Capstone_cut1

1 Imports

1.0.1 Used Libraries:

numpy

pandas

matplotlib

[1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings

warnings.filterwarnings('ignore')
```

2 Reading the CSV file

[2]:
```
df=pd.read_csv('CBL_Modelling_Data_Capstone_cut1.csv')
```

3 Giving column name to index column

Naming unnamed column as 'ID'

[3]:
```
df=df.rename(columns={'Unnamed: 0':'ID'})
```

3.0.1 Total rows in Dataframe

[4]:
```
no_of_rows=df.shape[0]
no_of_rows
```

[4]:
```
111792
```

3.0.2 Total columns in Dataframe

[5]:
```
no_of_columns=df.shape[1]
no_of_columns
```

[5]: 358

4 Dropping all duplicate columns

drop_duplicates removes the duplicates rows, so to remove duplicate columns, it can be converted to its transpose from and remove duplicates and then transpose again to get back to its original shape

[6]:
```
df2 = df.T.drop_duplicates().T
```

5 Saving all removed dupilcates columns data to new csv file

Saving all removed duplicates data into data.csv file

[7]:
```
df2.to_csv(r'C:\drive\self_training_projects\project_1\data.csv',header=True,␣
↪index=False)
```

6 Reading removed duplicated file

Reading data.csv file

[8]:
```
df = pd.read_csv('data.csv')
```

6.0.1 Total columns after removing duplicate columns

[9]:
```
no_of_columns=df.shape[1]
no_of_columns
```

[9]: 247

7 Accessing all columns and the count of null values which has at least 1 null value in its respective column

nan_counts is the dictionary which stores :

its keys as column names which has at least 1 null value.

its values as number of null values in their corresponding column.

[10]:
```
nan_counts = df[[i for i in df.columns if df[i].isna().any()]].isna().sum().
↪to_dict()
```

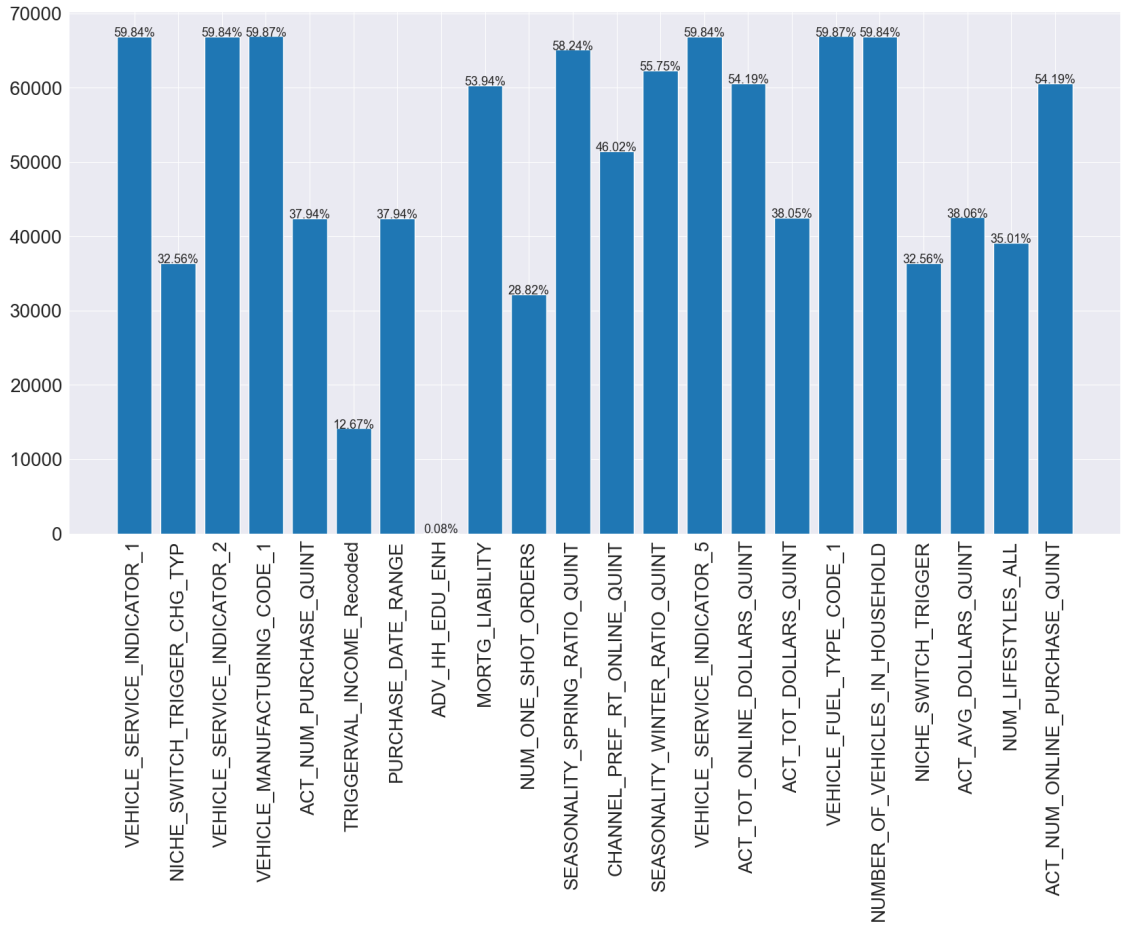
8 Calculating the percentage of null values in its respective column

[11]:
```
per = []
for key, value in nan_counts.items():
    per.append((value / df.shape[0]) * 100)
```

9 Plotting columns and number of null values corresponding to it

Line 4-8 code was written to represent the null value percentage on each bar to their corresponding column

