	18	3327	0.179605	0.541028
1113	fraud_Schmitt Inc	1	20	3747	0.202279	0.533760
619	fraud_Kihn Inc	1	18	3373	0.182089	0.533650
567	fraud_Jast-McDermott	1	17	3222	0.173937	0.527623
922	fraud_Okuneva, Schneider and Rau	1	17	3300	0.178148	0.515152
417	fraud_Greenholt, Jacobi and Gleason	1	19	3794	0.204816	0.500791

307	fraud_Eichmann, Bogan and Rodriguez	1	19	3798	0.205032	0.500263
569	fraud_Jenkins, Hauck and Friesen	1	19	3817	0.206058	0.497773
486	fraud_Heller-Langosh	1	18	3648	0.196934	0.493421
235	fraud_Cummerata-Jones	1	19	3860	0.208379	0.492228
1012	fraud_Reilly, Heaney and Cole	1	18	3698	0.199634	0.486750
729	fraud_Kuvalis Ltd	1	18	3700	0.199742	0.486486
95	fraud_Bernhard Inc	1	18	3741	0.201955	0.481155
1057	fraud_Roob, Conn and Tremblay	1	16	3335	0.180037	0.479760
111	fraud_Bins-Rice	1	18	3784	0.204276	0.475687
715	fraud_Kutch LLC	1	18	3828	0.206652	0.470219
1333	fraud_Willms, Kris and Bergnaum	1	16	3408	0.183978	0.469484
1294	fraud_VonRueden Group	1	6	1282	0.069208	0.468019
581	fraud_Johnston, Nikolaus and Maggio	1	6	1299	0.070125	0.461894
1172	fraud_Spinka-Welch	1	6	1305	0.070449	0.459770
442	fraud_Hahn, Douglas and Schowalter	1	5	1091	0.058897	0.458295
1238	fraud_Tillman, Dickinson and Labadie	1	17	3746	0.202225	0.453817
127	fraud_Boehm, Predovic and Reinger	1	10	2219	0.119791	0.450653
255	fraud_Daugherty, Pouros and Beahan	1	15	3337	0.180145	0.449506
1055	fraud_Roob LLC	1	6	1336	0.072123	0.449102
185	fraud_Christiansen, Goyette and Schamberger	1	17	3794	0.204816	0.448076
725	fraud_Kutch-Hegmann	1	6	1340	0.072339	0.447761
103	fraud_Bernier, Streich and Jewess	1	6	1353	0.073041	0.443459
347	fraud_Ferry, Reichel and DuBuque	1	6	1354	0.073095	0.443131
43	fraud_Balistreri-Nader	1	10	2270	0.122544	0.440529
458	fraud_Harber Inc	1	16	3640	0.196502	0.439560
1168	fraud_Spencer-Runolfsson	1	10	2285	0.123354	0.437637
181	fraud_Champlin, Rolfson and Connelly	1	5	1143	0.061704	0.437445
769	fraud_Lemke and Sons	1	5	1146	0.061866	0.436300
1091	fraud_Schaefer, Maggio and Daugherty	1	16	3671	0.198176	0.435849
677	fraud_Kozey-McDermott	1	5	1152	0.062190	0.434028
19	fraud_Ankunding-Carroll	1	5	1155	0.062352	0.432900
1031	fraud_Robel, Cummerata and Prosacco	1	16	3701	0.199796	0.432316
797	fraud_Luettgen PLC	1	16	3719	0.200767	0.430223
880	fraud_Mosciski Group	1	5	1167	0.063000	0.428449
444	fraud_Haley Group	1	10	2340	0.126323	0.427350
359	fraud_Friesen Inc	1	14	3281	0.177122	0.426699
97	fraud_Bernhard, Grant and Langworth	1	14	3285	0.177338	0.426180
1029	fraud_Ritchie, Oberbrunner and Cremin	1	5	1178	0.063593	0.424448
367	fraud_Friesen-Stamm	1	16	3774	0.203736	0.423953
309	fraud_Eichmann, Hayes and Treutel	1	5	1180	0.063701	0.423729
49	fraud_Bartoletti-Wunsch	1	16	3793	0.204762	0.421830
325	fraud_Erdman-Kertzmann	1	16	3839	0.207245	0.416775
1373	fraud_Ziemann-Waters	1	10	2438	0.131613	0.410172
221	fraud_Cremin, Hamill and Reichel	1	9	2203	0.118927	0.408534
1184	fraud_Stark-Batz	1	11	2702	0.145865	0.407106
239	fraud_Cummings LLC	1	15	3721	0.200875	0.403117
761	fraud_Ledner-Pfannerstill	1	15	3764	0.203197	0.398512
1190	fraud_Stiedemann Inc	1	9	2265	0.122274	0.397351
533	fraud_Hudson-Grady	1	13	3273	0.176690	0.397189
1341	fraud_Wisozk and Sons	1	9	2275	0.122814	0.395604

1208	fraud_Stroman, Hudson and Erdman	1	15	3829	0.206705	0.391747
848	fraud_Metz-Boehm	1	13	3323	0.179389	0.391213
1039	fraud_Rodriguez Group	1	15	3843	0.207461	0.390320
215	fraud_Corwin-Collins	1	15	3853	0.208001	0.389307
450	fraud_Halvorson Group	1	9	2312	0.124811	0.389273
193	fraud_Collier Inc	1	5	1290	0.069640	0.387597
559	fraud_Jaskolski-Dibbert	1	5	1292	0.069748	0.386997
1150	fraud_Shields Inc	1	5	1292	0.069748	0.386997
303	fraud_Effertz LLC	1	10	2600	0.140359	0.384615
424	fraud_Gutmann Ltd	1	5	1300	0.070179	0.384615
813	fraud_Marks Inc	1	14	3643	0.196664	0.384299
890	fraud_Murray Ltd	1	5	1315	0.070989	0.380228
545	fraud_Huels-Nolan	1	14	3714	0.200497	0.376952
1170	fraud_Spinka Inc	1	5	1328	0.071691	0.376506
508	fraud_Hills-Boyer	1	10	2663	0.143760	0.375516
983	fraud_Ratke and Sons	1	9	2433	0.131344	0.369914
199	fraud_Connelly, Reichert and Fritsch	1	14	3788	0.204492	0.369588
886	fraud_Mraz-Herzog	1	14	3788	0.204492	0.369588
9	fraud_Adams, Kovacek and Kuhlman	1	5	1354	0.073095	0.369276
924	fraud_Olson, Becker and Koch	1	14	3806	0.205464	0.367840
383	fraud_Gislason Group	1	13	3556	0.191968	0.365579
1083	fraud_Satterfield-Lowe	1	4	1095	0.059113	0.365297
428	fraud_Gutmann-Upton	1	8	2218	0.119737	0.360685
1230	fraud_Thiel PLC	1	8	2232	0.120493	0.358423
488	fraud_Herman Inc	1	8	2238	0.120817	0.357462
1022	fraud_Reynolds-Schinner	1	8	2240	0.120925	0.357143
898	fraud_Nicolas, Hills and McGlynn	1	10	2806	0.151480	0.356379
1276	fraud_Turner, Ruecker and Parisian	1	8	2250	0.121464	0.355556
275	fraud_Dickinson-Rempel	1	8	2255	0.121734	0.354767
874	fraud_Morissette PLC	1	12	3391	0.183060	0.353878
165	fraud_Bruen-Yost	1	8	2266	0.122328	0.353045
462	fraud_Harris Inc	1	13	3700	0.199742	0.351351
167	fraud_Buckridge PLC	1	8	2278	0.122976	0.351185
1093	fraud_Schaefer, McGlynn and Bosco	1	13	3742	0.202009	0.347408
1160	fraud_Smith-Stokes	1	8	2306	0.124488	0.346921
143	fraud_Boyer-Haley	1	4	1156	0.062406	0.346021
587	fraud_Kassulke Inc	1	9	2601	0.140413	0.346021
1244	fraud_Torp-Labadie	1	13	3769	0.203466	0.344919
862	fraud_Monahan, Hermann and Johns	1	4	1162	0.062730	0.344234
1204	fraud_Streich, Hansen and Veum	1	13	3785	0.204330	0.343461
187	fraud_Christiansen-Gusikowski	1	8	2330	0.125783	0.343348
741	fraud_Langworth LLC	1	9	2624	0.141655	0.342988
494	fraud_Hermann-Gaylord	1	8	2338	0.126215	0.342173
381	fraud_Gibson-Deckow	1	9	2647	0.142896	0.340008
229	fraud_Crooks and Sons	1	9	2649	0.143004	0.339751
641	fraud_Kling Inc	1	13	3841	0.207353	0.338454
1178	fraud_Stamm-Rodriguez	1	8	2364	0.127619	0.338409
315	fraud_Emard Inc	1	13	3867	0.208757	0.336178
369	fraud_Fritsch LLC	1	4	1195	0.064511	0.334728
436	fraud_Hagenes, Hermann and Stroman	1	4	1199	0.064727	0.333611

107	fraud_Bins, Balistreri and Beatty	1	11	3321	0.179282	0.331226
7	fraud_Abshire PLC	1	9	2733	0.147539	0.329308
759	fraud_Ledner, Hartmann and Feest	1	8	2436	0.131506	0.328407
117	fraud_Block-Hauck	1	4	1229	0.066347	0.325468
219	fraud_Corwin-Romaguera	1	4	1231	0.066455	0.324939
777	fraud_Lind, Huel and McClure	1	12	3714	0.200497	0.323102
514	fraud_Hilpert-Conroy	1	11	3416	0.184410	0.322014
484	fraud_Heller-Abshire	1	4	1255	0.067750	0.318725
155	fraud_Brekke and Sons	1	12	3781	0.204114	0.317376
3	fraud_Abbott-Steuber	1	8	2529	0.136526	0.316331
339	fraud_Fahey Inc	1	10	3171	0.171184	0.315358
317	fraud_Emmerich-Luettgen	1	4	1274	0.068776	0.313972
1304	fraud_Watsica LLC	1	8	2549	0.137606	0.313849
543	fraud_Huels-Hahn	1	12	3835	0.207029	0.312907
1355	fraud_Wuckert-Walter	1	4	1286	0.069424	0.311042
109	fraud_Bins-Howell	1	8	2573	0.138901	0.310921
523	fraud_Homenick LLC	1	8	2578	0.139171	0.310318
464	fraud_Harris, Gusikowski and Heaney	1	4	1294	0.069856	0.309119
177	fraud_Cassin-Harvey	1	4	1296	0.069964	0.308642
1117	fraud_Schneider, Hayes and Nikolaus	1	8	2610	0.140899	0.306513
593	fraud_Keeling-Crist	1	7	2290	0.123624	0.305677
1224	fraud_Terry, Johns and Bins	1	7	2295	0.123894	0.305011
763	fraud_Leffler-Goldner	1	8	2626	0.141762	0.304646
1345	fraud_Wiza LLC	1	7	2305	0.124434	0.303688
251	fraud_Dare-Marvin	1	8	2641	0.142572	0.302916
767	fraud_Lehner, Reichert and Mills	1	7	2311	0.124757	0.302899
213	fraud_Cormier, Stracke and Thiel	1	8	2648	0.142950	0.302115
1339	fraud_Wintheiser, Dietrich and Schimmel	1	7	2319	0.125189	0.301854
1256	fraud_Towne, Walker and Borer	1	4	1332	0.071907	0.300300
1166	fraud_Spencer PLC	1	8	2672	0.144246	0.299401
647	fraud_Kling-Grant	1	4	1336	0.072123	0.299401
1325	fraud_Wilkinson Ltd	1	8	2688	0.145110	0.297619
1357	fraud_Yost, Block and Koepp	1	7	2355	0.127133	0.297240
231	fraud_Cruickshank-Mills	1	8	2692	0.145325	0.297177
995	fraud_Reichel Inc	1	8	2726	0.147161	0.293470
993	fraud_Raynor, Reinger and Hagenes	1	11	3763	0.203143	0.292320
87	fraud_Berge LLC	1	11	3786	0.204384	0.290544
1075	fraud_Runte, Green and Emard	1	7	2410	0.130102	0.290456
245	fraud_Dare, Casper and Bartoletti	1	9	3169	0.171076	0.284001
745	fraud_Larkin Ltd	1	9	3171	0.171184	0.283822
31	fraud_Bahringer, Bergnaum and Quitzon	1	10	3552	0.191752	0.281532
1206	fraud_Streich, Rolfson and Wilderman	1	9	3213	0.173451	0.280112
1099	fraud_Schiller, Blanda and Johnson	1	10	3585	0.193533	0.278940
123	fraud_Bode-Schuster	1	9	3257	0.175827	0.276328
363	fraud_Friesen-D'Amore	1	10	3640	0.196502	0.274725
997	fraud_Reichel LLC	1	7	2560	0.138200	0.273438
1148	fraud_Shanahan-Lehner	1	7	2563	0.138361	0.273117
287	fraud_Douglas, DuBuque and McKenzie	1	3	1101	0.059437	0.272480
1045	fraud_Rolfson-Kunde	1	7	2578	0.139171	0.271528
860	fraud_Monahan, Bogisich and Ledner	1	6	2223	0.120007	0.269906

