

SUBMISSION DATE: 05.10.2023

# ASSIGNMENT – 5

Artificial Intelligence & Machine Learning in collaboration  
with Google (Applied Data Science)

**Name:** Diya Maria Sanjay

**Regno:** 21BML0113

**Branch:** Btech ECE with specialization in Biomedical  
Engineering

**Campus:** VIT Vellore

# Market Basket Magic: Extracting Insights for Retail Success

## TASK

### Understand the data

```
7s ▶ import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

UNDERSTANDING THE DATA

```
0s ▶ df = pd.read_csv('/content/Mall_Customers.csv')
df.head()
```

	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

```
0s [3] df.shape
(200, 5)

0s [4] df.info()
<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
```

```
[5] df.isnull().sum()
```

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

```
[6] df.describe()
```

	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

## Data Preprocessing

## DATA PREPROCESSING

```
[8] from sklearn import cluster
new_df = df.iloc[:, -2:]
new_df.head()
```

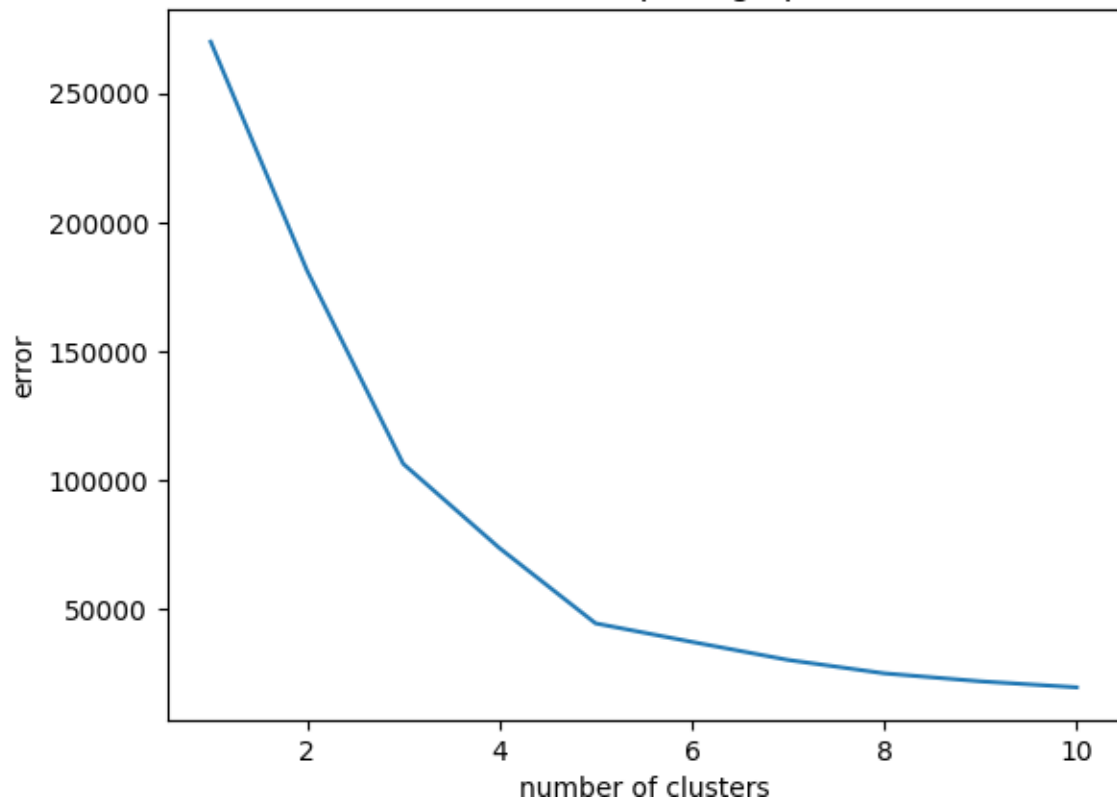
	Annual Income (k\$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40

```
[9] error=[]  
for i in range(1,11):  
    kmeans = cluster.KMeans(n_clusters=i,init = 'k-means+',random_state=4)  
    kmeans.fit(new_df)  
    error.append(kmeans.inertia)
```

[illegible]

```
plt.plot(range(1,11),error)
plt.title('The Elbow point graph')
plt.xlabel('number of clusters')
plt.ylabel('error')
plt.show()
```

The Elbow point graph



## Machine Learning approach with clustering algorithm

## Machine Learning approach with K-Means Clustering Algorithm

```
km_model = cluster.KMeans(n_clusters=5, init = 'k-means++', random_state=0)
km_model.fit(new_df)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in version 1.3. To suppress this warning, please pass `n_init` to the constructor.
  warnings.warn(
```

KMeans

KMeans(n\_clusters=5, random\_state=0)

[illegible]

```
[13] # Testing the model with random observation

km_model.predict([[60,50]])

/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
warnings.warn(
array([1], dtype=int32)
```

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

```
km_model.predict([[15,1]])
```

```
/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(
```

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

```
km_model.predict([[41,34]])
```

```
/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(
```

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

```
km_model.predict([[137,99]])
```

```
/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(
```

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

```
km model.predict([[78,73]])
```

```
/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(
```

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