[12]:
```
plt.figure(figsize=(20, 10))
ax = plt.bar(nan_counts.keys(), nan_counts.values())
plt.xticks(rotation='vertical',fontsize=20)
plt.yticks(fontsize=20)
patches = ax.patches
for i in range(len(patches)):
    x = patches[i].get_x() + patches[i].get_width() / 2
    y = patches[i].get_height() + .05
    plt.annotate('{:.2f}%'.format(per[i]), (x, y), ha='center',fontsize=13)
plt.show()
```



10 Columns which has more than 50% of null values

gt50nan stores the names of columns which has more than 50% of null values in it

[13]:
```
gt50nan = [i for i in df.columns if df[i].isna().sum() >= (df.shape[0] / 2)]
```

11 Dropping all columns with more than 50% of null values

[14]:
```
for col in gt50nan:
    df.drop(col, axis=1, inplace=True)
```

12 Dropping rows where the corresponding column has null value

Column 'ADV_HH_EDU_ENH' has only 0.08% which is 84 values, so which we can remove null values corresponding rows since 84 << 111792

[15]:
```
df = df[df['ADV_HH_EDU_ENH'].notna()]
```

12.0.1 Total rows after removing the null values corresponding row

[16]:
```
no_of_rows=df.shape[0]
no_of_rows
```

[16]: 111708

13 Accessing all columns and the count of null values which has at least 1 null value in its respective column

nan_counts is the dictionary which stores :

its keys as column names which has at least 1 null value.

its values as number of null values in their corresponding column.

[17]:
```
nan_counts = df[[i for i in df.columns if df[i].isna().any()]].isna().sum().
 ↪to_dict()
```

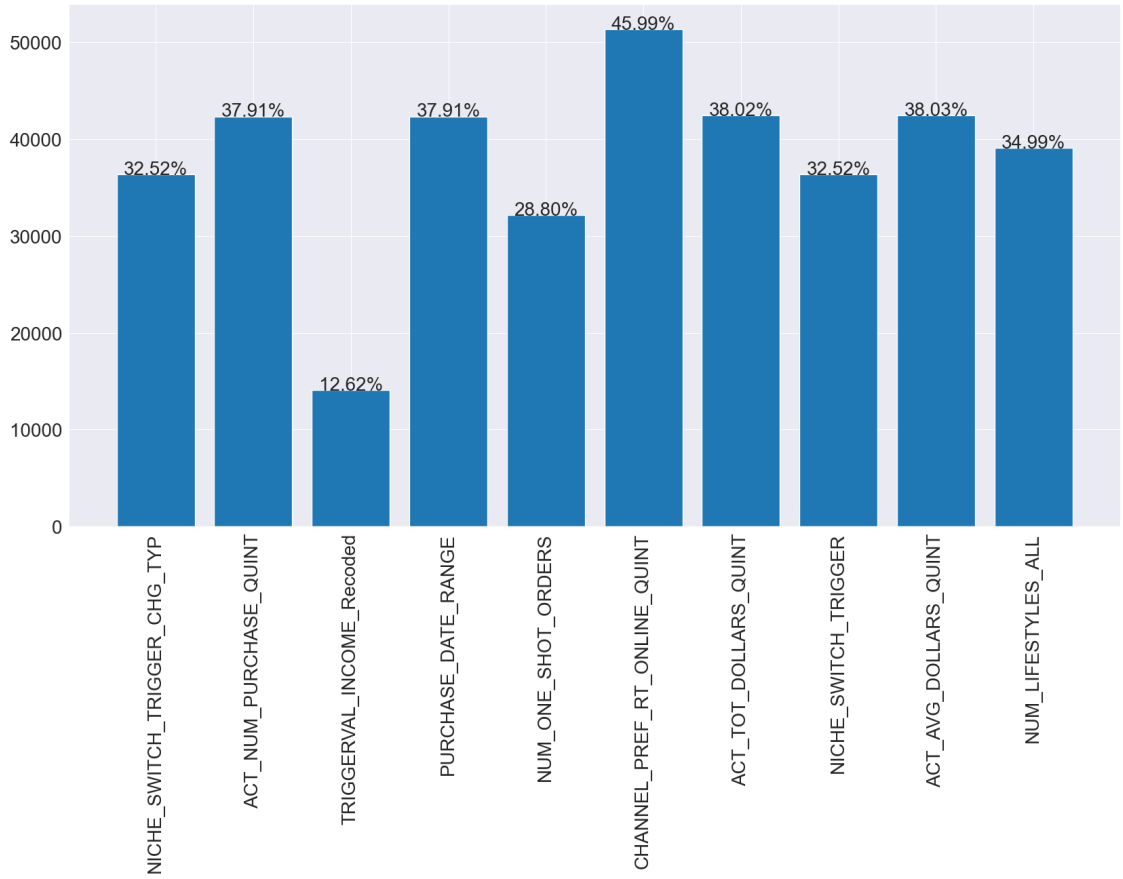
14 Calculating the percentage of null values of each column

[18]:
```
per = []
for key, value in nan_counts.items():
    per.append((value / df.shape[0]) * 100)
```

15 Plotting columns and number of null values corresponding to it

Line 4-8 code was written to represent the null value percentage on each bar to their corresponding column

[19]:
```
plt.figure(figsize=(20, 10))
ax = plt.bar(nan_counts.keys(), nan_counts.values())
plt.xticks(rotation='vertical',fontsize=20)
