credit-card-fraud-detection

November 5, 2023

1 Credit Card Fraud Detection using Machine Learning

1.1 Based on the factors provided, we are to determine if the credit card issued is fraud or not

2 Importing Required Libraries

[73]:
```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn import tree
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import␣
 ↪accuracy_score,confusion_matrix,classification_report
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

3 Importing The Dataset

[2]:
```
train_data = pd.read_csv('fraudTrain.csv')
train_data.head()
```

[2]:

	Unnamed: 0	trans_date_trans_time	cc_num	merchant	category	amt	first
0	0	2019-01-01 00:00:18	2703186189652095	fraud_Rippin, Kub and Mann	misc_net	4.97	Jennifer
1	1	2019-01-01 00:00:44	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.23	Stephanie
2	2	2019-01-01 00:00:51	38859492057661				
3	3	2019-01-01 00:01:16	3534093764340240				
4	4	2019-01-01 00:03:06	375534208663984				

```
2                fraud_Lind-Buckridge  entertainment  220.11     Edward
3  fraud_Kutch, Hermiston and Farrell  gas_transport   45.00     Jeremy
4                 fraud_Keeling-Crist       misc_pos   41.96      Tyler

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

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

                          trans_num   unix_time  merch_lat  merch_long  \
0  0b242abb623afc578575680df30655b9  1325376018  36.011293  -82.048315
1  1f76529f8574734946361c461b024d99  1325376044  49.159047 -118.186462
2  a1a22d70485983eac12b5b88dad1cf95  1325376051  43.150704 -112.154481
3  6b849c168bdad6f867558c3793159a81  1325376076  47.034331 -112.561071
4  a41d7549acf90789359a9aa5346dcb46  1325376186  38.674999  -78.632459

   is_fraud
0         0
1         0
2         0
3         0
4         0

[5 rows x 23 columns]
```

[3]:
```
test_data = pd.read_csv('fraudTest.csv')
test_data.head()
```

[3]:
```
   Unnamed: 0 trans_date_trans_time            cc_num  \
0           0   2020-06-21 12:14:25  2291163933867244
1           1   2020-06-21 12:14:33  3573030041201292
2           2   2020-06-21 12:14:53  3598215285024754
3           3   2020-06-21 12:15:15  3591919803438423
4           4   2020-06-21 12:15:17  3526826139003047

                              merchant        category    amt   first  \
0               fraud_Kirlin and Sons   personal_care   2.86    Jeff
1                fraud_Sporer-Keebler   personal_care  29.84  Joanne
```

```
2  fraud_Swaniawski, Nitzsche and Welch  health_fitness  41.28  Ashley
3                    fraud_Haley Group        misc_pos  60.05   Brian
4                fraud_Johnston-Casper          travel   3.19  Nathan

      last gender                        street  …      lat      long  \
0   Elliott      M            351 Darlene Green  …  33.9659  -80.9355
1  Williams      F             3638 Marsh Union  …  40.3207 -110.4360
2     Lopez      F         9333 Valentine Point  …  40.6729  -73.5365
3  Williams      M  32941 Krystal Mill Apt. 552  …  28.5697  -80.8191
4    Massey      M     5783 Evan Roads Apt. 465  …  44.2529  -85.0170

   city_pop                     job         dob  \
0    333497     Mechanical engineer  1968-03-19
1       302  Sales professional, IT  1990-01-17
2     34496       Librarian, public  1970-10-21
3     54767            Set designer  1987-07-25
4      1126      Furniture designer  1955-07-06

                          trans_num   unix_time  merch_lat  merch_long  \
0  2da90c7d74bd46a0caf3777415b3ebd3  1371816865  33.986391  -81.200714
1  324cc204407e99f51b0d6ca0055005e7  1371816873  39.450498 -109.960431
2  c81755dbbbea9d5c77f094348a7579be  1371816893  40.495810  -74.196111
3  2159175b9efe66dc301f149d3d5abf8c  1371816915  28.812398  -80.883061
4  57ff021bd3f328f8738bb535c302a31b  1371816917  44.959148  -85.884734

   is_fraud
0         0
1         0
2         0
3         0
4         0

[5 rows x 23 columns]
```

3.1 Creating a copy of the datasets (so that we can always refer back to the original data)

[4]:
```
train_data_ = train_data
test_data_ = test_data
```

[5]:
```
pd.set_option('display.max_columns', None)
train_data_.head()
```

[5]:
```
   Unnamed: 0 trans_date_trans_time            cc_num  \
0           0   2019-01-01 00:00:18  2703186189652095
1           1   2019-01-01 00:00:44      630423337322
```

```
2           2   2019-01-01 00:00:51    38859492057661
3           3   2019-01-01 00:01:16  3534093764340240
4           4   2019-01-01 00:03:06   375534208663984

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

      last gender                        street            city state    zip  \
0    Banks      F                561 Perry Cove  Moravian Falls    NC  28654
1     Gill      F  43039 Riley Greens Suite 393          Orient    WA  99160
2  Sanchez      M      594 White Dale Suite 530      Malad City    ID  83252
3    White      M   9443 Cynthia Court Apt. 038         Boulder    MT  59632
4   Garcia      M              408 Bradley Rest        Doe Hill    VA  24433

       lat      long  city_pop                                job         dob  \
0  36.0788  -81.1781      3495          Psychologist, counselling  1988-03-09
1  48.8878 -118.2105       149  Special educational needs teacher  1978-06-21
2  42.1808 -112.2620      4154        Nature conservation officer  1962-01-19
3  46.2306 -112.1138      1939                    Patent attorney  1967-01-12
4  38.4207  -79.4629        99     Dance movement psychotherapist  1986-03-28

                          trans_num   unix_time  merch_lat  merch_long  \
0  0b242abb623afc578575680df30655b9  1325376018  36.011293  -82.048315
1  1f76529f8574734946361c461b024d99  1325376044  49.159047 -118.186462
2  a1a22d70485983eac12b5b88dad1cf95  1325376051  43.150704 -112.154481
3  6b849c168bdad6f867558c3793159a81  1325376076  47.034331 -112.561071
4  a41d7549acf90789359a9aa5346dcb46  1325376186  38.674999  -78.632459

   is_fraud
0         0
1         0
2         0
3         0
4         0
```

[6]:
```
test_data_.head()
```

[6]:
```
   Unnamed: 0 trans_date_trans_time            cc_num  \
0           0   2020-06-21 12:14:25  2291163933867244
1           1   2020-06-21 12:14:33  3573030041201292
2           2   2020-06-21 12:14:53  3598215285024754
3           3   2020-06-21 12:15:15  3591919803438423
4           4   2020-06-21 12:15:17  3526826139003047
```

```
                                merchant        category    amt   first  \
0                 fraud_Kirlin and Sons   personal_care   2.86    Jeff
1                  fraud_Sporer-Keebler   personal_care  29.84  Joanne
2  fraud_Swaniawski, Nitzsche and Welch  health_fitness  41.28  Ashley
3                      fraud_Haley Group        misc_pos  60.05   Brian
4                  fraud_Johnston-Casper          travel   3.19  Nathan

      last gender                        street        city state    zip  \
0   Elliott      M             351 Darlene Green    Columbia    SC  29209
1  Williams      F              3638 Marsh Union     Altonah    UT  84002
2     Lopez      F          9333 Valentine Point    Bellmore    NY  11710
3  Williams      M  32941 Krystal Mill Apt. 552  Titusville    FL  32780
4    Massey      M      5783 Evan Roads Apt. 465    Falmouth    MI  49632

       lat      long  city_pop                     job         dob  \
0  33.9659  -80.9355    333497     Mechanical engineer  1968-03-19
1  40.3207 -110.4360       302  Sales professional, IT  1990-01-17
2  40.6729  -73.5365     34496       Librarian, public  1970-10-21
3  28.5697  -80.8191     54767            Set designer  1987-07-25
4  44.2529  -85.0170      1126      Furniture designer  1955-07-06

                          trans_num   unix_time  merch_lat  merch_long  \
0  2da90c7d74bd46a0caf3777415b3ebd3  1371816865  33.986391  -81.200714
1  324cc204407e99f51b0d6ca0055005e7  1371816873  39.450498 -109.960431
2  c81755dbbbea9d5c77f094348a7579be  1371816893  40.495810  -74.196111
3  2159175b9efe66dc301f149d3d5abf8c  1371816915  28.812398  -80.883061
4  57ff021bd3f328f8738bb535c302a31b  1371816917  44.959148  -85.884734

   is_fraud
0         0
1         0
2         0
3         0
4         0
```

3.2 Exploring the data and performing some analysis (EDA)

3.2.1 Analysing the training data

[7]:
```
train_data_.shape
```

[7]: (1296675, 23)

[8]:
```
test_data_.shape
```

[8]: (555719, 23)

we can clearly see, the dataset is quite big - it's beneficial as it'll make the model more accurate

[9]:
```
train_data_.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1296675 entries, 0 to 1296674
Data columns (total 23 columns):
 #   Column                 Non-Null Count    Dtype
---  ------                 --------------    -----
 0   Unnamed: 0             1296675 non-null  int64
 1   trans_date_trans_time  1296675 non-null  object
 2   cc_num                 1296675 non-null  int64
 3   merchant               1296675 non-null  object
 4   category               1296675 non-null  object
 5   amt                    1296675 non-null  float64
 6   first                  1296675 non-null  object
 7   last                   1296675 non-null  object
 8   gender                 1296675 non-null  object
 9   street                 1296675 non-null  object
 10  city                   1296675 non-null  object
 11  state                  1296675 non-null  object
 12  zip                    1296675 non-null  int64
 13  lat                    1296675 non-null  float64
 14  long                   1296675 non-null  float64
 15  city_pop               1296675 non-null  int64
 16  job                    1296675 non-null  object
 17  dob                    1296675 non-null  object
 18  trans_num              1296675 non-null  object
 19  unix_time              1296675 non-null  int64
 20  merch_lat              1296675 non-null  float64
 21  merch_long             1296675 non-null  float64
 22  is_fraud               1296675 non-null  int64
dtypes: float64(5), int64(6), object(12)
memory usage: 227.5+ MB
```

As we can see a lot of columns with categorical data, let us first analyse and transform them so that we can use them to train our model

[10]:
```
column_names = train_data_.columns

# now, we will bifurcate the columns with categorical data
categorical_columns = [var for var in column_names if train_data_[var].
 ↪dtype=='O'] # using list comprehension

print("The columns with categorical data are: {}".format(categorical_columns))
```

```
The columns with categorical data are: ['trans_date_trans_time', 'merchant',
'category', 'first', 'last', 'gender', 'street', 'city', 'state', 'job', 'dob',
```

```
'trans_num']
```

[11]:
```
train_data_[categorical_columns].head()
```

[11]:

	trans_date_trans_time	merchant	category	\
0	2019-01-01 00:00:18	fraud_Rippin, Kub and Mann	misc_net	
1	2019-01-01 00:00:44	fraud_Heller, Gutmann and Zieme	grocery_pos	
2	2019-01-01 00:00:51	fraud_Lind-Buckridge	entertainment	
3	2019-01-01 00:01:16	fraud_Kutch, Hermiston and Farrell	gas_transport	
4	2019-01-01 00:03:06	fraud_Keeling-Crist	misc_pos	

	first	last	gender	street	city	\
0	Jennifer	Banks	F	561 Perry Cove	Moravian Falls	
1	Stephanie	Gill	F	43039 Riley Greens Suite 393	Orient	
2	Edward	Sanchez	M	594 White Dale Suite 530	Malad City	
3	Jeremy	White	M	9443 Cynthia Court Apt. 038	Boulder	
4	Tyler	Garcia	M	408 Bradley Rest	Doe Hill	

	state	job	dob	\
0	NC	Psychologist, counselling	1988-03-09	
1	WA	Special educational needs teacher	1978-06-21	
2	ID	Nature conservation officer	1962-01-19	
3	MT	Patent attorney	1967-01-12	
4	VA	Dance movement psychotherapist	1986-03-28	

	trans_num
0	0b242abb623afc578575680df30655b9
1	1f76529f8574734946361c461b024d99
2	a1a22d70485983eac12b5b88dad1cf95
3	6b849c168bdad6f867558c3793159a81
4	a41d7549acf90789359a9aa5346dcb46

3.3 Observations on the categorical data

3.3.1 trans_date_trans_time - column giving information on the time

3.3.2 merchant - name of the merchant

3.3.3 category - category for which the credit card is being used

3.3.4 first, last - denotes the name of the customer

3.3.5 street, city, state - denotes the address of the customer

3.3.6 dob - date of birth of the customer

3.3.7 Now, I will analyze the categorical columns one by one

3.3.8 Column : trans_num

[12]:
```
train_data_['trans_num'].unique()
```

[12]:
```
array(['0b242abb623afc578575680df30655b9',
       '1f76529f8574734946361c461b024d99',
       'a1a22d70485983eac12b5b88dad1cf95', …,
       '483f52fe67fabef353d552c1e662974c',
       'd667cdcbadaaed3da3f4020e83591c83',
       '8f7c8e4ab7f25875d753b422917c98c9'], dtype=object)
```

[13]:
```
# We can clearly see the 'trans_num' doesn't have a proper pattern which will␣
↪help as a determination factor, so we drop it
train_data_ = train_data_.drop('trans_num', axis=1)
```

3.3.9 Column : dob

The complete date of birth is not important, so let's just extract the age of the customer from it

[14]:
```
train_data_['dob'].unique()
```

[14]:
```
array(['1988-03-09', '1978-06-21', '1962-01-19', '1967-01-12',
       '1986-03-28', '1961-06-19', '1993-08-16', '1947-08-21',
       '1941-03-07', '1974-03-28', '1990-07-13', '1966-02-14',
       '1989-02-28', '1945-12-21', '1967-08-30', '1965-06-30',
       '1952-07-06', '1938-03-15', '1946-02-02', '1980-12-21',
       '1980-11-22', '1961-02-14', '1974-07-19', '1965-07-26',
       '1946-01-02', '1962-08-13', '1971-11-05', '1967-08-02',
       '1966-12-03', '1945-03-15', '1961-05-19', '1964-12-30',
       '1964-04-22', '1977-02-22', '1970-07-20', '1984-06-04',
       '1970-10-21', '1984-12-24', '1998-10-01', '1988-04-27',
       '1987-07-18', '1971-10-14', '1987-04-23', '1942-01-06',
       '1971-01-28', '1972-07-25', '1984-09-01', '1960-01-06',
       '1986-11-06', '1954-01-05', '1970-09-27', '1994-02-09',
       '1942-11-24', '1994-11-05', '1993-10-25', '1976-10-18',
```

