Initial Data Analysis

Submitted by: Reet

INDEX:

Initial Data Analysis/Exploratory Data Analysis

- IDA and EDA are some techniques to understand the various aspects of the data. It involves many data exploration techniques to understand all aspects of data.
- We have to make sure that the data we are working with doesn't have any redundancies or outliers in it and it is clean.
- We also have to make sure that we identify the important values of the data set and variables as well.

Objective of performing EDA?

- It is basically performed to filter the data from redundancies.
- It helps us identify the quality points in data.

- It helps us understand the relationship between variables, which gives us a better understanding of data.

Steps involved in EDA:

1. Understand the data- variables, no. of columns, rows
2. Clean the data- redundancies, irregularities, data which is not necessary to help us reach a valid conclusion should be cleared, outliers that can cause over building or under building of the model
3. Analysis of relationship between variables

Performing EDA/IDA for each csv file separately

a) a.csv

- Printing the head and tail of data: Just to get an idea of how consistent the data is throughout. This will also make us understand how large is the data set we are dealing with.

```
b-IDA ×   a_IDA ×
C:\Users\VIRENDRA\anaconda3\envs\packt\python.exe C:\Users\VIRENDRA\PycharmProjects\packt\IDA-EDA\a_IDA.py
             log_time  phone    status  type        product pay_mode  marker
0  2019-05-03 15:57:56    0.0  assigned  1002            NaN      NaN       0
1  2019-05-03 15:57:56    1.0  assigned  1002            NaN      NaN       0
2  2019-05-03 11:34:57    2.0  purchase  1001  In-depth Book    cc-dc       1
3  2019-05-03 11:44:02    3.0  purchase  1001  In-depth Book    cp-nb       1
4  2019-05-03 15:57:56    4.0  assigned  1002            NaN      NaN       0
                  log_time     phone           status  ...  product pay_mode marker
998817  2021-07-31 23:58:52  607730.0         Assigned  ...      NaN      NaN     10
998818  2021-07-31 23:59:06  607730.0  Follow-up later  ...       LI      NaN      1
998819  2021-07-31 23:59:23  607731.0         Assigned  ...      NaN      NaN     10
998820  2021-07-31 23:59:37  607731.0  Follow-up later  ...       LI      NaN      1
998821  2021-07-31 23:59:45  607732.0         Assigned  ...      NaN      NaN     10
```

- Finding total rows and columns in the data set and the names of all the columns present, to understand the categories of the data well.

```
[5 rows x 7 columns]
(998822, 7)
Index(['log_time', 'phone', 'status', 'type', 'product', 'pay_mode', 'marker'], dtype='object')
               phone           type         marker
count  998814.000000  998822.000000  998822.000000
mean   260397.183060    1244.267067       1.623012
std    187624.002369     457.458217       2.865491
min         0.000000    1001.000000     -99.000000
25%     89754.250000    1001.000000       1.000000
50%    244574.500000    1002.000000       1.000000
75%    423747.000000    1002.000000       1.000000
max    607732.000000    2209.000000      10.000000
```

- Describing the total data entries and finding special entries in the column markers and status to analyse them further

```
log_time    559772
phone       607732
status          23
type            21
product        125
pay_mode        62
marker           6
dtype: int64
['assigned' 'purchase' 'AA' 'AB' 'AC' 'AD' 'Switched off' 'Not interested'
 'Already purchased' 'Follow-up later' 'User is Interested' 'Converted'
 'New product potential' 'Line Busy' 'Has complaints' 'Invalid Number'
 'Not reachable' 'Partially interested' 'Other' 'Assigned'
 'Could not call' 'Not picking up' 'none']
[  0   1 -99  -1 -10  10]
```