plt.yticks(fontsize=20)
patches = ax.patches
for i in range(len(patches)):
    x = patches[i].get_x() + patches[i].get_width() / 2
    y = patches[i].get_height() + .05
    plt.annotate('{:.2f}%'.format(per[i]), (x, y), ha='center',fontsize=20)
plt.show()
```



16 Finding correlation matrix of each column

corr_matrix is correlation matrix which stores the correlation values for every column

upper is the upper triangular matrix of corr_matrix

[20]:
```
corr_matrix = df.corr().abs()
upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(bool))
```

17 List of all columns which has > 0.7 correlation

drop_list stores the names of columns which has correlation higher than 0.7

[21]:
```
drop_list = [column for column in upper.columns if any(upper[column] > 0.7)]
```

18 Dropping all columns which has > 0.7 correlation

[22]:
```
df.drop(drop_list, axis=1, inplace=True)
```

19 List of all Column names with dtype object with 0 null values

obj_0nan stores the names of columns which has data type object and which do not have any null values

[23]:
```
obj_0nan = [i for i in df.columns if
            df[i].dtype == object and ~df[i].isna().any()]
```

20 List of all Column names with dtype object without 0 null values

obj_not0nan stores the names of columns which has data type object and which have null values

[24]:
```
obj_not0nan = [i for i in df.columns if
               df[i].dtype == object and df[i].isna().any()]
```

21 List of all Column names with dtype int or float with 0 null values

nonobj_0nan stores the names of columns which has data type int or float and which do not have any null values

[25]:
```
nonobj_0nan = [i for i in df.columns if df[i].dtype != object and ~df[
    i].isna().any()]
```

22 List of all Column names with dtype int or float without 0 null values

nonobj_0nan stores the names of columns which has data type int or float and which have any null values

[26]:
```
nonobj_not0nan = [i for i in df.columns if df[i].dtype != object and df[
    i].isna().any()]
```

23 Replacing all non-object null values with 0.0