323	fraud_Erdman-Ebert	1	7	2599	0.140305	0.269334
952	fraud_Parker-Kunde	1	7	2603	0.140521	0.268920
1375	fraud_Zieme, Bode and Dooley	1	10	3720	0.200821	0.268817
675	fraud_Kozey-Kuhlman	1	7	2606	0.140683	0.268611
61	fraud_Baumbach Ltd	1	7	2607	0.140737	0.268508
611	fraud_Kessler Group	1	3	1119	0.060408	0.268097
149	fraud_Bradtke, Torp and Bahringer	1	7	2623	0.141601	0.266870
531	fraud_Howe PLC	1	7	2629	0.141924	0.266261
399	fraud_Gottlieb-Hansen	1	7	2631	0.142032	0.266059
1128	fraud_Schroeder, Hauck and Treutel	1	7	2632	0.142086	0.265957
613	fraud_Kessler Inc	1	3	1130	0.061002	0.265487
305	fraud_Effertz, Welch and Schowalter	1	7	2639	0.142464	0.265252
785	fraud_Little-Gleichner	1	3	1131	0.061056	0.265252
279	fraud_Donnelly LLC	1	7	2647	0.142896	0.264450
269	fraud_Dibbert-Green	1	7	2661	0.143652	0.263059
751	fraud_Larson-Moen	1	7	2662	0.143706	0.262960
1300	fraud_Walter, Hettinger and Kessler	1	7	2664	0.143814	0.262763
679	fraud_Kris-Kertzmann	1	3	1143	0.061704	0.262467
595	fraud_Kemmer-Buckridge	1	6	2286	0.123408	0.262467
1361	fraud_Yost-Rogahn	1	7	2668	0.144030	0.262369
753	fraud_Leannon-Nikolaus	1	3	1146	0.061866	0.261780
151	fraud_Breitenberg LLC	1	3	1146	0.061866	0.261780
591	fraud_Kautzer and Sons	1	7	2680	0.144678	0.261194
57	fraud_Bauch-Blanda	1	7	2686	0.145002	0.260611
749	fraud_Larson, Quitzon and Spencer	1	3	1155	0.062352	0.259740
1142	fraud_Schuppe LLC	1	7	2722	0.146945	0.257164
900	fraud_Nienow PLC	1	7	2728	0.147269	0.256598
827	fraud_McDermott-Rice	1	6	2349	0.126809	0.255428
456	fraud_Hammes-Beatty	1	8	3150	0.170050	0.253968
836	fraud_McLaughlin, Armstrong and Koepp	1	3	1183	0.063863	0.253593
803	fraud_Lynch-Wisozk	1	9	3550	0.191644	0.253521
1101	fraud_Schimmel-Olson	1	8	3159	0.170536	0.253245
75	fraud_Bednar Inc	1	3	1190	0.064241	0.252101
157	fraud_Brown Inc	1	8	3174	0.171346	0.252048
253	fraud_Daugherty LLC	1	8	3201	0.172803	0.249922
1280	fraud_Ullrich Ltd	1	8	3202	0.172857	0.249844
301	fraud_Ebert-Daugherty	1	3	1203	0.064943	0.249377
247	fraud_Dare, Fritsch and Zboncak	1	6	2423	0.130804	0.247627
549	fraud_Hyatt-Blick	1	3	1219	0.065807	0.246103
1234	fraud_Thompson-Gleason	1	6	2439	0.131667	0.246002
1010	fraud_Reilly and Sons	1	6	2439	0.131667	0.246002
991	fraud_Raynor, Feest and Miller	1	9	3673	0.198284	0.245031
79	fraud_Beer-Jast	1	8	3279	0.177014	0.243977
1087	fraud_Schaefer Ltd	1	8	3290	0.177608	0.243161
205	fraud_Conroy, Balistreri and Gorczany	1	6	2472	0.133449	0.242718
1053	fraud_Romaguera, Wehner and Tromp	1	8	3302	0.178256	0.242277
719	fraud_Kutch, Hermiston and Farrell	1	9	3725	0.201091	0.241611
781	fraud_Little Ltd	1	8	3312	0.178796	0.241546
1152	fraud_Shields-Wunsch	1	6	2512	0.135608	0.238854
955	fraud_Pfeffer LLC	1	3	1260	0.068020	0.238095

1162	fraud_Smitham-Boehm	1	3	1265	0.068290	0.237154
665	fraud_Koss, Hansen and Lueilwitz	1	8	3383	0.182629	0.236477
842	fraud_Medhurst, Cartwright and Ebert	1	6	2547	0.137498	0.235571
661	fraud_Konopelski, Schneider and Hartmann	1	6	2556	0.137984	0.234742
39	fraud_Bahringer-Streich	1	6	2558	0.138092	0.234558
1018	fraud_Rempel PLC	1	3	1281	0.069154	0.234192
695	fraud_Kuhn Group	1	6	2563	0.138361	0.234101
33	fraud_Bahringer, Osinski and Block	1	6	2569	0.138685	0.233554
819	fraud_McCullough Group	1	3	1285	0.069370	0.233463
440	fraud_Hahn, Bahringer and McLaughlin	1	6	2570	0.138739	0.233463
815	fraud_Marvin-Lind	1	6	2575	0.139009	0.233010
713	fraud_Kutch Group	1	3	1289	0.069586	0.232739
967	fraud_Powlowski-Weimann	1	6	2581	0.139333	0.232468
518	fraud_Hintz-Bruen	1	3	1293	0.069802	0.232019
1158	fraud_Skiles-Ankunding	1	3	1298	0.070071	0.231125
137	fraud_Botsford PLC	1	8	3470	0.187325	0.230548
1176	fraud_Sporer-Keebler	1	6	2610	0.140899	0.229885
868	fraud_Moore, Williamson and Emmerich	1	8	3488	0.188297	0.229358
1351	fraud_Wuckert, Wintheiser and Friesen	1	8	3494	0.188621	0.228964
643	fraud_Kling, Howe and Schneider	1	8	3495	0.188675	0.228898
341	fraud_Feil, Hilpert and Koss	1	6	2624	0.141655	0.228659
1063	fraud_Rowe-Vandervort	1	3	1313	0.070881	0.228484
47	fraud_Bartoletti and Sons	1	6	2630	0.141978	0.228137
1073	fraud_Runolfsson and Sons	1	3	1315	0.070989	0.228137
1337	fraud_Windler, Goodwin and Kovacek	1	8	3507	0.189323	0.228115
1264	fraud_Tromp, Kerluke and Glover	1	3	1318	0.071151	0.227618
721	fraud_Kutch, Steuber and Gerhold	1	6	2639	0.142464	0.227359
1079	fraud_Rutherford, Homenick and Bergstrom	1	3	1320	0.071259	0.227273
460	fraud_Harris Group	1	6	2644	0.142734	0.226929
876	fraud_Morissette, Weber and Wiegand	1	3	1327	0.071637	0.226074
872	fraud_Morissette LLC	1	6	2660	0.143598	0.225564
1252	fraud_Towne LLC	1	5	2218	0.119737	0.225428
411	fraud_Graham, Hegmann and Hammes	1	6	2673	0.144300	0.224467
23	fraud_Auer LLC	1	6	2674	0.144354	0.224383
1020	fraud_Renner Ltd	1	8	3570	0.192724	0.224090
472	fraud_Heaney-Marquardt	1	6	2681	0.144732	0.223797
139	fraud_Botsford and Sons	1	8	3576	0.193047	0.223714
1077	fraud_Runte-Mohr	1	6	2684	0.144894	0.223547
15	fraud_Altenwerth-Kilback	1	8	3594	0.194019	0.222593
291	fraud_Douglas-White	1	6	2703	0.145919	0.221976
908	fraud_Nolan-Williamson	1	7	3160	0.170590	0.221519
902	fraud_Nienow, Ankunding and Collier	1	5	2268	0.122436	0.220459
1125	fraud_Schoen-Quigley	1	7	3201	0.172803	0.218682
267	fraud_Dibbert and Sons	1	6	2758	0.148888	0.217549
1246	fraud_Torp-Lemke	1	5	2302	0.124272	0.217202
1218	fraud_Swift PLC	1	7	3246	0.175233	0.215650
53	fraud_Barton LLC	1	7	3261	0.176042	0.214658
1296	fraud_Waelchi Inc	1	7	3262	0.176096	0.214592
1069	fraud_Ruecker-Mayert	1	7	3273	0.176690	0.213871
961	fraud_Pouros, Walker and Spencer	1	7	3279	0.177014	0.213480

353	fraud_Flatley Group	1	5	2366	0.127727	0.211327
179	fraud_Champlin and Sons	1	7	3344	0.180523	0.209330
551	fraud_Jacobi Inc	1	5	2405	0.129832	0.207900
29	fraud_Bahringer Group	1	5	2435	0.131452	0.205339
1156	fraud_Skiles LLC	1	7	3458	0.186677	0.202429
93	fraud_Berge-Ullrich	1	7	3465	0.187055	0.202020
319	fraud_Emmerich-Rau	1	5	2510	0.135500	0.199203
896	fraud_Nader-Maggio	1	5	2540	0.137120	0.196850
520	fraud_Hirthe-Beier	1	5	2541	0.137174	0.196773
637	fraud_Kirlin and Sons	1	5	2552	0.137768	0.195925
1278	fraud_Turner, Ziemann and Lehner	1	5	2554	0.137876	0.195771
1222	fraud_Terry Ltd	1	7	3583	0.193425	0.195367
739	fraud_Langosh, Wintheiser and Hyatt	1	5	2569	0.138685	0.194628
1123	fraud_Schoen, Nienow and Bauch	1	5	2571	0.138793	0.194477
365	fraud_Friesen-Ortiz	1	5	2571	0.138793	0.194477
1035	fraud_Roberts, Ryan and Smith	1	5	2575	0.139009	0.194175
1107	fraud_Schmeler-Howe	1	5	2575	0.139009	0.194175
975	fraud_Prosacco LLC	1	5	2589	0.139765	0.193125
1323	fraud_Wilkinson LLC	1	5	2589	0.139765	0.193125
747	fraud_Larkin, Stracke and Greenfelder	1	5	2602	0.140467	0.192160
327	fraud_Erdman-Schaden	1	5	2605	0.140629	0.191939
667	fraud_Koss, McLaughlin and Mayer	1	5	2608	0.140791	0.191718
257	fraud_Daugherty-Thompson	1	5	2609	0.140845	0.191644
878	fraud_Morissette-Schaefer	1	5	2614	0.141115	0.191278
629	fraud_Kilback Group	1	5	2620	0.141439	0.190840
430	fraud_Haag-Blanda	1	5	2624	0.141655	0.190549
1097	fraud_Schiller Ltd	1	5	2625	0.141709	0.190476
846	fraud_Metz, Russel and Metz	1	6	3163	0.170752	0.189693
1033	fraud_Roberts, Daniel and Macejkovic	1	6	3166	0.170914	0.189514
153	fraud_Breitenberg-Hermiston	1	6	3168	0.171022	0.189394
779	fraud_Lind-Buckridge	1	5	2642	0.142626	0.189251
99	fraud_Bernhard-Lesch	1	5	2645	0.142788	0.189036
1174	fraud_Sporer Inc	1	7	3719	0.200767	0.188223
391	fraud_Goldner-Lemke	1	5	2665	0.143868	0.187617
1365	fraud_Zboncak Ltd	1	5	2679	0.144624	0.186637
904	fraud_Nienow, Barrows and Romaguera	1	5	2684	0.144894	0.186289
1318	fraud_Welch, Rath and Koepp	1	5	2684	0.144894	0.186289
1268	fraud_Turcotte, McKenzie and Koss	1	5	2689	0.145164	0.185943
335	fraud_Fadel, Mertz and Rippin	1	5	2693	0.145379	0.185667
395	fraud_Gottlieb Group	1	6	3245	0.175179	0.184900
1232	fraud_Thiel-Thiel	1	5	2712	0.146405	0.184366
357	fraud_Frami Group	1	5	2714	0.146513	0.184230
133	fraud_Bogisich-Weimann	1	6	3260	0.175988	0.184049
527	fraud_Hoppe-Parisian	1	6	3288	0.177500	0.182482
163	fraud_Brown-Greenholt	1	5	2742	0.148025	0.182349
989	fraud_Rau-Robel	1	6	3292	0.177716	0.182260
1260	fraud_Treutel-King	1	2	1098	0.059275	0.182149
313	fraud_Eichmann-Russel	1	4	2196	0.118549	0.182149
1314	fraud_Weimann-Lockman	1	6	3300	0.178148	0.181818
1119	fraud_Schoen Ltd	1	6	3302	0.178256	0.181708

