```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

dataset = pd.read_csv('CC_GENERAL.csv')

dataset.info()
```

```
     CUST_ID      BALANCE  BALANCE_FREQUENCY  PURCHASES
ONEOFF_PURCHASES  \
0     C10001    40.900749           0.818182      95.40
0.00
1     C10002  3202.467416           0.909091       0.00
0.00
2     C10003  2495.148862           1.000000     773.17
773.17
3     C10004  1666.670542           0.636364    1499.00
1499.00
4     C10005   817.714335           1.000000      16.00
16.00
...      ...          ...                ...        ...
...
8945  C19186    28.493517           1.000000     291.12
0.00
8946  C19187    19.183215           1.000000     300.00
0.00
8947  C19188    23.398673           0.833333     144.40
0.00
8948  C19189    13.457564           0.833333       0.00
0.00
8949  C19190   372.708075           0.666667    1093.25
1093.25

      INSTALLMENTS_PURCHASES  CASH_ADVANCE  PURCHASES_FREQUENCY  \
0                      95.40      0.000000             0.166667
1                       0.00   6442.945483             0.000000
2                       0.00      0.000000             1.000000
3                       0.00    205.788017             0.083333
4                       0.00      0.000000             0.083333
...                      ...           ...                  ...
8945                  291.12      0.000000             1.000000
8946                  300.00      0.000000             1.000000
8947                  144.40      0.000000             0.833333
8948                    0.00     36.558778             0.000000
8949                    0.00    127.040008             0.666667

      ONEOFF_PURCHASES_FREQUENCY  PURCHASES_INSTALLMENTS_FREQUENCY  \
```

```
0                        0.000000                             0.083333
1                        0.000000                             0.000000
2                        1.000000                             0.000000
3                        0.083333                             0.000000
4                        0.083333                             0.000000
...                           ...                                  ...
8945                     0.000000                             0.833333
8946                     0.000000                             0.833333
8947                     0.000000                             0.666667
8948                     0.000000                             0.000000
8949                     0.666667                             0.000000

      CASH_ADVANCE_FREQUENCY  CASH_ADVANCE_TRX  PURCHASES_TRX
CREDIT_LIMIT  \
0                   0.000000                 0              2
1000.0
1                   0.250000                 4              0
7000.0
2                   0.000000                 0             12
7500.0
3                   0.083333                 1              1
7500.0
4                   0.000000                 0              1
1200.0
...                      ...               ...            ...
...
8945                0.000000                 0              6
1000.0
8946                0.000000                 0              6
1000.0
8947                0.000000                 0              5
1000.0
8948                0.166667                 2              0
