Mishra540_Project_Milestone_04

July 30, 2023

[1]:
```
# DSC540,Summer 2023 - T302 Data Preparation(2237-1)
# Assignment: Project Milestone 04
# Author by:  Debabrata Mishra
# Date: 2023-07-30

# Topic - Credit Card Transactional & Demographic Data
```

1 Milestone 4 Tasks

Connecting to an API/Pulling in the Data and Cleaning/Formatting Perform at least 5 data transformation and/or cleansing steps to your API data. The below examples are not required - they are just potential transformations you could do. If your data doesn't work for these scenarios, complete different transformations. You can do the same transformation multiple times if you needed to clean your data. The goal is a clean dataset at the end of the milestone.

```
Replace Headers
Format data into a more readable format
Identify outliers and bad data
Find duplicates
Fix casing or inconsistent values
Conduct Fuzzy Matching
```

Make sure you clearly label each transformation step (Step #1, Step #2, etc.) in your code and describe what it is doing in 1-2 sentences. You can submit a Jupyter Notebook or a PDF of your code. If you submit a .py file you need to also include a PDF or attachment of your results.

2 Cleaning and Formatting API Source Data

2.1 API Data

Descrition:

Typically, a reverse geocoding service is used to convert geographic coordinates (latitude and longitude) into a human-readable address or location. Popular geocoding services like Geocode, Google Maps API, Bing Maps API, or OpenStreetMap's Nominatim provide reverse geocoding capabilities

Link: https://geocode.maps.co/reverse?

Ethical Implications: Reading geolocation data through latitude and longitude using a geolocation API can raise various ethical implications, particularly when it comes to data privacy and secu-

rity.Geolocation data is highly sensitive as it reveals the precise location of individuals. Collecting and using such data without consent or a legitimate purpose can infringe on individuals' privacy rights. When collecting geolocation data, it is crucial to obtain informed consent from the individuals whose data is being collected. Users should be fully aware of why their location data is being collected, how it will be used, and how long it will be retained. Geolocation data, if mishandled, can lead to severe consequences for individuals. It is essential to implement robust security measures to protect this data from unauthorized access, hacking, or data breaches. The data collected through geolocation APIs should be used only for the specific purpose for which consent was obtained. Using the data for other purposes without obtaining further consent could be unethical. If possible, consider aggregating or anonymizing geolocation data to protect the identities of individuals. This helps prevent the identification of specific individuals while still enabling useful analysis. Clarify who owns the geolocation data collected through the API. Ensuring transparency about data ownership is essential, especially if the data is shared with third parties. We need to ensure that the geolocation data is accurate and reliable. Inaccurate data could lead to incorrect assumptions or decisions, potentially harming individuals. Be aware of and comply with relevant data protection and privacy regulations, such as the General Data Protection Regulation (GDPR) in Europe or the California Consumer Privacy Act (CCPA) in the United States.

As part of Project Milestone 4: I have considered below transformations.

- CReate a sample set from the final dataset from Project Milestone 01. This set will contain all fraud transactions and 10K+ random non fraud transaction.
- Read the API using the Merchant latitude and lognitude to get the address of merchant location. While calling the API , used the API error and No data found scenarios
- Format the data address and get the Key field like State , Country and Postal Code.
- Created a amount range which will be utilized to identify the testing txn and BOT attacks
- Create a ZIP/Postalcode match and state match flag between cardholder demographic vs Merchant demographic
- Identify outliers using IQR
- check for duplicate and drop those duplicates (if any)
- Missing value check

[2]:
```
#Load the Necessary Libraries

import pandas as pd
import numpy as np
import requests
import xlrd
from bs4 import BeautifulSoup
import numpy as np
import datapackage
import matplotlib.pyplot as plt
import seaborn as sns
import time
import concurrent.futures
```

```
import json
```

[3]:
```
# User defined function to make API Call and get data

def get_address_from_coordinates(lat, lng):
    url = f"https://geocode.maps.co/reverse?lat={latitude}&lon={longitude}"
    response = requests.get(url)
    status = response.status_code
    if status == 200:
        response_json = response.json()
        if "error" in response_json:
            add_data = "NAN - NO DATA"
        else:
            add_data = response_json["address"]
    else:
        add_data = "NAN - API ERROR"
    return add_data
```

[4]:
```
# Validation of API call function.
latitude = 39.202859
longitude = -78.247865
address = get_address_from_coordinates(latitude, longitude)
print(address)
```

```
{'road': 'Dicks Hollow Road', 'hamlet': 'Zeiger', 'county': 'Frederick County',
'state': 'Virginia', 'postcode': '22603', 'country': 'United States',
'country_code': 'us'}
```