273	fraud_Dickinson Ltd	1	9	4953	0.267384	0.181708
1359	fraud_Yost, Schamberger and Windler	1	6	3313	0.178850	0.181105
1154	fraud_Simonis-Prohaska	1	4	2220	0.119845	0.180180
844	fraud_Medhurst, Labadie and Gottlieb	1	2	1111	0.059976	0.180018
571	fraud_Jerde-Hermann	1	2	1112	0.060030	0.179856
1037	fraud_Roberts-Beahan	1	4	2249	0.121410	0.177857
833	fraud_McGlynn-Jaskolski	1	4	2250	0.121464	0.177778
1270	fraud_Turcotte-Halvorson	1	4	2271	0.122598	0.176134
723	fraud_Kutch-Ferry	1	6	3427	0.185004	0.175080
115	fraud_Block Group	1	4	2297	0.124002	0.174140
1103	fraud_Schmeler Inc	1	4	2300	0.124164	0.173913
407	fraud_Goyette-Herzog	1	2	1152	0.062190	0.173611
500	fraud_Herzog Ltd	1	4	2305	0.124434	0.173536
1331	fraud_Williamson LLC	1	4	2309	0.124650	0.173235
635	fraud_Kilback, Nitzsche and Leffler	1	2	1155	0.062352	0.173160
1200	fraud_Streich Ltd	1	6	3468	0.187217	0.173010
1343	fraud_Witting, Beer and Ernser	1	6	3468	0.187217	0.173010
807	fraud_Maggio-Fahey	1	4	2314	0.124919	0.172861
705	fraud_Kunze, Larkin and Mayert	1	2	1158	0.062514	0.172712
1130	fraud_Schroeder, Wolff and Hermiston	1	2	1167	0.063000	0.171380
1286	fraud_Veum-Koelpin	1	2	1169	0.063108	0.171086
329	fraud_Ernser-Feest	1	6	3516	0.189808	0.170648
529	fraud_Howe Ltd	1	4	2345	0.126593	0.170576
825	fraud_McDermott, Osinski and Morar	1	6	3527	0.190402	0.170116
375	fraud_Gaylord-Powlowski	1	6	3534	0.190780	0.169779
801	fraud_Lynch-Mohr	1	2	1180	0.063701	0.169492
1228	fraud_Thiel Ltd	1	2	1186	0.064025	0.168634
249	fraud_Dare-Gibson	1	4	2373	0.128104	0.168563
1236	fraud_Tillman LLC	1	2	1188	0.064133	0.168350
583	fraud_Johnston-Casper	1	2	1201	0.064835	0.166528
651	fraud_Klocko, Runolfsdottir and Breitenberg	1	4	2417	0.130480	0.165494
466	fraud_Hartmann, Rowe and Hermann	1	2	1215	0.065591	0.164609
1059	fraud_Roob-Okuneva	1	4	2442	0.131829	0.163800
452	fraud_Hamill-D'Amore	1	4	2462	0.132909	0.162470
37	fraud_Bahringer-Larson	1	2	1240	0.066940	0.161290
506	fraud_Hills, Hegmann and Schaefer	1	4	2482	0.133989	0.161160
361	fraud_Friesen Ltd	1	4	2489	0.134367	0.160707
237	fraud_Cummings Group	1	2	1250	0.067480	0.160000
510	fraud_Hills-Olson	1	2	1256	0.067804	0.159236
5	fraud_Abernathy and Sons	1	4	2513	0.135662	0.159172
77	fraud_Bednar PLC	1	5	3143	0.169672	0.159084
1302	fraud_Waters-Cruickshank	1	4	2524	0.136256	0.158479
547	fraud_Hyatt, Russel and Gleichner	1	4	2531	0.136634	0.158040
446	fraud_Haley, Batz and Auer	1	4	2534	0.136796	0.157853
11	fraud_Adams-Barrows	1	4	2535	0.136850	0.157791
864	fraud_Monahan-Morar	1	4	2542	0.137228	0.157356
373	fraud_Funk Group	1	2	1271	0.068614	0.157356
1258	fraud_Trantow PLC	1	2	1281	0.069154	0.156128
69	fraud_Bechtelar-Rippin	1	4	2566	0.138523	0.155885
918	fraud_O'Keefe-Wisoky	1	4	2570	0.138739	0.155642

565	fraud_Jast and Sons	1	4	2574	0.138955	0.155400
791	fraud_Lowe, Dietrich and Erdman	1	5	3222	0.173937	0.155183
685	fraud_Kub PLC	1	4	2581	0.139333	0.154979
225	fraud_Crona and Sons	1	5	3227	0.174207	0.154943
1363	fraud_Zboncak LLC	1	4	2586	0.139603	0.154679
1146	fraud_Schuppe-Schuppe	1	4	2593	0.139981	0.154261
331	fraud_Ernser-Lynch	1	4	2600	0.140359	0.153846
1164	fraud_Smitham-Schiller	1	2	1301	0.070233	0.153728
1377	fraud_Zulauf LLC	1	4	2605	0.140629	0.153551
241	fraud_Dach-Borer	1	2	1303	0.070341	0.153492
1212	fraud_Swaniawski, Bahringer and Ledner	1	2	1304	0.070395	0.153374
936	fraud_Pacocha-Weissnat	1	5	3271	0.176582	0.152858
914	fraud_O'Hara-Wilderman	1	4	2618	0.141331	0.152788
271	fraud_Dicki Ltd	1	2	1313	0.070881	0.152323
735	fraud_Lakin, Ferry and Beatty	1	4	2630	0.141978	0.152091
1329	fraud_Will Ltd	1	4	2636	0.142302	0.151745
1282	fraud_Upton PLC	1	4	2644	0.142734	0.151286
1	fraud_Abbott-Rogahn	1	4	2647	0.142896	0.151114
89	fraud_Berge, Kautzer and Harris	1	4	2648	0.142950	0.151057
755	fraud_Leannon-Ward	1	4	2656	0.143382	0.150602
633	fraud_Kilback and Sons	1	4	2657	0.143436	0.150546
1194	fraud_Stokes, Christiansen and Sipes	1	2	1329	0.071745	0.150489
639	fraud_Klein Group	1	4	2664	0.143814	0.150150
81	fraud_Beier LLC	1	4	2676	0.144462	0.149477
525	fraud_Hoppe, Harris and Bednar	1	4	2681	0.144732	0.149198
1014	fraud_Reinger, Weissnat and Strosin	1	4	2683	0.144840	0.149087
938	fraud_Padberg-Rogahn	1	4	2696	0.145541	0.148368
421	fraud_Grimes LLC	1	4	2707	0.146135	0.147765
645	fraud_Kling-Ernser	1	4	2709	0.146243	0.147656
1071	fraud_Runolfsdottir, Mueller and Hand	1	4	2727	0.147215	0.146681
912	fraud_O'Connell-Ullrich	1	5	3460	0.186785	0.144509
502	fraud_Hettinger, McCullough and Fay	1	5	3471	0.187379	0.144051
191	fraud_Cole, Hills and Jewess	1	5	3508	0.189377	0.142531
557	fraud_Jakubowski Inc	1	5	3517	0.189862	0.142167
727	fraud_Kutch-Wilderman	1	5	3562	0.192292	0.140371
1320	fraud_White and Sons	1	5	3570	0.192724	0.140056
910	fraud_O'Connell, Botsford and Hand	1	5	3578	0.193155	0.139743
920	fraud_O'Reilly, Mohr and Purdy	1	5	3605	0.194613	0.138696
125	fraud_Boehm, Block and Jakubowski	1	3	2214	0.119521	0.135501
821	fraud_McCullough LLC	1	3	2328	0.125675	0.128866
1298	fraud_Waelchi-Wolf	1	4	3117	0.168269	0.128329
1266	fraud_Turcotte, Batz and Buckridge	1	3	2349	0.126809	0.127714
496	fraud_Hermiston, Pacocha and Smith	1	3	2352	0.126971	0.127551
321	fraud_Erdman-Durgan	1	3	2391	0.129076	0.125471
1310	fraud_Weber, Thiel and Hammes	1	4	3189	0.172156	0.125431
609	fraud_Kertzmann LLC	1	3	2395	0.129292	0.125261
385	fraud_Gleason and Sons	1	4	3216	0.173613	0.124378
415	fraud_Greenholt Ltd	1	3	2419	0.130588	0.124018
737	fraud_Lang, Towne and Schuppe	1	4	3226	0.174153	0.123993
468	fraud_Hauck, Dietrich and Funk	1	4	3230	0.174369	0.123839
516	fraud_Hintz, Bauch and Smith	1	3	2427	0.131020	0.123609

316	fraud_Hintz, Bauch and Smith	1	3	2427	0.131020	0.123809
1047	fraud_Romaguera Ltd	1	3	2433	0.131344	0.123305
171	fraud_Cartwright PLC	1	4	3258	0.175881	0.122775
227	fraud_Cronin, Kshlerin and Weber	1	3	2446	0.132045	0.122649
169	fraud_Carroll PLC	1	3	2449	0.132207	0.122499
1242	fraud_Torp, Muller and Borer	1	3	2452	0.132369	0.122349
1216	fraud_Swaniawski, Nitzsche and Welch	1	3	2471	0.133395	0.121408
101	fraud_Bernier and Sons	1	4	3303	0.178310	0.121102
343	fraud_Feil-Morar	1	3	2482	0.133989	0.120870
121	fraud_Bode-Rempel	1	4	3319	0.179174	0.120518
333	fraud_Fadel Inc	1	3	2523	0.136202	0.118906
1371	fraud_Zemlak, Tillman and Cremin	1	3	2538	0.137012	0.118203
209	fraud_Conroy-Emard	1	3	2559	0.138146	0.117233
1008	fraud_Reilly LLC	1	4	3420	0.184626	0.116959
823	fraud_McCullough, Hudson and Schuster	1	3	2567	0.138577	0.116868
1353	fraud_Wuckert-Goldner	1	4	3425	0.184896	0.116788
627	fraud_Kihn-Schuster	1	3	2574	0.138955	0.116550
940	fraud_Padberg-Sauer	1	4	3432	0.185274	0.116550
888	fraud_Mueller, Gerhold and Mueller	1	3	2583	0.139441	0.116144
731	fraud_Labadie LLC	1	3	2586	0.139603	0.116009
438	fraud_Hagenes, Kohler and Hoppe	1	3	2590	0.139819	0.115830
1192	fraud_Stiedemann Ltd	1	3	2592	0.139927	0.115741
1308	fraud_Weber and Sons	1	3	2608	0.140791	0.115031
623	fraud_Kihn, Brakus and Goyette	1	3	2615	0.141169	0.114723
83	fraud_Beier and Sons	1	4	3492	0.188513	0.114548
928	fraud_Osinski Inc	1	3	2628	0.141870	0.114155
183	fraud_Champlin-Casper	1	4	3505	0.189215	0.114123
197	fraud_Connelly PLC	1	3	2633	0.142140	0.113938
1132	fraud_Schulist Ltd	1	3	2640	0.142518	0.113636
21	fraud_Armstrong, Walter and Gottlieb	1	3	2649	0.143004	0.113250
1186	fraud_Stark-Koss	1	4	3533	0.190726	0.113218
233	fraud_Cummerata-Hilpert	1	3	2658	0.143490	0.112867
201	fraud_Connelly-Carter	1	4	3555	0.191914	0.112518
575	fraud_Johns Inc	1	3	2676	0.144462	0.112108
577	fraud_Johns-Hoeger	1	3	2681	0.144732	0.111899
113	fraud_Bins-Tillman	1	3	2729	0.147323	0.109930
498	fraud_Hermiston, Russel and Price	1	3	2763	0.149158	0.108578
1327	fraud_Wilkinson PLC	1	3	3110	0.167891	0.096463
795	fraud_Lubowitz-Walter	1	3	3215	0.173559	0.093313
987	fraud_Rau-Grant	1	3	3249	0.175395	0.092336
297	fraud_Durgan, Gislason and Spencer	1	3	3375	0.182197	0.088889
1272	fraud_Turner LLC	1	1	1154	0.062298	0.086655
1347	fraud_Wiza, Schaden and Stark	1	2	2317	0.125081	0.086319
195	fraud_Collier LLC	1	3	3489	0.188351	0.085985
659	fraud_Kohler, Lindgren and Koelpin	1	1	1169	0.063108	0.085543
1049	fraud_Romaguera and Sons	1	1	1172	0.063269	0.085324
223	fraud_Crist, Jakubowski and Littel	1	3	3529	0.190510	0.085010
426	fraud_Gutmann, McLaughlin and Wiza	1	3	3531	0.190618	0.084962
432	fraud_Hackett Group	1	1	1185	0.063971	0.084388
959	fraud_Pollich LLC	1	3	3558	0.192076	0.084317
981	fraud_Quitzon-Goyette	1	3	3562	0.192292	0.084222

1250	fraud_Torphy-Kertzmann	1	2	2377	0.128320	0.084140
838	fraud_Medhurst Inc	1	3	3600	0.194343	0.083333
999	fraud_Reichel, Bradtke and Blanda	1	1	1201	0.064835	0.083264
337	fraud_Fadel-Hilpert	1	2	2402	0.129670	0.083264
265	fraud_Denesik, Powlowski and Pouros	1	3	3611	0.194937	0.083079
480	fraud_Heller PLC	1	2	2408	0.129994	0.083056
311	fraud_Eichmann-Kilback	1	3	3616	0.195207	0.082965
409	fraud_Graham and Sons	1	2	2435	0.131452	0.082136
371	fraud_Fritsch and Sons	1	2	2436	0.131506	0.082102
699	fraud_Kulas Group	1	2	2466	0.133125	0.081103
811	fraud_Mante, Luettgen and Hackett	1	2	2472	0.133449	0.080906
161	fraud_Brown, Homenick and Lesch	1	2	2483	0.134043	0.080548
793	fraud_Lubowitz, Terry and Stracke	1	2	2542	0.137228	0.078678
926	fraud_Ortiz Group	1	1	1291	0.069694	0.077459
870	fraud_Morar Inc	1	1	1292	0.069748	0.077399
555	fraud_Jakubowski Group	1	2	2599	0.140305	0.076953
597	fraud_Kemmer-Reinger	1	1	1327	0.071637	0.075358
1290	fraud_Volkman PLC	1	2	2684	0.144894	0.074516
625	fraud_Kihn-Fritsch	1	2	2697	0.145595	0.074156
71	fraud_Becker, Harris and Harvey	1	2	2737	0.147755	0.073073
405	fraud_Goyette-Gerhold	1	2	3300	0.178148	0.060606
91	fraud_Berge-Hills	1	2	3317	0.179066	0.060295
1111	fraud_Schmidt-Larkin	1	2	3423	0.184788	0.058428
377	fraud_Gerhold LLC	1	2	3441	0.185760	0.058123
1026	fraud_Rippin-VonRueden	1	1	2478	0.133773	0.040355
277	fraud_Dietrich-Fadel	1	1	2487	0.134259	0.040209
687	fraud_Kub-Heaney	1	1	2491	0.134475	0.040145

OneHotEncoding

259	fraud_Deckow-Dare	1	1	2588	0.139711	0.038640

```
category_onehot = pd.get_dummies(df1.category, prefix='category', drop_first=True)
gender_onehot = pd.get_dummies(df1.gender, prefix='gender', drop_first=True)
day_of_week_onehot = pd.get_dummies(df1.day_of_week, prefix='week',drop_first=True)
```

1089	fraud_Schaefer, Fay and Hilll	1	1	2664	0.143814	0.037538

```
df2 = pd.concat([df1, category_onehot,gender_onehot,day_of_week_onehot], axis=1)
```

```
df2.head()
```

	trans_date_trans_time	cc_num	merchant	category	amt	first	last	gender	street	city	state

```
def haversine(lat1, lon1, lat2, lon2, to_radians=True, earth_radius=6371):
    """
    Calculate the great circle distance between two points
    on the earth (specified in decimal degrees or in radians)

    All (lat, lon) coordinates must have numeric dtypes and be of equal length.

    """
    if to_radians:
        lat1, lon1, lat2, lon2 = np.radians([lat1, lon1, lat2, lon2])

    a = np.sin((lat2-lat1)/2.0)**2 + \
        np.cos(lat1) * np.cos(lat2) * np.sin((lon2-lon1)/2.0)**2

    return earth_radius * 2 * np.arcsin(np.sqrt(a))
```