500.0
8949                0.333333                 2             23
1200.0

         PAYMENTS  MINIMUM_PAYMENTS  PRC_FULL_PAYMENT  TENURE
0      201.802084         139.509787          0.000000      12
1     4103.032597        1072.340217          0.222222      12
2      622.066742         627.284787          0.000000      12
3        0.000000                NaN          0.000000      12
4      678.334763         244.791237          0.000000      12
...           ...                ...               ...     ...
8945   325.594462          48.886365          0.500000       6
8946   275.861322                NaN          0.000000       6
8947    81.270775          82.418369          0.250000       6
8948    52.549959          55.755628          0.250000       6
8949    63.165404          88.288956          0.000000       6
```

```
[8950 rows x 18 columns]
```

```
print(dataset.dtypes)
```

```
CUST_ID                            object
BALANCE                           float64
BALANCE_FREQUENCY                 float64
PURCHASES                         float64
ONEOFF_PURCHASES                  float64
INSTALLMENTS_PURCHASES            float64
CASH_ADVANCE                      float64
PURCHASES_FREQUENCY               float64
ONEOFF_PURCHASES_FREQUENCY        float64
PURCHASES_INSTALLMENTS_FREQUENCY  float64
CASH_ADVANCE_FREQUENCY            float64
CASH_ADVANCE_TRX                    int64
PURCHASES_TRX                       int64
CREDIT_LIMIT                      float64
PAYMENTS                          float64
MINIMUM_PAYMENTS                  float64
PRC_FULL_PAYMENT                  float64
TENURE                              int64
dtype: object
```

```
dataset.isnull().sum()
```

```
CUST_ID                            0
BALANCE                            0
BALANCE_FREQUENCY                  0
PURCHASES                          0
ONEOFF_PURCHASES                   0
INSTALLMENTS_PURCHASES             0
CASH_ADVANCE                       0
PURCHASES_FREQUENCY                0
ONEOFF_PURCHASES_FREQUENCY         0
PURCHASES_INSTALLMENTS_FREQUENCY   0
CASH_ADVANCE_FREQUENCY             0
CASH_ADVANCE_TRX                   0
PURCHASES_TRX                      0
CREDIT_LIMIT                       1
PAYMENTS                           0
MINIMUM_PAYMENTS                 313
PRC_FULL_PAYMENT                   0
TENURE                             0
dtype: int64
```

```
dataset.describe().T
```

```
                                   count         mean           std
min  \
```

```
BALANCE                             8950.0  1564.474828  2081.531879
0.000000
BALANCE_FREQUENCY                   8950.0     0.877271     0.236904
0.000000
PURCHASES                           8950.0  1003.204834  2136.634782
0.000000
ONEOFF_PURCHASES                    8950.0   592.437371  1659.887917
0.000000
INSTALLMENTS_PURCHASES              8950.0   411.067645   904.338115
0.000000
CASH_ADVANCE                        8950.0   978.871112  2097.163877
0.000000
PURCHASES_FREQUENCY                 8950.0     0.490351     0.401371
0.000000
ONEOFF_PURCHASES_FREQUENCY          8950.0     0.202458     0.298336
0.000000
PURCHASES_INSTALLMENTS_FREQUENCY    8950.0     0.364437     0.397448
0.000000