- Finding null values to see the redundancies in the data in each column

```
log_time         0
phone            8
status           0
type             0
product     668696
pay_mode    775819
marker           0
dtype: int64

Process finished with exit code 0
```

b) b.csv

Following the same steps as before, attaching the result

```
C:\Users\VIRENDRA\anaconda3\envs\packt\python.exe C:/Users/VIRENDRA/PycharmProjects/packt/IDA-EDA/b-IDA.py
   uuid beacon_type  beacon_value             log_date  status
0   0.0   user_stay          26.0  2019-02-26 16:19:08       1
1   0.0   user_stay          32.0  2019-02-26 16:30:08       1
2   1.0   user_stay           1.0  2019-02-26 16:33:39       1
3   2.0   user_stay           1.0  2019-02-26 16:42:00       1
4   3.0   user_stay           1.0  2019-02-26 16:42:00       1
                uuid        beacon_type  ...             log_date status
39009327  10058134.0  pay_button_paypal  ...  2021-07-31 23:59:58      1
39009328  10058148.0          user_stay  ...  2021-07-31 23:59:58      1
39009329  10058129.0          user_stay  ...  2021-07-31 23:59:59      1
39009330  10058118.0          user_stay  ...  2021-07-31 23:59:59      1
39009331  10058149.0          user_stay  ...  2021-07-31 23:59:59      1

[5 rows x 5 columns]
(39009332, 5)
Index(['uuid', 'beacon_type', 'beacon_value', 'log_date', 'status'], dtype='object')
               uuid  beacon_value      status
count  3.900932e+07  3.900933e+07  39009332.0
mean   4.922558e+06  5.610764e+00         1.0
std    2.985628e+06  1.102048e+01         0.0
min    0.000000e+00  1.000000e+00         1.0
25%    2.232751e+06  1.000000e+00         1.0
50%    4.901919e+06  2.000000e+00         1.0
75%    7.526520e+06  4.000000e+00         1.0
max    1.005815e+07  9.990000e+02         1.0
```

c) c.csv

```
C:\Users\VIRENDRA\anaconda3\envs\packt\python.exe C:/Users/VIRENDRA/PycharmProjects/packt/IDA-EDA/c-IDA.py
   id    email  primary_phone secondary_phones  profile_submit_count
0   1   537606           22.0             NaN                   592
1   5  1443908            NaN             NaN                     3
2   6   534973            NaN          588180                     6
3   7  3259797            NaN             NaN                     3
4   8  1701404            NaN             NaN                     5
              id    email  primary_phone secondary_phones  profile_submit_count
2295096  4867864  5554890      4869388.0             NaN                     1
2295097  4867865  5554891      4869389.0             NaN                     1
2295098  4867867  5554892      4869390.0             NaN                     1
2295099  4867879  5554893            NaN             NaN                     1
2295100  4867881  5554894      4869391.0             NaN                     1
(2295101, 5)
Index(['id', 'email', 'primary_phone', 'secondary_phones',
       'profile_submit_count'],
      dtype='object')
                id         email  primary_phone  profile_submit_count
count  2.295101e+06  2.295101e+06   1.502089e+06          2.295101e+06
mean   2.594772e+06  3.393918e+06   2.920364e+06          2.927417e+00
std    1.435092e+06  1.281969e+06   1.560694e+06          1.084325e+01
min    1.000000e+00  0.000000e+00   2.000000e+00          1.000000e+00
25%    1.285110e+06  2.580305e+06   1.799900e+06          2.000000e+00
50%    2.830071e+06  3.156446e+06   3.497062e+06          2.000000e+00
75%    3.868155e+06  4.638336e+06   3.926352e+06          3.000000e+00
max    4.867881e+06  5.554894e+06   4.869391e+06          9.842000e+03
id                     2295101
email                  2295101
primary_phone          1348348
secondary_phones         97881
```

```
  116   82  217   70  258   51   64  955   90   79  105  134  199   77
  329  308   76  496   74  150  670   56  100  167   69   93   92  187
  168   62  296   84  181  280 1092  415  123 1333  176 1396   83  320
  115  222   95  290   98  180 1693   80  171   99  146  200  513   96
  921  135  158   86  834  484  194  833  535   97   91  531   73  323
  376  233  124  155  203  143  195  104  220  165  107  125  311  128
  210  212  179  310  404  986  318  265   78  211  229  754  355  136
  101  470  255  238  132  213  319  193  353  344  425  262  139  288
  548  745 1072  161  373  144  239  219  102  362  183  214  751  274
  877  152  441  138  148 1440  216  287  337  454  201  129  929  303
   88  175  306  281  336  173  177  108  149  560  153  156  321  400
  164  140  169  122  516  184  227  131  286  142  119  676  254  127
  630  279  154  313  120  185  359  202  234  197  159  174  264  412
  367  133  299  504  370  205  260  471  261  157  270  397  147  218
  301  621  186  557  345  259  418  334  450  235  273  145  278  282
  198  109  380  237  207  178  462  330  566  444  568  208  247  307
  394  395  141  524  578  263  322  250  256  623  253  268  224  230
  466  196  331  285  304  401  413  225  588 1372 2933  269  351 3630
 1226 2401  472  347  710  221 1664  266  386  577  526  312  447  294
 1398  427  405  420  252  333  495 2390  204  192 1008  231  272  332
  449  328  357 1200  327  309  410]
id                            0
email                         0
primary_phone            793012
secondary_phones        2180343
profile_submit_count          0
dtype: int64

Process finished with exit code 0
```

