rahulml5new

October 18, 2024

[5]:
```
import pandas as pd
import numpy as np
```

[12]:
```
data=pd.read_csv("Mall_Customers.csv")
```

[ ]:
```
data.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

[ ]:
```
data.isnull().sum()
```

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

[16]:
```
from sklearn.preprocessing import LabelEncoder
from sklearn import metrics
le=LabelEncoder()
```

[17]:
```
data["Gender"]=le.fit_transform(data["Gender"])
```

[18]:
```
data.head()
```

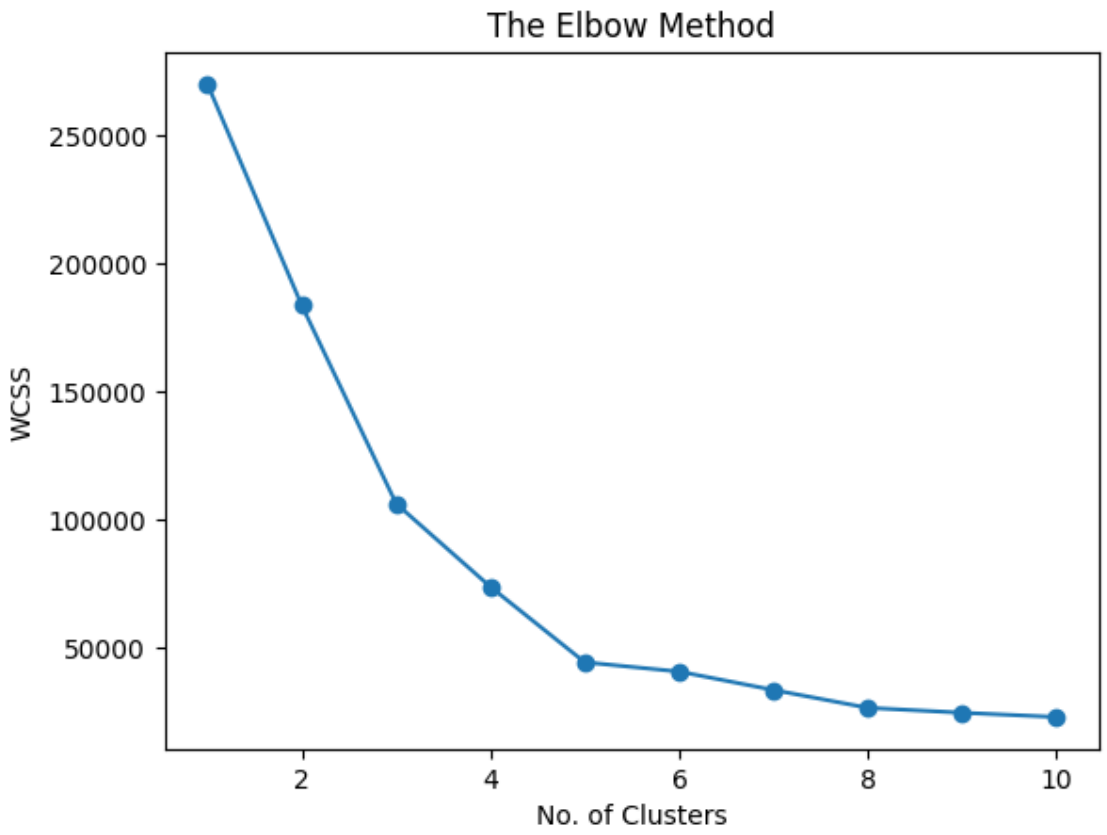
[18]:

	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	19	15	39
1	1	21	15	81
2	0	20	16	6
3	0	23	16	77
4	0	31	17	40

[19]:
```
df=data.copy()

x=df.iloc[:,[2,3]]
from sklearn.cluster import KMeans
wcss=[]
for i in range (1,11):
  kmeans=KMeans(n_clusters=i, init='k-means++',random_state=2)
  kmeans.fit(x);
  wcss.append(kmeans.inertia_)
  print("k:",i,"->wcss:",kmeans.inertia_)
```