3	2019-01-01 00:01:16	3534093764340240	Hermiston and	gas_transport	45.000000	Jeremy	White	M	Court	Boulder	MT

```
df2['dist'] = \
    haversine(df2['lat'], df2['long'],
              df2['merch_lat'], df2['merch_long'])
```

4	2019-01-01 00:03:06	375534208663984	_ Crist	misc_pos	41.960000	Tyler	Garcia	M	Bradley	Doe Hill	VA

```
df2['dist'].describe()
```

```
count    1852394.000000
mean          76.111726
std           29.116970
min            0.022255
25%           55.320087
50%           78.216380
75%           98.509467
max          152.117173
Name: dist, dtype: float64
```

```
df2.dtypes
```

```
trans_date_trans_time      datetime64[ns]
cc_num                              int64
merchant                           object
category                           object
amt                               float64
first                              object
last                               object
gender                             object
street                             object
city                               object
state                              object
zip                                 int64
lat                               float64
long                              float64
city_pop                            int64
job                                object
dob                        datetime64[ns]
trans_num                          object
unix_time                           int64
merch_lat                         float64
merch_long                        float64
is_fraud                            int64
trans_hour                          int64
day_of_week                        object
year_month                      period[M]
age                               float64
category_food_dining                uint8
category_gas_transport              uint8
category_grocery_net                uint8
category_grocery_pos                uint8
category_health_fitness             uint8
category_home                       uint8
category_kids_pets                  uint8
category_misc_net                   uint8
category_misc_pos                   uint8
category_personal_care              uint8
category_shopping_net               uint8
category_shopping_pos               uint8
category_travel                     uint8
gender_M                            uint8
week_Monday                         uint8
week_Saturday                       uint8
week_Sunday                         uint8
week_Thursday                       uint8
week_Tuesday                        uint8
week_Wednesday                      uint8
dist                              float64
dtype: object
```

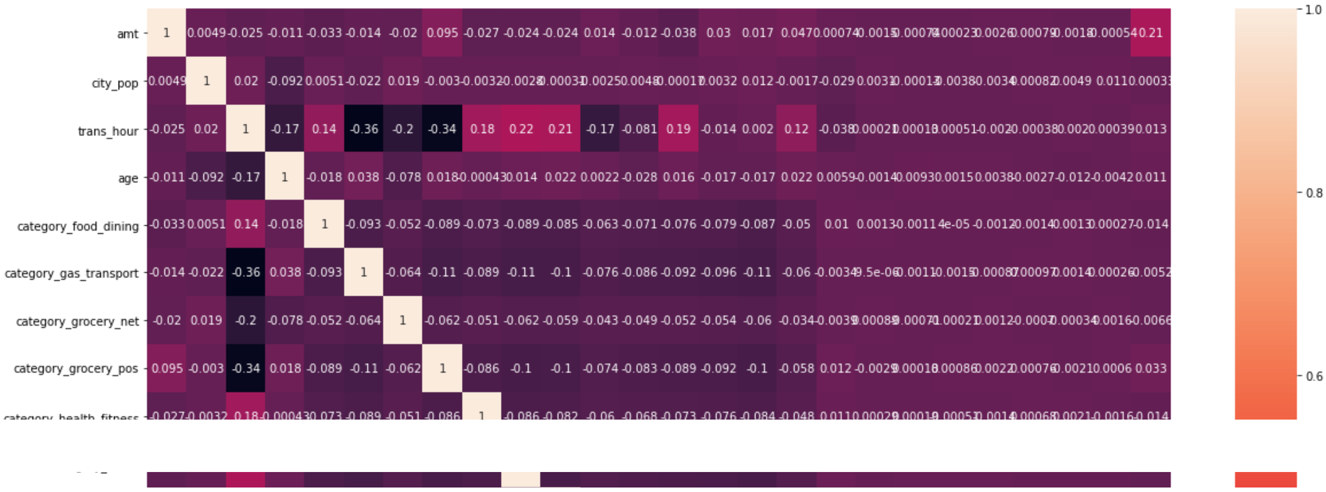
```
df2.columns
```

```
Index(['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', 'trans_hour', 'day_of_week', 'year_month',
       'age', 'category_food_dining', 'category_gas_transport',
       'category_grocery_net', 'category_grocery_pos',
       'category_health_fitness', 'category_home', 'category_kids_pets',
       'category_misc_net', 'category_misc_pos', 'category_personal_care',
       'category_shopping_net', 'category_shopping_pos', 'category_travel',
       'gender_M', 'week_Monday', 'week_Saturday', 'week_Sunday',
       'week_Thursday', 'week_Tuesday', 'week_Wednesday', 'dist'],
      dtype='object')
```

```
cols = ['amt','city_pop', 'trans_hour',
       'age', 'category_food_dining', 'category_gas_transport',
       'category_grocery_net', 'category_grocery_pos',
       'category_health_fitness', 'category_home', 'category_kids_pets',
       'category_misc_net', 'category_misc_pos', 'category_personal_care',
       'category_shopping_net', 'category_shopping_pos', 'category_travel',
       'gender_M','week_Monday','week_Tuesday', 'week_Wednesday','week_Thursday',
        'week_Saturday', 'week_Sunday','dist','is_fraud']
```

```
corr = df2[cols].corr()
```

```
import seaborn as sn

fig, ax = plt.subplots(figsize=(20,20))
sn.heatmap(corr, annot=True)
plt.show()
```



DA Track


```
import plotly
import plotly.graph_objects as go
```



```
df2.head()
```

	trans_date_trans_time	cc_num	merchant	category	amt	first	last	gender	street	city	state
0	2019-01-01 00:00:18	2703186189652095	fraud_Rippin, Kub and Mann	misc_net	4.970000	Jennifer	Banks	F	561 Perry Cove	Moravian Falls	NC
1	2019-01-01 00:00:44	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.230000	Stephanie	Gill	F	43039 Riley Greens Suite 393	Orient	WA
2	2019-01-01 00:00:51	38859492057661	fraud_Lind-Buckridge	entertainment	220.110000	Edward	Sanchez	M	594 White Dale Suite 530	Malad City	ID
3	2019-01-01 00:01:16	3534093764340240	fraud_Kutch, Hermiston and Farrell	gas_transport	45.000000	Jeremy	White	M	9443 Cynthia Court Apt. 038	Boulder	MT
4	2019-01-01 00:03:06	375534208663984	fraud_Keeling-Crist	misc_pos	41.960000	Tyler	Garcia	M	408 Bradley Rest	Doe Hill	VA

```
df_fraud_merchant = df1[['merchant','is_fraud','trans_num']].groupby(['merchant','is_fraud']).count().reset_index()
df_fraud_merchant.columns = ['Merchant','is_fraud','count']

df_fraud_merchant = df_fraud_merchant.merge(df_merchant[['Merchant','merchant_count','percent']],how='inner',\
                                    left_on='Merchant',right_on='Merchant')


df_fraud_merchant['percent_grp'] = (df_fraud_merchant['count']/df_fraud_merchant['merchant_count'])*100


df_job = df1[['job','trans_num']].groupby(['job']).count().reset_index()
df_job.columns = ['Job','tran_count_by_job']

df_job['percent'] = (df_job['tran_count_by_job']/df_job['tran_count_by_job'].sum())*100

df_job.sort_values(by = ['percent'], ascending=False)
```

	Job	tran_count_by_job	percent
194	Film/video editor	13898	0.750272
188	Exhibition designer	13167	0.710810
442	Surveyor, land/geomatics	12436	0.671347
309	Naval architect	12434	0.671239
287	Materials engineer	11711	0.632209
121	Designer, ceramics/pottery	11688	0.630967
179	Environmental consultant	10974	0.592423
195	Financial adviser	10963	0.591829
447	Systems developer	10962	0.591775
240	IT trainer	10943	0.590749
110	Copywriter, advertising	10241	0.552852
410	Scientist, audiological	10234	0.552474
71	Chartered public finance accountant	10211	0.551233
74	Chief Executive Officer	10199	0.550585
346	Podiatrist	9525	0.514199
102	Comptroller	9515	0.513660
278	Magazine features editor	9506	0.513174
16	Agricultural consultant	9500	0.512850
329	Paramedic	9494	0.512526
439	Sub	9488	0.512202
42	Audiological scientist	8801	0.475115
227	Historic buildings inspector/conservation officer	8787	0.474359
55	Building surveyor	8786	0.474305
272	Librarian, public	8773	0.473603
307	Musician	8772	0.473549
415	Scientist, research (maths)	8768	0.473333
44	Barrister	8767	0.473279
91	Clothing/textile technologist	8765	0.473171
299	Mining engineer	8762	0.473010
243	Immunologist	8760	0.472902
492	Water engineer	8740	0.471822
383	Quantity surveyor	8080	0.436192
288	Mechanical engineer	8062	0.435221
418	Secondary school teacher	8056	0.434897
196	Financial trader	8054	0.434789
353	Prison officer	8054	0.434789
261	Land/geomatics surveyor	8052	0.434681
406	Sales professional, IT	8052	0.434681
155	Engineer, automotive	8050	0.434573
113	Counsellor	8047	0.434411
335	Petroleum engineer	8046	0.434357
373	Psychologist, forensic	8044	0.434249
86	Claims inspector/assessor	8042	0.434141
137	Early years teacher	8041	0.434087
216	Geoscientist	8041	0.434087
151	Energy engineer	8038	0.433925
333	Pensions consultant	8036	0.433817
376	Psychotherapist, child	8036	0.433817
281	Make	8028	0.433385

198	Firefighter	8021	0.433007
72	Chemical engineer	7334	0.395920
408	Science writer	7332	0.395812
156	Engineer, biomedical	7330	0.395704
136	Drilling engineer	7321	0.395218
397	Research scientist (physical sciences)	7319	0.395110
292	Medical sales representative	7309	0.394570
271	Librarian, academic	7307	0.394463
413	Scientist, marine	7306	0.394409
479	Trade mark attorney	7304	0.394301
251	Insurance underwriter	7301	0.394139
147	Electrical engineer	7301	0.394139
115	Cytogeneticist	7297	0.393923
462	Television production assistant	7297	0.393923
70	Chartered loss adjuster	7296	0.393869
431	Special educational needs teacher	7283	0.393167
480	Trading standards officer	6611	0.356890
4	Accounting technician	6595	0.356026
472	Therapist, occupational	6594	0.355972
112	Counselling psychologist	6590	0.355756
443	Surveyor, minerals	6589	0.355702
146	Educational psychologist	6588	0.355648
120	Dealer	6586	0.355540
171	Engineer, production	6584	0.355432
385	Race relations officer	6583	0.355378
386	Radio broadcast assistant	6582	0.355324
301	Multimedia programmer	6582	0.355324
426	Social researcher	6580	0.355216
387	Radio producer	6579	0.355162
162	Engineer, control and instrumentation	6579	0.355162
456	Teacher, special educational needs	6578	0.355108
78	Chief Strategy Officer	6577	0.355054
458	Technical brewer	6576	0.355000
197	Fine artist	6576	0.355000
65	Ceramics designer	6569	0.354622
341	Physiotherapist	6566	0.354460
478	Toxicologist	6555	0.353866
421	Senior tax professional/tax inspector	5877	0.317265
463	Television/film/video producer	5871	0.316941
206	Further education lecturer	5865	0.316617
411	Scientist, biomedical	5862	0.316455
30	Archaeologist	5860	0.316347
207	Futures trader	5860	0.316347
56	Buyer, industrial	5857	0.316185
270	Lexicographer	5857	0.316185
176	Engineering geologist	5857	0.316185
355	Probation officer	5856	0.316131
124	Designer, industrial/product	5856	0.316131
130	Development worker, community	5852	0.315916
440	Surgeon	5852	0.315916

12	Advertising account planner	5852	0.315916
343	Pilot, airline	5851	0.315862
114	Curator	5851	0.315862
477	Town planner	5850	0.315808
350	Press photographer	5850	0.315808
316	Occupational psychologist	5848	0.315700
190	Facilities manager	5846	0.315592
380	Public relations account executive	5841	0.315322
189	Exhibitions officer, museum/gallery	5841	0.315322
389	Radiographer, therapeutic	5839	0.315214
81	Child psychotherapist	5839	0.315214
178	English as a second language teacher	5837	0.315106
223	Herbalist	5833	0.314890
366	Programme researcher, broadcasting/film/video	5832	0.314836
28	Applications developer	5826	0.314512
92	Colour technologist	5146	0.277803
52	Broadcast presenter	5143	0.277641
393	Research officer, trade union	5134	0.277155
250	Insurance risk surveyor	5133	0.277101
187	Exercise physiologist	5132	0.277047
446	Systems analyst	5132	0.277047
82	Chiropodist	5130	0.276939
266	Lecturer, further education	5130	0.276939
483	Travel agency manager	5128	0.276831
201	Forensic psychologist	5127	0.276777
232	Hospital doctor	5127	0.276777
392	Research officer, political party	5123	0.276561
264	Learning disability nurse	5122	0.276507
59	Call centre manager	5122	0.276507
199	Fisheries officer	5121	0.276453
372	Psychologist, counselling	5121	0.276453
402	Retail merchandiser	5120	0.276399
322	Operations geologist	5120	0.276399
224	Heritage manager	5119	0.276345
354	Private music teacher	5119	0.276345
140	Editor, commissioning	5117	0.276237
438	Structural engineer	5116	0.276183
6	Administrator	5115	0.276129
133	Dispensing optician	5113	0.276021
169	Engineer, mining	5112	0.275967
23	Amenity horticulturist	5111	0.275913
339	Physicist, medical	5110	0.275859
248	Insurance broker	5108	0.275751
164	Engineer, electronics	5107	0.275697
145	Education officer, museum	4404	0.237746
267	Lecturer, higher education	4404	0.237746
484	Tree surgeon	4403	0.237692
131	Development worker, international aid	4403	0.237692
45	Barrister's clerk	4401	0.237584
315	Occupational hygienist	4400	0.237530