null values can be replaced not only with 0.0 but can also be replaced with mean, median, mode and any aggregate function based on the feature

[27]:
```
df[nonobj_not0nan] = df[nonobj_not0nan].replace(np.nan, 0.0)
# df[nonobj_not0nan]
```

24 Encoding all object values which its column has 0 null values

Since number of unique values under each column is < 10 hence encoded with integers 0 - 7.

OneHotEnocoder or LabelEncoder can also be used for encoding.

[28]:
```
df[obj_0nan] = df[obj_0nan].replace(['H', 'S'], [1, 2])
df[obj_0nan] = df[obj_0nan].replace(['Some College', 'High School', 'College',␣
 ↪'Some High School or Less',
                                     'Graduate School'], [1, 2, 3, 4, 5])
df[obj_0nan] = df[obj_0nan].replace(['Single', 'Multiple with <5 surnames␣
 ↪without apt. #', 'Not Available',
                                     'Multiple with 5-9 surnames without apt.␣
 ↪#',
                                     'Multiple with 5-9 surnames with apt. #',
                                     'Multiple with <5 surnames with apt. #',
                                     'Multiple with 10+ surnames with apt. #',
                                     'Multiple with 10+ surnames without apt.␣
 ↪#'], [1, 2, 0, 4, 5, 3, 6, 7])
```

25 Encoding all object values which its column do not have 0 null values

All alphabet characters are encoded with integers 1-26 and null values are replaced with 0

OneHotEnocoder or LabelEncoder can also be used for encoding.

[29]:
```
mapping = {"A": 1, "B": 2, "C": 3, "D": 4, "E": 5, "F": 6, "G": 7, "H": 8, "I":␣
 ↪9, "J": 10, "K": 11, "L": 12, "M": 13,
           "N": 14, "O": 15, "P": 16, "Q": 17, "R": 18, "S": 19, "T": 20, "U":␣
 ↪21, "V": 22, "W": 23, "X": 24, "Y": 25,
           "Z": 26}
df[obj_not0nan] = df[obj_not0nan].replace(mapping)
df[obj_not0nan] = df[obj_not0nan].replace(np.nan, 0)
```

[30]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 111708 entries, 0 to 111791
Columns: 204 entries, ID to Y
dtypes: float64(7), int64(197)
memory usage: 174.7 MB
```