[5]:
```
# Read the final data set creatd from the CSV/Flat file ( Project Milestone 02)

csvfile_final_txn_data = pd.read_csv('final_txn_data.csv', sep=",")
csvfile_final_txn_data
```

[5]:

	row_id	trans_date_trans_time	merch_name	\
0	0	2019-01-01 00:00:18	fraud_Rippin, Kub and Mann	
1	1	2019-01-01 00:00:44	fraud_Heller, Gutmann and Zieme	
2	2	2019-01-01 00:00:51	fraud_Lind-Buckridge	
3	3	2019-01-01 00:01:16	fraud_Kutch, Hermiston and Farrell	
4	4	2019-01-01 00:03:06	fraud_Keeling-Crist	
…	…	…	…	
1283932	1296670	2020-06-21 12:12:08	fraud_Reichel Inc	
1283933	1296671	2020-06-21 12:12:19	fraud_Abernathy and Sons	
1283934	1296672	2020-06-21 12:12:32	fraud_Stiedemann Ltd	
1283935	1296673	2020-06-21 12:13:36	fraud_Reinger, Weissnat and Strosin	
1283936	1296674	2020-06-21 12:13:37	fraud_Langosh, Wintheiser and Hyatt	

	category	amount	first	last	gender	\
0	misc_net	4.97	Jennifer	Banks	F	

```
1          grocery_pos  107.23    Stephanie       Gill      F
2        entertainment  220.11       Edward    Sanchez      M
3        gas_transport   45.00       Jeremy      White      M
4             misc_pos   41.96        Tyler     Garcia      M
…                  …     …           …        …    …
1283932  entertainment   15.56         Erik  Patterson      M
1283933    food_dining   51.70      Jeffrey      White      M
1283934    food_dining  105.93  Christopher  Castaneda      M
1283935    food_dining   74.90       Joseph     Murray      M
1283936    food_dining    4.30      Jeffrey      Smith      M

                              street                     city  …  year  \
0                     561 Perry Cove            Moravian Falls  …  2019
1          43039 Riley Greens Suite 393                    Orient  …  2019
2              594 White Dale Suite 530                Malad City  …  2019
3           9443 Cynthia Court Apt. 038                   Boulder  …  2019
4                     408 Bradley Rest                  Doe Hill  …  2019
…                                  …                     …  …   …
1283932       162 Jessica Row Apt. 072                     Hatch  …  2020
1283933  8617 Holmes Terrace Suite 651                  Tuscarora  …  2020
1283934     1632 Cohen Drive Suite 639  High Rolls Mountain Park  …  2020
1283935           42933 Ryan Underpass                 Manderson  …  2020
1283936           135 Joseph Mountains                      Sula  …  2020

         month day hour weekday  dayofYear    txn_date  customer_age  \
0            1   1    0     Tue          1  2019-01-01          30.0
1            1   1    0     Tue          1  2019-01-01          40.0
2            1   1    0     Tue          1  2019-01-01          56.0
3            1   1    0     Tue          1  2019-01-01          51.0
4            1   1    0     Tue          1  2019-01-01          32.0
…        …   ..  …    …         …          …           …
1283932      6  21   12     Sun        173  2020-06-21          58.0
1283933      6  21   12     Sun        173  2020-06-21          40.0
1283934      6  21   12     Sun        173  2020-06-21          52.0
1283935      6  21   12     Sun        173  2020-06-21          39.0
1283936      6  21   12     Sun        173  2020-06-21          24.0

         masked_accountNumber     BIN
0            2703********2095  270318
1                6304****7322  630423
2              3885******7661  388594
3            3534********0240  353409
4             3755*******3984  375534
…                        …     …
1283932        3026******4123  302635
1283933      6011********6997  601114
1283934      3514********4695  351486
```

```
1283935      2720********6919  272001
1283936   4292**********3207  429290

[1283937 rows x 28 columns]
```

[6]:
```
# Check count of Fraud vs Not_Fraud Transactions. 1- Fraud , 0- Not_Fraud
value_counts = csvfile_final_txn_data['is_fraud'].value_counts()

print("Unique values and their counts in the column:")
print(value_counts)
```

```
Unique values and their counts in the column:
0    1280028
1       3909
Name: is_fraud, dtype: int64
```