```
'1981-02-15', '1974-03-13', '1926-07-12', '1966-12-21',
'1936-03-28', '1997-08-22', '1972-05-04', '1955-06-12',
'1990-08-13', '1967-02-04', '1974-06-21', '1962-11-11',
'1983-07-25', '1979-01-02', '2000-06-13', '1957-03-28',
'1955-01-05', '1976-02-26', '1982-01-07', '1935-09-08',
'1975-04-30', '1977-04-28', '1954-05-25', '1946-04-03',
'1975-07-31', '1998-03-19', '1974-12-23', '1995-07-12',
'1989-11-24', '1983-08-25', '1984-06-03', '1935-08-15',
'1995-04-19', '1976-09-08', '1946-08-24', '1971-08-20',
'1957-03-06', '1970-09-11', '1977-01-04', '1986-06-11',
'1989-04-08', '1986-06-20', '1980-12-16', '1978-07-08',
'1957-12-29', '1972-06-14', '1935-04-15', '1927-09-09',
'1928-10-01', '1950-08-19', '1958-06-26', '1926-09-14',
'1935-02-10', '1962-04-12', '1951-02-05', '1985-03-21',
'1990-05-03', '1945-09-20', '1961-01-21', '1972-11-28',
'1972-07-01', '1958-08-14', '1985-09-02', '2001-07-26',
'1929-05-06', '1960-06-14', '1954-06-14', '1977-10-19',
'1971-11-02', '1950-11-27', '1963-05-23', '1948-11-14',
'1966-11-10', '1990-10-15', '1963-04-22', '1954-12-10',
'1968-07-01', '1973-04-01', '1988-09-15', '1988-04-15',
'1959-09-27', '1999-06-06', '1985-01-01', '2003-05-07',
'1957-01-23', '1927-12-11', '1981-06-22', '1962-06-04',
'1990-06-25', '1960-01-16', '1954-07-15', '1984-07-03',
'1971-08-06', '1950-04-05', '1967-05-28', '1952-10-13',
'1983-10-14', '1969-03-02', '1968-10-06', '1931-01-26',
'1940-11-11', '1987-11-18', '1965-12-15', '1962-10-16',
'1981-07-05', '1934-03-19', '1989-07-17', '1992-05-09',
'1982-05-20', '1929-08-23', '1971-03-26', '1995-05-25',
'1997-06-04', '1951-12-04', '1967-05-27', '1939-09-19',
'1986-05-02', '1936-03-27', '1958-06-11', '1953-03-30',
'1997-03-12', '1963-12-29', '1957-04-17', '1998-10-07',
'1981-10-24', '1971-04-25', '1970-02-22', '1956-01-09',
'1968-06-18', '1969-10-30', '1995-10-17', '1978-01-15',
'1954-07-05', '1926-06-26', '1960-01-20', '1964-01-04',
'1945-08-19', '1934-10-06', '1974-10-27', '1951-11-08',
'1963-06-22', '1963-04-04', '1968-02-10', '1993-07-05',
'1978-03-06', '1980-05-18', '1970-11-12', '1977-06-07',
'1982-02-10', '1926-08-27', '1969-09-21', '1970-03-13',
'1974-03-10', '1939-11-09', '1986-12-13', '1958-04-06',
'1973-07-28', '1953-01-20', '1977-08-16', '2004-05-08',
'1992-01-20', '2005-01-29', '1974-02-15', '1972-05-23',
'1945-11-04', '1958-09-20', '1972-10-18', '1937-03-17',
'1961-10-24', '1983-07-24', '1955-05-06', '1951-01-15',
'1954-06-30', '1956-01-24', '1958-09-02', '1948-06-30',
'1969-07-24', '1954-08-22', '1961-09-10', '1980-08-18',
'1964-08-23', '1978-08-27', '1969-08-04', '1988-03-25',
'1979-06-24', '1985-04-15', '1973-05-07', '1975-07-07',
```

```
'1978-03-04', '1959-07-30', '1952-04-02', '1928-07-15',
'1980-09-15', '1956-09-15', '1986-12-17', '1969-11-22',
'1954-07-14', '1952-09-27', '1973-02-14', '1960-02-01',
'1942-04-03', '1933-03-15', '1964-02-13', '1963-02-09',
'1974-11-02', '1929-05-30', '1964-11-17', '1934-06-23',
'1960-04-03', '1987-05-19', '1955-07-25', '1976-09-17',
'1996-04-10', '1944-07-26', '1988-02-15', '1969-02-22',
'1993-04-08', '1970-06-09', '1991-03-13', '1964-08-08',
'1953-12-08', '1975-12-28', '1959-05-10', '1972-10-04',
'1997-07-05', '1990-11-23', '1962-09-27', '1975-07-13',
'1967-09-16', '1971-02-11', '1982-07-30', '1973-01-21',
'1993-10-12', '1964-02-15', '1969-11-01', '1973-05-16',
'1990-11-07', '1956-05-30', '1950-05-27', '1980-08-17',
'1974-10-15', '1961-01-31', '1973-06-09', '1969-12-22',
'1962-05-04', '1984-07-05', '1927-10-24', '1967-01-24',
'1976-03-26', '1983-06-23', '1990-01-24', '1965-04-13',
'1957-04-05', '1966-01-04', '1979-04-12', '1983-10-12',
'2001-06-22', '1991-10-13', '1993-11-17', '1974-11-20',
'1963-08-04', '1956-09-01', '1981-05-06', '1995-03-13',
'1976-09-29', '1958-01-01', '1979-12-11', '1976-05-16',
'1984-08-01', '1953-05-23', '1991-04-13', '1949-10-04',
'1958-10-29', '1955-11-07', '1973-10-19', '1968-06-24',
'1951-09-03', '1955-11-10', '1964-04-06', '1962-02-14',
'1991-10-04', '1976-06-15', '1962-04-05', '1999-09-11',
'1989-05-14', '1968-02-09', '1950-11-20', '1992-06-19',
'1936-11-05', '1966-09-16', '1954-01-06', '1987-09-26',
'1953-07-30', '1942-05-04', '1968-07-24', '1982-08-01',
'1963-06-13', '1993-05-14', '1986-03-14', '1969-09-15',
'1987-05-05', '1974-12-05', '1975-06-29', '1941-04-23',
'1948-05-01', '1961-09-03', '1986-04-28', '1943-05-28',
'1953-12-25', '1954-01-29', '1992-07-23', '1976-01-02',
'1941-10-16', '1972-04-18', '1993-11-02', '1991-06-05',
'1984-11-06', '1959-10-07', '1951-06-13', '1975-12-20',
'1960-01-13', '1991-01-28', '1970-03-14', '1961-04-25',
'1958-07-28', '1988-09-02', '1998-07-29', '1971-12-10',
'1973-07-13', '1999-10-25', '1958-10-26', '1967-06-20',
'1966-06-07', '1986-05-01', '1984-03-03', '1976-10-09',
'1955-01-20', '1956-10-08', '1965-04-27', '1999-11-30',
'1966-05-22', '1949-10-13', '1957-01-15', '1942-04-17',
'1941-11-16', '1995-07-08', '1983-02-08', '1961-04-22',
'1969-05-16', '1968-05-13', '1970-10-09', '1978-08-08',
'1968-11-22', '1960-11-19', '1976-06-30', '1952-03-08',
'1972-08-09', '1991-03-29', '1990-10-28', '1997-11-18',
'1952-01-29', '1992-11-27', '1987-09-19', '1985-12-27',
'1970-01-08', '1983-08-24', '1956-05-02', '1989-10-06',
'1965-09-15', '1979-07-03', '1986-01-30', '1968-05-16',
'1994-03-13', '1957-08-08', '1987-04-24', '1935-01-29',
```

```
'1941-03-30', '1978-01-22', '1982-07-02', '1959-05-28',
'1939-03-09', '1986-11-12', '1959-08-05', '1962-03-20',
'1985-08-29', '1980-03-18', '1990-11-09', '1974-05-18',
'1972-07-18', '1940-09-13', '1991-08-19', '1994-11-24',
'1974-12-24', '1981-08-29', '1988-09-19', '1973-02-07',
'1967-06-19', '1977-08-12', '1955-07-06', '1978-12-18',
'1965-11-11', '1946-08-30', '1949-03-20', '1986-07-23',
'1979-09-03', '1958-02-17', '1977-06-14', '1950-03-25',
'1957-11-12', '1994-04-22', '1982-02-08', '1974-09-14',
'1966-01-21', '1964-06-22', '1927-05-25', '1984-03-06',
'1973-05-27', '1938-09-08', '1987-02-22', '1971-08-05',
'1961-11-24', '1935-06-29', '1978-05-23', '1985-06-18',
'1976-11-21', '1957-08-30', '1953-04-19', '1999-12-27',
'1971-07-02', '1977-03-23', '1980-01-09', '1974-01-03',
'1992-10-07', '1988-10-26', '1980-07-12', '1930-08-13',
'1967-04-09', '1988-08-04', '1974-12-28', '1987-04-29',
'1956-06-22', '1983-06-14', '1985-05-25', '1986-03-31',
'1973-12-26', '1977-12-16', '1972-01-20', '1936-05-01',
'1976-09-12', '1966-02-21', '1991-05-01', '1989-03-09',
'1948-05-31', '1965-04-07', '1982-05-28', '1932-03-10',
'1946-11-01', '1993-02-26', '1943-06-30', '1990-06-08',
'1931-03-07', '2000-06-09', '1975-04-16', '1986-01-18',
'1981-08-10', '1987-01-27', '1989-10-28', '1978-09-30',
'1983-09-02', '1989-12-17', '1993-09-11', '1985-03-19',
'1979-04-30', '1962-02-13', '1946-08-11', '1985-12-03',
'1990-10-19', '1978-10-26', '1994-10-07', '1972-09-22',
'1973-11-14', '1960-03-01', '1995-12-28', '1937-02-01',
'1956-03-02', '1991-02-03', '1974-04-16', '1939-11-04',
'1997-01-02', '1979-01-21', '1966-02-13', '1983-06-12',
'1984-07-20', '1983-07-10', '2004-12-30', '1963-05-19',
'1993-10-07', '1983-06-13', '1999-03-05', '1993-10-05',
'1970-04-17', '1987-10-28', '1991-02-04', '1980-09-18',
'1986-02-17', '1988-07-28', '1974-05-30', '1984-02-14',
'1932-11-19', '1975-06-02', '1988-09-06', '1958-09-10',
'1978-10-05', '1989-02-08', '1949-11-17', '1994-11-12',
'1980-07-30', '2001-07-05', '1997-12-27', '1963-06-30',
'1940-08-23', '1933-04-02', '1949-11-16', '2003-09-14',
'1985-06-20', '1987-05-23', '1995-04-22', '1978-10-01',
'1966-08-08', '1986-11-24', '1982-02-11', '1992-07-24',
'1999-06-19', '1998-05-20', '1955-06-26', '1990-01-17',
'1987-07-25', '1961-12-14', '1990-12-18', '1959-06-28',
'1982-02-05', '1970-01-18', '1972-06-12', '1987-06-13',
'1992-10-03', '1973-05-04', '1988-01-04', '1943-12-15',
'1945-05-05', '1997-09-22', '1955-02-01', '1963-12-28',
'1978-12-25', '1960-08-05', '1965-03-25', '1998-02-03',
'1984-02-07', '1984-05-19', '1969-09-11', '1984-02-29',
'1994-07-27', '1964-02-18', '1991-10-22', '1995-08-16',
```

```
'1940-09-17', '1986-05-07', '1939-04-14', '1984-05-04',
'1941-07-31', '1995-11-29', '1960-03-12', '1955-04-03',
'2000-08-28', '1972-02-15', '1944-11-11', '2004-03-18',
'1986-12-31', '1991-04-11', '1995-09-11', '1986-04-15',
'1982-01-16', '1986-08-17', '1997-07-01', '1990-05-22',
'1960-12-13', '1977-06-12', '1969-01-14', '1985-08-31',
'1963-06-04', '1952-05-07', '1981-11-29', '1952-12-07',
'1959-06-06', '1987-09-08', '1993-11-24', '1993-05-10',
'1995-01-15', '1936-07-22', '1967-05-05', '1996-11-12',
'1957-04-25', '1977-07-17', '1973-04-06', '1968-03-19',
'1989-07-08', '1955-01-13', '1998-03-18', '1994-03-01',
'1994-02-16', '1990-02-25', '1992-12-29', '1988-11-01',
'1976-04-11', '1968-03-24', '1990-04-25', '1985-08-21',
'1978-11-30', '1930-10-21', '1992-11-20', '1970-11-09',
'1949-04-24', '1987-02-14', '1993-03-23', '1972-01-03',
'1965-04-01', '1973-10-14', '1972-09-12', '1991-01-01',
'1987-02-13', '1995-12-04', '1976-01-10', '1997-12-26',
'1961-05-13', '1984-12-16', '1973-09-22', '1988-03-21',
'1963-07-14', '1948-11-30', '1967-08-24', '1962-04-30',
'1968-05-29', '1966-09-19', '1980-03-24', '1947-07-15',
'1972-10-05', '1976-12-10', '1962-05-13', '1962-06-27',
'1962-11-18', '1956-05-15', '1972-03-05', '1956-12-13',
'1979-12-27', '1982-12-27', '1985-05-13', '1953-04-13',
'1985-12-08', '1985-07-08', '1975-10-11', '1991-04-22',
'1997-10-23', '1970-06-27', '1999-05-31', '1961-12-05',
'1971-10-19', '1945-11-26', '1987-10-26', '1968-01-28',
'1962-03-04', '1971-09-01', '1943-12-17', '1976-12-14',
'1979-10-22', '1968-09-19', '1939-06-01', '1975-11-30',
'1985-04-03', '1995-10-10', '1999-09-01', '1991-07-06',
'1985-03-31', '1972-12-31', '1929-03-19', '1989-12-10',
'1994-05-31', '1972-01-05', '1987-08-16', '1944-05-14',
'1949-06-09', '2000-02-20', '1985-04-04', '1972-03-28',
'1957-12-26', '1932-09-17', '1979-08-14', '1967-10-18',
'1969-05-01', '1967-03-17', '1982-04-19', '1950-09-15',
'1976-04-12', '1991-01-31', '1981-01-06', '1975-01-26',
'1993-05-27', '2004-06-19', '1967-10-04', '1993-09-29',
'1970-11-20', '1950-04-17', '1953-03-19', '1967-03-30',
'1993-04-29', '1948-04-11', '1929-04-22', '1997-11-23',
'1976-01-15', '1960-04-08', '1930-02-28', '1945-12-07',
'1960-10-28', '1979-01-26', '1985-09-01', '1961-12-18',
'2000-08-16', '1969-11-20', '1971-12-12', '1966-05-10',
'1949-08-14', '1979-01-08', '1946-03-21', '1965-02-03',
'1983-03-20', '1954-07-21', '1987-10-27', '1981-03-29',
'1928-04-02', '1981-03-04', '1952-12-05', '1997-08-04',
'1947-10-27', '1953-10-18', '1967-03-12', '1948-09-07',
'1966-06-24', '1956-09-14', '1965-11-06', '1972-09-13',
'1983-11-10', '1992-10-08', '1994-07-09', '1984-09-13',
```