d) ct.csv

```
C:\Users\VIRENDRA\anaconda3\envs\packt\python.exe C:/Users/VIRENDRA/PycharmProjects/packt/IDA-EDA/ct-IDA.py
   id  cid            timestamp   amount             status
0   4    1  2021-04-26 17:21:24    730.0  PAYMENT_COMPLETED
1   5    5  2021-01-01 05:57:10  17700.0  PAYMENT_COMPLETED
2   6    6  2021-01-01 10:33:22    849.0  PAYMENT_COMPLETED
3   7    7  2021-01-02 06:10:53   1685.0  PAYMENT_COMPLETED
4   8    8  2021-01-02 08:32:43   2000.0  PAYMENT_COMPLETED
              id      cid            timestamp  amount status
4174008  4174055  4867893  2021-08-25 20:10:18     0.0      N
4174009  4174056  2197111  2021-08-25 20:10:18     0.0      N
4174010  4174057  3423129  2021-08-25 20:10:23     0.0      N
4174011  4174058  4867896  2021-08-25 20:10:26     0.0      N
4174012  4174059   653124  2021-08-25 20:10:50     0.0      N
(4174013, 5)
Index(['id', 'cid', 'timestamp', 'amount', 'status'], dtype='object')
                 id           cid        amount
count  4.174013e+06  4.174013e+06  4.174013e+06
mean   2.087048e+06  1.767756e+06  3.592679e+01
std    1.204940e+06  1.375565e+06  3.090433e+02
min    4.000000e+00  1.000000e+00  0.000000e+00
25%    1.043544e+06  5.536010e+05  0.000000e+00
50%    2.087049e+06  1.517702e+06  0.000000e+00
75%    3.130556e+06  2.748210e+06  0.000000e+00
max    4.174059e+06  4.867896e+06  1.301137e+05
id           4174013
cid          1633595
timestamp    3240843
amount          1094
status            10
dtype: int64
Run   TODO   Problems   Terminal   Python Packages   Python Console
```

```
dtype: int64
id            0
cid           0
timestamp     0
amount        0
status        0
dtype: int64

Process finished with exit code 0
```