[7]:
```
# Create a new DataFrame to store the selected records
selected_data = pd.DataFrame()

# Copy all rows where 'is_fraud' is true (1) to the new DataFrame
selected_data = csvfile_final_txn_data[csvfile_final_txn_data['is_fraud'] == 1].
 ↪copy()

# Sample 16100 random records where 'is_fraud' is false (0) and add them to the␣
 ↪new DataFrame
selected_data = pd.concat([selected_data,␣
 ↪csvfile_final_txn_data[csvfile_final_txn_data['is_fraud'] == 0].
 ↪sample(n=16100, random_state=42)])

# Display the new DataFrame with the selected records
print(selected_data)
```

```
         row_id trans_date_trans_time                       merch_name  \
2436       2449   2019-01-02 01:06:37           fraud_Rutherford-Mertz
2459       2472   2019-01-02 01:47:29  fraud_Jenkins, Hauck and Friesen
2510       2523   2019-01-02 03:05:23           fraud_Goodwin-Nitzsche
2532       2546   2019-01-02 03:38:03           fraud_Erdman-Kertzmann
2539       2553   2019-01-02 03:55:47               fraud_Koepp-Parker
…           …                   …                                …
1103847  1114738   2020-04-08 15:33:40            fraud_Hermann-Gaylord
135276    136824   2019-03-16 10:32:54    fraud_Christiansen-Gusikowski
37199      37638   2019-01-22 18:04:04                fraud_Howe Ltd
449200    453698   2019-07-20 16:17:40            fraud_Gislason Group
737604    744851   2019-11-14 23:08:39              fraud_Dibbert-Green

             category  amount    first     last gender  \
2436       grocery_pos  281.06    Jason   Murphy      M
2459     gas_transport   11.52    Misty     Hart      F
```

```
2510       grocery_pos  276.31    Misty     Hart      F
2532     gas_transport    7.03    Jason   Murphy      M
2539       grocery_pos  275.73    Misty     Hart      F
…                 …     …       …       …    …
1103847       misc_pos    2.12  Sabrina  Johnson      F
135276        misc_pos   34.97    Craig     Dunn      M
37199         misc_pos    4.26   Sharon  Johnson      F
449200           travel    7.00   Daniel  Escobar      M
737604    entertainment   63.61   Joseph    Davis      M

                            street           city  …  year  month day hour  \
2436     542 Steve Curve Suite 011  Collettsville  …  2019      1   2    1
2459     27954 Hall Mill Suite 575    San Antonio  …  2019      1   2    1
2510     27954 Hall Mill Suite 575    San Antonio  …  2019      1   2    3
2532     542 Steve Curve Suite 011  Collettsville  …  2019      1   2    3
2539     27954 Hall Mill Suite 575    San Antonio  …  2019      1   2    3
…                              …            …  …   …    …  ..  …
1103847      320 Nicholson Orchard       Thompson  …  2020      4   8   15
135276       721 Jacqueline Brooks     New Boston  …  2019      3  16   10
37199           7202 Jeffrey Mills         Conway  …  2019      1  22   18
449200            61390 Hayes Port        Romulus  …  2019      7  20   16
737604          941 Adam Stravenue       Nazareth  …  2019     11  14   23

         weekday  dayofYear    txn_date  customer_age  masked_accountNumber  \
2436         Wed          2  2019-01-02          30.0         4613*****1966
2459         Wed          2  2019-01-02          58.0       3401*******0220
2510         Wed          2  2019-01-02          58.0       3401*******0220
2532         Wed          2  2019-01-02          30.0         4613*****1966
2539         Wed          2  2019-01-02          58.0       3401*******0220
…          …        …          …            …                   …
1103847      Wed         99  2020-04-08          32.0      4642********5942
135276       Sat         75  2019-03-16          25.0       1800*******0192
37199        Tue         22  2019-01-22          34.0      3553********4918
449200       Sat        201  2019-07-20          47.0       3749*******3758
737604       Thu        318  2019-11-14          39.0      4451********2894

            BIN
2436     461331
2459     340187
2510     340187
2532     461331
2539     340187
…         …
1103847  464225
135276   180011
37199    355362
449200   374930
737604   445195
```

```
[20009 rows x 28 columns]
```

[8]:
```
result = selected_data.groupby('category').agg(fraud_count=('is_fraud', 'sum'),␣
 ↪category_count=('category', 'size'))

# Create a new column 'fraud_percentage' which represents the fraud percentage␣
 ↪for each category
result['fraud_percentage'] = (result['fraud_count'] / result['category_count'])␣
 ↪* 100

# Round the fraud_percentage to two decimal points
result['fraud_percentage'] = result['fraud_percentage'].round(2)

# Sort the DataFrame by the fraud_percentage in descending order (high to low)
result_sorted = result.sort_values(by='fraud_percentage', ascending=False)

print(result_sorted)
```

```
               fraud_count  category_count  fraud_percentage
category
grocery_pos           1743            3252             53.60
gas_transport          618            2248             27.49
travel                 116             623             18.62
grocery_net            134             740             18.11
personal_care          220            1358             16.20
misc_pos               186            1199             15.51
kids_pets              239            1669             14.32
entertainment          171            1317             12.98
health_fitness         133            1103             12.06
food_dining            151            1271             11.88
home                   198            1747             11.33
misc_net                 0             786              0.00
shopping_net             0            1267              0.00
shopping_pos             0            1429              0.00
```