93	Commercial horticulturist	4399	0.237476
297	Metallurgist	4398	0.237422
203	Freight forwarder	4397	0.237369
364	Production manager	4397	0.237369
273	Licensed conveyancer	4396	0.237315
29	Arboriculturist	4396	0.237315
208	Gaffer	4395	0.237261
356	Producer, radio	4394	0.237207
388	Radiographer, diagnostic	4394	0.237207
152	Energy manager	4392	0.237099
213	Geologist, engineering	4392	0.237099
306	Music tutor	4392	0.237099
424	Site engineer	4391	0.237045
126	Designer, jewellery	4391	0.237045
259	Journalist, newspaper	4389	0.236937
489	Warden/ranger	4389	0.236937
342	Phytotherapist	4389	0.236937
77	Chief Operating Officer	4388	0.236883
62	Careers information officer	4387	0.236829
391	Regulatory affairs officer	4386	0.236775
486	Video editor	4386	0.236775
461	Television floor manager	4385	0.236721
48	Biomedical scientist	4384	0.236667
269	Leisure centre manager	4382	0.236559
94	Commercial/residential surveyor	4382	0.236559
165	Engineer, land	4381	0.236505
148	Electronics engineer	4380	0.236451
450	Tax inspector	4379	0.236397
122	Designer, exhibition/display	4378	0.236343
283	Manufacturing engineer	4378	0.236343
476	Tourist information centre manager	4375	0.236181
60	Camera operator	4375	0.236181
219	Health physicist	4374	0.236127
360	Product/process development scientist	4374	0.236127
123	Designer, furniture	4372	0.236019
351	Press sub	4369	0.235857
107	Contractor	4366	0.235695
312	Nurse, children's	3684	0.198878
10	Administrator, local government	3677	0.198500
212	Geochemist	3677	0.198500
470	Therapist, horticultural	3676	0.198446
235	Human resources officer	3675	0.198392
200	Fitness centre manager	3672	0.198230
36	Armed forces training and education officer	3672	0.198230
49	Bookseller	3672	0.198230
9	Administrator, education	3672	0.198230
53	Building control surveyor	3670	0.198122
495	Wellsite geologist	3669	0.198068
428	Soil scientist	3669	0.198068
79	Chief Technology Officer	3668	0.198014

403	Risk analyst	3665	0.197852
328	Paediatric nurse	3663	0.197744
75	Chief Financial Officer	3663	0.197744
8	Administrator, charities/voluntary organisations	3662	0.197690
330	Patent attorney	3662	0.197690
183	Equality and diversity officer	3662	0.197690
347	Police officer	3662	0.197690
295	Mental health nurse	3662	0.197690
494	Web designer	3662	0.197690
453	Teacher, early years/pre	3662	0.197690
280	Maintenance engineer	3661	0.197636
358	Product designer	3661	0.197636
449	Tax adviser	3658	0.197474
170	Engineer, petroleum	3658	0.197474
285	Market researcher	3658	0.197474
321	Operational researcher	3657	0.197420
3	Accountant, chartered public finance	3657	0.197420
323	Optician, dispensing	3657	0.197420
218	Health and safety adviser	3657	0.197420
119	Database administrator	3657	0.197420
368	Programmer, multimedia	3656	0.197366
221	Health service manager	3656	0.197366
491	Waste management officer	3656	0.197366
455	Teacher, secondary school	3655	0.197312
255	Investment analyst	3654	0.197258
97	Community arts worker	3653	0.197204
326	Osteopath	3651	0.197096
289	Media buyer	3651	0.197096
260	Land	3650	0.197042
468	Therapist, art	3650	0.197042
134	Doctor, general practice	3649	0.196988
475	Tourism officer	3649	0.196988
135	Doctor, hospital	3648	0.196934
473	Therapist, sports	3647	0.196880
14	Advice worker	3647	0.196880
437	Statistician	3645	0.196772
305	Music therapist	3643	0.196664
445	Surveyor, rural practice	3643	0.196664
252	Intelligence analyst	3641	0.196556
348	Politician's assistant	2944	0.158929
433	Sport and exercise psychologist	2941	0.158768
230	Horticultural therapist	2941	0.158768
96	Communications engineer	2941	0.158768
25	Animal nutritionist	2940	0.158714
217	Glass blower/designer	2940	0.158714
303	Museum/gallery conservator	2940	0.158714
374	Psychologist, sport and exercise	2936	0.158498
262	Landscape architect	2936	0.158498
31	Architect	2936	0.158498
377	Public affairs consultant	2935	0.158444

482	Transport planner	2935	0.158444
205	Furniture designer	2934	0.158390
311	Neurosurgeon	2934	0.158390
314	Nutritional therapist	2934	0.158390
210	General practice doctor	2934	0.158390
293	Medical secretary	2934	0.158390
66	Charity fundraiser	2933	0.158336
247	Information systems manager	2933	0.158336
76	Chief Marketing Officer	2933	0.158336
409	Scientific laboratory technician	2932	0.158282
191	Farm manager	2931	0.158228
419	Secretary/administrator	2931	0.158228
420	Seismic interpreter	2930	0.158174
256	Investment banker, corporate	2930	0.158174
370	Psychiatrist	2930	0.158174
304	Museum/gallery exhibitions officer	2930	0.158174
467	Theme park manager	2930	0.158174
87	Clinical biochemist	2929	0.158120
435	Sports development officer	2929	0.158120
85	Civil engineer, contracting	2929	0.158120
400	Retail buyer	2928	0.158066
141	Editor, film/video	2928	0.158066
225	Herpetologist	2927	0.158012
414	Scientist, physiological	2926	0.157958
38	Art therapist	2926	0.157958
222	Health visitor	2926	0.157958
242	Immigration officer	2925	0.157904
231	Horticulturist, commercial	2925	0.157904
336	Pharmacist, community	2924	0.157850
220	Health promotion specialist	2924	0.157850
214	Geologist, wellsite	2924	0.157850
90	Clinical research associate	2923	0.157796
174	Engineer, technical sales	2923	0.157796
394	Research scientist (life sciences)	2923	0.157796
40	Arts development officer	2923	0.157796
459	Telecommunications researcher	2922	0.157742
116	Dance movement psychotherapist	2922	0.157742
142	Editor, magazine features	2921	0.157688
375	Psychotherapist	2921	0.157688
63	Cartographer	2921	0.157688
163	Engineer, drilling	2920	0.157634
47	Biomedical engineer	2920	0.157634
245	Industrial/product designer	2920	0.157634
26	Animal technologist	2919	0.157580
99	Community education officer	2917	0.157472
17	Aid worker	2917	0.157472
111	Corporate investment banker	2917	0.157472
332	Pension scheme manager	2916	0.157418
488	Volunteer coordinator	2916	0.157418
21	Airline pilot	2916	0.157418

399	Retail banker	2916	0.157418
378	Public house manager	2916	0.157418
417	Scientist, research (physical sciences)	2914	0.157310
469	Therapist, drama	2913	0.157256
11	Advertising account executive	2213	0.119467
324	Optometrist	2212	0.119413
310	Network engineer	2212	0.119413
226	Higher education careers adviser	2210	0.119305
15	Aeronautical engineer	2208	0.119197
422	Set designer	2208	0.119197
150	Emergency planning/management officer	2207	0.119143
238	Hydrologist	2207	0.119143
405	Sales executive	2206	0.119089
100	Community pharmacist	2205	0.119035
68	Chartered accountant	2204	0.118981
34	Armed forces logistics/support/administrative ...	2203	0.118927
149	Embryologist, clinical	2203	0.118927
471	Therapist, music	2203	0.118927
465	Theatre director	2203	0.118927
338	Pharmacologist	2202	0.118873
95	Commissioning editor	2202	0.118873
192	Field seismologist	2201	0.118819
369	Psychiatric nurse	2201	0.118819
371	Psychologist, clinical	2200	0.118765
181	Environmental health practitioner	2199	0.118711
432	Special effects artist	2199	0.118711
27	Animator	2199	0.118711
390	Records manager	2199	0.118711
103	Conservation officer, historic buildings	2199	0.118711
125	Designer, interior/spatial	2198	0.118657
166	Engineer, maintenance	2198	0.118657
481	Training and development officer	2198	0.118657
5	Acupuncturist	2198	0.118657
186	Event organiser	2198	0.118657
132	Diagnostic radiographer	2198	0.118657
363	Production engineer	2198	0.118657
425	Social research officer, government	2197	0.118603
457	Teaching laboratory technician	2197	0.118603
291	Medical physicist	2196	0.118549
379	Public librarian	2196	0.118549
282	Management consultant	2195	0.118495
265	Learning mentor	2195	0.118495
33	Archivist	2195	0.118495
160	Engineer, civil (contracting)	2195	0.118495
154	Engineer, agricultural	2195	0.118495
276	Logistics and distribution manager	2195	0.118495
454	Teacher, primary school	2194	0.118441
340	Physiological scientist	2194	0.118441
43	Barista	2193	0.118387
209	Garment/textile technologist	2193	0.118387

275	Location manager	2193	0.118387
325	Orthoptist	2193	0.118387
241	Illustrator	2193	0.118387
318	Oceanographer	2192	0.118333
384	Quarry manager	2192	0.118333
344	Planning and development surveyor	2192	0.118333
444	Surveyor, mining	2191	0.118279
54	Building services engineer	2190	0.118225
184	Equities trader	2190	0.118225
404	Rural practice surveyor	2190	0.118225
161	Engineer, communications	2190	0.118225
300	Mudlogger	2189	0.118171
215	Geophysicist/field seismologist	2188	0.118117
108	Control and instrumentation engineer	2188	0.118117
466	Theatre manager	2187	0.118063
254	Interpreter	2186	0.118009
395	Research scientist (maths)	2185	0.117955
337	Pharmacist, hospital	1483	0.080059
294	Medical technical officer	1481	0.079951
158	Engineer, building services	1480	0.079897
258	Jewellery designer	1479	0.079843
139	Economist	1477	0.079735
84	Civil Service fast streamer	1476	0.079681
349	Presenter, broadcasting	1474	0.079573
127	Designer, multimedia	1473	0.079519
302	Museum education officer	1472	0.079465
284	Manufacturing systems engineer	1472	0.079465
46	Biochemist, clinical	1472	0.079465
80	Chief of Staff	1471	0.079411
290	Media planner	1471	0.079411
180	Environmental education officer	1471	0.079411
416	Scientist, research (medical)	1470	0.079357
396	Research scientist (medical)	1470	0.079357
233	Hospital pharmacist	1470	0.079357
37	Art gallery manager	1470	0.079357
153	Engineer, aeronautical	1470	0.079357
487	Visual merchandiser	1470	0.079357
401	Retail manager	1470	0.079357
451	Teacher, English as a foreign language	1469	0.079303
237	Hydrographic surveyor	1469	0.079303
129	Designer, textile	1468	0.079249
138	Ecologist	1468	0.079249
412	Scientist, clinical (histocompatibility and im...	1467	0.079195
434	Sports administrator	1467	0.079195
0	Academic librarian	1467	0.079195
101	Company secretary	1467	0.079195
143	Education administrator	1467	0.079195
277	Loss adjuster, chartered	1467	0.079195
18	Air broker	1467	0.079195
331	Pathologist	1467	0.079195

13	Advertising copywriter	1466	0.079141
361	Production assistant, radio	1466	0.079141
345	Plant breeder/geneticist	1466	0.079141
32	Architectural technologist	1466	0.079141
327	Outdoor activities/education manager	1466	0.079141
367	Programmer, applications	1465	0.079087
249	Insurance claims handler	1465	0.079087
157	Engineer, broadcasting (operations)	1465	0.079087
22	Ambulance person	1465	0.079087
69	Chartered legal executive (England and Wales)	1464	0.079033
58	Cabin crew	1463	0.078979
73	Chemist, analytical	1463	0.078979
359	Product manager	1463	0.078979
204	Furniture conservator/restorer	1462	0.078925
239	IT consultant	1462	0.078925
89	Clinical psychologist	1461	0.078871
313	Nurse, mental health	1461	0.078871
436	Stage manager	1460	0.078817
448	TEFL teacher	760	0.041028
263	Lawyer	757	0.040866
98	Community development worker	751	0.040542
2	Accountant, chartered certified	751	0.040542
229	Horticultural consultant	746	0.040272
211	Geneticist, molecular	745	0.040218
88	Clinical cytogeneticist	744	0.040164
279	Magazine journalist	744	0.040164
308	Nature conservation officer	743	0.040110
105	Conservator, museum/gallery	743	0.040110
460	Television camera operator	742	0.040056
496	Writer	741	0.040002
253	Interior and spatial designer	740	0.039948
430	Solicitor, Scotland	740	0.039948
118	Data scientist	740	0.039948
128	Designer, television/film set	740	0.039948
67	Charity officer	740	0.039948
298	Minerals surveyor	740	0.039948
193	Field trials officer	740	0.039948
83	Civil Service administrator	739	0.039894
474	Tour manager	739	0.039894
19	Air cabin crew	739	0.039894
398	Restaurant manager, fast food	739	0.039894
381	Public relations officer	738	0.039840
144	Education officer, community	738	0.039840
39	Artist	738	0.039840
173	Engineer, structural	738	0.039840
382	Purchasing manager	738	0.039840
236	Hydrogeologist	738	0.039840
362	Production assistant, television	738	0.039840
319	Oncologist	738	0.039840
317	Occupational therapist	738	0.039840

