```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
```

```
df=pd.read_csv('/content/Mall_Customers.csv')
df
```

	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
...	...	...	...	...	...
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

200 rows × 5 columns

```
#preprocessing
```

```
df.isnull().sum()
```

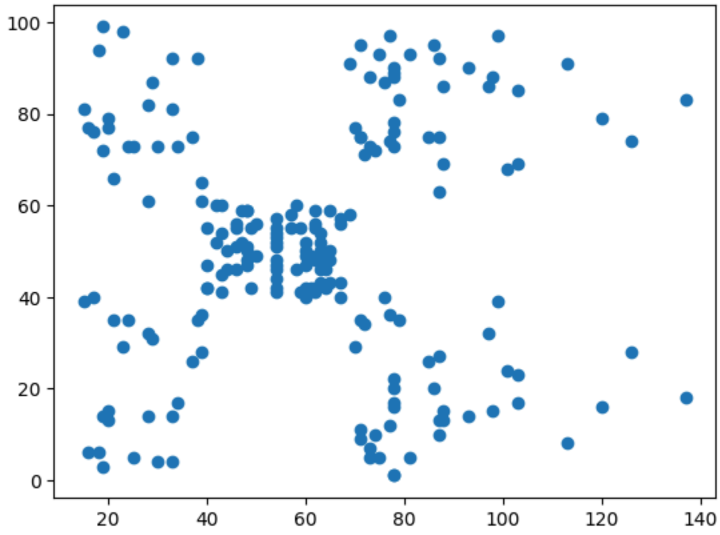
```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

```
df.duplicated().sum()
```

```
0
```

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

```
<matplotlib.collections.PathCollection at 0x7eeb0c997160>
```



```
#using elbow method to decide the number of cluster to be formed
krange=range(1,9)
sse=[]
for k in krange:
  model1=KMeans(n_clusters=k)
  model1.fit(df[['Annual Income (k$)','Spending Score (1-100)']])
  sse.append(model1.inertia_)
```

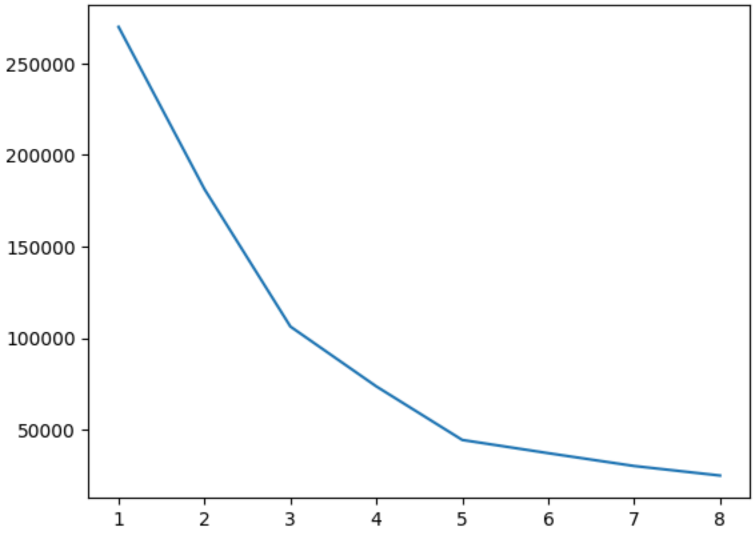
```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
```

```
sse
```

```
[269981.28,
 181363.59595959593,
 106348.37306211122,
 73679.78903948836,
 44448.4554479337,
 37239.835542456036,
 30227.606513152008,
 25018.57633477634]
```

```
plt.plot(krange,sse)
```

```
[<matplotlib.lines.Line2D at 0x7eeb0a7c99c0>]
```



```
#n_cluster=5

scaler=MinMaxScaler()
scaler.fit(df[['Annual Income (k$)']])
df['Annual Income (k$)']=scaler.transform(df[['Annual Income (k$)']])

scaler.fit(df[['Spending Score (1-100)']])
df['Spending Score (1-100)']=scaler.transform(df[['Spending Score (1-100)']])
```