```
       '1966-06-19', '1967-09-30', '1990-06-21', '1994-12-08',
       '1990-09-12', '1966-05-29', '1995-08-30', '1982-06-27',
       '1978-10-04', '1959-03-31', '1975-09-11', '1977-05-18',
       '1991-08-21', '1999-06-28', '1975-12-24', '1931-09-12',
       '1947-08-14', '1962-03-19', '1962-12-06', '1961-09-28',
       '1987-02-26', '1955-12-04', '1934-02-09', '1986-04-03',
       '1964-03-15', '1969-03-20', '1948-03-22', '1966-12-15',
       '1933-03-01', '1987-09-22', '1981-01-26', '1959-01-15',
       '2001-12-19', '1982-02-19', '1928-06-26', '1964-03-16',
       '1964-06-25', '1965-09-27', '1971-01-19', '1976-05-24',
       '1927-08-25', '1973-10-09', '1975-06-25', '1988-04-09',
       '1975-06-01', '1942-04-02', '1981-02-18', '1964-08-18',
       '1967-09-23', '1969-12-12', '1929-04-07', '1940-09-06',
       '1937-02-06', '1990-01-13', '1990-06-13', '1986-10-17',
       '1984-08-31', '1950-12-14', '1969-09-08', '1989-08-16',
       '1956-05-01', '1961-11-07', '1959-06-18', '1997-01-18',
       '1975-10-07', '1924-10-30', '1959-10-19', '1952-05-26',
       '1967-10-28', '1996-01-11', '1983-01-21', '1971-11-26',
       '1959-05-30', '2001-07-10', '1985-01-02', '1972-08-15',
       '1965-11-21', '1963-09-11', '1983-01-08', '1961-07-31',
       '1972-07-29', '1961-09-13', '1957-06-12', '1987-11-30',
       '1967-08-28', '1951-03-31', '1989-10-19', '2000-02-15',
       '1962-03-14', '1997-08-08', '1970-06-25', '1943-07-25',
       '1961-06-16', '1968-10-26', '1996-01-10', '1944-06-17',
       '1966-08-03', '1925-08-29', '1996-04-04', '1937-04-16',
       '1962-11-15', '1940-11-08', '1964-11-18', '1963-02-26',
       '1927-02-03', '1996-04-01', '1999-09-29', '1938-08-07',
       '2001-07-17', '1965-08-18', '2000-03-16', '1940-10-27',
       '1956-03-20', '1963-03-13', '1961-06-18', '1949-02-15',
       '1939-04-13', '1998-12-29', '1961-01-30', '1949-02-25',
       '1963-02-20', '1941-09-30', '1957-12-17', '1948-05-16',
       '1996-07-04', '1940-03-06', '1957-07-27', '1970-03-16',
       '1964-02-21', '1957-04-20', '1946-05-28', '1999-10-26',
       '1949-04-28', '1998-11-12', '1952-11-23', '1966-07-25',
       '1957-04-15', '1934-05-04', '1960-02-21', '1966-06-12',
       '1958-03-18', '1963-03-08', '1968-10-14', '1931-04-21',
       '1946-11-20', '1964-09-17', '1932-08-10', '1997-04-06',
       '1941-07-06', '1936-05-04', '1930-06-26', '1943-08-04',
       '1962-01-24', '1940-07-30', '1970-01-09', '1963-03-14',
       '1954-07-07', '1934-05-24', '1997-04-17', '1949-12-22'],
      dtype=object)
```

So, the data format is uniform

[15]:
```
# Defining a function to extract the age of customer from their date of birth
def find_age(date_of_birth):
    year = int(date_of_birth[:4])
```

```
    return (2023 - year)

# Applying the function to the required column
train_data_['dob'] = train_data_['dob'].apply(find_age)

train_data_.rename(columns={'dob':'age'}, inplace=True)
train_data_['age'].unique()
```

[15]:
```
array([35, 45, 61, 56, 37, 62, 30, 76, 82, 49, 33, 57, 34, 78, 58, 71, 85,
       77, 43, 52, 59, 46, 53, 39, 25, 36, 81, 51, 63, 69, 29, 47, 42, 97,
       87, 26, 68, 40, 44, 23, 66, 41, 88, 48, 28, 96, 95, 73, 65, 72, 38,
       22, 94, 60, 75, 55, 50, 64, 24, 20, 54, 92, 83, 89, 31, 84, 70, 67,
       19, 18, 86, 90, 27, 79, 32, 74, 80, 93, 91, 99, 98], dtype=int64)
```

3.3.10 Column : merchant

[16]:
```
train_data_['merchant'].unique()
```

[16]:
```
array(['fraud_Rippin, Kub and Mann', 'fraud_Heller, Gutmann and Zieme',
       'fraud_Lind-Buckridge', 'fraud_Kutch, Hermiston and Farrell',
       'fraud_Keeling-Crist', 'fraud_Stroman, Hudson and Erdman',
       'fraud_Rowe-Vandervort', 'fraud_Corwin-Collins',
       'fraud_Herzog Ltd', 'fraud_Schoen, Kuphal and Nitzsche',
       'fraud_Rutherford-Mertz', 'fraud_Kerluke-Abshire',
       'fraud_Lockman Ltd', 'fraud_Kiehn Inc', 'fraud_Beier-Hyatt',
       'fraud_Schmidt and Sons', 'fraud_Lebsack and Sons',
       'fraud_Mayert Group', 'fraud_Konopelski, Schneider and Hartmann',
       'fraud_Schultz, Simonis and Little', 'fraud_Bauch-Raynor',
       'fraud_Harris Inc', 'fraud_Kling-Grant', 'fraud_Pacocha-Bauch',
       'fraud_Lesch Ltd', 'fraud_Kunde-Sanford', "fraud_Deckow-O'Conner",
       'fraud_Bruen-Yost', 'fraud_Kunze Inc',
       'fraud_Nitzsche, Kessler and Wolff',
       'fraud_Kihn, Abernathy and Douglas', 'fraud_Torphy-Goyette',
       'fraud_Balistreri-Nader', 'fraud_Bahringer, Schoen and Corkery',
       'fraud_Hudson-Ratke', 'fraud_Heidenreich PLC',
       'fraud_Halvorson Group', 'fraud_Harber Inc',
       'fraud_Mosciski, Gislason and Mertz',
       'fraud_Christiansen, Goyette and Schamberger', 'fraud_Howe Ltd',
       'fraud_Ledner-Pfannerstill', 'fraud_Koepp-Witting',
       'fraud_Doyle Ltd', 'fraud_Schaefer, Maggio and Daugherty',
       'fraud_Stracke-Lemke', 'fraud_Mosciski, Ziemann and Farrell',
       'fraud_Cartwright-Harris', "fraud_Pacocha-O'Reilly",
       'fraud_Pfeffer LLC', 'fraud_Huels-Hahn', 'fraud_Volkman PLC',
       'fraud_Kiehn-Emmerich', 'fraud_Kemmer-Buckridge',
       'fraud_Cummerata-Jones', 'fraud_Goldner, Kovacek and Abbott',
       'fraud_Spinka-Welch', 'fraud_Huel-Langworth',
       'fraud_Macejkovic-Lesch', 'fraud_Morar Inc',
```

```
'fraud_Eichmann, Bogan and Rodriguez',
'fraud_Huel, Hammes and Witting', 'fraud_Ferry, Lynch and Kautzer',
'fraud_Heathcote LLC', 'fraud_Little, Gutmann and Lynch',
'fraud_Jaskolski-Dibbert', 'fraud_Hackett-Lueilwitz',
'fraud_Cummings LLC', 'fraud_Swaniawski, Lowe and Robel',
'fraud_Osinski, Ledner and Leuschke',
'fraud_Reichert, Huels and Hoppe', 'fraud_Ankunding LLC',
'fraud_Pfeffer and Sons', 'fraud_Greenholt, Jacobi and Gleason',
'fraud_Connelly, Reichert and Fritsch', 'fraud_Torp-Labadie',
'fraud_Wolf Inc', 'fraud_VonRueden Group', 'fraud_Vandervort-Funk',
'fraud_Bernhard Inc', 'fraud_Morissette, Weber and Wiegand',
'fraud_Brekke and Sons', 'fraud_Mraz-Herzog',
'fraud_Padberg-Welch', 'fraud_Kutch-Hegmann',
'fraud_Corwin-Gorczany', 'fraud_Huels-Nolan', 'fraud_DuBuque LLC',
'fraud_Schmeler Inc', 'fraud_Schaefer, McGlynn and Bosco',
'fraud_Williamson LLC', 'fraud_Pouros-Conroy',
'fraud_Erdman-Kertzmann', 'fraud_Kuhn LLC',
'fraud_Fisher-Schowalter', 'fraud_Medhurst PLC',
'fraud_Kerluke Inc', 'fraud_Stark-Batz',
'fraud_Gottlieb, Considine and Schultz',
'fraud_Adams, Kovacek and Kuhlman', 'fraud_Grimes LLC',
'fraud_Rodriguez Group', 'fraud_Wuckert-Walter',
'fraud_Weber and Sons', 'fraud_Herman, Treutel and Dickens',
'fraud_Rau and Sons', 'fraud_Koss and Sons',
'fraud_Smitham-Schiller', 'fraud_Hills-Olson', 'fraud_Durgan-Auer',
'fraud_McDermott-Rice', 'fraud_Raynor, Reinger and Hagenes',
'fraud_Powlowski-Weimann', 'fraud_Langosh, Wintheiser and Hyatt',
'fraud_Baumbach, Hodkiewicz and Walsh', 'fraud_Rempel Inc',
'fraud_Bins-Rice', 'fraud_Emard Inc', 'fraud_Kutch and Sons',
'fraud_Fisher Inc', 'fraud_Olson, Becker and Koch',
"fraud_Friesen-D'Amore", 'fraud_Reilly, Heaney and Cole',
'fraud_Boyer PLC', 'fraud_Luettgen PLC', 'fraud_Cassin-Harvey',
'fraud_Funk Group', 'fraud_Goodwin-Nitzsche',
'fraud_Parisian, Schiller and Altenwerth', 'fraud_Metz-Boehm',
'fraud_Monahan, Bogisich and Ledner', 'fraud_Koelpin and Sons',
'fraud_Kuhic LLC', 'fraud_Koepp-Parker',
'fraud_Stehr, Jewess and Schimmel',
'fraud_Quitzon, Green and Bashirian',
'fraud_Nicolas, Hills and McGlynn', 'fraud_Leannon-Ward',
'fraud_Friesen-Stamm', 'fraud_Botsford Ltd', 'fraud_Bradtke PLC',
'fraud_Zieme, Bode and Dooley', 'fraud_Hills-Witting',
'fraud_Brown-Greenholt', 'fraud_Bernhard, Grant and Langworth',
'fraud_Wiza, Schaden and Stark', 'fraud_Buckridge PLC',
'fraud_Kilback LLC', 'fraud_Spinka Inc',
'fraud_Kerluke, Kertzmann and Wiza',
'fraud_Ferry, Reichel and DuBuque', 'fraud_Price Inc',
'fraud_Johnston, Nikolaus and Maggio', 'fraud_Berge LLC',
```

```
'fraud_Conroy-Cruickshank', 'fraud_Abshire PLC',
'fraud_Gleason-Macejkovic', 'fraud_McGlynn-Jaskolski',
'fraud_Bartoletti-Wunsch', 'fraud_Cole PLC',
'fraud_Lehner, Reichert and Mills', 'fraud_McCullough LLC',
'fraud_Schmitt Ltd', 'fraud_Pouros-Haag',
'fraud_Strosin-Cruickshank', 'fraud_Weimann, Kuhic and Beahan',
'fraud_Donnelly PLC', 'fraud_Lockman, West and Runte',
'fraud_Torp-Lemke', 'fraud_Kris-Weimann',
'fraud_Rodriguez, Yost and Jenkins', 'fraud_Wiegand-Lowe',
'fraud_Auer-Mosciski', 'fraud_Okuneva, Schneider and Rau',
'fraud_Kassulke PLC', 'fraud_Dach-Nader', 'fraud_Cummings Group',
'fraud_Reichert, Shanahan and Hayes',
'fraud_Tromp, Kerluke and Glover',
'fraud_Bins, Balistreri and Beatty', 'fraud_Trantow PLC',
'fraud_Miller-Hauck', 'fraud_Kuvalis Ltd', 'fraud_Dickinson Ltd',
'fraud_Stanton, Jakubowski and Baumbach',
'fraud_Rohan, White and Aufderhar',
'fraud_Cormier, Stracke and Thiel',
'fraud_Tillman, Fritsch and Schmitt',
'fraud_Christiansen-Gusikowski',
'fraud_Jenkins, Hauck and Friesen', 'fraud_Reichel Inc',
'fraud_Prohaska-Murray', 'fraud_Parker, Nolan and Trantow',
'fraud_Volkman Ltd', 'fraud_Streich, Hansen and Veum',
'fraud_Kutch LLC', 'fraud_Barton Inc', 'fraud_Dooley-Thompson',
'fraud_Flatley Group', 'fraud_Turcotte-Halvorson',
'fraud_Lynch Ltd', 'fraud_Romaguera, Cruickshank and Greenholt',
'fraud_Gutmann Ltd', 'fraud_Sporer Inc',
'fraud_Runolfsson and Sons', 'fraud_Block Group',
'fraud_Bahringer-Larson', 'fraud_Rowe, Batz and Goodwin',
'fraud_Schmitt Inc', 'fraud_Maggio-Fahey',
'fraud_Haley, Jewess and Bechtelar',
'fraud_Ruecker, Beer and Collier',
'fraud_Robel, Cummerata and Prosacco', 'fraud_Jast-McDermott',
'fraud_Bernier, Volkman and Hoeger', 'fraud_Heaney-Marquardt',
'fraud_Hudson-Grady', 'fraud_Towne, Walker and Borer',
'fraud_Terry-Huel', 'fraud_Kihn-Fritsch', 'fraud_Eichmann-Russel',
'fraud_Miller-Harris', 'fraud_Schumm PLC',
'fraud_Greenfelder, Bartoletti and Davis',
'fraud_Kovacek, Dibbert and Ondricka', 'fraud_Hermann-Gaylord',
'fraud_Bailey-Morar', 'fraud_McDermott-Weimann', 'fraud_Welch Inc',
'fraud_Auer-West', 'fraud_Boehm, Predovic and Reinger',
'fraud_Dibbert and Sons', 'fraud_Lind, Huel and McClure',
'fraud_Casper, Hand and Zulauf', 'fraud_McCullough Group',
'fraud_Streich, Dietrich and Barton', "fraud_O'Keefe-Hudson",
'fraud_Pagac LLC', 'fraud_Bernier, Streich and Jewess',
"fraud_Greenholt, O'Hara and Balistreri", 'fraud_Dickinson-Rempel',
'fraud_Dare-Marvin', 'fraud_Spencer PLC', 'fraud_Stamm-Rodriguez',
```