CASH_ADVANCE_FREQUENCY              8950.0     0.135144     0.200121
0.000000
CASH_ADVANCE_TRX                    8950.0     3.248827     6.824647
0.000000
PURCHASES_TRX                       8950.0    14.709832    24.857649
0.000000
CREDIT_LIMIT                        8949.0  4494.449450  3638.815725
50.000000
PAYMENTS                            8950.0  1733.143852  2895.063757
0.000000
MINIMUM_PAYMENTS                    8637.0   864.206542  2372.446607
0.019163
PRC_FULL_PAYMENT                    8950.0     0.153715     0.292499
0.000000
TENURE                              8950.0    11.517318     1.338331
6.000000

                                           25%          50%
75%  \
BALANCE                             128.281915   873.385231
2054.140036
BALANCE_FREQUENCY                     0.888889     1.000000
1.000000
PURCHASES                            39.635000   361.280000
1110.130000
ONEOFF_PURCHASES                      0.000000    38.000000
577.405000
INSTALLMENTS_PURCHASES                0.000000    89.000000
468.637500
CASH_ADVANCE                          0.000000     0.000000
1113.821139
```

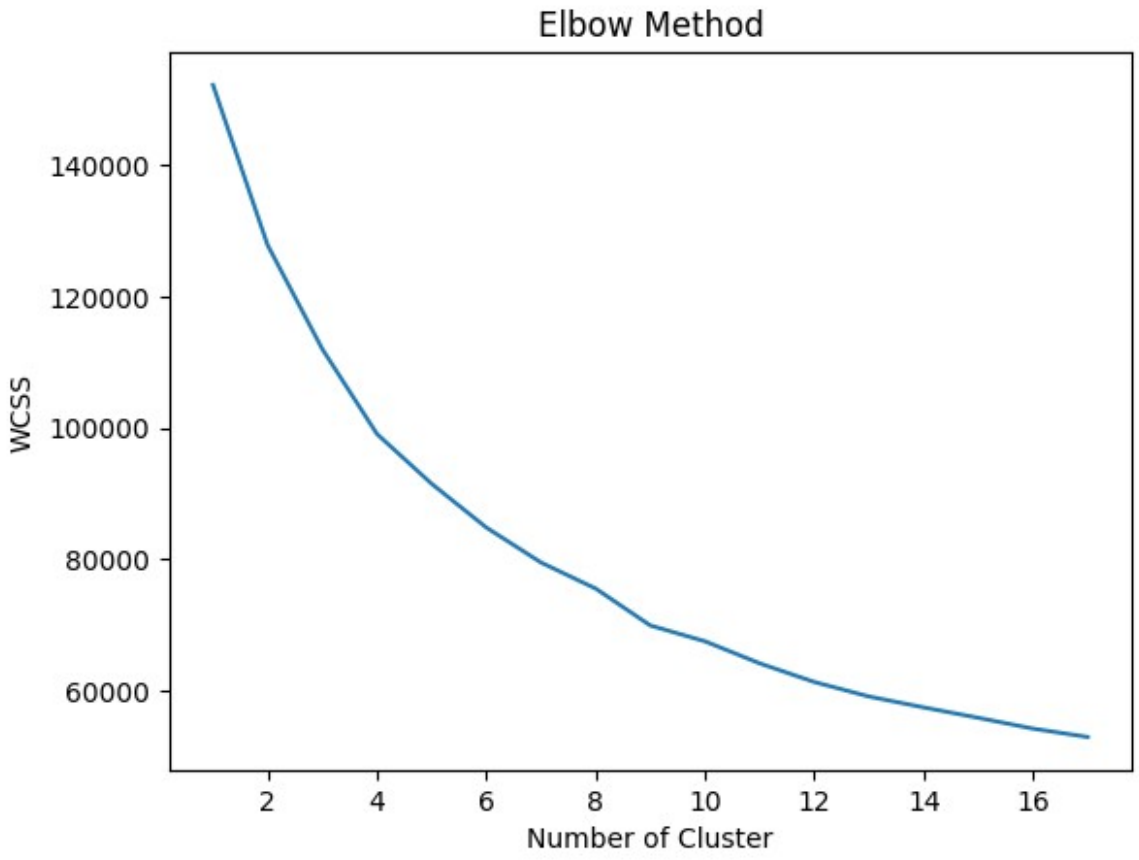
```
PURCHASES_FREQUENCY                     0.083333     0.500000
0.916667
ONEOFF_PURCHASES_FREQUENCY              0.000000     0.083333
0.300000
PURCHASES_INSTALLMENTS_FREQUENCY        0.000000     0.166667
0.750000
CASH_ADVANCE_FREQUENCY                  0.000000     0.000000
0.222222
CASH_ADVANCE_TRX                        0.000000     0.000000
4.000000
PURCHASES_TRX                           1.000000     7.000000
17.000000
CREDIT_LIMIT                         1600.000000  3000.000000
6500.000000
PAYMENTS                              383.276166   856.901546
1901.134317
MINIMUM_PAYMENTS                      169.123707   312.343947
825.485459
PRC_FULL_PAYMENT                        0.000000     0.000000
0.142857
TENURE                                12.000000    12.000000
12.000000
```

```
                                          max
BALANCE                              19043.13856
BALANCE_FREQUENCY                        1.00000
PURCHASES                            49039.57000
ONEOFF_PURCHASES                     40761.25000
INSTALLMENTS_PURCHASES               22500.00000
CASH_ADVANCE                         47137.21176
PURCHASES_FREQUENCY                      1.00000
ONEOFF_PURCHASES_FREQUENCY               1.00000
PURCHASES_INSTALLMENTS_FREQUENCY         1.00000
CASH_ADVANCE_FREQUENCY                   1.50000
CASH_ADVANCE_TRX                       123.00000
PURCHASES_TRX                          358.00000
CREDIT_LIMIT                         30000.00000
PAYMENTS                             50721.48336
MINIMUM_PAYMENTS                     76406.20752
PRC_FULL_PAYMENT                         1.00000
TENURE                                  12.00000
```

```
X = dataset.iloc[:, 1:].values

plt.figure(figsize=(12,12))
sns.heatmap(dataset.corr(), annot=True)
plt.show()
```


```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values= np.NAN, strategy= 'mean',
fill_value=None, verbose=0, copy=True)
imputer = imputer.fit(X)
X = imputer.transform(X)

from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X = sc_X.fit_transform(X)
```

```
from sklearn.cluster import KMeans