[9]:
```
# Initialize 'merch_address' column with NaN
selected_data['merch_address'] = np.nan
print(selected_data)
```

```
         row_id trans_date_trans_time                      merch_name  \
2436       2449   2019-01-02 01:06:37          fraud_Rutherford-Mertz
2459       2472   2019-01-02 01:47:29  fraud_Jenkins, Hauck and Friesen
2510       2523   2019-01-02 03:05:23          fraud_Goodwin-Nitzsche
2532       2546   2019-01-02 03:38:03          fraud_Erdman-Kertzmann
2539       2553   2019-01-02 03:55:47              fraud_Koepp-Parker
…           …                   …                               …
1103847  1114738   2020-04-08 15:33:40           fraud_Hermann-Gaylord
```

```
135276    136824   2019-03-16 10:32:54     fraud_Christiansen-Gusikowski
37199      37638   2019-01-22 18:04:04                   fraud_Howe Ltd
449200    453698   2019-07-20 16:17:40              fraud_Gislason Group
737604    744851   2019-11-14 23:08:39               fraud_Dibbert-Green

              category  amount    first     last gender  \
2436       grocery_pos  281.06    Jason   Murphy      M
2459     gas_transport   11.52    Misty     Hart      F
2510       grocery_pos  276.31    Misty     Hart      F
2532     gas_transport    7.03    Jason   Murphy      M
2539       grocery_pos  275.73    Misty     Hart      F
…                  …      …       …      …    …
1103847       misc_pos    2.12  Sabrina  Johnson      F
135276        misc_pos   34.97    Craig     Dunn      M
37199         misc_pos    4.26   Sharon  Johnson      F
449200           travel    7.00   Daniel  Escobar      M
737604    entertainment   63.61   Joseph    Davis      M

                           street           city  … month  day hour  \
2436     542 Steve Curve Suite 011  Collettsville  …     1    2    1
2459     27954 Hall Mill Suite 575    San Antonio  …     1    2    1
2510     27954 Hall Mill Suite 575    San Antonio  …     1    2    3
2532     542 Steve Curve Suite 011  Collettsville  …     1    2    3
2539     27954 Hall Mill Suite 575    San Antonio  …     1    2    3
…                              …            …  …   …  …  …
1103847      320 Nicholson Orchard       Thompson  …     4    8   15
135276       721 Jacqueline Brooks     New Boston  …     3   16   10
37199           7202 Jeffrey Mills         Conway  …     1   22   18
449200            61390 Hayes Port        Romulus  …     7   20   16
737604          941 Adam Stravenue       Nazareth  …    11   14   23

         weekday dayofYear    txn_date  customer_age  masked_accountNumber  \
2436         Wed         2  2019-01-02          30.0         4613*****1966
2459         Wed         2  2019-01-02          58.0       3401*******0220
2510         Wed         2  2019-01-02          58.0       3401*******0220
2532         Wed         2  2019-01-02          30.0         4613*****1966
2539         Wed         2  2019-01-02          58.0       3401*******0220
…          …       …          …           …                   …
1103847      Wed        99  2020-04-08          32.0      4642********5942
135276       Sat        75  2019-03-16          25.0       1800*******0192
37199        Tue        22  2019-01-22          34.0      3553********4918
449200       Sat       201  2019-07-20          47.0       3749*******3758
737604       Thu       318  2019-11-14          39.0      4451********2894

            BIN  merch_address
2436     461331            NaN
2459     340187            NaN
2510     340187            NaN
```

```
2532      461331             NaN
2539      340187             NaN
…          …                 …
1103847   464225             NaN
135276    180011             NaN
37199     355362             NaN
449200    374930             NaN
737604    445195             NaN

[20009 rows x 29 columns]
```

[10]:
```python
# Make the API call to get the address of merchant based on their langitude and␣
 ↪latitude

for index, row in selected_data.iterrows():
    latitude = row['merch_lat']
    longitude = row['merch_long']
    address_data = get_address_from_coordinates(latitude, longitude)
    time.sleep(0.1)
    # Add the merchant address as a column to data frame
    selected_data.at[index, 'merch_address'] = json.dumps(address_data)


api_ff_comb_data = selected_data.copy()


def extract_address(json_str):
    try:
        json_obj = json.loads(json_str)
        if isinstance(json_obj, dict):
            state = json_obj.get('state')
            postcode = json_obj.get('postcode')
            country_code = json_obj.get('country_code')
            return pd.Series([state,postcode, country_code])

    except (json.JSONDecodeError, AttributeError):
        pass
    return pd.Series([np.nan, np.nan, np.nan])


# 01- Extract 'state', 'postcode' and 'country' from the 'merch_address' column␣
 ↪and create new columns
api_ff_comb_data[['merch_state', 'merch_postcode', 'merch_country_code']] =␣
 ↪api_ff_comb_data['merch_address'].apply(extract_address)


# Print data to check the new columns.
```

```
api_ff_comb_data
```