177	English as a foreign language teacher	737	0.039786
352	Primary school teacher	737	0.039786
452	Teacher, adult education	737	0.039786
64	Catering manager	737	0.039786
234	Hotel manager	737	0.039786
7	Administrator, arts	736	0.039732
357	Producer, television/film/video	736	0.039732
286	Marketing executive	736	0.039732
182	Environmental manager	736	0.039732
167	Engineer, manufacturing	736	0.039732
464	Textile designer	735	0.039678
104	Conservator, furniture	735	0.039678
441	Surveyor, hydrographic	735	0.039678
24	Analytical chemist	735	0.039678
50	Broadcast engineer	735	0.039678
365	Professor Emeritus	734	0.039624
109	Copy	734	0.039624
493	Water quality scientist	734	0.039624
41	Associate Professor	734	0.039624
57	Buyer, retail	734	0.039624
159	Engineer, civil (consulting)	734	0.039624
185	Estate manager/land agent	733	0.039570
257	Investment banker, operational	733	0.039570
296	Merchandiser, retail	732	0.039516
274	Local government officer	732	0.039516
168	Engineer, materials	731	0.039462
117	Dancer	19	0.001026
20	Air traffic controller	17	0.000918
61	Careers adviser	15	0.000810
407	Sales promotion account executive	14	0.000756
268	Legal secretary	12	0.000648
172	Engineer, site	12	0.000648
334	Personnel officer	12	0.000648
429	Solicitor	11	0.000594
427	Software engineer	11	0.000594
320	Operational investment banker	11	0.000594
228	Homeopath	11	0.000594
1	Accountant, chartered	11	0.000594
244	Industrial buyer	10	0.000540
202	Forest/woodland manager	9	0.000486
51	Broadcast journalist	9	0.000486
35	Armed forces technical officer	8	0.000432
246	Information officer	8	0.000432

```
df_fraud_job = df1[['job','is_fraud','trans_num']].groupby(['job','is_fraud']).count().reset_index()
df_fraud_job.columns = ['Job','is_fraud','count']

df_fraud_job =  df_fraud_job.merge(df_job[['Job','tran_count_by_job','percent']],how='inner',\
                                  left_on='Job',right_on='Job')


df_fraud_job['percent_grp'] = (df_fraud_job['count']/df_fraud_job['tran_count_by_job'])*100


job_plt_data = df_fraud_job.sort_values(by = ["tran_count_by_job"], ascending = False).head(20)
```

```
job_plt_data
```

	Job	is_fraud	count	tran_count_by_job	percent	percent_grp
378	Film/video editor	1	52	13898	0.750272	0.374155
377	Film/video editor	0	13846	13898	0.750272	99.625845
366	Exhibition designer	1	51	13167	0.710810	0.387332
365	Exhibition designer	0	13116	13167	0.710810	99.612668
861	Surveyor, land/geomatics	0	12386	12436	0.671347	99.597941
862	Surveyor, land/geomatics	1	50	12436	0.671347	0.402059
603	Naval architect	1	66	12434	0.671239	0.530803
602	Naval architect	0	12368	12434	0.671239	99.469197
559	Materials engineer	1	62	11711	0.632209	0.529417
558	Materials engineer	0	11649	11711	0.632209	99.470583
235	Designer, ceramics/pottery	0	11665	11688	0.630967	99.803217
236	Designer, ceramics/pottery	1	23	11688	0.630967	0.196783
347	Environmental consultant	0	10937	10974	0.592423	99.662839
348	Environmental consultant	1	37	10974	0.592423	0.337161
379	Financial adviser	0	10921	10963	0.591829	99.616893
380	Financial adviser	1	42	10963	0.591829	0.383107
871	Systems developer	0	10920	10962	0.591775	99.616858
872	Systems developer	1	42	10962	0.591775	0.383142
467	IT trainer	0	10906	10943	0.590749	99.661884
468	IT trainer	1	37	10943	0.590749	0.338116

```
job_plt_data['label'] = 'Not Fraud'
job_plt_data.loc[job_plt_data['is_fraud']==1,['label']]= 'Fraud'
job_plt_data
```

	Job	is_fraud	count	tran_count_by_job	percent	percent_grp	label
378	Film/video editor	1	52	13898	0.750272	0.374155	Fraud
377	Film/video editor	0	13846	13898	0.750272	99.625845	Not Fraud
366	Exhibition designer	1	51	13167	0.710810	0.387332	Fraud
365	Exhibition designer	0	13116	13167	0.710810	99.612668	Not Fraud
861	Surveyor, land/geomatics	0	12386	12436	0.671347	99.597941	Not Fraud
862	Surveyor, land/geomatics	1	50	12436	0.671347	0.402059	Fraud
603	Naval architect	1	66	12434	0.671239	0.530803	Fraud
602	Naval architect	0	12368	12434	0.671239	99.469197	Not Fraud
559	Materials engineer	1	62	11711	0.632209	0.529417	Fraud
558	Materials engineer	0	11649	11711	0.632209	99.470583	Not Fraud
235	Designer, ceramics/pottery	0	11665	11688	0.630967	99.803217	Not Fraud
236	Designer, ceramics/pottery	1	23	11688	0.630967	0.196783	Fraud
347	Environmental consultant	0	10937	10974	0.592423	99.662839	Not Fraud
348	Environmental consultant	1	37	10974	0.592423	0.337161	Fraud
379	Financial adviser	0	10921	10963	0.591829	99.616893	Not Fraud
380	Financial adviser	1	42	10963	0.591829	0.383107	Fraud
871	Systems developer	0	10920	10962	0.591775	99.616858	Not Fraud
872	Systems developer	1	42	10962	0.591775	0.383142	Fraud
467	IT trainer	0	10906	10943	0.590749	99.661884	Not Fraud
468	IT trainer	1	37	10943	0.590749	0.338116	Fraud

```
ne_grp = job_plt_data['Job'].unique()
print(ne_grp)

rm_grp = job_plt_data['label'].unique()
print(rm_grp)
```

```
['Film/video editor' 'Exhibition designer' 'Surveyor, land/geomatics'
 'Naval architect' 'Materials engineer' 'Designer, ceramics/pottery'
 'Environmental consultant' 'Financial adviser' 'Systems developer'
 'IT trainer']
['Fraud' 'Not Fraud']
```

```
fig = go.Figure(data=[
    go.Bar(name=rm_grp[0], x = ne_grp, y = job_plt_data[job_plt_data['label'] == rm_grp[0]]['percent_grp']),
    #go.Bar(name=rm_grp[1], x = ne_grp, y = job_plt_data[job_plt_data['label'] == rm_grp[1]]['percent_grp'])
])
# Change the bar mode
fig.update_layout(xaxis_title="Neighbourhood Group - Stacked with Room Type"\
                  ,yaxis_title="Percent of Listings")
fig.show()
```

Interactive Dashboard to Understand the Transaction Amt Distribution Based on City, Age and Gender

```
from ipywidgets import interact
fig = go.FigureWidget()
scatt = fig.add_histogram()

xs = df2
@interact(state = df2['state'].unique(), \
          gender = df2['gender'].unique(),\
          age = (14,100,5),
          is_fraud = [0,1])

def update(state = 'NC',gender = 'M', age = 14,is_fraud=1):
    with fig.batch_update():
        scatt = df2[(df2['state'] == state) \
                    & (df2['gender'] == gender) \
                    & (df2['age'] >= age) \
                    & (df2['is_fraud'] == is_fraud)]['amt']
        fig.data[0].x=scatt
fig.update_layout(xaxis_title="Number of transaction"
                  ,yaxis_title="Transaction amount")
```

```
FigureWidget({
    'data': [{'type': 'histogram',
              'uid': '5efb55d2-3a3a-440b-b44a-a99b36c65985',…
```

```
fig
```

```
FigureWidget({
```

Ploting Fraudulent Transactions in the Map

```
dftemp_fraud = df2[df2['is_fraud'] == 1]
fig = go.Figure()
fig.add_trace(go.Scattergeo(
        locationmode = 'USA-states',
        lon = dftemp_fraud['long'],
        lat = dftemp_fraud['lat'],
        #text = df_sub['text'],
        marker = dict(
            #size = df_sub['total_cases']/scale,
            color = dftemp_fraud['is_fraud'],
            line_color='rgb(40,40,40)',
            line_width=0.5,
            sizemode = 'area'
        ),
        name = 'test'))
fig.update_layout( title_text = 'test',
                 geo = dict(
                     landcolor = 'rgb(217,217,217)',),
                  mapbox_style="open-street-map"
                 )
```

```
import plotly.express as px

df2_fraud = df2[df2['is_fraud'] == 1]

fig = px.scatter_mapbox(df2_fraud, lat="lat", lon="long", hover_name="city",
                        zoom=3, height=500,
                        color="is_fraud",  color_discrete_sequence=px.colors.cyclical.IceFire)
fig.update_layout(mapbox_style="open-street-map")
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```

Train and Test Split

```
X_cols = ['amt','city_pop', 'trans_hour',
       'age', 'category_food_dining', 'category_gas_transport',
       'category_grocery_net', 'category_grocery_pos',
       'category_health_fitness', 'category_home', 'category_kids_pets',
       'category_misc_net', 'category_misc_pos', 'category_personal_care',
       'category_shopping_net', 'category_shopping_pos', 'category_travel',
       'gender_M','week_Monday','week_Tuesday', 'week_Wednesday','week_Thursday',
        'week_Saturday', 'week_Sunday','dist']

Y_cols = ['is_fraud']
```

```
df2_initial_train = df2.loc[(df2['trans_date_trans_time'] >= '2019-03-01 00:00:00') \
    & (df2['trans_date_trans_time'] <= '2020-06-30 23:23:00')][cols]
```

Oversampling

```
cnt_non_fraud_initial = df2_initial_train[df2_initial_train['is_fraud'] == 0]['amt'].count()
df2_class_fraud_initial = df2_initial_train[df2_initial_train['is_fraud'] == 1]
df2_class_nonfraud_initial = df2_initial_train[df2_initial_train['is_fraud'] == 0]


df2_class_fraud_oversample_initial = df2_class_fraud_initial.sample(cnt_non_fraud_initial, replace=True)
df2_oversampled_initial = pd.concat([df2_class_nonfraud_initial, df2_class_fraud_oversample_initial], axis=0)

print('Random over-sampling:')
print(df2_oversampled_initial['is_fraud'].value_counts())
```

```
Random over-sampling:
1    1217587
0    1217587
Name: is_fraud, dtype: int64
```

```
df2_oversampled_initial[X_cols].head()
```

	amt	city_pop	trans_hour	age	category_food_dining	category_gas_transport	category_grocery_net	category_gr
102391	59.560000	965	0	37.000000	0	1	0	
102392	81.210000	24536	0	35.000000	0	0	0	
102393	196.040000	4056	0	29.000000	0	0	0	
102394	86.050000	760	0	33.000000	0	0	0	
102395	60.930000	5512	0	39.000000	0	0	1	

Train and Test Split

```
X_train_initial = \
df2_oversampled_initial[X_cols]

y_train_initial = \
df2_oversampled_initial[Y_cols]
```

```
X_test_initial = df2[(df2['trans_date_trans_time'] >= '2019-05-01 00:00:00') \
    & (df2['trans_date_trans_time'] <= '2020-08-30 23:23:00')][X_cols]

y_test_initial = df2[(df2['trans_date_trans_time'] >= '2019-05-01 00:00:00') \
    & (df2['trans_date_trans_time'] <= '2020-08-30 23:23:00')][Y_cols]
```

Model Training

Logistic Regression

```
from sklearn.linear_model import LogisticRegression
```

```
logreg = LogisticRegression(random_state=42)
```

```
logreg.fit(X_train_initial, y_train_initial)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72: DataConversionWarning:

A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel()

LogisticRegression(random_state=42)
```

Evaluating the model

```
y_train_pred_initial = logreg.predict(X_train_initial)
y_test_pred_initial = logreg.predict(X_test_initial)
```

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

Logistic Model Train Results

```
print(confusion_matrix(y_train_initial, y_train_pred_initial))
print(classification_report(y_train_initial, y_train_pred_initial))
```

```
[[1128645   88942]
 [ 291306  926281]]
              precision    recall  f1-score   support

           0       0.79      0.93      0.86   1217587
           1       0.91      0.76      0.83   1217587

    accuracy                           0.84   2435174
   macro avg       0.85      0.84      0.84   2435174
weighted avg       0.85      0.84      0.84   2435174
```

Logistic Model Test Results

```
print(confusion_matrix(y_test_initial, y_test_pred_initial))
print(classification_report(y_test_initial, y_test_pred_initial))
```

```
[[1158289   90984]
 [   1547    4931]]
              precision    recall  f1-score   support

           0       1.00      0.93      0.96   1249273
           1       0.05      0.76      0.10      6478

    accuracy                           0.93   1255751
   macro avg       0.53      0.84      0.53   1255751
weighted avg       0.99      0.93      0.96   1255751
```

Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
dt_clf = DecisionTreeClassifier(criterion = 'gini', max_depth = 20, random_state=0)
dt_clf.fit(X_train_initial, y_train_initial)
```

```
DecisionTreeClassifier(max_depth=20, random_state=0)
```

Decision Tree - Model Evaluation

```
print("Train Results")
pred_train_initial = dt_clf.predict(X_train_initial)

print(confusion_matrix(y_train_initial, pred_train_initial))
print(classification_report(y_train_initial, pred_train_initial))
```

```
Train Results
[[1210223    7364]
 [      0 1217587]]
              precision    recall  f1-score   support

           0       1.00      0.99      1.00   1217587
           1       0.99      1.00      1.00   1217587

    accuracy                           1.00   2435174
   macro avg       1.00      1.00      1.00   2435174
weighted avg       1.00      1.00      1.00   2435174
```

```
print("Test Results")
pred_test_initial = dt_clf.predict(X_test_initial)

print(confusion_matrix(y_test_initial, pred_test_initial))
print(classification_report(y_test_initial, pred_test_initial))
```

```
Test Results
[[1241718    7555]
 [     90    6388]]
              precision    recall  f1-score   support

           0       1.00      0.99      1.00   1249273
           1       0.46      0.99      0.63      6478

    accuracy                           0.99   1255751
   macro avg       0.73      0.99      0.81   1255751
weighted avg       1.00      0.99      1.00   1255751
```