```
'fraud_Langworth, Boehm and Gulgowski',
'fraud_Larkin, Stracke and Greenfelder',
'fraud_Yost, Block and Koepp',
'fraud_Douglas, Schneider and Turner', 'fraud_Kuphal-Predovic',
'fraud_Parisian and Sons', 'fraud_Flatley-Durgan',
'fraud_Gislason Group', 'fraud_Bednar Group',
'fraud_Heller-Langosh', 'fraud_Zboncak Ltd', 'fraud_Gerlach Inc',
'fraud_Wilkinson Ltd', 'fraud_Moen, Reinger and Murphy',
'fraud_Kerluke, Considine and Macejkovic', 'fraud_Upton PLC',
'fraud_Bogisich Inc', 'fraud_Marks Inc', 'fraud_Murray-Smitham',
'fraud_Frami Group', 'fraud_Ortiz Group', 'fraud_Goldner-Lemke',
'fraud_Hickle Group', 'fraud_Conroy Ltd',
'fraud_Schumm, Bauch and Ondricka', 'fraud_McGlynn-Heathcote',
'fraud_Herman Inc', 'fraud_Gutmann-Upton',
'fraud_Volkman-Predovic',
'fraud_Wintheiser, Dietrich and Schimmel',
'fraud_Raynor, Feest and Miller', 'fraud_Bogisich-Homenick',
'fraud_Jast and Sons', 'fraud_Baumbach, Strosin and Nicolas',
'fraud_Towne, Greenholt and Koepp', "fraud_Schamberger-O'Keefe",
'fraud_Nienow PLC', 'fraud_Emmerich-Luettgen', 'fraud_Mante Group',
'fraud_Baumbach, Feeney and Morar', 'fraud_Thiel PLC',
'fraud_Mohr Inc', 'fraud_Hagenes, Kohler and Hoppe',
'fraud_Block-Parisian', 'fraud_Dooley Inc',
'fraud_Bashirian Group', 'fraud_Cremin, Hamill and Reichel',
'fraud_Sawayn PLC', 'fraud_Jewess LLC', 'fraud_Roberts-Beahan',
'fraud_Brown PLC', 'fraud_Tillman, Dickinson and Labadie',
'fraud_Reichert, Rowe and Mraz', 'fraud_Kilback Group',
'fraud_Spencer-Runolfsson', 'fraud_Labadie, Treutel and Bode',
'fraud_Dach-Borer', 'fraud_Johnson, Runolfsdottir and Mayer',
'fraud_Smitham-Boehm', 'fraud_Moore, Dibbert and Koepp',
'fraud_Hamill-Daugherty', 'fraud_Watsica, Haag and Considine',
'fraud_Smith-Stokes', 'fraud_Kuhic, Bins and Pfeffer',
'fraud_Nader-Heller', 'fraud_Kozey-Boehm', 'fraud_Stiedemann Inc',
'fraud_Lehner, Mosciski and King', 'fraud_Reynolds-Schinner',
'fraud_Koss, McLaughlin and Mayer', 'fraud_Terry, Johns and Bins',
'fraud_Kemmer-Reinger', 'fraud_Larson-Moen', 'fraud_Kuhic Inc',
'fraud_Kris-Padberg', 'fraud_Schmeler, Bashirian and Price',
'fraud_Klocko LLC', 'fraud_Schaefer, Fay and Hilll',
'fraud_Kling Inc', 'fraud_Turner and Sons', 'fraud_Mohr-Bayer',
'fraud_Wiza LLC', 'fraud_Skiles-Ankunding',
'fraud_Jaskolski-Vandervort', 'fraud_Kerluke PLC',
'fraud_Ernser-Lynch', 'fraud_Zboncak, Rowe and Murazik',
'fraud_Stoltenberg-Beatty', 'fraud_Kuphal-Bartoletti',
'fraud_Rempel PLC', 'fraud_Denesik and Sons', 'fraud_Goyette Inc',
'fraud_Dicki Ltd', 'fraud_Stokes, Christiansen and Sipes',
'fraud_Hermann and Sons', 'fraud_Gibson-Deckow',
'fraud_Douglas-White', 'fraud_Murray Ltd', 'fraud_McKenzie-Huels',
```

```
'fraud_Runte-Mohr', 'fraud_Roob, Conn and Tremblay',
'fraud_Boyer-Reichert', 'fraud_Shanahan-Lehner',
'fraud_Greenholt Ltd', 'fraud_Kulas Group',
'fraud_Dare, Fritsch and Zboncak', 'fraud_Waters-Cruickshank',
'fraud_Kunze, Larkin and Mayert', 'fraud_Bednar Inc',
'fraud_Schimmel-Olson', 'fraud_Lubowitz, Terry and Stracke',
'fraud_Schoen-Quigley', 'fraud_Kihn-Schuster',
'fraud_Padberg-Rogahn', 'fraud_Lynch-Mohr', 'fraud_Hilpert-Conroy',
'fraud_Jacobi Inc', 'fraud_Berge, Kautzer and Harris',
'fraud_Hammes-Beatty', 'fraud_Gulgowski LLC',
'fraud_Schuppe, Nolan and Hoeger', 'fraud_Pollich LLC',
'fraud_Bernier and Sons', 'fraud_Wilkinson PLC',
'fraud_Kling-Ernser', "fraud_O'Connell, Botsford and Hand",
'fraud_Little-Gleichner', 'fraud_Champlin and Sons',
'fraud_Hoppe-Parisian', 'fraud_Schuppe LLC', 'fraud_Emmerich-Rau',
'fraud_Beier LLC', 'fraud_Champlin-Casper', 'fraud_Gerhold LLC',
'fraud_Renner Ltd', 'fraud_Gaylord-Powlowski',
'fraud_Prosacco, Kreiger and Kovacek', 'fraud_Eichmann-Kilback',
"fraud_O'Hara-Wilderman", 'fraud_Terry Ltd',
'fraud_Beier and Sons', "fraud_O'Keefe-Wisoky",
'fraud_Simonis-Prohaska', 'fraud_Breitenberg-Hermiston',
'fraud_Brown Inc', 'fraud_Fahey Inc',
'fraud_Boehm, Block and Jakubowski',
'fraud_Mante, Luettgen and Hackett', 'fraud_Thompson-Gleason',
'fraud_Turcotte, Batz and Buckridge',
'fraud_Goyette, Howell and Collier',
'fraud_Jones, Sawayn and Romaguera', 'fraud_Medhurst Inc',
'fraud_Bode-Schuster', 'fraud_Connelly-Carter',
'fraud_Gottlieb-Hansen', 'fraud_Schroeder, Wolff and Hermiston',
'fraud_Hettinger, McCullough and Fay',
'fraud_Pouros, Walker and Spencer', 'fraud_Kub PLC',
'fraud_Watsica LLC', 'fraud_Stiedemann Ltd',
'fraud_Hauck, Dietrich and Funk', 'fraud_Labadie LLC',
'fraud_Friesen Inc', 'fraud_Crist, Jakubowski and Littel',
'fraud_Hodkiewicz, Prohaska and Paucek',
'fraud_McDermott, Osinski and Morar', 'fraud_Fritsch and Sons',
'fraud_Swaniawski, Nitzsche and Welch',
'fraud_Windler, Goodwin and Kovacek',
'fraud_Yost, Schamberger and Windler',
'fraud_Metz, Russel and Metz', 'fraud_Erdman-Durgan',
'fraud_Bahringer, Bergnaum and Quitzon', 'fraud_Veum-Koelpin',
'fraud_Becker, Harris and Harvey', 'fraud_Monahan-Morar',
'fraud_Johns Inc', 'fraud_Mosciski Group', 'fraud_Abbott-Steuber',
'fraud_Schaefer Ltd', 'fraud_Dibbert-Green',
'fraud_Weimann-Lockman', 'fraud_Kub-Heaney', 'fraud_Zulauf LLC',
'fraud_Ratke and Sons', 'fraud_Jakubowski Inc', 'fraud_Beer-Jast',
'fraud_Kautzer and Sons', 'fraud_Feil-Morar',
```

```
'fraud_Johnston-Casper', 'fraud_Medhurst, Labadie and Gottlieb',
"fraud_Lesch, D'Amore and Brown", 'fraud_Botsford PLC',
'fraud_Bins-Howell', 'fraud_Kihn Inc',
'fraud_Hartmann, Rowe and Hermann', 'fraud_Towne LLC',
'fraud_Lynch-Wisozk', 'fraud_Kutch-Ferry', 'fraud_Goyette-Gerhold',
'fraud_Bradtke, Torp and Bahringer', 'fraud_Homenick LLC',
'fraud_Zemlak, Tillman and Cremin',
'fraud_Schneider, Hayes and Nikolaus',
'fraud_Schumm, McLaughlin and Carter', 'fraud_Nader-Maggio',
'fraud_Haley, Batz and Auer', 'fraud_Yost-Rogahn',
'fraud_Schoen, Nienow and Bauch',
'fraud_Ledner, Hartmann and Feest', 'fraud_Collier LLC',
'fraud_Schuppe-Schuppe', 'fraud_Walter, Hettinger and Kessler',
'fraud_Bechtelar-Rippin', "fraud_Hamill-D'Amore",
'fraud_Swift PLC', 'fraud_Cronin, Kshlerin and Weber',
'fraud_Romaguera, Wehner and Tromp',
'fraud_Feil, Hilpert and Koss', 'fraud_White and Sons',
'fraud_Mueller, Gerhold and Mueller', 'fraud_Botsford and Sons',
'fraud_Kirlin and Sons', 'fraud_Bednar PLC',
'fraud_Runolfsdottir, Mueller and Hand', 'fraud_Kuphal-Toy',
'fraud_Bahringer-Streich', 'fraud_Wuckert, Wintheiser and Friesen',
'fraud_Crona and Sons', 'fraud_Prosacco LLC', 'fraud_Schiller Ltd',
'fraud_Waelchi-Wolf', 'fraud_Torphy-Kertzmann',
'fraud_McCullough, Hudson and Schuster', 'fraud_Baumbach Ltd',
'fraud_Schiller, Blanda and Johnson', 'fraud_Cartwright PLC',
'fraud_Reilly LLC', 'fraud_Cruickshank-Mills',
'fraud_Altenwerth, Cartwright and Koss',
'fraud_Effertz, Welch and Schowalter',
'fraud_Klocko, Runolfsdottir and Breitenberg',
'fraud_Ruecker-Mayert', 'fraud_Schroeder, Hauck and Treutel',
'fraud_Lemke-Gutmann', 'fraud_Graham, Hegmann and Hammes',
'fraud_Reilly and Sons', 'fraud_Stark-Koss', 'fraud_Daugherty LLC',
'fraud_Denesik, Powlowski and Pouros', 'fraud_Rippin-VonRueden',
'fraud_Heller PLC', 'fraud_Hills-Boyer', 'fraud_Cormier LLC',
'fraud_Erdman-Ebert', 'fraud_Bogisich-Weimann',
'fraud_Gutmann, McLaughlin and Wiza', 'fraud_Little Ltd',
'fraud_Bode-Rempel', 'fraud_Kutch, Steuber and Gerhold',
'fraud_Kessler Inc', 'fraud_Deckow-Dare', 'fraud_Rolfson-Kunde',
'fraud_Marvin-Lind', 'fraud_Barrows PLC', 'fraud_Abbott-Rogahn',
'fraud_Ziemann-Waters', 'fraud_Reinger, Weissnat and Strosin',
'fraud_Heathcote, Yost and Kertzmann', 'fraud_Will Ltd',
'fraud_Kutch-Wilderman', 'fraud_Hermiston, Russel and Price',
'fraud_Schmidt-Larkin', 'fraud_Lang, Towne and Schuppe',
'fraud_Weber, Thiel and Hammes',
'fraud_Hahn, Bahringer and McLaughlin',
'fraud_Koss, Hansen and Lueilwitz',
'fraud_Roberts, Ryan and Smith', 'fraud_Hintz, Bauch and Smith',
```

```
'fraud_Monahan, Hermann and Johns',
'fraud_Bahringer, Osinski and Block',
'fraud_Douglas, DuBuque and McKenzie', 'fraud_Hackett Group',
'fraud_Schmeler-Howe', 'fraud_Predovic Inc', 'fraud_Langworth LLC',
'fraud_Bartoletti and Sons', 'fraud_Bernhard-Lesch',
'fraud_Satterfield-Lowe', 'fraud_Runte, Green and Emard',
'fraud_Lowe, Dietrich and Erdman', 'fraud_Jast Ltd',
'fraud_Welch, Rath and Koepp', 'fraud_Skiles LLC',
'fraud_Ernser-Feest', 'fraud_Klein Group',
'fraud_Torp, Muller and Borer', 'fraud_Kuhn Group',
'fraud_Streich, Rolfson and Wilderman', 'fraud_Bins-Tillman',
'fraud_Ankunding-Carroll', 'fraud_Hahn, Douglas and Schowalter',
'fraud_Witting, Beer and Ernser', 'fraud_Morissette LLC',
'fraud_Berge-Hills', 'fraud_Donnelly LLC', 'fraud_Zboncak LLC',
'fraud_Turner, Ziemann and Lehner', 'fraud_Padberg-Sauer',
'fraud_Schulist Ltd', 'fraud_Rau-Grant', 'fraud_Osinski Inc',
'fraud_Berge-Ullrich', 'fraud_Wuckert-Goldner',
'fraud_Kertzmann LLC', 'fraud_Daugherty, Pouros and Beahan',
'fraud_Armstrong, Walter and Gottlieb', 'fraud_Conroy-Emard',
'fraud_Moore, Williamson and Emmerich', 'fraud_Gleason and Sons',
'fraud_Roberts, Daniel and Macejkovic', 'fraud_Turner LLC',
'fraud_Crooks and Sons', 'fraud_Waelchi Inc',
'fraud_Hoppe, Harris and Bednar', 'fraud_Effertz LLC',
'fraud_Lubowitz-Walter', 'fraud_Hyatt-Blick', 'fraud_Carroll PLC',
'fraud_Lemke and Sons', 'fraud_Treutel-King',
'fraud_Fadel-Hilpert', 'fraud_Altenwerth-Kilback',
'fraud_Bauch-Blanda', 'fraud_Sporer-Keebler', 'fraud_Hirthe-Beier',
'fraud_Wisozk and Sons', 'fraud_Streich Ltd', 'fraud_Schoen Ltd',
'fraud_Windler LLC', 'fraud_Nolan-Williamson',
'fraud_Roob-Okuneva', 'fraud_Lakin, Ferry and Beatty',
'fraud_Gottlieb Group', 'fraud_Harris Group', 'fraud_Fritsch LLC',
'fraud_Corwin-Romaguera', 'fraud_Dietrich-Fadel',
'fraud_Kling, Howe and Schneider', 'fraud_Kozey-Kuhlman',
'fraud_Graham and Sons', 'fraud_Jacobi and Sons',
'fraud_Eichmann, Hayes and Treutel',
'fraud_Brown, Homenick and Lesch', 'fraud_Erdman-Schaden',
'fraud_Durgan, Gislason and Spencer', 'fraud_Friesen-Ortiz',
'fraud_Morissette-Schaefer', 'fraud_Nienow, Ankunding and Collier',
'fraud_Cole, Hills and Jewess',
'fraud_Conroy, Balistreri and Gorczany',
'fraud_Reichel, Bradtke and Blanda',
"fraud_O'Reilly, Mohr and Purdy", 'fraud_Ullrich Ltd',
'fraud_Schroeder Group', 'fraud_Boyer-Haley',
'fraud_Dare, Casper and Bartoletti',
'fraud_Medhurst, Cartwright and Ebert', 'fraud_Quitzon-Goyette',
'fraud_Wilkinson LLC', 'fraud_Romaguera and Sons',
'fraud_Larkin Ltd', 'fraud_Fadel, Mertz and Rippin',
```

