```
# 6A Clustering algorithms for unsupervised classification.
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
df=pd.read_csv("/content/drive/MyDrive/KRAI/Mall_Customers_dataset.csv")
df.head()
```

	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

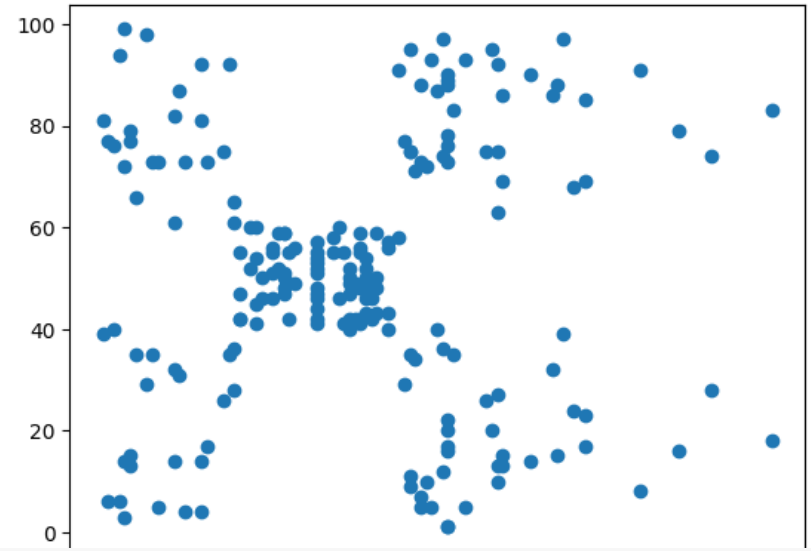
```
X=df[['Annual Income (k$)','Spending Score (1-100)']]
X
```

	Annual Income (k$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40
...	...	...
195	120	79
196	126	28
197	126	74
198	137	18
199	137	83

200 rows × 2 columns

```
plt.scatter(X['Annual Income (k$)'],X['Spending Score (1-100)'])
```

```
<matplotlib.collections.PathCollection at 0x7d84d5a3d930>
```




```
from sklearn.cluster import KMeans
model= KMeans(n_clusters=5)
model.fit(X)
```

```
/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 1.4. Set the value of `n_
  warnings.warn(
```

▾ KMeans

KMeans(n_clusters=5)

```
model.cluster_centers_
```

```
array([[88.2       , 17.11428571],
       [55.2962963 , 49.51851852],
       [25.72727273, 79.36363636],
       [86.53846154, 82.12820513],
       [26.30434783, 20.91304348]])
```

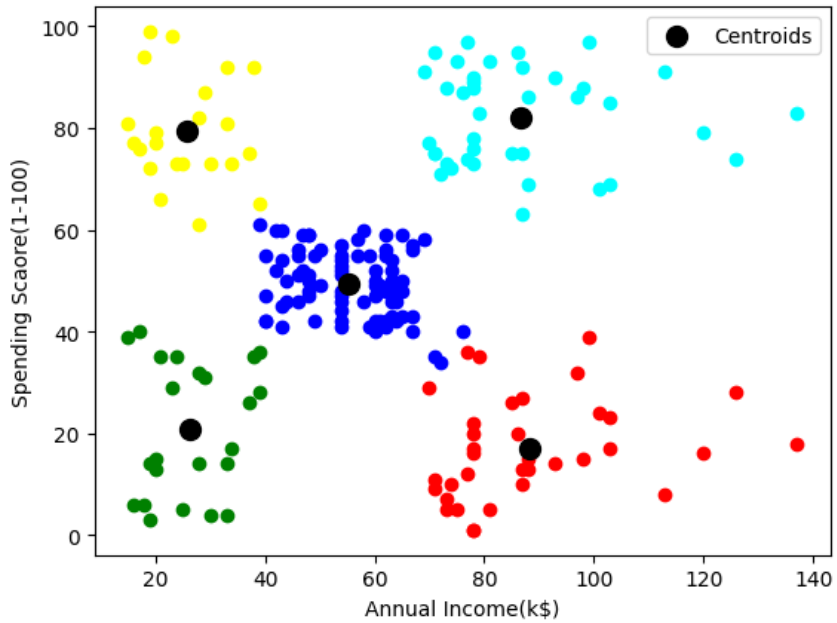
```
cluster_number=model.predict(X)
```

```
len(cluster_number)
```

```
200
```

```
c0=X[cluster_number==0]
c1=X[cluster_number==1]
c2=X[cluster_number==2]
c3=X[cluster_number==3]
c4=X[cluster_number==4]
```

```
plt.scatter(c0['Annual Income (k$)'],c0['Spending Score (1-100)'],c='red')
plt.scatter(c1['Annual Income (k$)'],c1['Spending Score (1-100)'],c='blue')
plt.scatter(c2['Annual Income (k$)'],c2['Spending Score (1-100)'],c='yellow')
plt.scatter(c3['Annual Income (k$)'],c3['Spending Score (1-100)'],c='cyan')
plt.scatter(c4['Annual Income (k$)'],c4['Spending Score (1-100)'],c='green')
plt.scatter(model.cluster_centers_[:,0],model.cluster_centers_[:,1],s=100,c='black',label='Centroids')
plt.xlabel('Annual Income(k$)')
plt.ylabel('Spending Scaore(1-100)')
plt.legend()
plt.show()
```