Random Forest classifier - Model Training

```
from sklearn.model_selection import GridSearchCV
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
rf_clf = RandomForestClassifier(random_state=345)

param_grid = {
    'n_estimators': [50],
    'max_depth' : [8,16,20]
}
```

Grid Search - For Random Forest

```
#best_rf = GridSearchCV(estimator=rf_clf, param_grid=param_grid)
#best_rf.fit(X_train_initial, y_train_initial)
```

```
#best_rf.best_estimator_
```

```
rf_clf = RandomForestClassifier(n_estimators = 50,max_depth = 20,
                                random_state=345, verbose = 1)
rf_clf.fit(X_train_initial, y_train_initial)
```

```
<ipython-input-101-bd745524c8d6>:3: DataConversionWarning:

A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:  5.4min finished
RandomForestClassifier(max_depth=20, n_estimators=50, random_state=345,
                       verbose=1)
```

Random Forest Classifier - Model Evaluation

```
print("Train Results")
pred_train_initial = rf_clf.predict(X_train_initial)

print(confusion_matrix(y_train_initial, pred_train_initial))
print(classification_report(y_train_initial, pred_train_initial))
```

```
Train Results
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:   22.0s finished
[[1215083    2504]
 [      0 1217587]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00   1217587
           1       1.00      1.00      1.00   1217587

    accuracy                           1.00   2435174
   macro avg       1.00      1.00      1.00   2435174
weighted avg       1.00      1.00      1.00   2435174
```

```
print("Test Results")
pred_test_initial = rf_clf.predict(X_test_initial)

print(confusion_matrix(y_test_initial, pred_test_initial))
print(classification_report(y_test_initial, pred_test_initial))
```

```
Test Results
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:   10.3s finished
[[1246648    2625]
 [    102    6376]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00   1249273
           1       0.71      0.98      0.82      6478

    accuracy                           1.00   1255751
   macro avg       0.85      0.99      0.91   1255751
weighted avg       1.00      1.00      1.00   1255751
```

XGBoost - Model Training

```
#pip install xgboost
from xgboost import XGBClassifier

# fit model no training data
xbt_model = XGBClassifier(n_estimators = 100, learning_rate = 0.1, max_depth = 3, random_state=345, verbose = 1)
xbt_model.fit(X_train_initial, y_train_initial)

xbt_model.fit(X_train_initial, y_train_initial)
```

```
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\utils\validation.py:73: DataConversionWarning: A column-vector y was pass
  return f(**kwargs)
[21:05:04] WARNING: C:\Users\Administrator\workspace\xgboost-win64_release_1.2.0\src\learner.cc:516:
Parameters: { verbose } might not be used.

  This may not be accurate due to some parameters are only used in language bindings but
  passed down to XGBoost core.  Or some parameters are not used but slip through this
  verification. Please open an issue if you find above cases.


[21:07:43] WARNING: C:\Users\Administrator\workspace\xgboost-win64_release_1.2.0\src\learner.cc:516:
Parameters: { verbose } might not be used.

  This may not be accurate due to some parameters are only used in language bindings but
  passed down to XGBoost core.  Or some parameters are not used but slip through this
  verification. Please open an issue if you find above cases.


XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
              learning_rate=0.1, max_delta_step=0, max_depth=3,
              min_child_weight=1, missing=nan, monotone_constraints='()',
              n_estimators=100, n_jobs=0, num_parallel_tree=1, random_state=345,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
              tree_method='exact', validate_parameters=1, verbose=1,
              verbosity=None)
```

XGBoost - Model Evaluation

Train Results

```
pred_train_initial = xbt_model.predict(X_train_initial)
```

```
from sklearn.metrics import classification_report,confusion_matrix
```

```
print(confusion_matrix(y_train_initial, pred_train_initial))
print(classification_report(y_train_initial, pred_train_initial))
```

```
[[1184254   33333]
 [  47257 1170330]]
              precision    recall  f1-score   support

           0       0.96      0.97      0.97   1217587
           1       0.97      0.96      0.97   1217587

    accuracy                           0.97   2435174
   macro avg       0.97      0.97      0.97   2435174
weighted avg       0.97      0.97      0.97   2435174
```

Test Results

```
pred_test_initial = xbt_model.predict(X_test_initial)
```

```
print(confusion_matrix(y_test_initial, pred_test_initial))
print(classification_report(y_test_initial, pred_test_initial))
```

```
[[1214977   34296]
 [    254    6224]]
              precision    recall  f1-score   support

           0       1.00      0.97      0.99   1249273
           1       0.15      0.96      0.26      6478

    accuracy                           0.97   1255751
   macro avg       0.58      0.97      0.63   1255751
weighted avg       1.00      0.97      0.98   1255751
```

Derive Historical Variables

```
df2.index = pd.to_datetime(df2['trans_date_trans_time'])
df2 = df2.rename_axis(index={'trans_date_trans_time': 'time_index'})
df2 = df2.sort_index()
df2.head()
```

trans_date_trans_time cc_num merchant category amt first last gender street c:

Orders in last 2 Months

2019-01-01 fraud_Rippin, Morav

```
df2['val_for_agg'] = 1
```

60 Day Transactions by Customers

Zieme Suite

```
df_hist_trans_60d = \
    df2 \
    .groupby(['cc_num'])['val_for_agg']\
    .rolling('60D')\
    .count()\
    .shift()\
    .reset_index()\
    .fillna(0)

df_hist_trans_60d.columns = ['cc_num','trans_date','hist_trans_60d']

df_hist_trans_60d['trans_date'] = df_hist_trans_60d['trans_date'].dt.date
```

00:03:06 Crist

```
df_hist_trans_60d = df_hist_trans_60d.groupby(['cc_num','trans_date'])['hist_trans_60d'].min().reset_index()

df_hist_trans_60d.head()
```

	cc_num	trans_date	hist_trans_60d
0	60416207185	2019-01-01	0.000000
1	60416207185	2019-01-02	1.000000
2	60416207185	2019-01-03	5.000000
3	60416207185	2019-01-04	7.000000
4	60416207185	2019-01-05	9.000000

24 Hours Orders by Customers

```
df_hist_orders_24h = \
    df2 \
    .groupby(['cc_num'])['val_for_agg']\
    .rolling('24H')\
    .count()\
    .shift()\
    .reset_index()\
    .fillna(0)

df_hist_orders_24h.columns = ['cc_num','trans_date_trans_time','hist_trans_24h']

df_hist_orders_24h.head()
```

	cc_num	trans_date_trans_time	hist_trans_24h
0	60416207185	2019-01-01 12:47:15	0.000000
1	60416207185	2019-01-02 08:44:57	1.000000
2	60416207185	2019-01-02 08:47:36	2.000000
3	60416207185	2019-01-02 12:38:14	3.000000
4	60416207185	2019-01-02 13:10:46	4.000000

24 Hours Fraud Orders by Customers

```
df_hist_fraud_trans_24h = \
    df2[df2['is_fraud']== 1]\
    .groupby(['cc_num'])['val_for_agg']\
    .rolling('24H')\
    .count()\
    .shift()\
    .reset_index()\
    .fillna(0)
```

```
df_hist_fraud_trans_24h.columns = ['cc_num','trans_date_trans_time','hist_fraud_trans_24h']
```

```
df_hist_fraud_trans_24h.head()
```

	cc_num	trans_date_trans_time	hist_fraud_trans_24h
0	60416207185	2019-03-01 01:32:53	0.000000
1	60416207185	2019-03-01 02:42:25	1.000000
2	60416207185	2019-03-01 23:06:58	2.000000
3	60416207185	2019-03-02 22:10:38	3.000000
4	60416207185	2019-03-02 22:10:59	2.000000

2 Hour Fraud Orders by Customers

```
df_hist_fraud_trans_2h = \
    df2[df2['is_fraud']== 1]\
    .groupby(['cc_num'])['val_for_agg']\
    .rolling('2H')\
    .count()\
    .shift()\
    .reset_index()\
    .fillna(0)

df_hist_fraud_trans_2h.columns = ['cc_num','trans_date_trans_time','hist_fraud_trans_2h']
```

```
df_hist_fraud_trans_2h.head()
```

	cc_num	trans_date_trans_time	hist_fraud_trans_2h
0	60416207185	2019-03-01 01:32:53	0.000000
1	60416207185	2019-03-01 02:42:25	1.000000
2	60416207185	2019-03-01 23:06:58	2.000000
3	60416207185	2019-03-02 22:10:38	1.000000
4	60416207185	2019-03-02 22:10:59	1.000000

60 Day Orders Amt Avg by Customers

```
df_hist_trans_amt_avg_60d = \
    df2 \
    .groupby(['cc_num'])['amt']\
    .rolling('60D')\
    .mean()\
    .shift(1)\
    .reset_index()\
    .fillna(0)

df_hist_trans_amt_avg_60d.columns = ['cc_num','trans_date','hist_trans_avg_amt_60d']
```

```
df_hist_trans_amt_avg_60d['trans_date'] = df_hist_trans_amt_avg_60d['trans_date'].dt.date
```

```
df_hist_trans_amt_avg_60d = df_hist_trans_amt_avg_60d.groupby(['cc_num','trans_date'])\
['hist_trans_avg_amt_60d'].min().reset_index()
```

```
df_hist_trans_amt_avg_60d.head(10)
```

	cc_num	trans_date	hist_trans_avg_amt_60d
0	60416207185	2019-01-01	0.000000
1	60416207185	2019-01-02	7.270000

Merge Historical Variables with Transactions by ['cc_num','trans_date_trans_time']

```
df2['trans_date'] = df2['trans_date_trans_time'].dt.date
```

```
df3 = df2.merge(df_hist_trans_60d,left_on = ['cc_num','trans_date'], \
         right_on = ['cc_num','trans_date'],how = 'left')
```

```
df3 = df3.merge(df_hist_orders_24h,left_on = ['cc_num','trans_date_trans_time'], \
         right_on = ['cc_num','trans_date_trans_time'],how = 'left')

df3 = df3.merge(df_hist_fraud_trans_24h,left_on = ['cc_num','trans_date_trans_time'], \
         right_on = ['cc_num','trans_date_trans_time'],how = 'left')

df3 = df3.merge(df_hist_fraud_trans_2h,left_on = ['cc_num','trans_date_trans_time'], \
         right_on = ['cc_num','trans_date_trans_time'],how = 'left')

df3 = df3.merge(df_hist_trans_amt_avg_60d,left_on = ['cc_num','trans_date'], \
         right_on = ['cc_num','trans_date'],how = 'left')
```

```
df3[['hist_trans_60d','hist_trans_24h','hist_fraud_trans_24h','hist_fraud_trans_2h','hist_trans_avg_amt_60d']]= \
df3[['hist_trans_60d','hist_trans_24h','hist_fraud_trans_24h','hist_fraud_trans_2h','hist_trans_avg_amt_60d']].fillna(0)
```

```
df3.head()
```

	trans_date_trans_time	cc_num	merchant	category	amt	first	last	gender	street	city	state
0	2019-01-01 00:00:18	2703186189652095	fraud_Rippin, Kub and Mann	misc_net	4.970000	Jennifer	Banks	F	561 Perry Cove	Moravian Falls	NC
1	2019-01-01 00:00:44	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.230000	Stephanie	Gill	F	43039 Riley Greens Suite 393	Orient	WA
2	2019-01-01 00:00:51	38859492057661	fraud_Lind-Buckridge	entertainment	220.110000	Edward	Sanchez	M	594 White Dale Suite 530	Malad City	ID
3	2019-01-01 00:01:16	3534093764340240	fraud_Kutch, Hermiston and Farrell	gas_transport	45.000000	Jeremy	White	M	9443 Cynthia Court Apt. 038	Boulder	MT
4	2019-01-01 00:03:06	375534208663984	fraud_Keeling-Crist	misc_pos	41.960000	Tyler	Garcia	M	408 Bradley Rest	Doe Hill	VA

