

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

```
df=pd.read_csv('/content/Mall_Customers.csv')
df.columns = df.columns.str.replace(' ', '_')
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

```
df.shape
```

(200, 5)

```
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
```

```
df.isnull().any()
```

```
CustomerID      False
Gender          False
Age             False
Annual_Income_(k$) False
Spending_Score_(1-100) False
dtype: bool
```

```
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

```
df.corr()
```

```
<ipython-input-10-2f6f6606aa2c>:1: FutureWarning: The default value of numeric only in DataFrame.corr is deprecated. In a future ver
```

DATA PREPROCESSING

```
from sklearn import cluster
```

Age	-0.026763	1.000000	-0.012398	-0.327227
------------	-----------	----------	-----------	-----------

```
df2=df.iloc[:, -2:]
```

```
df2.head()
```

	Annual_Income_(k\$)	Spending_Score_(1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40

```
error=[]
```

```
for i in range(1,11):
```

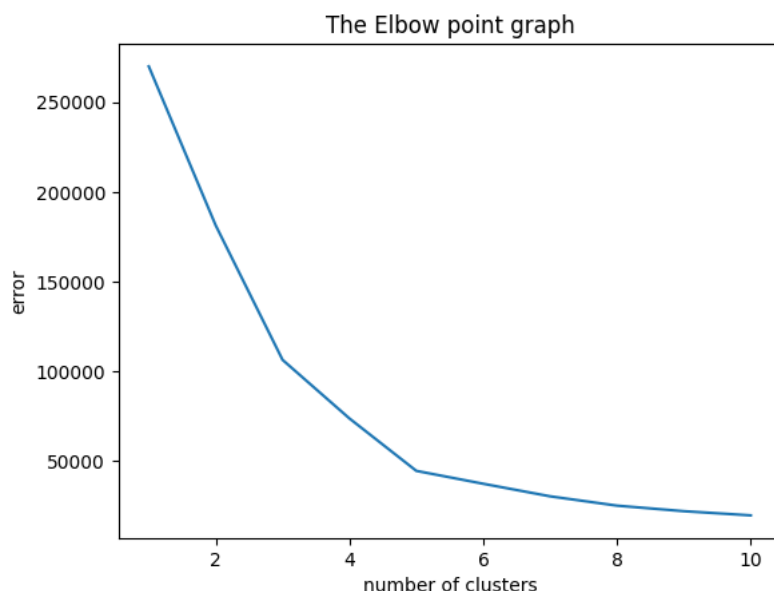
```
kmeans = cluster.KMeans(n_clusters=i,init = 'k-means++',random_state=4)
```

```
kmeans.fit(df2)
```

```
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()
```



ML APPROACH

```
km_model = cluster.KMeans(n_clusters=5,init = 'k-means++',random_state=0)
km_model.fit(df2)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
warnings.warn(
```

```
▼ KMeans
KMeans(n_clusters=5, random_state=0)
```

```
pred = km_model.predict(df2)
pred
```

```
array([4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3,
       4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 1,
       4, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
       0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
       0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
       0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
       0, 2], dtype=int32)
```

TESTING WITH RANDOM OBSERVATIONS

```
km_model.predict([[50,10]])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but KMeans was fitted
warnings.warn(
array([4], dtype=int32)
```

```
km_model.predict([[7,11]])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but KMeans was fitted
warnings.warn(
array([4], dtype=int32)
```

```
km_model.predict([[10,6]])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but KMeans was fitted
warnings.warn(
array([4], dtype=int32)
```