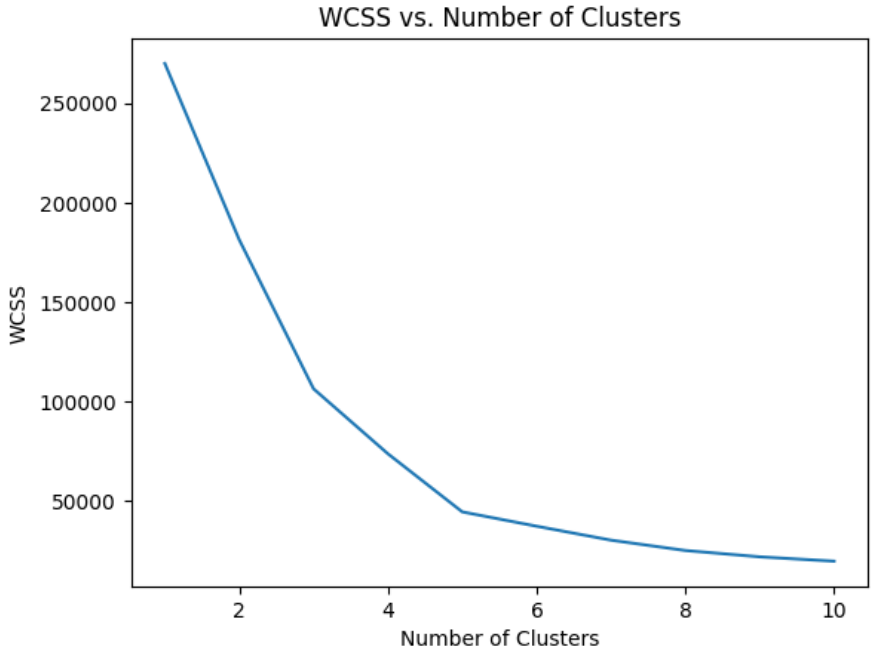



```
model.inertia_
```

```
44448.4554479337
```

```
WCSS = []
for i in range(1, 11):
  model = KMeans(n_clusters=i, n_init=10)  # Explicitly set n_init to suppress warning
  model.fit(X)
  WCSS.append(model.inertia_)

plt.plot(range(1, 11), WCSS)
plt.title('WCSS vs. Number of Clusters')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.show()
```




```
WCSS
```

```
[269981.28,
 181363.59595959593,
 106348.37306211122,
 73679.78903948836,
 44448.4554479337,
 37233.814510710006,
 30273.394312070042,
 25043.89004329005,
 21838.86369282892,
 19636.753964898147]
```

```
plt.plot(range(1,11),WCSS,marker='x')
```

```
[<matplotlib.lines.Line2D at 0x7d84c85ac370>]
```