```
'fraud_Huel Ltd', 'fraud_Cummerata-Hilpert', 'fraud_Zemlak Group',
'fraud_Dare-Gibson', 'fraud_Adams-Barrows', 'fraud_Block-Hauck',
'fraud_Howe PLC', 'fraud_Leffler-Goldner', 'fraud_Tillman LLC',
'fraud_Pacocha-Weissnat', 'fraud_Morissette PLC',
'fraud_Jerde-Hermann', 'fraud_Kihn, Brakus and Goyette',
'fraud_Hills, Hegmann and Schaefer', 'fraud_Friesen Ltd',
'fraud_Kilback and Sons', 'fraud_Ebert-Daugherty',
'fraud_Hermiston, Pacocha and Smith', 'fraud_Auer LLC',
'fraud_Fadel Inc', 'fraud_Daugherty-Thompson',
'fraud_Romaguera Ltd', 'fraud_Parker-Kunde', 'fraud_Barton LLC',
'fraud_Kassulke Inc', 'fraud_McLaughlin, Armstrong and Koepp',
'fraud_Abernathy and Sons', 'fraud_Bahringer Group',
'fraud_Connelly PLC', 'fraud_Willms, Kris and Bergnaum',
'fraud_Thiel Ltd', 'fraud_Kris-Kertzmann',
"fraud_O'Connell-Ullrich", 'fraud_Kozey-McDermott',
'fraud_Reichel LLC', 'fraud_Thiel-Thiel', 'fraud_Rau-Robel',
'fraud_Haley Group', 'fraud_Turcotte, McKenzie and Koss',
'fraud_Stamm-Witting', 'fraud_Ritchie, Bradtke and Stiedemann',
'fraud_Nienow, Barrows and Romaguera', 'fraud_Shields-Wunsch',
'fraud_Goyette-Herzog', 'fraud_Shields Inc',
'fraud_Hayes, Marquardt and Dibbert',
'fraud_Swaniawski, Bahringer and Ledner', 'fraud_Roob LLC',
'fraud_Rutherford, Homenick and Bergstrom',
'fraud_Harris, Gusikowski and Heaney', 'fraud_Hintz-Bruen',
'fraud_Turner, Ruecker and Parisian', 'fraud_Ruecker Group',
'fraud_Johns-Hoeger', 'fraud_Ritchie, Oberbrunner and Cremin',
'fraud_Haag-Blanda', 'fraud_Hagenes, Hermann and Stroman',
'fraud_Reichert-Weissnat', 'fraud_Hyatt, Russel and Gleichner',
'fraud_Champlin, Rolfson and Connelly', 'fraud_Leannon-Nikolaus',
'fraud_Tromp Group', 'fraud_Kovacek Ltd', 'fraud_Kutch Group',
'fraud_Kohler, Lindgren and Koelpin', 'fraud_Heller-Abshire',
'fraud_Swift, Bradtke and Marquardt',
'fraud_Larson, Quitzon and Spencer',
'fraud_Kilback, Nitzsche and Leffler', 'fraud_Jakubowski Group',
'fraud_Breitenberg LLC', 'fraud_Collier Inc', 'fraud_Paucek-Wiza',
'fraud_Kessler Group'], dtype=object)
```

3.3.11 Column : category

[17]:
```
train_data_['category'].unique()
```

[17]:
```
array(['misc_net', 'grocery_pos', 'entertainment', 'gas_transport',
       'misc_pos', 'grocery_net', 'shopping_net', 'shopping_pos',
       'food_dining', 'personal_care', 'health_fitness', 'travel',
       'kids_pets', 'home'], dtype=object)
```

3.3.12 Columns : first, last - Full name can be found by joining them

[18]:
```
train_data_['name'] = train_data_['first'] + ' ' + train_data_['last'].
 ↪astype(str)
train_data_['name'].unique()
train_data_ = train_data_.drop(['first'], axis=1)
train_data_ = train_data_.drop(['last'], axis=1)
train_data_.head()
```

[18]:

	Unnamed: 0	trans_date_trans_time	cc_num	merchant	category	amt	gender	street	city	state	zip	lat	long	city_pop	job	age	unix_time	merch_lat	merch_long	is_fraud	name
0	0	2019-01-01 00:00:18	2703186189652095	fraud_Rippin, Kub and Mann	misc_net	4.97	F	561 Perry Cove	Moravian Falls	NC	28654	36.0788	-81.1781	3495	Psychologist, counselling	35	1325376018	36.011293	-82.048315	0	Jennifer Banks
1	1	2019-01-01 00:00:44	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.23	F	43039 Riley Greens Suite 393	Orient	WA	99160	48.8878	-118.2105	149	Special educational needs teacher	45	1325376044	49.159047	-118.186462	0	Stephanie Gill
2	2	2019-01-01 00:00:51	38859492057661	fraud_Lind-Buckridge	entertainment	220.11	M	594 White Dale Suite 530	Malad City	ID	83252	42.1808	-112.2620	4154	Nature conservation officer	61	1325376051	43.150704	-112.154481	0	Edward Sanchez
3	3	2019-01-01 00:01:16	3534093764340240	fraud_Kutch, Hermiston and Farrell	gas_transport	45.00	M	9443 Cynthia Court Apt. 038	Boulder	MT	59632	46.2306	-112.1138	1939	Patent attorney	56	1325376076	47.034331	-112.561071	0	Jeremy White
4	4	2019-01-01 00:03:06	375534208663984	fraud_Keeling-Crist	misc_pos	41.96	M	408 Bradley Rest	Doe Hill	VA	24433	38.4207	-79.4629	99	Dance movement psychotherapist	37	1325376186	38.674999	-78.632459	0	Tyler Garcia

3.3.13 Column : Street

We have already been provided with the longitude and latitude of the customer's place, so we can drop this

[19]:
```
train_data_ = train_data_.drop(['street'], axis=1)
train_data_.sample(7)
```

[19]:

	Unnamed: 0	trans_date_trans_time	cc_num	merchant	category	amt	gender	city	state	zip	lat	long	city_pop	job	age	unix_time	merch_lat	merch_long	is_fraud	name
438291	438291	2019-07-14 18:15:57	568279015842	fraud_Lynch-Wisozk	home	87.61	M	Christine	ND	58015	46.5522	-96.7909	291	Television floor manager	32	1342289757	46.559827	-95.871516	0	Micheal Hernandez
1059467	1059467	2020-03-16 00:16:33	630412733309	fraud_Koelpin and Sons	misc_net	7.77	F	Republic	MI	49879	46.3680	-87.9938	1038	Armed forces training and education officer	59	1363392993	45.754075	-87.399680	0	Heather Stanton
861612	861612	2019-12-17 19:08:34	3576144910346950	fraud_Hahn, Douglas and Schowalter	travel	3.71	M	Huslia	AK	99746	65.6899	-156.2920	277	Engineer, civil (consulting)	58	1355771314	65.238462	-155.802593	0	Thomas Payne
167740	167740	2019-03-30 03:14:34	4277232699798846	fraud_Schmidt and Sons	shopping_net	4.09	F	Rhame	ND	58651	46.1664	-103.7079	475	Illustrator	68	1333077274	46.841962	-102.810812	0	Jennifer Vance
396062	396062	2019-06-30 00:33:55	4010002218955876	fraud_Durgan-Auer	misc_net	22.60	F	Houston	TX	77026	29.7972	-95.3288	2906700	Animator	40	1341016435	29.526850	-95.689022	0	Jennifer Bishop
479009	479009	2019-07-29 01:54:03	676245600876	fraud_Hickle Group	shopping_pos	6.93	F	West Palm Beach	FL	33417	26.7197	-80.1248	459921	Historic buildings inspector/conservation officer	63	1343526843	26.217177	-80.604211	0	Ashley Blanchard
208076	208076	2019-04-16 00:19:51	2719496466799416	fraud_Berge LLC	gas_transport	110.39	F	Belle Fourche	SD	57717	44.6723	-103.8396	8007	Hospital pharmacist	83	1334535591	44.457024	-104.294658	0	Sheila Baker

3.3.14 Column : trans_date_trans_time

[20]:
```
# This column is written as strings, I will parse it to date time format
train_data_['trans_date_trans_time'] = pd.
 ↪to_datetime(train_data_['trans_date_trans_time'])
train_data_['trans_date_trans_time'].unique()
```

[20]:
```
array(['2019-01-01T00:00:18.000000000', '2019-01-01T00:00:44.000000000',
       '2019-01-01T00:00:51.000000000', …,
       '2020-06-21T12:12:32.000000000', '2020-06-21T12:13:36.000000000',
       '2020-06-21T12:13:37.000000000'], dtype='datetime64[ns]')
```

[21]:
```
# Extracting years, months and dates from this column
train_data_['year'] = train_data_['trans_date_trans_time'].dt.year
train_data_['month'] = train_data_['trans_date_trans_time'].dt.month
train_data_['day'] = train_data_['trans_date_trans_time'].dt.day

train_data_.drop('trans_date_trans_time', axis=1, inplace=True)
train_data_.sample(5)
```

[21]:

	Unnamed: 0	cc_num	merchant	category	amt	gender	city	state	zip
1013037	1013037	4247921790666	fraud_Heller, Gutmann and Zieme	grocery_pos	68.45	F	Washington Court House	OH	43160
1197369	1197369	4570636521433188	fraud_Romaguera, Wehner and Tromp	kids_pets	39.90	F	Deltona	FL	32725
468278	468278	3573030041201292	fraud_Gerlach Inc	shopping_net	6.22	F	Altonah	UT	84002
943526	943526	2231186809828225	fraud_Welch Inc	misc_net	2.20	F	Vienna	GA	31092
389472	389472	630423337322	fraud_Wiza, Schaden and Stark	misc_pos	3.61	F	Orient	WA	99160

	lat	long	city_pop	job	age
1013037	39.5370	-83.4550	22305	Television floor manager	84
1197369	28.8989	-81.2473	88735	Commercial horticulturist	35
468278	40.3207	-110.4360	302	Sales professional, IT	33
943526	32.0913	-83.7922	7420	Phytotherapist	66
389472	48.8878	-118.2105	149	Special educational needs teacher	45

	unix_time	merch_lat	merch_long	is_fraud	name
1013037	1361503212	40.100264	-83.059485	0	Judith Moss
1197369	1368636017	28.123024	-81.287401	0	Christine Leblanc
468278	1343235390	39.435702	-110.431067	0	Joanne Williams
943526	1357974404	32.755646	-83.545248	0	Katherine Pennington
389472	1340811977	49.526160	-117.456809	0	Stephanie Gill

```
         year  month  day
1013037  2020      2   22
1197369  2020      5   15
468278   2019      7   25
943526   2020      1   12
389472   2019      6   27
```

[22]:
```
# Checking out the categorical features once again
column_names = train_data_.columns

categorical_columns = [var for var in column_names if train_data_[var].
 ↪dtype=='O']

print("The columns with categorical data are: {}".format(categorical_columns))
```

```
The columns with categorical data are: ['merchant', 'category', 'gender',
'city', 'state', 'job', 'name']
```

[23]:
```
# It's safe to drop the 'name' column as it doesn't help in differentiating␣
 ↪between customers
categorical_columns = categorical_columns[:-1]
categorical_columns
```

[23]:
```
['merchant', 'category', 'gender', 'city', 'state', 'job']
```

3.4 Numerical Data

3.4.1 The distance between the merchant and the customer may well be a factor, so we use the latitudes and longitudes to find the distance between them

[24]:
```
train_data_['latitude difference'] =␣
 ↪abs(train_data_['lat']-train_data_['merch_lat'])
train_data_['longitude difference'] =␣
 ↪abs(train_data_['long']-train_data_['merch_long'])
```

3.4.2 Now, we have to find the displacement between the longitude difference and the latitude difference

3.4.3 We can use Pythagorean Theorem for finding this value

[25]:
```
# Difference b/w. consecutive longitudes and latitudes is 110km
train_data_['displacement'] = np.sqrt(pow((train_data_['latitude␣
 ↪difference']*110),2) + pow((train_data_['longitude difference']*110),2))
train_data_.sample(5)
```

[25]:

	Unnamed: 0	cc_num	merchant	category	amt	gender	city	state	zip	lat	long	city_pop	job	age	unix_time	merch_lat	merch_long	is_fraud	name	year	month	day	latitude difference	longitude difference	displacement
1242672	1242672	3504178999463051	fraud_Bartoletti and Sons	personal_care	4.03	M	Lima	OH	45801	40.7641	-84.0973	86954	Copywriter, advertising	31	1370126175	40.041198	-84.001290	0	Drew Garcia	2020	6	1	0.722902	0.096010	80.217474
1158236	1158236	4586810168620942	fraud_Zieme, Bode and Dooley	gas_transport	55.58	F	Edisto Island	SC	29438	32.5486	-80.3070	2408	Sales professional, IT	26	1367107482	31.891791	-81.134077	0	Michelle Gregory	2020	4	28	0.656809	0.827077	116.176584
806104	806104	4169759661243568	fraud_Hagenes, Hermann and Stroman	travel	1.02	F	Lawn	PA	17041	40.2236	-76.5380	213	Special educational needs teacher	51	1354749432	41.131434	-76.399526	0	Jamie Carr	2019	12	5	0.907834	0.138474	101.016757
62098	62098	180017442990269	fraud_Botsford PLC	home	20.34	F	Albany	NY	12222	42.6853	-73.8253	151022	Designer, textile	84	1328482099	42.188389	-74.719332	0	Michelle Anderson	2019	2	5	0.496911	0.894032	112.513050
814745	814745	676248282243	fraud_Champlin-Casper	home	42.17	F	Cape Coral	FL	33909	26.6939	-81.9452	156391	Higher education careers adviser	55	1354918765	26.241425	-82.862144	0	Amanda Wheeler	2019	12	7	0.452475	0.916944	112.475736

[26]:
```
train_data_['displacement'] = round(train_data_['displacement'])
train_data_ = train_data_.
 ↪drop(columns=['lat','long','merch_lat','merch_long','latitude␣
 ↪difference','longitude difference'], axis=1)
train_data_.head()
```

[26]:

	Unnamed: 0	cc_num	merchant	category	amt	gender	city	state	zip	city_pop	job	age	unix_time	is_fraud	name	year	month	day	latitude difference	longitude difference	displacement
0	0	2703186189652095	fraud_Rippin, Kub and Mann	misc_net	4.97	F	Moravian Falls	NC	28654	3495	Psychologist, counselling	32	1325376018	0	Jennifer Banks	2020	6	1	0.722902	0.096010	80.217474
1	1	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.23	F	Orient	WA	99160	149	Special educational needs teacher	49	1325376044	0	Stephanie Gill	2020	6	1	0.656809	0.827077	116.176584
2	2	38859492057661	fraud_Lind-Buckridge	entertainment	220.11	M	Malad City	ID	83252	4154	Nature conservation officer	43	1325376051	0	Edward Sanchez	2020	6	1	0.907834	0.138474	101.016757
3	3	3534093764340240	fraud_Kutch, Hermiston and Farrell	gas_transport	45.00	M	Boulder	MT	59632	1939	Patent attorney	57	1325376076	0	Jeremy White	2020	6	1	0.496911	0.894032	112.513050
4	4	375534208663984	fraud_Keeling-Crist	misc_pos	41.96	M	Doe Hill	VA	24433	99	Dance movement psychotherapist	38	1325376186	0	Brandon Conner	2020	6	1	0.452475	0.916944	112.475736

```
        category     amt gender            city state    zip  city_pop  \
0       misc_net    4.97      F  Moravian Falls    NC  28654      3495
1    grocery_pos  107.23      F          Orient    WA  99160       149
2  entertainment  220.11      M      Malad City    ID  83252      4154
3  gas_transport   45.00      M         Boulder    MT  59632      1939
4       misc_pos   41.96      M        Doe Hill    VA  24433        99