wcss = []
for i in range(1,18):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300,
n_init=10, random_state=0).fit(X)
    wcss.append(kmeans.inertia_)

plt.plot(range(1,18), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of Cluster')
plt.ylabel('WCSS')
plt.show()
```




```
kmeans = KMeans(n_clusters=8, init='k-means++', max_iter=300,
n_init=10, random_state=0)
y_kmeans = kmeans.fit_predict(X)

y_kmeans
```

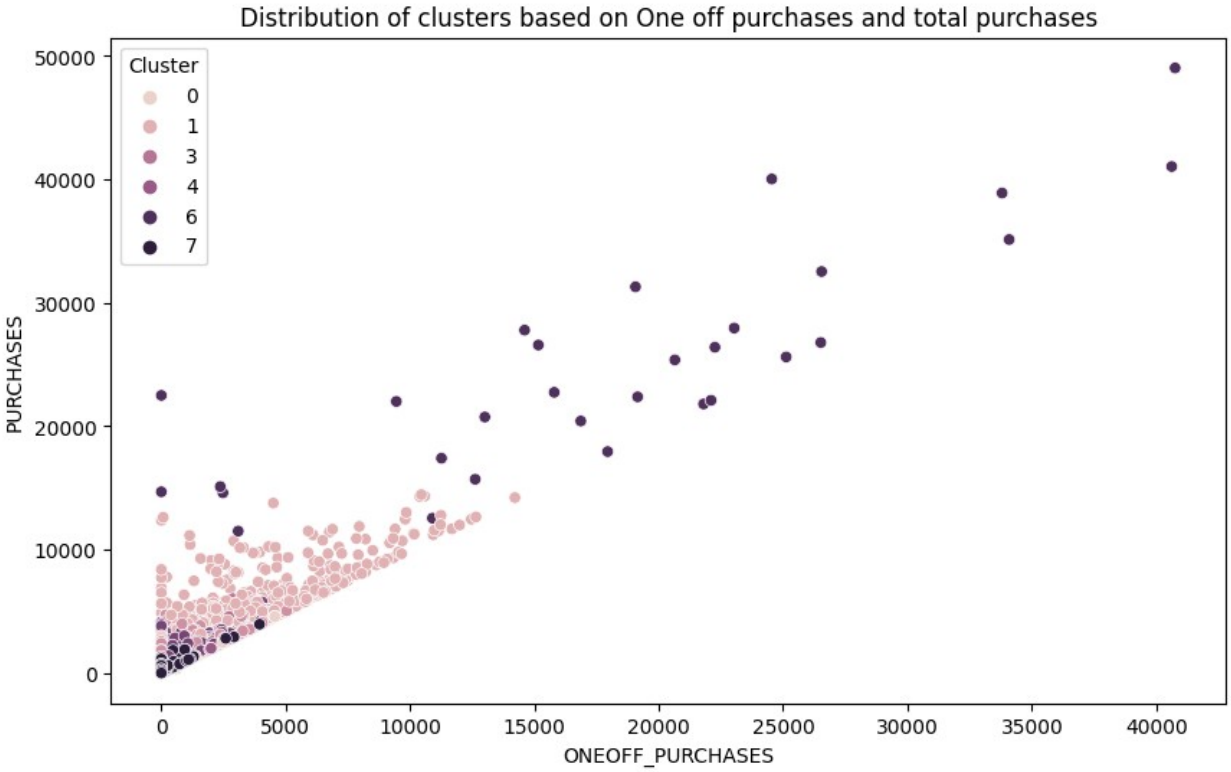
```
array([0, 3, 5, ..., 7, 7, 7], dtype=int32)
```

```
dataset['Cluster'] = y_kmeans
dataset.head()
```

```
  CUST_ID      BALANCE  BALANCE_FREQUENCY  PURCHASES  ONEOFF_PURCHASES
\
0  C10001    40.900749           0.818182      95.40              0.00

1  C10002  3202.467416           0.909091       0.00              0.00

2  C10003  2495.148862           1.000000     773.17            773.17

3  C10004  1666.670542           0.636364    1499.00           1499.00

4  C10005   817.714335           1.000000      16.00             16.00


   INSTALLMENTS_PURCHASES  CASH_ADVANCE  PURCHASES_FREQUENCY  \
0                    95.4      0.000000             0.166667
1                     0.0   6442.945483             0.000000
2                     0.0      0.000000             1.000000
3                     0.0    205.788017             0.083333
4                     0.0      0.000000             0.083333

   ONEOFF_PURCHASES_FREQUENCY  PURCHASES_INSTALLMENTS_FREQUENCY  \
0                    0.000000                          0.083333
1                    0.000000                          0.000000
2                    1.000000                          0.000000
3                    0.083333                          0.000000
4                    0.083333                          0.000000

   CASH_ADVANCE_FREQUENCY  CASH_ADVANCE_TRX  PURCHASES_TRX
CREDIT_LIMIT  \
0                0.000000                 0              2
1000.0
1                0.250000                 4              0
7000.0
2                0.000000                 0             12
7500.0
3                0.083333                 1              1
7500.0
4                0.000000                 0              1
1200.0

      PAYMENTS  MINIMUM_PAYMENTS  PRC_FULL_PAYMENT  TENURE  Cluster
0   201.802084        139.509787          0.000000      12        0
1  4103.032597       1072.340217          0.222222      12        3
2   622.066742        627.284787          0.000000      12        5
3     0.000000               NaN          0.000000      12        0
4   678.334763        244.791237          0.000000      12        0
```

```
plt.figure(figsize=(10,6))
sns.scatterplot(data=dataset, x='ONEOFF_PURCHASES', y='PURCHASES',
hue='Cluster')
```

```
plt.title('Distribution of clusters based on One off purchases and
total purchases')
plt.show()
```



Conclusion:

By segmenting the data into 7 clusters, we have effectively categorized the dataset based on distinct characteristics or patterns. These segments can provide valuable insights for formulating targeted marketing strategies. Since each segment exhibits different purchase capacities, understanding these variations allows for tailored approaches to maximize effectiveness and meet the unique needs of each customer group.