[10]:
```
        row_id trans_date_trans_time                       merch_name  \
0         2449   2019-01-02 01:06:37           fraud_Rutherford-Mertz
1         2472   2019-01-02 01:47:29  fraud_Jenkins, Hauck and Friesen
2         2523   2019-01-02 03:05:23           fraud_Goodwin-Nitzsche
3         2546   2019-01-02 03:38:03           fraud_Erdman-Kertzmann
4         2553   2019-01-02 03:55:47               fraud_Koepp-Parker
…          …                    …                                …
20004  1114738   2020-04-08 15:33:40            fraud_Hermann-Gaylord
20005   136824   2019-03-16 10:32:54    fraud_Christiansen-Gusikowski
20006    37638   2019-01-22 18:04:04                fraud_Howe Ltd
20007   453698   2019-07-20 16:17:40             fraud_Gislason Group
20008   744851   2019-11-14 23:08:39              fraud_Dibbert-Green

            category  amount    first     last gender  \
0        grocery_pos  281.06    Jason   Murphy      M
1      gas_transport   11.52    Misty     Hart      F
2        grocery_pos  276.31    Misty     Hart      F
3      gas_transport    7.03    Jason   Murphy      M
4        grocery_pos  275.73    Misty     Hart      F
…              …     …      …       …    …
20004       misc_pos    2.12  Sabrina  Johnson      F
20005       misc_pos   34.97    Craig     Dunn      M
20006       misc_pos    4.26   Sharon  Johnson      F
20007         travel    7.00   Daniel  Escobar      M
20008  entertainment   63.61   Joseph    Davis      M

                         street           city  … weekday  dayofYear  \
0      542 Steve Curve Suite 011  Collettsville  …     Wed          2
1      27954 Hall Mill Suite 575    San Antonio  …     Wed          2
2      27954 Hall Mill Suite 575    San Antonio  …     Wed          2
3      542 Steve Curve Suite 011  Collettsville  …     Wed          2
4      27954 Hall Mill Suite 575    San Antonio  …     Wed          2
…                          …            …  …     …        …
20004     320 Nicholson Orchard       Thompson  …     Wed         99
20005     721 Jacqueline Brooks     New Boston  …     Sat         75
20006        7202 Jeffrey Mills         Conway  …     Tue         22
20007          61390 Hayes Port        Romulus  …     Sat        201
20008        941 Adam Stravenue       Nazareth  …     Thu        318

        txn_date customer_age masked_accountNumber     BIN  \
0     2019-01-02         30.0        4613*****1966  461331
1     2019-01-02         58.0      3401******0220  340187
2     2019-01-02         58.0      3401******0220  340187
3     2019-01-02         30.0        4613*****1966  461331
```

```
4      2019-01-02         58.0      3401*******0220  340187
…           …          …                  …     …
20004  2020-04-08         32.0     4642********5942  464225
20005  2019-03-16         25.0      1800*******0192  180011
20006  2019-01-22         34.0     3553********4918  355362
20007  2019-07-20         47.0      3749*******3758  374930
20008  2019-11-14         39.0     4451********2894  445195

                                              merch_address     merch_state  \
0      {"road": "Bluff Mountain Trail", "county": "Al…  North Carolina
1      {"county": "Bandera County", "state": "Texas",…           Texas
2      {"road": "County Road 5716", "county": "Medina…           Texas
3      {"building": "BRP US Inc", "house_number": "12…  North Carolina
4      {"road": "Ammann Road", "town": "Boerne", "cou…           Texas
…                                                    …              …
20004  {"county": "Uintah County", "state": "Utah", "…            Utah
20005  {"road": "268th Street", "county": "Jefferson …            Iowa
20006  {"county": "Jefferson County", "state": "Washi…      Washington
20007  {"road": "Carlton Rockwood Road", "town": "Ash…        Michigan
20008  {"road": "County Road 17", "county": "Deaf Smi…           Texas

       merch_postcode  merch_country_code
0                 NaN                  us
1                 NaN                  us
2               78059                  us
3               28777                  us
4               78006                  us
…                   …                   …
20004             NaN                  us
20005             NaN                  us
20006             NaN                  us
20007           48179                  us
20008             NaN                  us

[20009 rows x 32 columns]
```

[11]:
```
# Size before duplicate check
print("Size of the dataset before to duplicate check: ",api_ff_comb_data.shape)
```

```
Size of the dataset before to duplicate check:  (20009, 32)
```