```
df
```

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	0.000000	0.387755
1	2	Male	21	0.000000	0.816327
2	3	Female	20	0.008197	0.051020
3	4	Female	23	0.008197	0.775510
4	5	Female	31	0.016393	0.397959
...	...	...	...	...	...
195	196	Female	35	0.860656	0.795918
196	197	Female	45	0.909836	0.275510
197	198	Male	32	0.909836	0.744898
198	199	Male	32	1.000000	0.173469
199	200	Male	30	1.000000	0.836735

200 rows × 5 columns

```
model1=KMeans(n_clusters=5)
model1.fit(df[['Annual Income (k$)','Spending Score (1-100)']])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
▾      KMeans
KMeans(n_clusters=5)
```

```
y=model1.predict(df[['Annual Income (k$)','Spending Score (1-100)']])
y
```

```
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, 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, 0, 2, 0, 1, 0, 2, 0, 2, 0,
       1, 0, 2, 0, 2, 0, 2, 0, 2, 0, 1, 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], dtype=int32)
```

```
model1.cluster_centers_
```

```
array([[0.58638083, 0.82783883],
       [0.33029751, 0.49508692],
       [0.6       , 0.16443149],
       [0.08792846, 0.79962894],
       [0.09265859, 0.20319432]])
```

```
df['New_Cluster']=y
df
```

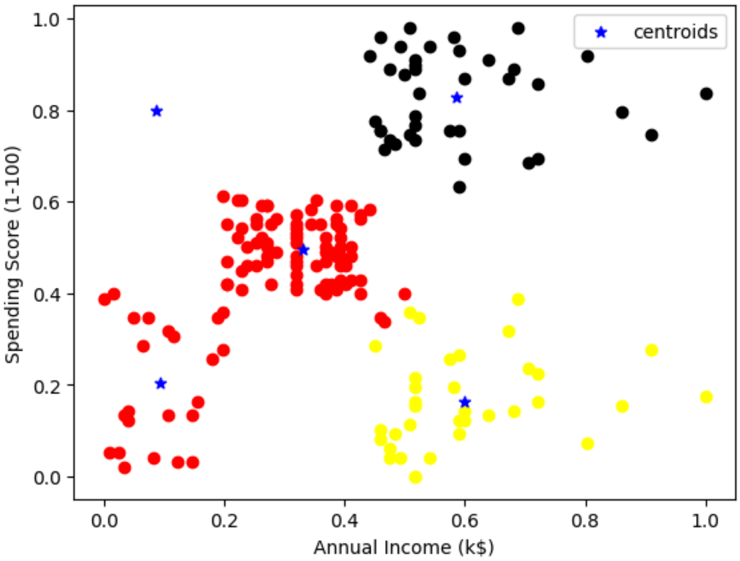
	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	New_Cluster
0	1	Male	19	0.000000	0.387755	4
1	2	Male	21	0.000000	0.816327	3
2	3	Female	20	0.008197	0.051020	4
3	4	Female	23	0.008197	0.775510	3
4	5	Female	31	0.016393	0.397959	4
...	...	...	...	...	...	...
195	196	Female	35	0.860656	0.795918	0
196	197	Female	45	0.909836	0.275510	2
197	198	Male	32	0.909836	0.744898	0
198	199	Male	32	1.000000	0.173469	2
199	200	Male	30	1.000000	0.836735	0

200 rows × 6 columns

```
df1=df[df.New_Cluster==0]
df2=df[df.New_Cluster==1]
df3=df[df.New_Cluster==2]
df4=df[df.New_Cluster==4]
df5=df[df.New_Cluster==5]


plt.scatter(df1['Annual Income (k$)'],df1['Spending Score (1-100)'],color="black")
plt.scatter(df2['Annual Income (k$)'],df2['Spending Score (1-100)'],color="red")
plt.scatter(df3['Annual Income (k$)'],df3['Spending Score (1-100)'],color="yellow")
plt.scatter(df4['Annual Income (k$)'],df4['Spending Score (1-100)'],color="red")
plt.scatter(df5['Annual Income (k$)'],df5['Spending Score (1-100)'],color="gray")
plt.scatter(model1.cluster_centers_[:,0],model1.cluster_centers_[:,1],color="blue",marker="*",label="centroids")
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7eeb07e77850>
```



Start coding or generate with AI.