                               job  age   unix_time  is_fraud  \
0         Psychologist, counselling   35  1325376018         0
1  Special educational needs teacher   45  1325376044         0
2       Nature conservation officer   61  1325376051         0
3                   Patent attorney   56  1325376076         0
4    Dance movement psychotherapist   37  1325376186         0

             name  year  month  day  displacement
0  Jennifer Banks  2019      1    1          96.0
1  Stephanie Gill  2019      1    1          30.0
2  Edward Sanchez  2019      1    1         107.0
3    Jeremy White  2019      1    1         101.0
4    Tyler Garcia  2019      1    1          96.0
```

[27]:
```
## dropping the city,state and zip columns as now we have the distance required
train_data_ = train_data_.drop(['city','state','zip'], axis=1)
train_data_.head()
```

[27]:
```
   Unnamed: 0            cc_num                             merchant  \
0           0  2703186189652095          fraud_Rippin, Kub and Mann
1           1     630423337322     fraud_Heller, Gutmann and Zieme
2           2   38859492057661                fraud_Lind-Buckridge
3           3  3534093764340240  fraud_Kutch, Hermiston and Farrell
4           4   375534208663984                 fraud_Keeling-Crist

        category     amt gender  city_pop                               job  \
0       misc_net    4.97      F      3495          Psychologist, counselling
1    grocery_pos  107.23      F       149  Special educational needs teacher
2  entertainment  220.11      M      4154       Nature conservation officer
3  gas_transport   45.00      M      1939                   Patent attorney
4       misc_pos   41.96      M        99    Dance movement psychotherapist

   age   unix_time  is_fraud           name  year  month  day  displacement
0   35  1325376018         0  Jennifer Banks  2019      1    1          96.0
1   45  1325376044         0  Stephanie Gill  2019      1    1          30.0
2   61  1325376051         0  Edward Sanchez  2019      1    1         107.0
3   56  1325376076         0    Jeremy White  2019      1    1         101.0
4   37  1325376186         0    Tyler Garcia  2019      1    1          96.0
```

[28]:
```
## dropping the Unnamed column as it is repetition
train_data_ = train_data_.drop(['Unnamed: 0'], axis=1)
train_data_.head()
```

[28]:

	cc_num	merchant	category	amt	gender	city_pop	job	age	unix_time	is_fraud	name	year	month	day	displacement
0	2703186189652095	fraud_Rippin, Kub and Mann	misc_net	4.97	F	3495	Psychologist, counselling	35	1325376018	0	Jennifer Banks	2019	1	1	96.0
1	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.23	F	149	Special educational needs teacher	45	1325376044	0	Stephanie Gill	2019	1	1	30.0
2	38859492057661	fraud_Lind-Buckridge	entertainment	220.11	M	4154	Nature conservation officer	61	1325376051	0	Edward Sanchez	2019	1	1	107.0
3	3534093764340240	fraud_Kutch, Hermiston and Farrell	gas_transport	45.00	M	1939	Patent attorney	56	1325376076	0	Jeremy White	2019	1	1	101.0
4	375534208663984	fraud_Keeling-Crist	misc_pos	41.96	M	99	Dance movement psychotherapist	37	1325376186	0	Tyler Garcia	2019	1	1	96.0

3.5 UNIX TIME & CC_NUM

3.5.1 The number of seconds passed from the UNIX EPOCH i.e. 00:00:00 UTC on 1 January 1970

3.5.2 These two features will help us in differentiating between customers

3.5.3 So we can drop the name column which doesn't really contribute much

[29]:
```
train_data_ = train_data_.drop(['name'], axis=1)
train_data_.head()
```

[29]:

	cc_num	merchant	category	amt	gender	city_pop	job	age	unix_time	is_fraud	year	month	day	displacement
0	2703186189652095	fraud_Rippin, Kub and Mann	misc_net	4.97	F	3495	Psychologist, counselling	35	1325376018	0	2019	1	1	96.0
1	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.23	F	149	Special educational needs teacher	45	1325376044	0	2019	1	1	30.0
2	38859492057661	fraud_Lind-Buckridge	entertainment	220.11	M	4154	Nature conservation officer	61	1325376051	0	2019	1	1	107.0
3	3534093764340240	fraud_Kutch, Hermiston and Farrell	gas_transport	45.00	M	1939	Patent attorney	56	1325376076	0	2019	1	1	101.0
4	375534208663984	fraud_Keeling-Crist	misc_pos	41.96	M	99	Dance movement psychotherapist	37	1325376186	0	2019	1	1	96.0

```
3   45.00      M      1939                    Patent attorney   56
4   41.96      M        99     Dance movement psychotherapist   37

    unix_time  is_fraud  year  month  day  displacement
0  1325376018         0  2019      1    1          96.0
1  1325376044         0  2019      1    1          30.0
2  1325376051         0  2019      1    1         107.0
3  1325376076         0  2019      1    1         101.0
4  1325376186         0  2019      1    1          96.0
```

[30]:
```
### Now, we can categorize the displacement column
train_data_['displacement'].unique()
```

[30]:
```
array([ 96.,  30., 107., 101., 109., 130.,  13.,  33.,  74., 119.,  49.,
        78., 100.,  95.,  51.,  97.,  68.,  53., 122.,  26.,  94.,  84.,
        80.,  93.,  39.,  58.,  65.,  48.,  63.,  71.,  37.,  91.,  66.,
        99.,  77.,  36.,  87.,  56.,  34.,  24.,   9.,  88.,  57., 105.,
        15.,  69.,  59., 116., 126., 128.,  76., 102.,  81.,  46., 132.,
       112.,  90., 111., 103., 121.,  75., 108.,  82.,  52.,  98.,  70.,
       129., 124.,  43.,  62.,  92., 133., 106.,   5., 115.,  20.,  72.,
        27., 117.,  60.,  42., 139.,  16.,  10.,  79.,  47.,  44.,  40.,
       118., 113.,  86., 114., 110., 104., 149.,  67.,  89., 123.,  55.,
       143.,  73., 131.,  83.,  28., 136.,  85.,  38.,  54., 125., 135.,
        35.,  14.,  31.,  29., 138., 134., 147., 137.,  45.,  64.,  61.,
       152.,   8., 140.,  22., 141., 144., 146.,  12.,  18.,  25.,  23.,
        19., 120.,   7., 148.,  41., 145., 127.,   2., 153.,  21., 150.,
        50.,  32.,  11.,   3., 142., 151.,   0.,   6.,   4.,  17., 154.,
       155.,   1.])
```

[31]:
```
train_data_['displacement'].describe()
```

[31]:
```
count    1.296675e+06
mean     8.422295e+01
std      3.132332e+01
min      0.000000e+00
25%      6.200000e+01
50%      8.800000e+01
75%      1.080000e+02
max      1.550000e+02
Name: displacement, dtype: float64
```

3.5.4 Now, we can categorize the displacement column into 'near', 'far', 'very far' using the lower quartile and the upper quartile

[32]:
```
train_data_.loc[(train_data_['displacement']<62),['proximity']] = "near"
train_data_.loc[((train_data_['displacement']>=62) &␣
 ↪(train_data_['displacement']<108)), ['proximity']] = "far"
train_data_.loc[(train_data_['displacement']>=108), ['proximity']] = "very far"

train_data_.sample(10)
```

[32]:

	cc_num	merchant	category	amt	gender	city_pop	job	age	unix_time	is_fraud	year	month	day	displacement	proximity
111185	6011348830550197	fraud_O'Reilly, Mohr and Purdy	home	33.95	M	493806	Musician	43	1330862491	0	2019	3	4	135.0	very far
490762	2720012583106919	fraud_Altenwerth-Kilback	home	87.36	M	1126	Volunteer coordinator	43	1343936169	0	2019	8	2	97.0	far
671616	3585740823295298	fraud_Stark-Batz	entertainment	40.66	M	3202	Librarian, public	25	1350134376	0	2019	10	13	7.0	near
375474	4716561796955522	fraud_Romaguera, Wehner and Tromp	kids_pets	6.16	F	743	Water engineer	51	1340387500	0	2019	6	22	18.0	near
1028571	4988304376504	fraud_Balistreri-Nader	misc_pos	1.63	M	2258	Building surveyor	86	1362217779	0	2020	3	2	62.0	far
1185543	6593939509150446	fraud_Kerluke-Abshire	shopping_net	181.01	F	59705	Civil Service fast streamer	40	1368200408	0	2020	5	10	92.0	far
1149204	4355790796238264643	fraud_Bashirian Group	shopping_net	2.45	M	1656	Exhibition designer	54	1366791106	0	2020	4	24	81.0	far
175050	340953839692349	fraud_Hahn, Douglas and Schowalter	travel	4.21	M	134056	Doctor, hospital	43	1333293611	0	2019	4	1	121.0	very far
782419	2305336922781618	fraud_Hackett Group	travel	7812.76	M	1132	Probation officer	62	1354302183	0	2019	11	30	128.0	very far
1207471	373905417449658	fraud_Jast-McDermott	shopping_pos	185.68	F	2526	Phytotherapist	53	1368974705	0	2020	5	19	89.0	far

3.5.5 Similarly, categorizing the city population as “less dense”,“normal”,“over-crowded”

[33]:
```
train_data_['city_pop'].describe()
```

[33]:
```
count    1.296675e+06
mean     8.882444e+04
std      3.019564e+05
min      2.300000e+01
25%      7.430000e+02
50%      2.456000e+03
75%      2.032800e+04
max      2.906700e+06
Name: city_pop, dtype: float64
```

[34]:
```
train_data_.loc[(train_data_['city_pop']<10000), ['city_population_status']] =␣
 ↪"less dense"
train_data_.loc[((train_data_['city_pop']>1000) &␣
 ↪(train_data_['city_pop']<50000)),['city_population_status']] = "normal"
train_data_.loc[(train_data_['city_pop']>50000),['city_population_status']] =␣
 ↪"over crowded"

train_data_.sample(5)
```

[34]:

	cc_num	merchant	category	amt	gender	city_pop	job	age	unix_time	is_fraud	year	month	day	displacement	proximity	city_population_status
401863	2227671554547514	fraud_Kihn Inc	shopping_pos	445.41	F	172247	Geneticist, molecular	44	1341151194	0	2019	7	1	85.0	far	over crowded
1167885	3598215285024754	fraud_Haley, Jewess and Bechtelar	shopping_pos	1105.93	F	34496	Librarian, public	53	1367527358	0	2020	5	2	90.0	far	
1185147	6011109736646996	fraud_Eichmann-Kilback	home	245.06	F	186140	English as a second language teacher	40	1368189599	0	2020	5	10	111.0	very far	
652874	3559679414981506	fraud_Zboncak LLC	food_dining	21.98	F	8830	Theme park manager	32	1349445056	0	2019	10	5	116.0	very far	
724219	6011438889172900	fraud_Friesen-Ortiz	personal_care	21.42	F	5161	Electrical engineer	30	1352072624	0	2019	11	4	132.0	very far	

```
1167885                normal
1185147          over crowded
652874                 normal
724219                 normal
```

[35]:
```
train_data_ = train_data_.drop(['city_pop'], axis=1)
```

[36]:
```
train_data_.sample(20)
```

[36]:

	cc_num	merchant	category	amt	gender	job	age
1050686	4561546772499	fraud_Nitzsche, Kessler and Wolff	shopping_pos	1.33	M	Therapist, art	35
483406	38947654498698	fraud_Padberg-Sauer	home	20.17	F	Copywriter, advertising	44
1261898	3533800906065217	fraud_Larkin Ltd	kids_pets	84.29	F	Surveyor, minerals	83
1182991	2233882705243596	fraud_Zboncak Ltd	food_dining	3.56	F	Medical physicist	63
348992	4862293128558	fraud_Weimann, Kuhic and Beahan	shopping_pos	4.73	F	Pension scheme manager	67
85732	379141394109214	fraud_Collier LLC	home	44.15	M	Chief Strategy Officer	57
868611	3525590521269779	fraud_Christiansen-Gusikowski	misc_pos	47.48	M	Paramedic	39
1216259	344342339068828	fraud_Boyer PLC	shopping_net	3.72	F	Tax adviser	56
806715	4653879239169997	fraud_Harber Inc	gas_transport	70.62	F	Therapist, sports	24
595897	676372984911	fraud_Little, Gutmann and Lynch	shopping_net	7.86	F	Tourism officer	36
677135	372382441451095	fraud_Cormier LLC	health_fitness	8.79	M	Heritage manager	57
548074	38588538868506	fraud_Stiedemann Ltd	food_dining	24.02	F	Lexicographer	33
352544	180067784565096	fraud_Champlin, Rolfson and Connelly	travel	8.87	F	Applications developer	81
44105	341542810616333	fraud_Cole PLC	grocery_pos	139.87	M	Colour technologist	64
1220004	4715741951931168360	fraud_McDermott-Weimann	grocery_pos	114.19	M	Sub	82
1009163	343668971234893	fraud_O'Hara-Wilderman					
158213	3549202406645667	fraud_Connelly, Reichert and Fritsch					
1275434	675909898057	fraud_McDermott-Rice					
1161819	3514865930894695	fraud_Keeling-Crist					
596014	3585052663373890	fraud_Rau-Robel					

```
1009163     food_dining   60.00      F                     Chiropodist   74
158213    gas_transport   78.87      M                            Make   90
1275434        misc_pos    2.08      M  Television/film/video producer   59
1161819        misc_pos    9.25      M                 Naval architect   56
596014        kids_pets   46.81      M                  Science writer   60

          unix_time  is_fraud  year  month  day  displacement proximity  \
1050686  1363019110         0  2020      3   11          94.0       far
483406   1343653351         0  2019      7   30          55.0      near
1261898  1370718914         0  2020      6    8          74.0       far
1182991  1368104311         0  2020      5    9         111.0  very far
348992   1339559689         0  2019      6   13          21.0      near
85732    1329682120         0  2019      2   19         118.0  very far
868611   1355947985         0  2019     12   19          53.0      near
1216259  1369370453         0  2020      5   24          98.0       far
806715   1354767110         0  2019     12    6          97.0       far
595897   1347209878         0  2019      9    9          51.0      near
677135   1350306110         0  2019     10   15          93.0       far
548074   1345670865         0  2019      8   22         106.0       far
352544   1339693546         0  2019      6   14          44.0      near
44105    1327659138         0  2019      1   27         125.0  very far
1220004  1369464457         0  2020      5   25         143.0  very far
1009163  1361267620         0  2020      2   19          55.0      near
158213   1332657396         0  2019      3   25          10.0      near
1275434  1371170108         0  2020      6   14          69.0       far
1161819  1367230324         0  2020      4   29          95.0       far
596014   1347212714         0  2019      9    9          40.0      near

        city_population_status
1050686                 normal
483406              less dense
1261898             less dense
1182991                 normal
348992              less dense
85732                   normal
868611                  normal
1216259                 normal
806715                  normal
595897              less dense
677135                  normal
548074              less dense
352544                  normal
44105               less dense
1220004             less dense
1009163                 normal
158213                  normal
1275434                 normal
```

```
1161819             less dense
596014              less dense
```

[37]:
```
train_data_['is_fraud'].value_counts()
```

[37]:
```
0    1289169
1       7506
Name: is_fraud, dtype: int64
```

3.5.6 Now that we are done with cleaning and processing our training data, we'll clean the test dataset as well

[38]:
```
test_data_.head()
```

[38]:

	Unnamed: 0	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
0	0	2020-06-21 12:14:25	2291163933867244	fraud_Kirlin and Sons	personal_care	2.86	Jeff	Elliott	M	351 Darlene Green	Columbia	SC	29209	33.9659	-80.9355	333497	Mechanical engineer	1968-03-19	2da90c7d74bd46a0caf3777415b3ebd3	1371816865	33.986391	-81.200714
1	1	2020-06-21 12:14:33	3573030041201292	fraud_Sporer-Keebler	personal_care	29.84	Joanne	Williams	F	3638 Marsh Union	Altonah	UT	84002	40.3207	-110.4360	302	Sales professional, IT	1990-01-17	324cc204407e99f51b0d6ca0055005e7	1371816873	39.450498	-109.960431
2	2	2020-06-21 12:14:53	3598215285024754	fraud_Swaniawski, Nitzsche and Welch	health_fitness	41.28	Ashley	Lopez	F	9333 Valentine Point	Bellmore	NY	11710	40.6729	-73.5365	34496	Librarian, public	1970-10-21	c81755dbbbea9d5c77f094348a7579be	1371816893	40.495810	-74.196111
3	3	2020-06-21 12:15:15	3591919803438423	fraud_Haley Group	misc_pos	60.05	Brian	Williams	M	32941 Krystal Mill Apt. 552	Titusville	FL	32780	28.5697	-80.8191	54767	Set designer	1987-07-25	2159175b9efe66dc301f149d3d5abf8c	1371816915	28.812398	-80.883061
4	4	2020-06-21 12:15:17	3526826139003047	fraud_Johnston-Casper	travel	3.19	Nathan	Massey	M	5783 Evan Roads Apt. 465	Falmouth	MI	49632	44.2529	-85.0170	1126	Furniture designer	1955-07-06				