[12]:
```
# 02 : Identify any duplicate rows
api_ff_comb_data_duplicates = api_ff_comb_data[api_ff_comb_data.
 ↪duplicated(subset=api_ff_comb_data.columns[:-1], keep=False)]
print(api_ff_comb_data_duplicates)
```

```
Empty DataFrame
Columns: [row_id, trans_date_trans_time, merch_name, category, amount, first,
```

```
last, gender, street, city, state, zip, job, trans_num, unix_time, merch_lat,
merch_long, is_fraud, year, month, day, hour, weekday, dayofYear, txn_date,
customer_age, masked_accountNumber, BIN, merch_address, merch_state,
merch_postcode, merch_country_code]
Index: []

[0 rows x 32 columns]
```

[13]:
```
# 03: Identify any missing values
missing_values = api_ff_comb_data.isnull().sum()
print("Missing Values:\n", missing_values)
```

```
Missing Values:
 row_id                       0
trans_date_trans_time         0
merch_name                    0
category                      0
amount                        0
first                         0
last                          0
gender                        0
street                        0
city                          0
state                         0
zip                           0
job                           0
trans_num                     0
unix_time                     0
merch_lat                     0
merch_long                    0
is_fraud                      0
year                          0
month                         0
day                           0
hour                          0
weekday                       0
dayofYear                     0
txn_date                      0
customer_age                  0
masked_accountNumber          0
BIN                           0
merch_address                 0
merch_state                 849
merch_postcode             8949
merch_country_code          685
dtype: int64
```

[14]:
```
# 04 : Find the Outilier for the Amount

# Calculate summary statistics for the transaction amount column
amount_stats = api_ff_comb_data['amount'].describe()

# Calculate the interquartile range (IQR)
Q1 = amount_stats['25%']
Q3 = amount_stats['75%']
IQR = Q3 - Q1

# Find the lower and upper bounds for outliers
lower_bound = Q1 - (1.5 * IQR)
upper_bound = Q3 + (1.5 * IQR)

# Identify the rows with transaction amounts outside the bounds
outliers = api_ff_comb_data[(api_ff_comb_data['amount'] < lower_bound) |␣
 ↪(api_ff_comb_data['amount'] > upper_bound)]

# Print the number of outliers found
print("Number of outliers found: ", len(outliers))

# Remove the outliers

api_ff_comb_data_no_outliers = api_ff_comb_data[(api_ff_comb_data['amount'] >=␣
 ↪lower_bound) &  (api_ff_comb_data['amount'] <= upper_bound)]

print("Size of the dataset after removal of outliers: ",␣
 ↪len(api_ff_comb_data_no_outliers))
```

```
Number of outliers found:  2410
Size of the dataset after removal of outliers:  17599
```