e) s.csv

```
C:\Users\VIRENDRA\anaconda3\envs\packt\python.exe C:/Users/VIRENDRA/PycharmProjects/packt/IDA-EDA/s-IDA.py
       uuid     phone  status  ... report_type  device             log_date
0  10058150     145.0       1  ...       LS-MT  mobile  2019-02-26 16:07:25
1         0     145.0       1  ...       LS-MT  mobile  2019-02-26 16:12:08
2         1     145.0       1  ...       LS-MT  mobile  2019-02-26 16:33:00
3  10058153  607734.0       1  ...       LS-MP  mobile  2019-02-26 16:44:19
4        26  607735.0       1  ...       LS-MT  mobile  2019-02-26 16:44:32

[5 rows x 10 columns]
             uuid      phone  status  ... report_type  device             log_date
9095597  19153747  4007596.0       1  ...       LS-MT  mobile  2021-07-31 23:59:45
9095598  19153748   607007.0       1  ...       LS-MT  mobile  2021-07-31 23:59:49
9095599  19153749  4007729.0       1  ...       LS-MT  mobile  2021-07-31 23:59:51
9095600  19153750  4007717.0       1  ...       LS-MT  mobile  2021-07-31 23:59:54
9095601  19153751  4007730.0       1  ...       LS-MT  mobile  2021-07-31 23:59:54

[5 rows x 10 columns]
(9095602, 10)
Index(['uuid', 'phone', 'status', 'gender', 'dob', 'language', 'email',
       'report_type', 'device', 'log_date'],
      dtype='object')
               uuid         phone     status         email
count  9.095602e+06  9.094625e+06  9095602.0  9.094869e+06
mean   6.474746e+06  1.415220e+06        1.0  1.240107e+06
std    4.422480e+06  1.215990e+06        0.0  9.714803e+05
min    0.000000e+00  0.000000e+00        1.0  0.000000e+00
25%    2.790156e+06  2.984330e+05        1.0  3.941350e+05
50%    6.237638e+06  1.125327e+06        1.0  1.030454e+06
75%    9.048801e+06  2.339591e+06        1.0  2.018648e+06
max    1.915375e+07  4.007730e+06        1.0  3.259793e+06
```

```
uuid           9095602
phone          3399997
status               1
gender               6
dob              36934
language            17
email          3259793
report_type         81
device               5
log_date       8285461
dtype: int64
uuid               0
phone            977
status             0
gender          4765
dob               20
language         398
email            733
report_type       70
device           187
log_date           0
dtype: int64

Process finished with exit code 0
```

f) tp.csv

```
tp-IDA
C:\Users\VIRENDRA\anaconda3\envs\packt\python.exe C:/Users/VIRENDRA/PycharmProjects/packt/IDA-EDA/tp-IDA.py
   ctid  variant language status
0     4  premium      tel    NaN
1     5  premium      eng    NaN
2     6  premium      eng    NaN
3     7  premium      eng    NaN
4     8  premium      eng    NaN
             ctid variant language status
4179019  4174090   basic      eng    NaN
4179020  4174091   basic      eng    NaN
4179021  4174092   basic      eng    NaN
4179022  4174093   basic      tam    NaN
4179023  4174094   basic      eng    NaN
(4179024, 4)
Index(['ctid', 'variant', 'language', 'status'], dtype='object')
               ctid
count  4.179024e+06
mean   2.086676e+06
std    1.204867e+06
min    4.000000e+00
25%    1.043103e+06
50%    2.086494e+06
75%    3.129968e+06
max    4.174094e+06
ctid        4170263
variant           6
language         30
status            5
dtype: int64
ctid              0
Run   TODO   Problems   Terminal   Python Packages   Python Console
```

Attaching the code spinnept that I used to get these results

```python
b-IDA.py
import pandas as pd
import numpy as np
import seaborn as sns
data = pd.read_csv(r"C:\Users\VIRENDRA\Downloads\data_packt_gp1_customer_prioritisation (1)\b.csv")
# 1 understanding the data
data.head()
print(data.head())
data.tail()
print(data.tail())
data.shape
print(data.shape)
data.columns
print(data.columns)
data.describe()
print(data.describe())
data.nunique()
print(data.nunique())
data['status'].unique()
print(data['status'].unique())
data['beacon_value'].unique()
print(data['beacon_value'].unique())
# cleaning the data
data.isnull().sum()
print(data.isnull().sum())
```