

Customer_Segmentation_Using_K-Means_Clustering

```
In [70]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Import and Understand the data

```
In [71]: df = pd.read_csv('Mall_Customers.csv')
```

```
In [72]: df.head()
```

```
Out[72]:
```

	CustomerID	Genre	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

```
In [73]: df.tail()
```

```
Out[73]:
```

	CustomerID	Genre	Age	Annual Income (k\$)	Spending Score (1-100)
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

```
In [74]: df.shape
```

```
Out[74]: (200, 5)
```

In [75]: 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   Genre                                 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
```

In [76]: X = df.iloc[:,[3,4]].values

In [77]: X

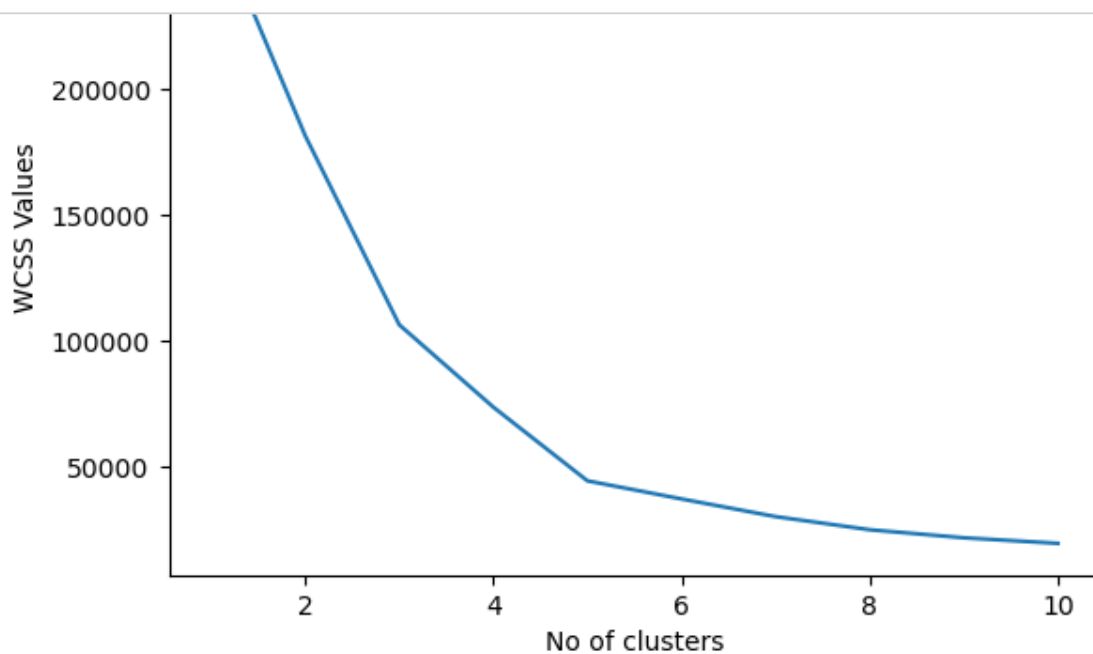
```
[ 57,  58],
[ 57,  55],
[ 58,  60],
[ 58,  46],
[ 59,  55],
[ 59,  41],
[ 60,  49],
[ 60,  40],
[ 60,  42],
[ 60,  52],
[ 60,  47],
[ 60,  50],
[ 61,  42],
[ 61,  49],
[ 62,  41],
[ 62,  48],
[ 62,  59],
[ 62,  55],
[ 62,  56],
[ 62,  42]
```

Perform Elbow Method to find Optimal No. of Clusters

In [78]: **from** sklearn.cluster **import** KMeans
wcss = []

In [79]: **for** i **in** range(1, 11):
 kmeans = KMeans(n_clusters=i, init='k-means++', random_state=0)
 kmeans.fit(X)
 wcss.append(kmeans.inertia_)

```
In [80]: plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('No of clusters')
plt.ylabel('WCSS Values')
plt.show()
```