```
k: 1 ->wcss: 269981.28000000014
k: 2 ->wcss: 183653.3289473683
k: 3 ->wcss: 106348.37306211119
k: 4 ->wcss: 73880.64496247198
k: 5 ->wcss: 44448.45544793369
k: 6 ->wcss: 40825.16946386947
k: 7 ->wcss: 33642.57922077922
k: 8 ->wcss: 26686.837785187785
k: 9 ->wcss: 24766.471609793436
k: 10 ->wcss: 23103.122085983905
```

[20]:
```
import matplotlib.pyplot as plt
plt.plot(range(1,11),wcss,marker='o')
plt.title("The Elbow Method")
plt.xlabel("No. of Clusters")
plt.ylabel("WCSS")
plt.show()
```



[21]:
```
kmeans=KMeans(n_clusters=5)
kmeans.fit(df)
y=kmeans.predict(df)
df["label"]=y
df.head()
```

[21]:

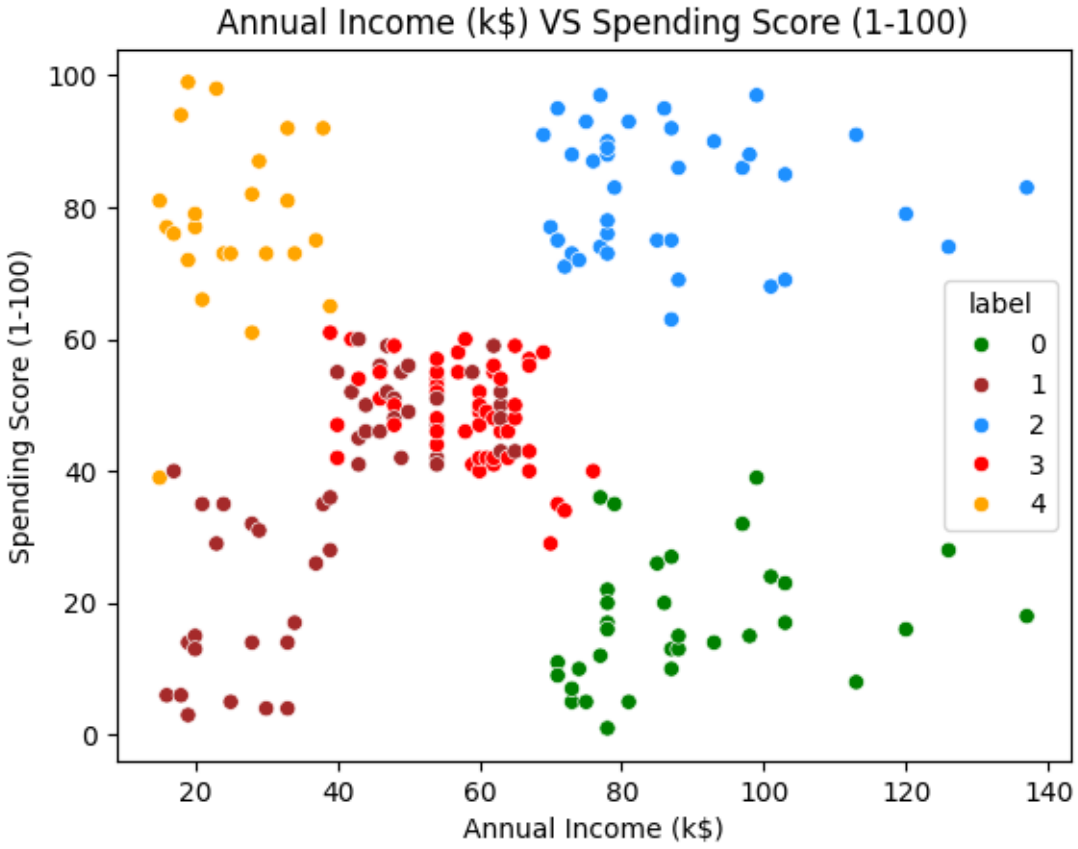
	Gender	Age	Annual Income (k$)	Spending Score (1-100)	label
0	1	19	15	39	4
1	1	21	15	81	4
2	0	20	16	6	1
3	0	23	16	77	4
4	0	31	17	40	1

[22]:
```
import seaborn as sns
```

[30]:
```
from sklearn.cluster import KMeans
km=KMeans(n_clusters=5)
# Assuming 'x' from your previous code (ipython-input-19) contains the features␣
  ↪you want to cluster
km.fit(x) # Use 'x' instead of 'X_train'
```

[30]:
```
KMeans(n_clusters=5)
```

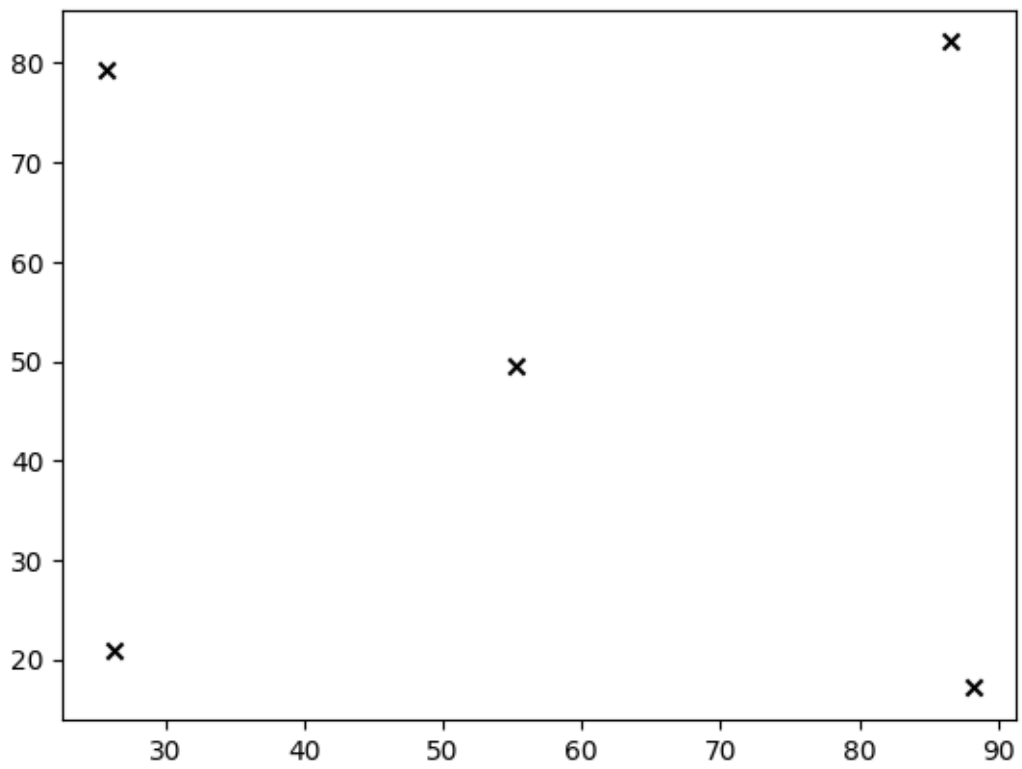
[24]:
```
sns.scatterplot(x='Annual Income (k$)',y='Spending Score␣
↪(1-100)',hue="label",palette=['green','brown','dodgerblue','red','orange'],data=df)

plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('Annual Income (k$) VS Spending Score (1-100)')
plt.show()
```



[31]:
```
centers = np.array(km.cluster_centers_)
```

[35]:
```
plt.scatter(centers[:,0], centers[:,1], marker="x", color='black')
```

[35]:
```
<matplotlib.collections.PathCollection at 0x7e739bd82740>
```



[33]:
```
Y_train_km=km.predict(X_train)
Y_test_km=km.predict(X_test)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-33-467d1b5328db> in <cell line: 1>()
----> 1 Y_train_km=km.predict(X_train)
      2 Y_test_km=km.predict(X_test)

NameError: name 'X_train' is not defined
```

[28]:
```
from sklearn.metrics.cluster import adjusted_rand_score
acc_train=adjusted_rand_score(Y_train,Y_train_km)
acc_test=adjusted_rand_score(Y_test,Y_test_km)
```

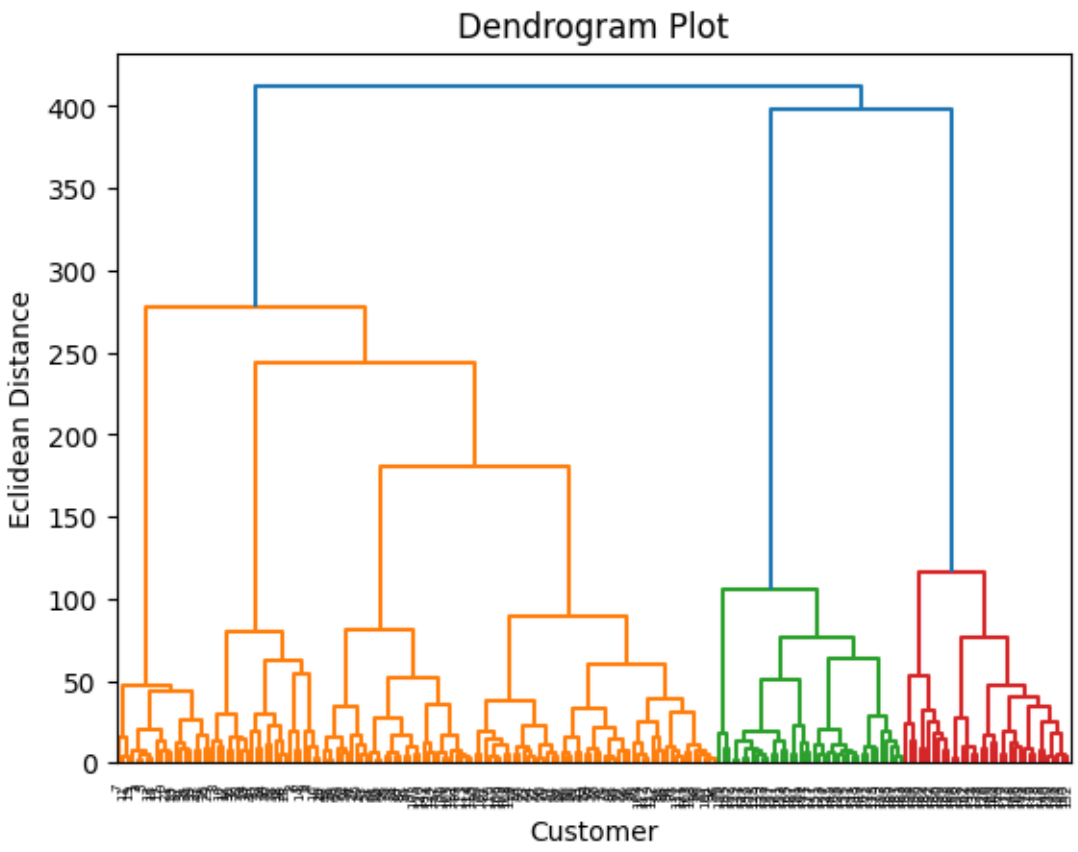
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-28-d412d52aeff9> in <cell line: 2>()
      1 from sklearn.metrics.cluster import adjusted_rand_score
----> 2 acc_train=adjusted_rand_score(Y_train,Y_train_km)
      3 acc_test=adjusted_rand_score(Y_test,Y_test_km)
```

```
NameError: name 'Y_train' is not defined
```

[29]:
```
print("K mean : Accuracy on training Data: {:,3f}",format(acc_train) )
print("K mean : Accuracy on testing Data: {:,3f}",format(acc_test) )
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-29-02c9451bd431> in <cell line: 1>()
----> 1 print("K mean : Accuracy on training Data: {:,3f}",format(acc_train) )
      2 print("K mean : Accuracy on testing Data: {:,3f}",format(acc_test) )
      3

NameError: name 'acc_train' is not defined
```

[ ]:
```
import scipy.cluster.hierarchy as shc
dendogram=shc.dendrogram(shc.linkage(df,method="ward"))
plt.title("Dendrogram Plot")
plt.xlabel("Customer")
plt.ylabel("Eclidean Distance")
plt.grid(False)
```



[ ]:
```
from sklearn.cluster import AgglomerativeClustering
agc=AgglomerativeClustering(n_clusters=5)
```

[11]:
```
df["label"]=agc.fit_predict(df);
df.head()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-11-092b81288176> in <cell line: 1>()
----> 1 df["label"]=agc.fit_predict(df);
      2 df.head()

NameError: name 'agc' is not defined
```

[10]:
```
sns.scatterplot(x='Annual Income (k$)',y='Spending Score␣
↪(1-100)',hue="label",palette=['green','brown','dodgerblue','red','orange'],data=df)

plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('Annual Income (k$) VS Spending Score (1-100)')
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-10-d5587fea3bc4> in <cell line: 1>()
----> 1 sns.scatterplot(x='Annual Income (k$)',y='Spending Score␣
↪(1-100)',hue="label",palette=['green','brown','dodgerblue','red','orange'],da:a=df)
      2
      3 plt.xlabel('Annual Income (k$)')
      4 plt.ylabel('Spending Score (1-100)')
      5 plt.title('Annual Income (k$) VS Spending Score (1-100)')

NameError: name 'sns' is not defined
```

[38]:
```
from sklearn.cluster import AgglomerativeClustering,KMeans
from sklearn.metrics import silhouette_score

silhouette_scores = []
# Start the loop from 2 instead of 1
for n_clusters in range(2, 12):
    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
    cluster_labels = kmeans.fit_predict(x)
    silhouette_avg = silhouette_score(x, cluster_labels)
```

```
    silhouette_scores.append(silhouette_avg)
    print("For n_clusters =", n_clusters,
          "The average silhouette_score is :", silhouette_avg)

plt.plot(range(2, 12), silhouette_scores, marker='o')
plt.xlabel("Number of Clusters")
plt.ylabel("Silhouette Score")
plt.title("Silhouette Score for Different Number of Clusters")
plt.show()
```

```
For n_clusters = 2 The average silhouette_score is : 0.39564531743995546
For n_clusters = 3 The average silhouette_score is : 0.46761358158775435
For n_clusters = 4 The average silhouette_score is : 0.4937945814354117
For n_clusters = 5 The average silhouette_score is : 0.553931997444648
For n_clusters = 6 The average silhouette_score is : 0.5128405328004378
For n_clusters = 7 The average silhouette_score is : 0.5017174409749505
For n_clusters = 8 The average silhouette_score is : 0.4962769338093321
For n_clusters = 9 The average silhouette_score is : 0.45587414130065596
For n_clusters = 10 The average silhouette_score is : 0.4426214845978157
For n_clusters = 11 The average silhouette_score is : 0.41413838935154096
```