26 Accessing all columns except id column

Making a copy of dataframe except 1^{st} column

[31]:
```
df1 = df.iloc[:, 1:]
```

27 Normalizing all columns except id column

normalized_df is a dataframe which is normalized between 0 - 1 except id

can also be used L2 normalization

not only normalization but standardization can also be used

making a copy of orginal dataframe df to df3

[32]:
```
normalized_df = (df1 - df1.min()) / (df1.max() - df1.min())
df3 = df.copy()
df3.iloc[:, 1:] = normalized_df
```

28 Normalized dataframe

[33]:
```
df3
```

[33]:

	ID	MT_RETAILER_EMAIL_SUBSCRIBERS	MT_SPRINT_CELL_PHONE_CUSTOMER	\
0	110505	0.500000	0.540816	
1	32429	0.612245	0.459184	
2	2760	0.163265	0.897959	
3	80006	0.234694	0.510204	
4	40392	0.877551	0.163265	
…	…	…	…	
111787	69168	0.071429	0.408163	
111788	117425	0.663265	0.826531	
111789	119403	0.061224	0.000000	
111790	4981	0.234694	0.193878	
111791	44728	0.183673	0.367347	

	PROPENSITY_TO_BUY_LUX_VEH_SUV	MT_ACTIVE_ON_PINTEREST	\
0	0.387755	0.275510	
1	0.306122	0.040816	
2	0.295918	0.244898	
3	0.326531	0.234694	
4	0.785714	0.448980	
…	…	…	
111787	0.408163	0.489796	
111788	0.795918	0.816327	
111789	0.602041	0.020408	
111790	0.020408	0.091837	

```
111791                      0.816327                0.306122

        MT_LIKELY_CRUISER  MT_OPENING_WEEKEND_MOVIE_ENTHUSIASTS  \
0                0.459184                              0.642857
1                0.806122                              0.500000
2                0.316327                              0.122449
3                0.214286                              0.295918
4                0.061224                              0.826531
…                   …                                     …
111787           0.581633                              0.112245
111788           0.306122                              0.071429
111789           0.010204                              0.020408
111790           0.000000                              0.479592
111791           0.265306                              0.163265

        MT_DISCOUNT_MOVIE_ENTHUSIASTS  MT_SELF_PAY_HEALTH_INSURANCE  \
0                            0.928571                      0.459184
1                            0.489796                      0.765306
2                            0.306122                      0.540816
3                            0.030612                      0.734694
4                            1.000000                      0.204082
…                               …                             …
111787                       0.673469                      0.775510
111788                       0.061224                      0.438776
111789                       0.459184                      0.693878
111790                       0.867347                      0.183673
111791                       0.020408                      0.887755

        MT_STOCK_UP_AT_GROCERY_STORES  …  MT_PREMIUM_NATURAL_HOME_CLEANERS  \
0                            0.571429  …                          0.428571
1                            0.683673  …                          0.122449
2                            0.632653  …                          0.306122
3                            0.673469  …                          0.173469
4                            0.632653  …                          0.846939
…                               …  …                             …
111787                       0.826531  …                          0.265306
111788                       0.622449  …                          0.846939
111789                       0.091837  …                          0.000000
111790                       0.561224  …                          0.510204
111791                       0.112245  …                          0.316327

        ACT_AVG_DOLLARS_QUINT  MT_BAR_AND_LOUNGE_FOOD_ENTHUSIASTS  \
0                         0.4                            0.408163
1                         0.0                            0.255102
2                         0.8                            0.612245
3                         0.6                            0.459184
4                         0.4                            0.030612
```

```
…                      …                                  …
111787                   1.0                           0.142857
111788                   0.0                           0.540816
111789                   0.4                           0.510204
111790                   0.8                           0.428571
111791                   0.0                           0.428571

        MT_PRE_SHOP_PLANNERS  MT_CHRISTMAS_ORNAMENTSCOLLECTIBLES_BUYER  \
0                   0.622449                                  0.602041
1                   0.377551                                  0.255102
2                   0.632653                                  0.775510
3                   0.877551                                  0.551020
4                   0.663265                                  0.132653
…                      …                                       …
111787              0.469388                                  0.367347
111788              0.948980                                  0.469388
111789              0.010204                                  0.000000
111790              0.153061                                  0.020408
111791              0.500000                                  0.520408

        MT_COFFEE_ENTHUSIASTS  MT_SCENT_SEEKERS  MT_HOME_WARRANTY_PURCHASERS  \
0                    0.163265          0.816327                     1.000000
1                    0.530612          0.153061                     1.000000
2                    0.795918          0.265306                     1.000000
3                    0.316327          0.091837                     1.000000
4                    0.357143          0.836735                     1.000000
…                       …                 …                            …
111787               0.826531          0.193878                     0.336735
111788               0.540816          0.428571                     1.000000
111789               0.000000          0.020408                     0.448980
111790               0.306122          0.000000                     0.908163
111791               0.591837          0.255102                     1.000000

        MT_RETAIL_TEXTERS    Y
0                0.316327  0.0
1                0.183673  0.0
2                0.051020  1.0
3                0.275510  0.0
4                0.683673  0.0
…                   …  …
111787           0.030612  0.0
111788           0.102041  0.0
111789           0.000000  0.0
111790           0.489796  1.0
111791           0.010204  0.0

[111708 rows x 204 columns]
```

29 Finding correlation matrix of each column

corr_matrix is correlation matrix which stores the correlation values for every column

upper is the upper triangular matrix of corr_matrix

[34]:
```
corr_matrix = df3.corr().abs()
upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(bool))
```

30 List of all columns which has > 0.7 correlation

drop_list stores the names of columns which has correlation higher than 0.7

[35]:
```
drop_list = [column for column in upper.columns if any(upper[column] > 0.7)]
```

31 Dropping all columns which has > 0.7 correlation

[36]:
```
df3.drop(drop_list, axis=1, inplace=True)
```

[37]:
```
df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 111708 entries, 0 to 111791
Columns: 201 entries, ID to Y
dtypes: float64(200), int64(1)
memory usage: 172.2 MB
```

[38]:
```
df3.describe()
```

[38]:

	ID	MT_RETAILER_EMAIL_SUBSCRIBERS	MT_SPRINT_CELL_PHONE_CUSTOMER	PROPENSITY_TO_BUY_LUX_VEH_SUV
count	111708.000000	111708.000000	111708.000000	111708.000000
mean	62103.662916	0.532566	0.556956	0.460521
std	35853.725636	0.279448	0.282921	0.280119
min	0.000000	0.000000	0.000000	0.000000
25%	31083.750000	0.306122	0.326531	0.214286
50%	62130.000000	0.540816	0.591837	0.438776
75%	93133.250000	0.775510	0.795918	0.693878
max	124212.000000	1.000000		

```
max                          1.000000                          1.000000

       MT_ACTIVE_ON_PINTEREST  MT_LIKELY_CRUISER  \
count           111708.000000      111708.000000
mean                 0.531581           0.524158
std                  0.278723           0.287947
min                  0.000000           0.000000
25%                  0.306122           0.275510
50%                  0.540816           0.530612
75%                  0.765306           0.775510
max                  1.000000           1.000000

       MT_OPENING_WEEKEND_MOVIE_ENTHUSIASTS  MT_DISCOUNT_MOVIE_ENTHUSIASTS  \
count                         111708.000000                  111708.000000
mean                              0.474082                       0.422383
std                               0.282898                       0.290581
min                               0.000000                       0.000000
25%                               0.234694                       0.163265
50%                               0.459184                       0.377551
75%                               0.714286                       0.663265
max                               1.000000                       1.000000

       MT_SELF_PAY_HEALTH_INSURANCE  MT_STOCK_UP_AT_GROCERY_STORES  …  \
count                 111708.000000                  111708.000000  …
mean                      0.547173                       0.441354  …
std                       0.281913                       0.288337  …
min                       0.000000                       0.000000  …
25%                       0.316327                       0.183673  …
50%                       0.571429                       0.418367  …
75%                       0.795918                       0.683673  …
max                       1.000000                       1.000000  …

       PROPENSITY_TO_BUY_COMPACT_TRUCK  MT_PREMIUM_NATURAL_HOME_CLEANERS  \
count                    111708.000000                     111708.000000
mean                         0.578114                          0.494549
std                          0.277622                          0.280190
min                          0.000000                          0.000000
25%                          0.357143                          0.255102
50%                          0.602041                          0.489796
75%                          0.816327                          0.734694
max                          1.000000                          1.000000

       ACT_AVG_DOLLARS_QUINT  MT_BAR_AND_LOUNGE_FOOD_ENTHUSIASTS  \
count          111708.00000                      111708.000000
mean                0.38003                           0.468716
std                 0.37101                           0.276347
min                 0.00000                           0.000000
```

```
25%                  0.00000                              0.234694
50%                  0.40000                              0.459184
75%                  0.80000                              0.693878
max                  1.00000                              1.000000

       MT_PRE_SHOP_PLANNERS  MT_CHRISTMAS_ORNAMENTSCOLLECTIBLES_BUYER  \
count         111708.000000                            111708.000000
mean               0.543820                                 0.552153
std                0.277922                                 0.280601
min                0.000000                                 0.000000
25%                0.316327                                 0.336735
50%                0.561224                                 0.581633
75%                0.785714                                 0.785714
max                1.000000                                 1.000000

       MT_COFFEE_ENTHUSIASTS  MT_SCENT_SEEKERS  MT_RETAIL_TEXTERS  \
count          111708.000000     111708.000000      111708.000000
mean                0.458339          0.462945           0.460721
std                 0.283873          0.279199           0.297180
min                 0.000000          0.000000           0.000000
25%                 0.214286          0.224490           0.193878
50%                 0.438776          0.448980           0.438776
75%                 0.693878          0.693878           0.714286
max                 1.000000          1.000000           1.000000

                   Y
count  111708.000000
mean        0.149936
std         0.357010
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max         1.000000

[8 rows x 201 columns]
```