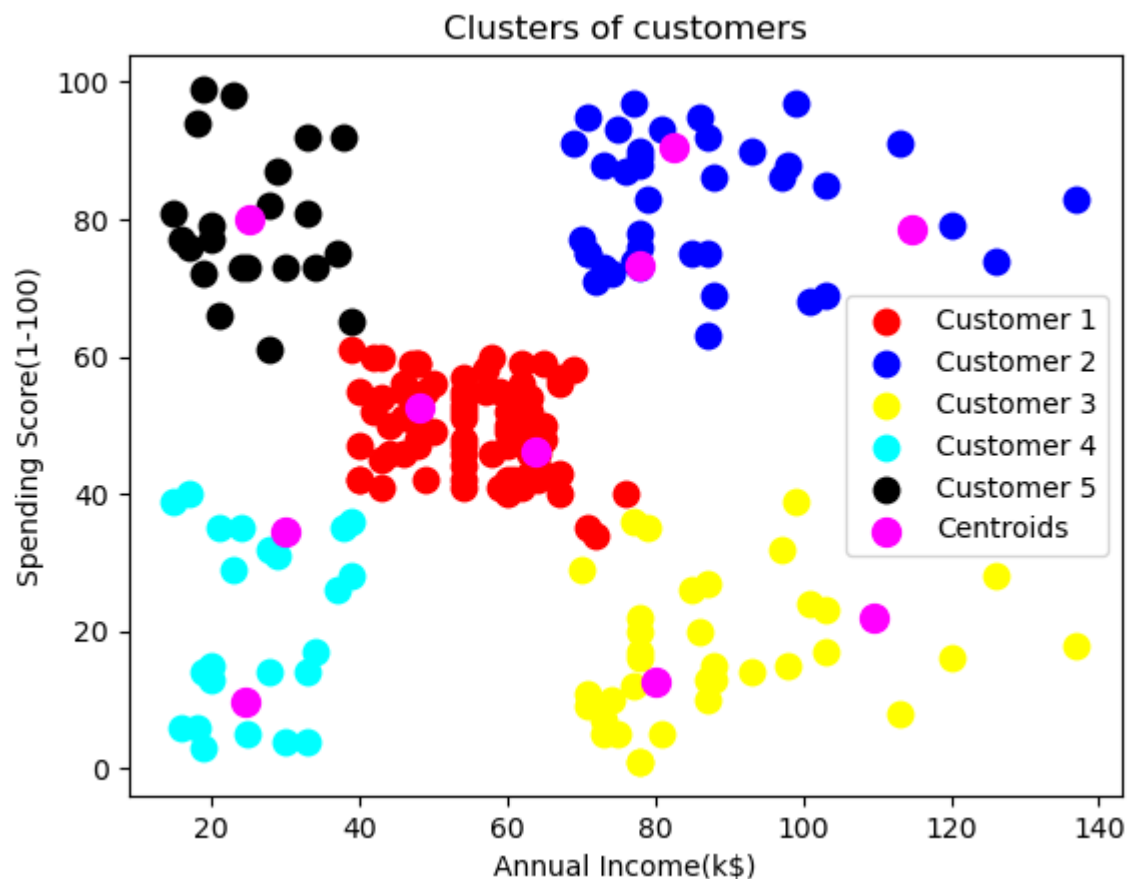
Training a model using Unsupervised Learning Algorithm

```
In [81]: kmeansmodel = KMeans(n_clusters=5, init='k-means++', random_state=0)
```

```
In [82]: y_kmeans = kmeansmodel.fit_predict(X)
```

```
In [83]: plt.scatter(X[y_kmeans == 0,0], X[y_kmeans == 0,1], s=80, c='red', label='Cu
plt.scatter(X[y_kmeans == 1,0], X[y_kmeans == 1,1], s=80, c='blue', label='C
plt.scatter(X[y_kmeans == 2,0], X[y_kmeans == 2,1], s=80, c='yellow', label=
plt.scatter(X[y_kmeans == 3,0], X[y_kmeans == 3,1], s=80, c='cyan', label='C
plt.scatter(X[y_kmeans == 4,0], X[y_kmeans == 4,1], s=80, c='black', label='

plt.scatter(kmeans.cluster_centers[:,0], kmeans.cluster_centers[:,1], s=100, c='m
plt.title('Clusters of customers')
plt.xlabel('Annual Income(k$)')
plt.ylabel('Spending Score(1-100)')
plt.legend()
plt.show()
```



THANK YOU!!!

GitHub: <https://github.com/anujtiwari21?tab=repositories>
[\(https://github.com/anujtiwari21?tab=repositories\)](https://github.com/anujtiwari21?tab=repositories)