

NAVNEEL MONDAL

Reg No: 21BCE2654

DATE: 05-10-2023

AI ML ASSIGNMENT-5

1. Download the dataset: mailcustomersegmentaion.csv is downloaded.

2. Load The dataset:

```
1 # import required libraries
2
3 import pandas as pd
4 import numpy as np
5 import matplotlib.pyplot as plt
6 import seaborn as sns
```

3. Understanding the Data

```
8
9 df = pd.read_csv('/content/Mall_Customers.csv')
10 df.head()
11
```

	CustomerID	Gender	Age	Annual Income (k\$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

```
11 df.head()
12 df.shape
13
```

```
: (200, 5)
```

```
14 df.info()
15
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
#   Column                Non-Null Count  Dtype
---  -
0   CustomerID            200 non-null    int64
1   Gender                200 non-null    object
2   Age                   200 non-null    int64
3   Annual Income (k$)    200 non-null    int64
4   Spending Score (1-100) 200 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

```
15
16 df.isnull().sum()
17
```

```
CustomerID      0
Gender          0
Age             0
Annual Income (k$) 0
Spending Score (1-100) 0
dtype: int64
```

```
17 df.describe()
18
19
```

	CustomerID	Age	Annual Income (k\$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

4.Data preprocessing:

```
19
20 from sklearn import cluster
21
22 ew_df = df.iloc[:, -2:]
23 new_df.head()
24
```

	Annual Income (k\$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40

```
25 error=[]
26 for i in range(1,11):
27     kmeans = cluster.KMeans(n_clusters=i,init = 'k-means++',random_state=4)
28     kmeans.fit(new_df)
29     error.append(kmeans.inertia_)
30
```

[illegible]


```
43 # Testing the model with random observation
44
45 km_model.predict([[60,50]])
```

```
array([1], dtype=int32)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but KMeans was fitted with feature names
  warnings.warn(
```

```
46
47 km_model.predict([[15,1]])
48
```

```
array([4], dtype=int32)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but KMeans was fitted with feature names
  warnings.warn(
```

```
48
49 km_model.predict([[41,34]])
50
```

```
array([4], dtype=int32)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but KMeans was fitted with feature names
  warnings.warn(
```

```
50
51 km_model.predict([[137,99]])
52
```

```
array([2], dtype=int32)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but KMeans was fitted with feature names
  warnings.warn(
```

```
52
53 km_model.predict([[78,73]])
54
55
```

```
array([2], dtype=int32)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but KMeans was fitted with feature names
  warnings.warn(
```