[39]:
```
df3
```

[39]:
```
            ID  MT_RETAILER_EMAIL_SUBSCRIBERS  MT_SPRINT_CELL_PHONE_CUSTOMER  \
0       110505                       0.500000                       0.540816
1        32429                       0.612245                       0.459184
2         2760                       0.163265                       0.897959
3        80006                       0.234694                       0.510204
4        40392                       0.877551                       0.163265
…        …                            …                             …
111787   69168                       0.071429                       0.408163
```

```
111788  117425                      0.663265                      0.826531
111789  119403                      0.061224                      0.000000
111790    4981                      0.234694                      0.193878
111791   44728                      0.183673                      0.367347

        PROPENSITY_TO_BUY_LUX_VEH_SUV  MT_ACTIVE_ON_PINTEREST  \
0                            0.387755                0.275510
1                            0.306122                0.040816
2                            0.295918                0.244898
3                            0.326531                0.234694
4                            0.785714                0.448980
…                               …                       …
111787                       0.408163                0.489796
111788                       0.795918                0.816327
111789                       0.602041                0.020408
111790                       0.020408                0.091837
111791                       0.816327                0.306122

        MT_LIKELY_CRUISER  MT_OPENING_WEEKEND_MOVIE_ENTHUSIASTS  \
0                0.459184                              0.642857
1                0.806122                              0.500000
2                0.316327                              0.122449
3                0.214286                              0.295918
4                0.061224                              0.826531
…                   …                                     …
111787           0.581633                              0.112245
111788           0.306122                              0.071429
111789           0.010204                              0.020408
111790           0.000000                              0.479592
111791           0.265306                              0.163265

        MT_DISCOUNT_MOVIE_ENTHUSIASTS  MT_SELF_PAY_HEALTH_INSURANCE  \
0                            0.928571                      0.459184
1                            0.489796                      0.765306
2                            0.306122                      0.540816
3                            0.030612                      0.734694
4                            1.000000                      0.204082
…                               …                             …
111787                       0.673469                      0.775510
111788                       0.061224                      0.438776
111789                       0.459184                      0.693878
111790                       0.867347                      0.183673
111791                       0.020408                      0.887755

        MT_STOCK_UP_AT_GROCERY_STORES  …  PROPENSITY_TO_BUY_COMPACT_TRUCK  \
0                            0.571429  …                        0.459184
1                            0.683673  …                        0.408163
```

```
2                          0.632653  …                          0.051020
3                          0.673469  …                          0.530612
4                          0.632653  …                          0.551020
…                             …   …                             …
111787                     0.826531  …                          0.081633
111788                     0.622449  …                          0.224490
111789                     0.091837  …                          0.479592
111790                     0.561224  …                          0.765306
111791                     0.112245  …                          0.877551

        MT_PREMIUM_NATURAL_HOME_CLEANERS  ACT_AVG_DOLLARS_QUINT  \
0                               0.428571                    0.4
1                               0.122449                    0.0
2                               0.306122                    0.8
3                               0.173469                    0.6
4                               0.846939                    0.4
…                                  …                      …
111787                          0.265306                    1.0
111788                          0.846939                    0.0
111789                          0.000000                    0.4
111790                          0.510204                    0.8
111791                          0.316327                    0.0

        MT_BAR_AND_LOUNGE_FOOD_ENTHUSIASTS  MT_PRE_SHOP_PLANNERS  \
0                                 0.408163              0.622449
1                                 0.255102              0.377551
2                                 0.612245              0.632653
3                                 0.459184              0.877551
4                                 0.030612              0.663265
…                                    …                    …
111787                            0.142857              0.469388
111788                            0.540816              0.948980
111789                            0.510204              0.010204
111790                            0.428571              0.153061
111791                            0.428571              0.500000

        MT_CHRISTMAS_ORNAMENTSCOLLECTIBLES_BUYER  MT_COFFEE_ENTHUSIASTS  \
0                                       0.602041               0.163265
1                                       0.255102               0.530612
2                                       0.775510               0.795918
3                                       0.551020               0.316327
4                                       0.132653               0.357143
…                                          …                      …
111787                                  0.367347               0.826531
111788                                  0.469388               0.540816
111789                                  0.000000               0.000000
111790                                  0.020408               0.306122
```

```
111791                                0.520408                0.591837

        MT_SCENT_SEEKERS  MT_RETAIL_TEXTERS    Y
0               0.816327           0.316327  0.0
1               0.153061           0.183673  0.0
2               0.265306           0.051020  1.0
3               0.091837           0.275510  0.0
4               0.836735           0.683673  0.0
…                    …                  …  …
111787          0.193878           0.030612  0.0
111788          0.428571           0.102041  0.0
111789          0.020408           0.000000  0.0
111790          0.000000           0.489796  1.0
111791          0.255102           0.010204  0.0

[111708 rows x 201 columns]
```

[40]:
```
#heatmap of upper triangular matrix
# upper.style.background_gradient(cmap='coolwarm')
```