```
4  57ff021bd3f328f8738bb535c302a31b  1371816917  44.959148  -85.884734

   is_fraud
0         0
1         0
2         0
3         0
4         0
```

[39]:
```
test_data_ = test_data_.drop(['trans_num'], axis=1)
test_data_.head()
```

[39]:
```
   Unnamed: 0 trans_date_trans_time            cc_num  \
0           0   2020-06-21 12:14:25  2291163933867244
1           1   2020-06-21 12:14:33  3573030041201292
2           2   2020-06-21 12:14:53  3598215285024754
3           3   2020-06-21 12:15:15  3591919803438423
4           4   2020-06-21 12:15:17  3526826139003047

                             merchant        category    amt   first  \
0                fraud_Kirlin and Sons   personal_care   2.86    Jeff
1                 fraud_Sporer-Keebler   personal_care  29.84  Joanne
2  fraud_Swaniawski, Nitzsche and Welch  health_fitness  41.28  Ashley
3                    fraud_Haley Group        misc_pos  60.05   Brian
4                fraud_Johnston-Casper          travel   3.19  Nathan

      last gender                        street        city state    zip  \
0   Elliott      M            351 Darlene Green    Columbia    SC  29209
1  Williams      F             3638 Marsh Union     Altonah    UT  84002
2     Lopez      F         9333 Valentine Point    Bellmore    NY  11710
3  Williams      M  32941 Krystal Mill Apt. 552  Titusville    FL  32780
4    Massey      M     5783 Evan Roads Apt. 465    Falmouth    MI  49632

      lat      long  city_pop                     job         dob  \
0  33.9659  -80.9355    333497     Mechanical engineer  1968-03-19
1  40.3207 -110.4360       302  Sales professional, IT  1990-01-17
2  40.6729  -73.5365     34496       Librarian, public  1970-10-21
3  28.5697  -80.8191     54767            Set designer  1987-07-25
4  44.2529  -85.0170      1126      Furniture designer  1955-07-06

    unix_time  merch_lat  merch_long  is_fraud
0  1371816865  33.986391  -81.200714         0
1  1371816873  39.450498 -109.960431         0
2  1371816893  40.495810  -74.196111         0
3  1371816915  28.812398  -80.883061         0
4  1371816917  44.959148  -85.884734         0
```

[40]:
```
def find_age(date_of_birth):
    year = int(date_of_birth[:4])
    return (2023 - year)

test_data_['dob'] = test_data_['dob'].apply(find_age)

test_data_.rename(columns={'dob':'age'}, inplace=True)
test_data_['age'].unique()
```

[40]:
```
array([55, 33, 53, 36, 68, 32, 72, 51, 50, 67, 27, 47, 46, 86, 52, 35, 31,
       26, 38, 66, 75, 93, 49, 59, 65, 28, 43, 54, 48, 62, 80, 44, 37, 29,
       94, 89, 30, 24, 41, 25, 39, 45, 34, 56, 23, 74, 85, 57, 58, 78, 40,
       19, 71, 42, 73, 20, 69, 63, 82, 61, 77, 70, 99, 60, 22, 64, 97, 76,
       87, 88, 84, 81, 18, 79, 92, 96, 83, 95, 90, 91, 21], dtype=int64)
```

[41]:
```
test_data_ = test_data_.drop(['first','last','city','street','zip'],axis=1)
test_data_.head()
```

[41]:

	Unnamed: 0	trans_date_trans_time	cc_num	merchant	category	amt	gender	state	lat	long	city_pop	job	age	unix_time	merch_lat	merch_long	is_fraud
0	0	2020-06-21 12:14:25	2291163933867244	fraud_Kirlin and Sons	personal_care	2.86	M	SC	33.9659	-80.9355	333497	Mechanical engineer	55	1371816865	33.986391	-81.200714	0
1	1	2020-06-21 12:14:33	3573030041201292	fraud_Sporer-Keebler	personal_care	29.84	F	UT	40.3207	-110.4360	302	Sales professional, IT	33	1371816873	39.450498	-109.960431	0
2	2	2020-06-21 12:14:53	3598215285024754	fraud_Swaniawski, Nitzsche and Welch	health_fitness	41.28	F	NY	40.6729	-73.5365	34496	Librarian, public	53	1371816893	40.495810	-74.196111	0
3	3	2020-06-21 12:15:15	3591919803438423	fraud_Haley Group	misc_pos	60.05	M	FL	28.5697	-80.8191	54767	Set designer	36	1371816915	28.812398	-80.883061	0
4	4	2020-06-21 12:15:17	3526826139003047	fraud_Johnston-Casper	travel	3.19	M	MI	44.2529	-85.0170	1126	Furniture designer	68	1371816917	44.959148	-85.884734	0

[42]:
```
test_data_['trans_date_trans_time'] = pd.
↪to_datetime(test_data_['trans_date_trans_time'])
```

[43]:
```
test_data_['year'] = test_data_['trans_date_trans_time'].dt.year
test_data_['month'] = test_data_['trans_date_trans_time'].dt.month
test_data_['day'] = test_data_['trans_date_trans_time'].dt.day

test_data_.drop('trans_date_trans_time', axis=1, inplace=True)
test_data_.sample(5)
```

[43]:

	Unnamed: 0	cc_num	merchant	category	amt	gender	state	lat	long	city_pop	job	age	unix_time	merch_lat	merch_long	is_fraud	year	month	day
5436	5436	3568255211412877	fraud_Kuhic LLC	shopping_net	8.46	M	IN	41.2249	-85.0301	5341	Biomedical engineer	29	1371933768	42.152337	-86.013584	0	2020	6	22
189119	189119	213193596103206	fraud_Schaefer, Maggio and Daugherty	gas_transport	66.85	M	MI	45.7549	-84.4470	95	Electrical engineer	29	1377495069	45.375669	-85.008559	0	2020	8	26
201880	201880	343746486082492	fraud_Rutherford-Mertz	grocery_pos	55.94	M	MI	44.8605	-85.8138	3096	Social research officer, government	48	1377943008	44.799760	-86.278329	0	2020	8	31
267479	267479	4599285557366057	fraud_Streich, Hansen and Veum	gas_transport	73.82	F	KS	38.0261	-97.6666	1689	Gaffer	26	1380361302	37.767936	-97.904970	0	2020	9	28
307692	307692	4555104582813474	fraud_Torphy-Kertzmann	health_fitness	77.46	M	PA	40.9540	-76.1747	143	Health and safety adviser	41	1381873990	40.771788	-75.750541	0	2020	10	15

[44]:
```
test_data_['latitude difference'] =␣
↪abs(test_data_['lat']-test_data_['merch_lat'])
test_data_['longitude difference'] =␣
↪abs(test_data_['long']-test_data_['merch_long'])
```

[45]:
```
test_data_['displacement'] = np.sqrt(pow((test_data_['latitude␣
↪difference']*110),2) + pow((test_data_['longitude difference']*110),2))
test_data_.sample(5)
```

[45]:
```
        Unnamed: 0               cc_num  \
344380      344380       38530489946071
549728      549728  4497913965512794052
58388        58388     6011492816282597
312980      312980     2222001896600109
416976      416976     4997733566924489

                                      merchant        category    amt gender  \
344380  fraud_Kerluke, Considine and Macejkovic        misc_net   4.87      F
549728                          fraud_Kuhn LLC        misc_net   2.89      M
58388                          fraud_Terry Ltd            home  50.68      M
312980                      fraud_Hirthe-Beier  health_fitness  79.59      F
416976                       fraud_Schmitt Inc   gas_transport  44.12      F

       state      lat     long  city_pop                  job  age  \
344380    SC  34.9572 -81.9916       530  Animal technologist   34
549728    NC  34.6902 -79.1834     14783      Hospital doctor   68
58388     AR  33.3398 -92.7442      2501    Financial adviser   37
312980    IL  38.9318 -89.9618      2401    Buyer, industrial   51
416976    MN  44.9913 -92.9487    753116    Fisheries officer   52

         unix_time  merch_lat  merch_long  is_fraud  year  month  day  \
344380  1383300108  34.129963  -81.503498         0  2020     11    1
549728  1388374614  34.485316  -79.593209         0  2020     12   30
58388   1373574923  32.936537  -92.713484         0  2020      7   11
312980  1382104612  38.238683  -89.621818         0  2020     10   18
416976  1385870097  44.340464  -93.452439         0  2020     12    1

        latitude difference  longitude difference  displacement
344380             0.827237              0.488102    105.655250
549728             0.204884              0.409809     50.398835
58388              0.403263              0.030716     44.487422
312980             0.693117              0.339982     84.921064
416976             0.650836              0.503739     90.530767
```

[46]:
```
test_data_['displacement'] = round(test_data_['displacement'])
test_data_ = test_data_.
 ↪drop(columns=['lat','long','merch_lat','merch_long','latitude␣
 ↪difference','longitude difference'], axis=1)
test_data_.head()
```

[46]:
```
   Unnamed: 0           cc_num                              merchant  \
0           0  2291163933867244                 fraud_Kirlin and Sons
1           1  3573030041201292                  fraud_Sporer-Keebler
2           2  3598215285024754  fraud_Swaniawski, Nitzsche and Welch
3           3  3591919803438423                     fraud_Haley Group
4           4  3526826139003047                 fraud_Johnston-Casper
```

```
          category    amt gender state  city_pop                     job  age  \
0   personal_care   2.86      M    SC    333497     Mechanical engineer   55
1   personal_care  29.84      F    UT       302  Sales professional, IT   33
2  health_fitness  41.28      F    NY     34496       Librarian, public   53
3        misc_pos  60.05      M    FL     54767            Set designer   36
4          travel   3.19      M    MI      1126      Furniture designer   68

    unix_time  is_fraud  year  month  day  displacement
0  1371816865         0  2020      6   21          29.0
1  1371816873         0  2020      6   21         109.0
2  1371816893         0  2020      6   21          75.0
3  1371816915         0  2020      6   21          28.0
4  1371816917         0  2020      6   21         123.0
```

[47]:
```
test_data_.loc[(test_data_['displacement']<62),['proximity']] = "near"
test_data_.loc[((test_data_['displacement']>=62) &␣
 ↪(test_data_['displacement']<108)), ['proximity']] = "far"
test_data_.loc[(test_data_['displacement']>=108), ['proximity']] = "very far"

test_data_.sample(10)
```

[47]:
```
        Unnamed: 0               cc_num                             merchant  \
434379      434379      36722699017270                fraud_Wuckert-Goldner
32289        32289       4186530744674                  fraud_Pouros-Conroy
20417        20417  4792627764422477317                   fraud_Homenick LLC
175201      175201     4826655832045236      fraud_Dare, Fritsch and Zboncak
549854      549854      345832460465610                  fraud_Hills-Witting
187419      187419     3564839259330465  fraud_Douglas, DuBuque and McKenzie
382443      382443      377113842678100    fraud_Bins, Balistreri and Beatty
467705      467705  4956828990005111019              fraud_Cummerata-Hilpert
406522      406522     6592861994408652               fraud_Heaney-Marquardt
31537        31537       30446018552504    fraud_Hyatt, Russel and Gleichner

             category      amt gender state  city_pop  \
434379           home    43.75      F    AZ      2872
32289     shopping_pos     3.29      F    SC      4913
20417    personal_care     2.63      F    KS      5760
175201  health_fitness    30.42      F    MA      1850
549854    shopping_net   106.57      M    VA    104396
187419          travel  5484.73      M    CA       198
382443    shopping_pos     2.41      M    NJ       804
467705   entertainment     2.65      M    NJ    124967
406522   entertainment    32.24      F    NY      1166
31537   health_fitness    50.11      F    OK       608

                                                job  age   unix_time  \
```

```
434379                              Petroleum engineer   36  1386255306
32289                                 Technical brewer   57  1372742917
20417                           Chief Executive Officer   45  1372430890
175201                        Counselling psychologist   60  1377123865
549854                              Production engineer   73  1388379589
187419  Armed forces training and education officer   64  1377450270
382443                           Insurance risk surveyor   58  1384681514
467705                            Operational researcher   43  1386885881
406522                              Colour technologist   40  1385667078
31537                               Seismic interpreter   30  1372708125

        is_fraud  year  month  day  displacement proximity
434379         0  2020     12    5          86.0       far
32289          0  2020      7    2          26.0      near
20417          0  2020      6   28          54.0      near
175201         0  2020      8   21         100.0       far
549854         0  2020     12   30          74.0       far
187419         0  2020      8   25          95.0       far
382443         0  2020     11   17          94.0       far
467705         0  2020     12   12          93.0       far
406522         0  2020     11   28         127.0  very far
31537          0  2020      7    1          50.0      near
```

[48]:
```
train_data_ = train_data_.drop(['displacement'], axis=1)
test_data_ = test_data_.drop(['displacement'], axis=1)
```

[49]:
```
test_data_.loc[(test_data_['city_pop']<10000), ['city_population_status']] =␣
 ↪"less dense"
test_data_.loc[((test_data_['city_pop']>1000) &␣
 ↪(test_data_['city_pop']<50000)),['city_population_status']] = "normal"
test_data_.loc[(test_data_['city_pop']>50000),['city_population_status']] =␣
 ↪"over crowded"

test_data_.sample(5)
```

[49]:
```
        Unnamed: 0            cc_num                            merchant  \
134727      134727     4783226709001                       fraud_Jacobi Inc
90793        90793  3528407217576457  fraud_Schmeler, Bashirian and Price
106905      106905  2719496466799416                    fraud_Pacocha-Bauch
418871      418871   213199865312311                fraud_Champlin and Sons
163679      163679   180067784565096                      fraud_Kilback LLC

              category    amt gender state  city_pop                     job  \
134727  health_fitness  38.42      F    KS       320       Film/video editor
90793     shopping_net  17.21      F    TN     87124           Warden/ranger
106905    shopping_pos   7.17      F    SD      8007     Hospital pharmacist
418871            home  43.44      M    GA        74     Pensions consultant
```

```
163679     food_dining  39.43      F    WI      2328  Applications developer

        age   unix_time  is_fraud  year  month  day proximity  \
134727   62  1375887817         0  2020      8    7       far
90793    36  1374498595         0  2020      7   22       far
106905   83  1374999030         0  2020      7   28       far
418871   79  1385902414         0  2020     12    1  very far
163679   81  1376758030         0  2020      8   17       far

       city_population_status
134727             less dense
90793            over crowded
106905                normal
418871             less dense
163679                normal
```

[50]:
```
test_data_ = test_data_.drop(['city_pop'], axis=1)
```

[51]:
```
test_data_ = test_data_.drop(['state'], axis=1)
test_data_.sample(10)
```

[51]:
```
        Unnamed: 0              cc_num  \
43305        43305     2450829271795901
495880      495880  4220495028289516646
453364      453364     3557442674264531
161345      161345  4488258847705225890
244313      244313  4457732997086323466
224809      224809      378904938837132
175940      175940     4859525594182537
135366      135366      180040027502291
348337      348337     6538441737335434
104318      104318         676369110710

                                      merchant       category     amt gender  \
43305                      fraud_Dach-Borer    grocery_net   35.10      M
495880                       fraud_Auer LLC  personal_care   35.51      F
453364  fraud_Runolfsdottir, Mueller and Hand  entertainment   80.20      F
161345                   fraud_Kutch and Sons    grocery_pos  104.47      F
244313                      fraud_Kozey-Boehm   shopping_net  101.36      M
224809                fraud_Rutherford-Mertz    grocery_pos   84.30      F
175940                     fraud_Schmitt Ltd       misc_net   19.89      F
135366        fraud_O'Reilly, Mohr and Purdy           home    4.93      F
348337                       fraud_Lynch Ltd   shopping_pos  129.53      F
104318    fraud_Daugherty, Pouros and Beahan   shopping_pos    1.60      M

                             job  age   unix_time  is_fraud  year  month  \
43305       Hydrographic surveyor   36  1373103268         0  2020      7
```

```
495880          Wellsite geologist   57  1387405802         0  2020     12
453364  Lecturer, higher education   19  1386538234         0  2020     12
161345           Chemical engineer   94  1376701930         0  2020      8
244313         Immigration officer   36  1379478297         0  2020      9
224809                   Barrister   37  1378701498         0  2020      9
175940      Psychotherapist, child   33  1377157191         0  2020      8
135366      Audiological scientist   66  1375907364         0  2020      8
348337              Energy manager   26  1383413354         0  2020     11
104318        Engineer, automotive   49  1374942700         0  2020      7

        day proximity city_population_status
43305     6      near                normal
495880   18      near          over crowded
453364    8      near            less dense
161345   17       far          over crowded
244313   18      near            less dense
224809    9  very far                normal
175940   22       far                normal
135366    7       far          over crowded
348337    2      near            less dense
104318   27       far          over crowded
```

[52]:
```
train_data_.sample(10)
```

[52]:
```
                    cc_num                                merchant  \
221938   4715741951931168360     fraud_Bins, Balistreri and Beatty
1289961       4862293128558   fraud_Denesik, Powlowski and Pouros
146100     4586810168620942     fraud_Turcotte, McKenzie and Koss
1163753  4956828990005111019  fraud_Moore, Williamson and Emmerich
621958     2222157926772399       fraud_Zboncak, Rowe and Murazik
993200        4562827002127                 fraud_Denesik and Sons
21971       180048185037117                   fraud_Gislason Group
743895         567868110212                       fraud_Skiles LLC
490341     3576021480694169                   fraud_Crona and Sons
707413     4514242065619750   fraud_Baumbach, Strosin and Nicolas

              category    amt gender                             job  age  \
221938    shopping_pos   4.18      M                             Sub   82
1289961           home  20.81      F          Pension scheme manager   67
146100   entertainment  91.49      F          Sales professional, IT   26
1163753           home  46.10      M          Operational researcher   43
621958    shopping_net   9.43      F  Investment banker, operational   36
993200    shopping_pos   4.54      M                   Media planner   52
21971         misc_pos  36.43      F          Leisure centre manager   49
743895            home  42.11      F         Copywriter, advertising   39
490341       kids_pets  21.86      F        Secondary school teacher   19
707413    shopping_pos   2.94      F                 Pilot, airline   35
```

```
          unix_time  is_fraud  year  month  day proximity  \
221938   1335090737         0  2019      4   22       far
1289961  1371580347         0  2020      6   18       far
146100   1332197953         0  2019      3   19       far
1163753  1367332757         0  2020      4   30  very far
621958   1348259638         0  2019      9   21      near
993200   1360451015         0  2020      2    9       far
21971    1326510912         0  2019      1   14  very far
743895   1352898627         0  2019     11   14       far
490341   1343923548         0  2019      8    2       far
707413   1351447662         0  2019     10   28       far

        city_population_status
221938              less dense
1289961             less dense
146100                  normal
1163753           over crowded
621958                  normal
993200                  normal
21971             over crowded
743895            over crowded
490341            over crowded
707413                  normal
```

3.5.7 Now, the two sets look uniform

3.5.8 We are ready to train our models and predict

3.5.9 We'll remove some columns to avoid overfitting which prodcues high variance

[53]:
```
temp_data = train_data_.drop(['job','merchant'], axis=1)
data_train_ = pd.get_dummies(temp_data, columns =␣
 ↪['gender','proximity','city_population_status'], drop_first=True)

temp_data = test_data_.drop(['job','merchant'], axis=1)
data_test_ = pd.get_dummies(temp_data, columns =␣
 ↪['gender','proximity','city_population_status'], drop_first=True)

data_test_.head()
```

[53]:
```
   Unnamed: 0            cc_num        category    amt  age   unix_time  \
0           0  2291163933867244   personal_care   2.86   55  1371816865
1           1  3573030041201292   personal_care  29.84   33  1371816873
2           2  3598215285024754  health_fitness  41.28   53  1371816893
3           3  3591919803438423        misc_pos  60.05   36  1371816915
4           4  3526826139003047          travel   3.19   68  1371816917
```

```
   is_fraud  year  month  day  gender_M  proximity_near  proximity_very far  \
0         0  2020      6   21         1               1                   0
1         0  2020      6   21         0               0                   1
2         0  2020      6   21         0               0                   0
3         0  2020      6   21         1               1                   0
4         0  2020      6   21         1               0                   1

   city_population_status_normal  city_population_status_over crowded
0                              0                                   1
1                              0                                   0
2                              1                                   0
3                              0                                   1
4                              1                                   0
```

[54]:
```
data_train_.head()
```

[54]:
```
             cc_num       category     amt  age   unix_time  is_fraud  year  \
0  2703186189652095       misc_net    4.97   35  1325376018         0  2019
1     630423337322    grocery_pos  107.23   45  1325376044         0  2019
2   38859492057661  entertainment  220.11   61  1325376051         0  2019
3  3534093764340240  gas_transport   45.00   56  1325376076         0  2019
4   375534208663984       misc_pos   41.96   37  1325376186         0  2019

   month  day  gender_M  proximity_near  proximity_very far  \
0      1    1         0               0                   0
1      1    1         0               1                   0
2      1    1         1               0                   0
3      1    1         1               0                   0
4      1    1         1               0                   0

   city_population_status_normal  city_population_status_over crowded
0                              1                                   0
1                              0                                   0
2                              1                                   0
3                              1                                   0
4                              0                                   0
```

[55]:
```
data_train_ = pd.get_dummies(data_train_, columns = ['category'], drop_first =␣
↪True)
data_test_ = pd.get_dummies(data_test_, columns = ['category'], drop_first =␣
↪True)

data_train_.head()
```

[55]:
```
             cc_num     amt  age   unix_time  is_fraud  year  month  day  \
0  2703186189652095    4.97   35  1325376018         0  2019      1    1
```

```
1      630423337322  107.23   45  1325376044         0  2019      1    1
2    38859492057661  220.11   61  1325376051         0  2019      1    1
3  3534093764340240   45.00   56  1325376076         0  2019      1    1
4   375534208663984   41.96   37  1325376186         0  2019      1    1

   gender_M  proximity_near  proximity_very far  \
0         0               0                   0
1         0               1                   0
2         1               0                   0
3         1               0                   0
4         1               0                   0

   city_population_status_normal  city_population_status_over crowded  \
0                              1                                    0
1                              0                                    0
2                              1                                    0
3                              1                                    0
4                              0                                    0

   category_food_dining  category_gas_transport  category_grocery_net  \
0                     0                       0                     0
1                     0                       0                     0
2                     0                       0                     0
3                     0                       1                     0
4                     0                       0                     0

   category_grocery_pos  category_health_fitness  category_home  \
0                     0                        0              0
1                     1                        0              0
2                     0                        0              0
3                     0                        0              0
4                     0                        0              0

   category_kids_pets  category_misc_net  category_misc_pos  \
0                   0                  1                  0
1                   0                  0                  0
2                   0                  0                  0
3                   0                  0                  0
4                   0                  0                  1

   category_personal_care  category_shopping_net  category_shopping_pos  \
0                       0                      0                      0
1                       0                      0                      0
2                       0                      0                      0
3                       0                      0                      0
4                       0                      0                      0
```

```
   category_travel
0                0
1                0
2                0
3                0
4                0
```

[56]:
```
X = data_train_.drop(['is_fraud'], axis=1)
y = data_train_['is_fraud']

X_test = data_test_.drop(['is_fraud','Unnamed: 0'], axis=1)
y_test = data_test_['is_fraud']
```

3.6 Feature Scaling

[57]:
```
scaler = MinMaxScaler()
```

[58]:
```
data_train_ = scaler.fit_transform(data_train_)
data_test_ = scaler.fit_transform(data_test_)
```

3.7 Model Training

4 Logistic Regression

[59]:
```
logreg = LogisticRegression(solver='liblinear', random_state=0)

logreg.fit(X,y)
```

[59]:
```
LogisticRegression(random_state=0, solver='liblinear')
```

4.1 Predicting Results

[60]:
```
y_pred_test = logreg.predict(X_test)
y_pred_test
```

[60]:
```
array([0, 0, 0, …, 0, 0, 0], dtype=int64)
```

[61]:
```
X_test
```

[61]:
```
                  cc_num     amt  age   unix_time  year  month  day  gender_M  \
0       2291163933867244    2.86   55  1371816865  2020      6   21         1
1       3573030041201292   29.84   33  1371816873  2020      6   21         0
2       3598215285024754   41.28   53  1371816893  2020      6   21         0
3       3591919803438423   60.05   36  1371816915  2020      6   21         1
4       3526826139003047    3.19   68  1371816917  2020      6   21         1
…                    …       …   …           …     …      …   …         …
```

```
555714    30560609640617   43.77   57  1388534347  2020     12   31         1
555715  3556613125071656  111.84   24  1388534349  2020     12   31         1
555716  6011724471098086   86.88   42  1388534355  2020     12   31         0
555717     4079773899158    7.99   58  1388534364  2020     12   31         1
555718  4170689372027579   38.13   30  1388534374  2020     12   31         1

        proximity_near  proximity_very far  city_population_status_normal  \
0                    1                   0                              0
1                    0                   1                              0
2                    0                   0                              1
3                    1                   0                              0
4                    0                   1                              1
…                  …                   …                              …
555714               0                   0                              0
555715               0                   0                              1
555716               0                   0                              1
555717               0                   0                              0
555718               0                   0                              0

        city_population_status_over crowded  category_food_dining  \
0                                         1                    0
1                                         0                    0
2                                         0                    0
3                                         1                    0
4                                         0                    0
…                                       …                    …
555714                                    0                    0
555715                                    0                    0
555716                                    0                    0
555717                                    0                    0
555718                                    1                    0

        category_gas_transport  category_grocery_net  category_grocery_pos  \
0                            0                     0                     0
1                            0                     0                     0
2                            0                     0                     0
3                            0                     0                     0
4                            0                     0                     0
…                          …                     …                     …
555714                       0                     0                     0
555715                       0                     0                     0
555716                       0                     0                     0
555717                       0                     0                     0
555718                       0                     0                     0

        category_health_fitness  category_home  category_kids_pets  \
0                             0              0                   0
```

```
1                              0                0                     0
2                              1                0                     0
3                              0                0                     0
4                              0                0                     0
…                            …                …                     …
555714                         1                0                     0
555715                         0                0                     1
555716                         0                0                     1
555717                         0                0                     0
555718                         0                0                     0

        category_misc_net  category_misc_pos  category_personal_care  \
0                       0                  0                       1
1                       0                  0                       1
2                       0                  0                       0
3                       0                  1                       0
4                       0                  0                       0
…                     …                  …                       …
555714                  0                  0                       0
555715                  0                  0                       0
555716                  0                  0                       0
555717                  0                  0                       0
555718                  0                  0                       0

        category_shopping_net  category_shopping_pos  category_travel
0                           0                      0                0
1                           0                      0                0
2                           0                      0                0
3                           0                      0                0
4                           0                      0                1
…                         …                      …              …
555714                      0                      0                0
555715                      0                      0                0
555716                      0                      0                0
555717                      0                      0                1
555718                      0                      0                0

[555719 rows x 25 columns]
```

4.2 Checking Accuracy Scores

[63]:
```
print('The accuracy score is: {0:0.4f}'.
  ↪format(accuracy_score(y_test,y_pred_test)))
```

```
The accuracy score is: 0.9961
```

4.3 99.61% is great accuracy, which means our model is performing wonderful!!

4.4 Confusion Matrix

[64]:
```
cm = confusion_matrix(y_test,y_pred_test)

cm
```

[64]:
```
array([[553574,      0],
       [  2145,      0]], dtype=int64)
```

[65]:
```
cr = classification_report(y_test,y_pred_test)

print(cr)
```

```
C:\MY FOLDERS\ml\lib\site-packages\sklearn\metrics\_classification.py:1344:
UndefinedMetricWarning: Precision and F-score are ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to
control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\MY FOLDERS\ml\lib\site-packages\sklearn\metrics\_classification.py:1344:
UndefinedMetricWarning: Precision and F-score are ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to
control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    553574
           1       0.00      0.00      0.00      2145

    accuracy                           1.00    555719
   macro avg       0.50      0.50      0.50    555719
weighted avg       0.99      1.00      0.99    555719


C:\MY FOLDERS\ml\lib\site-packages\sklearn\metrics\_classification.py:1344:
UndefinedMetricWarning: Precision and F-score are ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to
control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

5 Decision Tree

5.0.1 First, we'll create a decision tree with a GINI index

[67]:
```
dec = DecisionTreeClassifier(criterion='gini', max_depth=3, random_state=0)
dec.fit(X,y)
```

[67]:
```
DecisionTreeClassifier(max_depth=3, random_state=0)
```

[68]:
```
y_pred_test_gini = dec.predict(X_test)
y_pred_test_gini
```

[68]:
```
array([0, 0, 0, …, 0, 0, 0], dtype=int64)
```

[69]:
```
print('The accuracy score is: {0:0.4f}'.
  ↪format(accuracy_score(y_test,y_pred_test_gini)))
```

```
The accuracy score is: 0.9965
```

5.1 99.65%, just a tad bit more accurate than Logistic Regression

[70]:
```
confusion_matrix(y_test,y_pred_test_gini)
```

[70]:
```
array([[552367,   1207],
       [   738,   1407]], dtype=int64)
```

[72]:
```
print(classification_report(y_test,y_pred_test_gini))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    553574
           1       0.54      0.66      0.59      2145

    accuracy                           1.00    555719
   macro avg       0.77      0.83      0.79    555719
weighted avg       1.00      1.00      1.00    555719
```

6 Random Forest

[75]:
```
randf = RandomForestClassifier(n_estimators=10,random_state=0)

randf.fit(X,y)
```

[75]:
```
RandomForestClassifier(n_estimators=10, random_state=0)
```

[76]:
```
y_pred_randf = randf.predict(X_test)
```

[77]:
```
print('The accuracy score is: {0:0.4f}'.
  ↪format(accuracy_score(y_test,y_pred_randf)))
```

```
The accuracy score is: 0.9976
```

6.1 99.76% accuracy better than the other two algorithms

[ ]:
```

```