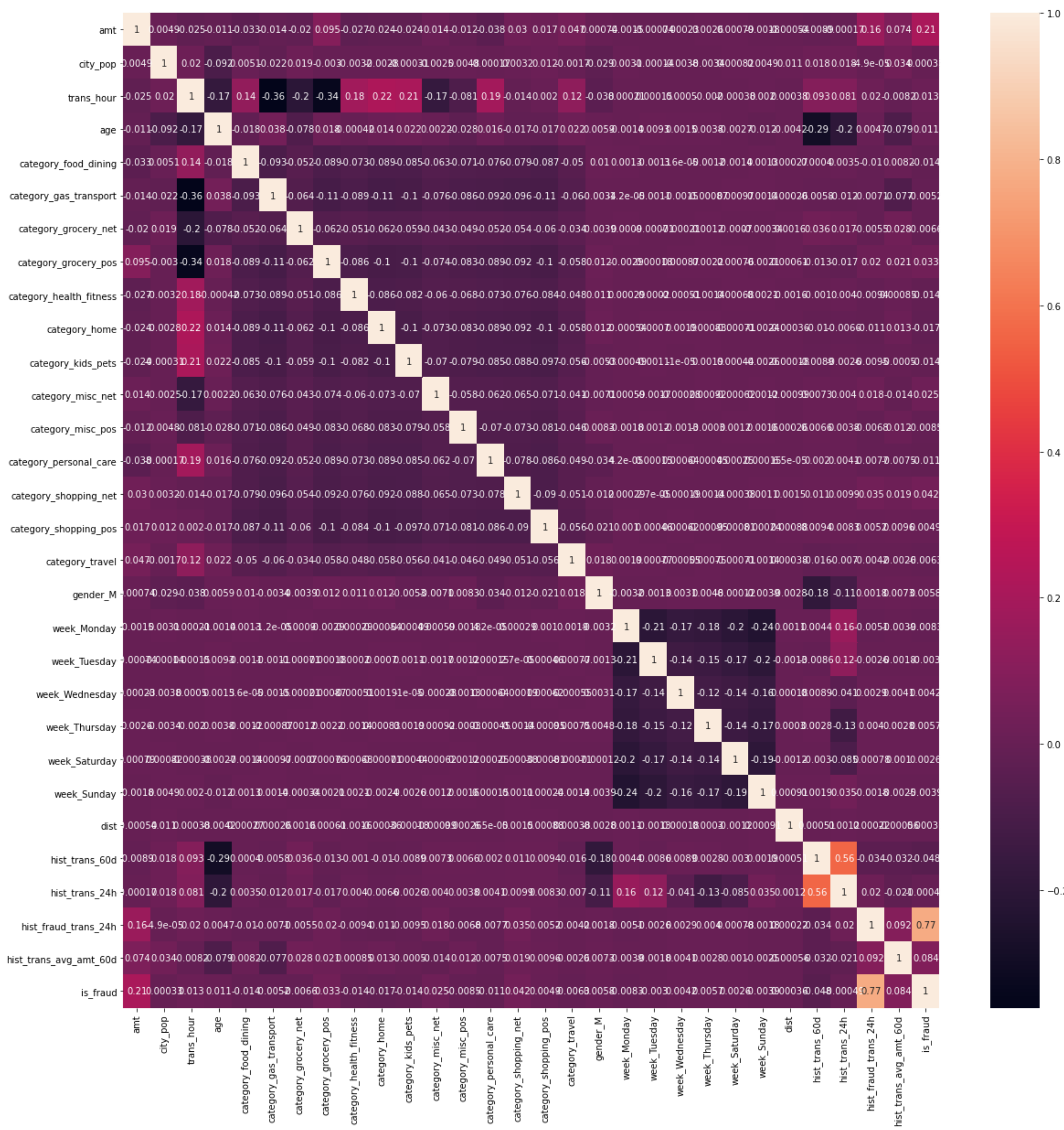
```
df3['hist_fraud_trans_24h'] = df3['hist_fraud_trans_24h'] - df3['hist_fraud_trans_2h']
```

```
cols = ['amt','city_pop', 'trans_hour',
       'age', 'category_food_dining', 'category_gas_transport',
       'category_grocery_net', 'category_grocery_pos',
       'category_health_fitness', 'category_home', 'category_kids_pets',
       'category_misc_net', 'category_misc_pos', 'category_personal_care',
       'category_shopping_net', 'category_shopping_pos', 'category_travel',
       'gender_M','week_Monday','week_Tuesday', 'week_Wednesday','week_Thursday',
        'week_Saturday', 'week_Sunday','dist','hist_trans_60d','hist_trans_24h',
       'hist_fraud_trans_24h','hist_trans_avg_amt_60d','is_fraud']
```

```
corr = df3[cols].corr()
```

```
import seaborn as sn

fig, ax = plt.subplots(figsize=(20,20))
```

```
sn.heatmap(corr, annot=True)
plt.show()
```



Train and Test Split

```
X_cols = ['amt','city_pop', 'trans_hour',
       'age', 'category_food_dining', 'category_gas_transport',
       'category_grocery_net', 'category_grocery_pos',
       'category_health_fitness', 'category_home', 'category_kids_pets',
       'category_misc_net', 'category_misc_pos', 'category_personal_care',
       'category_shopping_net', 'category_shopping_pos', 'category_travel',
       'gender_M','week_Monday','week_Tuesday', 'week_Wednesday','week_Thursday',
        'week_Saturday', 'week_Sunday','dist','hist_trans_60d','hist_trans_24h',
       'hist_fraud_trans_24h','hist_trans_avg_amt_60d'] #,
```

```
Y_cols = ['is_fraud']
```

```
df3_train = df3.loc[(df3['trans_date_trans_time'] >= '2019-03-01 00:00:00') \
    & (df3['trans_date_trans_time'] <= '2020-06-30 23:23:00')][cols]
```

Oversampling

```
cnt_non_fraud = df3_train[df3_train['is_fraud'] == 0]['amt'].count()
df3_class_fraud = df3_train[df3_train['is_fraud'] == 1]
df3_class_nonfraud = df3_train[df3_train['is_fraud'] == 0]
```

```
df3_class_fraud_oversample = df3_class_fraud.sample(cnt_non_fraud, replace=True)
df3_oversampled = pd.concat([df3_class_nonfraud, df3_class_fraud_oversample], axis=0)

print('Random over-sampling:')
print(df3_oversampled['is_fraud'].value_counts())
```

```
Random over-sampling:
1    1217627
0    1217627
Name: is_fraud, dtype: int64
```

```
df3_oversampled[X_cols].head()
```

	amt	city_pop	trans_hour	age	category_food_dining	category_gas_transport	category_grocery_net	category_gr
102393	59.560000	965	0	37.000000	0	1	0	
102394	81.210000	24536	0	35.000000	0	0	0	
102395	196.040000	4056	0	29.000000	0	0	0	
102396	86.050000	760	0	33.000000	0	0	0	
102397	60.930000	5512	0	39.000000	0	0	1	

Train and Test Split

```
X_train = \
df3_oversampled[X_cols]

y_train = \
df3_oversampled[Y_cols]
```

```
X_test = df3[(df3['trans_date_trans_time'] >= '2019-05-01 00:00:00') \
    & (df3['trans_date_trans_time'] <= '2020-08-30 23:23:00')][X_cols]

y_test = df3[(df3['trans_date_trans_time'] >= '2019-05-01 00:00:00') \
    & (df3['trans_date_trans_time'] <= '2020-08-30 23:23:00')][Y_cols]
```

Model Training

Logistic Regression

```
from sklearn.linear_model import LogisticRegression
```

```
logreg = LogisticRegression(random_state=42)
```

```
logreg.fit(X_train, y_train)
```

```
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\utils\validation.py:73: DataConversionWarning: A column-vector y was pass
  return f(**kwargs)
LogisticRegression(random_state=42)
```

Evaluating the model

```
y_train_pred = logreg.predict(X_train)
y_test_pred = logreg.predict(X_test)
```

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

Logistic Model Train Results

```
print(confusion_matrix(y_train, y_train_pred))
print(classification_report(y_train, y_train_pred))
```

```
[[1137891   79736]
 [ 291795  925832]]
              precision    recall  f1-score   support

           0       0.80      0.93      0.86   1217627
           1       0.92      0.76      0.83   1217627

    accuracy                           0.85   2435254
   macro avg       0.86      0.85      0.85   2435254
weighted avg       0.86      0.85      0.85   2435254
```

Logistic Model Test Results

```
print(confusion_matrix(y_test, y_test_pred))
print(classification_report(y_test, y_test_pred))
```

```
[[1172880   76441]
 [   1595    4883]]
              precision    recall  f1-score   support

           0       1.00      0.94      0.97   1249321
           1       0.06      0.75      0.11      6478

    accuracy                           0.94   1255799
   macro avg       0.53      0.85      0.54   1255799
weighted avg       0.99      0.94      0.96   1255799
```

Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
dt_clf = DecisionTreeClassifier(criterion = 'gini', max_depth = 20, random_state=0)
dt_clf.fit(X_train, y_train)
```

```
DecisionTreeClassifier(max_depth=20, random_state=0)
```

Decision Tree - Model Evaluation

```
print("Train Results")
pred_train = dt_clf.predict(X_train)

print(confusion_matrix(y_train, pred_train))
print(classification_report(y_train, pred_train))
```

```
Train Results
[[1216105    1522]
 [      0 1217627]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00   1217627
           1       1.00      1.00      1.00   1217627

    accuracy                           1.00   2435254
   macro avg       1.00      1.00      1.00   2435254
weighted avg       1.00      1.00      1.00   2435254
```

```
print("Test Results")
pred_test = dt_clf.predict(X_test)

print(confusion_matrix(y_test, pred_test))
print(classification_report(y_test, pred_test))
```

```
Test Results
[[1247825    1496]
```

```
 [     29    6449]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00   1249321
           1       0.81      1.00      0.89      6478

    accuracy                           1.00   1255799
   macro avg       0.91      1.00      0.95   1255799
weighted avg       1.00      1.00      1.00   1255799
```

Random Forest classifier - Model Training

```
from sklearn.model_selection import GridSearchCV
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
rf_clf = RandomForestClassifier(random_state=345)

param_grid = {
    'n_estimators': [50],
    'max_depth' : [8,16,20]
}
```

Grid Search - For Random Forest

```
best_rf = GridSearchCV(estimator=rf_clf, param_grid=param_grid)
best_rf.fit(X_train, y_train)
```

```
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:531: DataConversionWarning: A column-vecto
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:531: DataConversionWarning: A column-vecto
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:531: DataConversionWarning: A column-vecto
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:531: DataConversionWarning: A column-vecto
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:531: DataConversionWarning: A column-vecto
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:531: DataConversionWarning: A column-vecto
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:531: DataConversionWarning: A column-vecto
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:531: DataConversionWarning: A column-vecto
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:531: DataConversionWarning: A column-vecto
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:531: DataConversionWarning: A column-vecto
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:531: DataConversionWarning: A column-vecto
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:531: DataConversionWarning: A column-vecto
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:531: DataConversionWarning: A column-vecto
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:531: DataConversionWarning: A column-vecto
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:531: DataConversionWarning: A column-vecto
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:765: DataConversionWarning: A column-vector y w
  self.best_estimator_.fit(X, y, **fit_params)
GridSearchCV(estimator=RandomForestClassifier(random_state=345),
             param_grid={'max_depth': [8, 16, 20], 'n_estimators': [50]})
```

```
best_rf.best_estimator_
```

```
RandomForestClassifier(max_depth=20, n_estimators=50, random_state=345)
```

```
rf_clf = RandomForestClassifier(n_estimators = 50,max_depth = 20,
                                random_state=345, verbose = 1)
rf_clf.fit(X_train, y_train)
```

```
<ipython-input-356-8fa160f0888f>:3: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change
  rf_clf.fit(X_train, y_train)
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed: 22.1min finished
RandomForestClassifier(max_depth=20, n_estimators=50, random_state=345,
                       verbose=1)
```

Random Forest Classifier - Model Evaluation

```
print("Train Results")
pred_train = rf_clf.predict(X_train)

print(confusion_matrix(y_train, pred_train))
print(classification_report(y_train, pred_train))
```

```
Train Results
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:   36.3s finished
[[1217541      86]
 [      0 1217627]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00   1217627
           1       1.00      1.00      1.00   1217627

    accuracy                           1.00   2435254
   macro avg       1.00      1.00      1.00   2435254
weighted avg       1.00      1.00      1.00   2435254
```

```
print("Test Results")
pred_test = rf_clf.predict(X_test)

print(confusion_matrix(y_test, pred_test))
print(classification_report(y_test, pred_test))
```

```
Test Results
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:   20.6s finished
[[1249224      97]
 [     46    6432]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00   1249321
           1       0.99      0.99      0.99      6478

    accuracy                           1.00   1255799
   macro avg       0.99      1.00      0.99   1255799
weighted avg       1.00      1.00      1.00   1255799
```

XGBoost - Model Training

```
from xgboost import XGBClassifier

# fit model no training data
xbt_model = XGBClassifier(n_estimators = 100, learning_rate = 0.1, max_depth = 3, random_state=345, verbose = 1)
xbt_model.fit(X_train, y_train)

xbt_model.fit(X_train, y_train)
```

```
C:\Users\nimmy.samson\Anaconda3\lib\site-packages\sklearn\utils\validation.py:73: DataConversionWarning: A column-vector y was pass
  return f(**kwargs)
[10:59:24] WARNING: C:\Users\Administrator\workspace\xgboost-win64_release_1.2.0\src\learner.cc:516:
Parameters: { verbose } might not be used.

  This may not be accurate due to some parameters are only used in language bindings but
  passed down to XGBoost core.  Or some parameters are not used but slip through this
  verification. Please open an issue if you find above cases.


[11:05:01] WARNING: C:\Users\Administrator\workspace\xgboost-win64_release_1.2.0\src\learner.cc:516:
Parameters: { verbose } might not be used.

  This may not be accurate due to some parameters are only used in language bindings but
  passed down to XGBoost core.  Or some parameters are not used but slip through this
  verification. Please open an issue if you find above cases.


XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
              learning_rate=0.1, max_delta_step=0, max_depth=3,
              min_child_weight=1, missing=nan, monotone_constraints='()',
              n_estimators=100, n_jobs=0, num_parallel_tree=1, random_state=345,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
```

```
              tree_method='exact', validate_parameters=1, verbose=1,
              verbosity=None)
```

XGBoost Trees - Model Evaluation

Train Results

```
pred_train = xbt_model.predict(X_train)
```

```
from sklearn.metrics import classification_report,confusion_matrix
```

```
print(confusion_matrix(y_train, pred_train))
print(classification_report(y_train, pred_train))
```

```
[[1209268    8359]
 [  25233 1192394]]
              precision    recall  f1-score   support

           0       0.98      0.99      0.99   1217627
           1       0.99      0.98      0.99   1217627

    accuracy                           0.99   2435254
   macro avg       0.99      0.99      0.99   2435254
weighted avg       0.99      0.99      0.99   2435254
```

Test Results

```
pred_test = xbt_model.predict(X_test)
```

```
print(confusion_matrix(y_test, pred_test))
print(classification_report(y_test, pred_test))
```

```
[[1240829    8492]
 [    145    6333]]
              precision    recall  f1-score   support

           0       1.00      0.99      1.00   1249321
           1       0.43      0.98      0.59      6478

    accuracy                           0.99   1255799
   macro avg       0.71      0.99      0.80   1255799
weighted avg       1.00      0.99      0.99   1255799
```