[15]:
```
# 05 : Drop colmuns those are not required

# Drop the columns that are not needed and create a new DataFrame without those␣
 ↪columns
api_ff_comb_data_S01 = api_ff_comb_data_no_outliers.
 ↪drop(columns=api_ff_comb_data_no_outliers.filter(like='V').columns)

# Display the new DataFrame with the unnecessary columns dropped
print(api_ff_comb_data_S01)
```

```
        row_id trans_date_trans_time                          merch_name  \
1         2472   2019-01-02 01:47:29     fraud_Jenkins, Hauck and Friesen
3         2546   2019-01-02 03:38:03                fraud_Erdman-Kertzmann
5         3580   2019-01-03 01:05:27              fraud_Conroy-Cruickshank
9         4693   2019-01-03 22:58:44                  fraud_Mosciski Group
11        4808   2019-01-04 00:58:03  fraud_Stokes, Christiansen and Sipes
```

```
…        …                    …                                    …
20004  1114738   2020-04-08 15:33:40                 fraud_Hermann-Gaylord
20005   136824   2019-03-16 10:32:54          fraud_Christiansen-Gusikowski
20006    37638   2019-01-22 18:04:04                       fraud_Howe Ltd
20007   453698   2019-07-20 16:17:40                  fraud_Gislason Group
20008   744851   2019-11-14 23:08:39                   fraud_Dibbert-Green

            category  amount    first     last gender  \
1      gas_transport   11.52    Misty     Hart      F
3      gas_transport    7.03    Jason   Murphy      M
5      gas_transport   10.76    Misty     Hart      F
9             travel    4.50  Heather    Chase      F
11       grocery_net   14.37     Mark    Brown      M
…                …     …       …       …    …
20004       misc_pos    2.12  Sabrina  Johnson      F
20005       misc_pos   34.97    Craig     Dunn      M
20006       misc_pos    4.26   Sharon  Johnson      F
20007         travel    7.00   Daniel  Escobar      M
20008  entertainment   63.61   Joseph    Davis      M

                          street           city  … weekday  dayofYear  \
1        27954 Hall Mill Suite 575    San Antonio  …     Wed          2
3        542 Steve Curve Suite 011  Collettsville  …     Wed          2
5        27954 Hall Mill Suite 575    San Antonio  …     Thu          3
9      6888 Hicks Stream Suite 954          Manor  …     Thu          3
11                8580 Moore Cove          Wales  …     Fri          4
…                             …              …  …     …        …
20004        320 Nicholson Orchard       Thompson  …     Wed         99
20005        721 Jacqueline Brooks     New Boston  …     Sat         75
20006           7202 Jeffrey Mills         Conway  …     Tue         22
20007             61390 Hayes Port        Romulus  …     Sat        201
20008           941 Adam Stravenue       Nazareth  …     Thu        318

         txn_date customer_age masked_accountNumber     BIN  \
1      2019-01-02         58.0      3401*******0220  340187
3      2019-01-02         30.0        4613*****1966  461331
5      2019-01-03         58.0      3401*******0220  340187
9      2019-01-03         77.0     4922********1201  492271
11     2019-01-04         79.0      3415*******6537  341546
…             …            …                   …     …
20004  2020-04-08         32.0     4642********5942  464225
20005  2019-03-16         25.0      1800*******0192  180011
20006  2019-01-22         34.0     3553********4918  355362
20007  2019-07-20         47.0      3749*******3758  374930
20008  2019-11-14         39.0     4451********2894  445195

                                          merch_address     merch_state  \
1      {"county": "Bandera County", "state": "Texas",…           Texas
```

```
3      {"building": "BRP US Inc", "house_number": "12…  North Carolina
5      {"county": "Karnes County", "state": "Texas", …           Texas
9      {"county": "Mineral County", "state": "West Vi…   West Virginia
11     {"road": "Nome-Taylor Highway", "county": "Nom…          Alaska
…                                                    …             …
20004  {"county": "Uintah County", "state": "Utah", "…            Utah
20005  {"road": "268th Street", "county": "Jefferson …            Iowa
20006  {"county": "Jefferson County", "state": "Washi…      Washington
20007  {"road": "Carlton Rockwood Road", "town": "Ash…        Michigan
20008  {"road": "County Road 17", "county": "Deaf Smi…           Texas

       merch_postcode  merch_country_code
1                 NaN                  us
3               28777                  us
5                 NaN                  us
9                 NaN                  us
11                NaN                  us
…                 …                    …
20004             NaN                  us
20005             NaN                  us
20006             NaN                  us
20007           48179                  us
20008             NaN                  us

[17599 rows x 32 columns]
```

[16]:
```python
api_ff_comb_data_S01['zip_match'] = np.nan
api_ff_comb_data_S01['state_match'] = np.nan

for index, row in api_ff_comb_data_S01.iterrows():
    zip_code1 = row['zip']
    zip_code2 = row['merch_postcode']

    state1 = row['state']
    state2 = row['merch_state']

    if zip_code2 == zip_code1:
        api_ff_comb_data_S01.at[index, 'zip_match'] = 'Y'
    else:
        api_ff_comb_data_S01.at[index, 'zip_match'] = 'N'

    if state1 == state2:
        api_ff_comb_data_S01.at[index, 'state_match'] = 'Y'
    else:
        api_ff_comb_data_S01.at[index, 'state_match'] = 'N'
```

```
api_ff_comb_data_final = api_ff_comb_data_S01.copy()
api_ff_comb_data_final
```

[16]:

```
        row_id trans_date_trans_time                          merch_name  \
1         2472   2019-01-02 01:47:29     fraud_Jenkins, Hauck and Friesen
3         2546   2019-01-02 03:38:03                fraud_Erdman-Kertzmann
5         3580   2019-01-03 01:05:27              fraud_Conroy-Cruickshank
9         4693   2019-01-03 22:58:44                   fraud_Mosciski Group
11        4808   2019-01-04 00:58:03  fraud_Stokes, Christiansen and Sipes
…          …                     …                                     …
20004  1114738   2020-04-08 15:33:40                fraud_Hermann-Gaylord
20005   136824   2019-03-16 10:32:54         fraud_Christiansen-Gusikowski
20006    37638   2019-01-22 18:04:04                        fraud_Howe Ltd
20007   453698   2019-07-20 16:17:40                  fraud_Gislason Group
20008   744851   2019-11-14 23:08:39                   fraud_Dibbert-Green

            category  amount    first     last gender  \
1      gas_transport   11.52    Misty     Hart      F
3      gas_transport    7.03    Jason   Murphy      M
5      gas_transport   10.76    Misty     Hart      F
9             travel    4.50  Heather    Chase      F
11       grocery_net   14.37     Mark    Brown      M
…                …      …       …       …    …
20004       misc_pos    2.12  Sabrina  Johnson      F
20005       misc_pos   34.97    Craig     Dunn      M
20006       misc_pos    4.26   Sharon  Johnson      F
20007         travel    7.00   Daniel  Escobar      M
20008  entertainment   63.61   Joseph    Davis      M

                          street           city  …    txn_date  \
1        27954 Hall Mill Suite 575    San Antonio  …  2019-01-02
3        542 Steve Curve Suite 011  Collettsville  …  2019-01-02
5        27954 Hall Mill Suite 575    San Antonio  …  2019-01-03
9      6888 Hicks Stream Suite 954          Manor  …  2019-01-03
11                8580 Moore Cove          Wales  …  2019-01-04
…                             …              …  …         …
20004       320 Nicholson Orchard       Thompson  …  2020-04-08
20005       721 Jacqueline Brooks     New Boston  …  2019-03-16
20006          7202 Jeffrey Mills         Conway  …  2019-01-22
20007            61390 Hayes Port        Romulus  …  2019-07-20
20008          941 Adam Stravenue       Nazareth  …  2019-11-14

       customer_age masked_accountNumber     BIN  \
1              58.0      3401******0220  340187
3              30.0        4613*****1966  461331
5              58.0      3401******0220  340187
9              77.0     4922*******1201  492271
```

```
11             79.0      3415*******6537  341546
…              …                    …     …
20004          32.0     4642********5942  464225
20005          25.0      1800*******0192  180011
20006          34.0     3553********4918  355362
20007          47.0      3749*******3758  374930
20008          39.0     4451********2894  445195

                                                merch_address     merch_state  \
1      {"county": "Bandera County", "state": "Texas",…            Texas
3      {"building": "BRP US Inc", "house_number": "12…  North Carolina
5      {"county": "Karnes County", "state": "Texas", …            Texas
9      {"county": "Mineral County", "state": "West Vi…   West Virginia
11     {"road": "Nome-Taylor Highway", "county": "Nom…           Alaska
…                                                    …              …
20004  {"county": "Uintah County", "state": "Utah", "…             Utah
20005  {"road": "268th Street", "county": "Jefferson …             Iowa
20006  {"county": "Jefferson County", "state": "Washi…      Washington
20007  {"road": "Carlton Rockwood Road", "town": "Ash…         Michigan
20008  {"road": "County Road 17", "county": "Deaf Smi…            Texas

       merch_postcode  merch_country_code  zip_match  state_match
1                 NaN                  us          N            N
3               28777                  us          N            N
5                 NaN                  us          N            N
9                 NaN                  us          N            N
11                NaN                  us          N            N
…                 …                    …          …            …
20004             NaN                  us          N            N
20005             NaN                  us          N            N
20006             NaN                  us          N            N
20007           48179                  us          N            N
20008             NaN                  us          N            N

[17599 rows x 34 columns]
```

[17]:
```
# Overview of the structure and characteristics
print(api_ff_comb_data_final.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 17599 entries, 1 to 20008
Data columns (total 34 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   row_id                 17599 non-null  int64
 1   trans_date_trans_time  17599 non-null  object
 2   merch_name             17599 non-null  object
 3   category               17599 non-null  object
```

```
 4   amount                 17599 non-null  float64
 5   first                  17599 non-null  object
 6   last                   17599 non-null  object
 7   gender                 17599 non-null  object
 8   street                 17599 non-null  object
 9   city                   17599 non-null  object
 10  state                  17599 non-null  object
 11  zip                    17599 non-null  int64
 12  job                    17599 non-null  object
 13  trans_num              17599 non-null  object
 14  unix_time              17599 non-null  object
 15  merch_lat              17599 non-null  float64
 16  merch_long             17599 non-null  float64
 17  is_fraud               17599 non-null  int64
 18  year                   17599 non-null  int64
 19  month                  17599 non-null  int64
 20  day                    17599 non-null  int64
 21  hour                   17599 non-null  int64
 22  weekday                17599 non-null  object
 23  dayofYear              17599 non-null  int64
 24  txn_date               17599 non-null  object
 25  customer_age           17599 non-null  float64
 26  masked_accountNumber   17599 non-null  object
 27  BIN                    17599 non-null  int64
 28  merch_address          17599 non-null  object
 29  merch_state            16918 non-null  object
 30  merch_postcode         9798 non-null   object
 31  merch_country_code     17061 non-null  object
 32  zip_match              17599 non-null  object
 33  state_match            17599 non-null  object
dtypes: float64(4), int64(9), object(21)
memory usage: 5.2+ MB
None
```

[ ]: