ml-assignment-5-k-means-clustering

October 16, 2024

problem statement– This dataset gives the data of Income and money spent by the customers visiting a Shopping Mall. The data set contains Customer ID, Gender, Age, Annual Income, Spending Score. Therefore, as a mall owner you need to find the group of people who are the profitable customers for the mall owner. Apply at least two clustering algorithms (based on Spending Score) to find the group of customers. A. Apply Data pre-processing (Label Encoding , Data Transformation….) techniques if necessary. B. Perform data-preparation(Train-Test Split) C. Apply Machine Learning Algorithm D. Evaluate Model. E. Apply Cross-Validation and Evaluate Model

[1]:
```
import pandas as pd
```

[2]:
```
data=pd.read_csv('/content/Mall_Customers.csv')
```

[3]:
```
print(data.isnull().sum())
```

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

[4]:
```
print(data)
```

```
     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
..          …     …  …                 …                       …
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 x 5 columns]
```

[5]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

Label-Encoder for encoding binary categories in a column

[6]:
```
from sklearn.preprocessing import LabelEncoder
from sklearn import metrics
encoder=LabelEncoder()
```

[8]:
```
data['Genre']=encoder.fit_transform(data['Genre'])
```

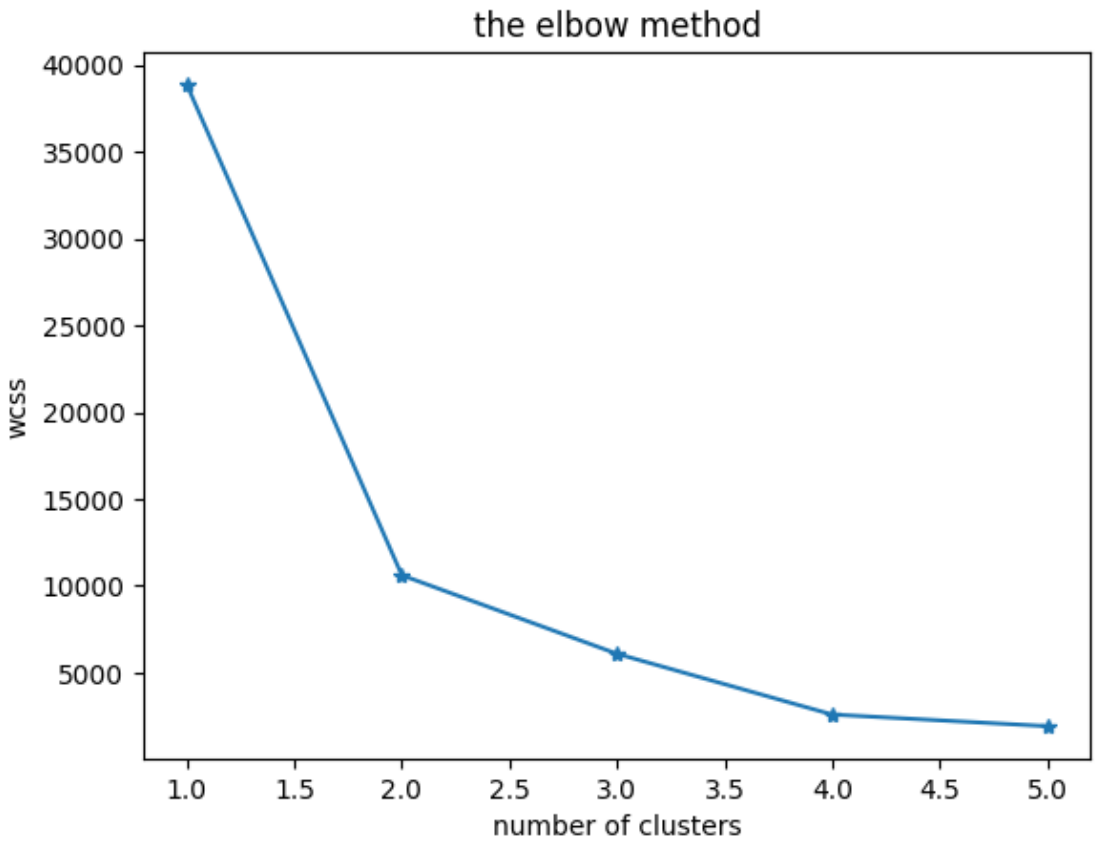
[9]:
```
df=data.copy()
x=df.iloc[:,[2,1]]
```

Applying K-Means clustering

[10]:
```
from sklearn.cluster import KMeans
wcss=[]
for i in range(1,6):
  kmeans=KMeans(n_clusters=i,init='k-means++',random_state=42)
  kmeans.fit(x)
  wcss.append(kmeans.inertia_)
  print("wcss:",wcss)
```

```
wcss: [38880.78000000002]
wcss: [38880.78000000002, 10606.741312741315]
wcss: [38880.78000000002, 10606.741312741315, 6081.062770562774]
wcss: [38880.78000000002, 10606.741312741315, 6081.062770562774,
2581.889381571298]
wcss: [38880.78000000002, 10606.741312741315, 6081.062770562774,
2581.889381571298, 1910.989964681525]
```

[11]:
```
plt.plot(range(1,6),wcss,marker='*')
plt.title('the elbow method')
plt.xlabel('number of clusters')
plt.ylabel('wcss')
plt.show()
```



[13]:
```
kmeans=KMeans(n_clusters=4,random_state=42)
kmeans.fit(x)
y=kmeans.fit_predict(x)
df["lable"]=y
```

[14]:
```
df.head()
```

[14]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)	lable
0	1	1	19	15	39	2
1	2	1	21	15	81	2
2	3	0	20	16	6	2
3	4	0	23	16	77	2
4	5	0	31	17	40	1

[15]:
```
unique_clusters=df['lable'].unique()
```

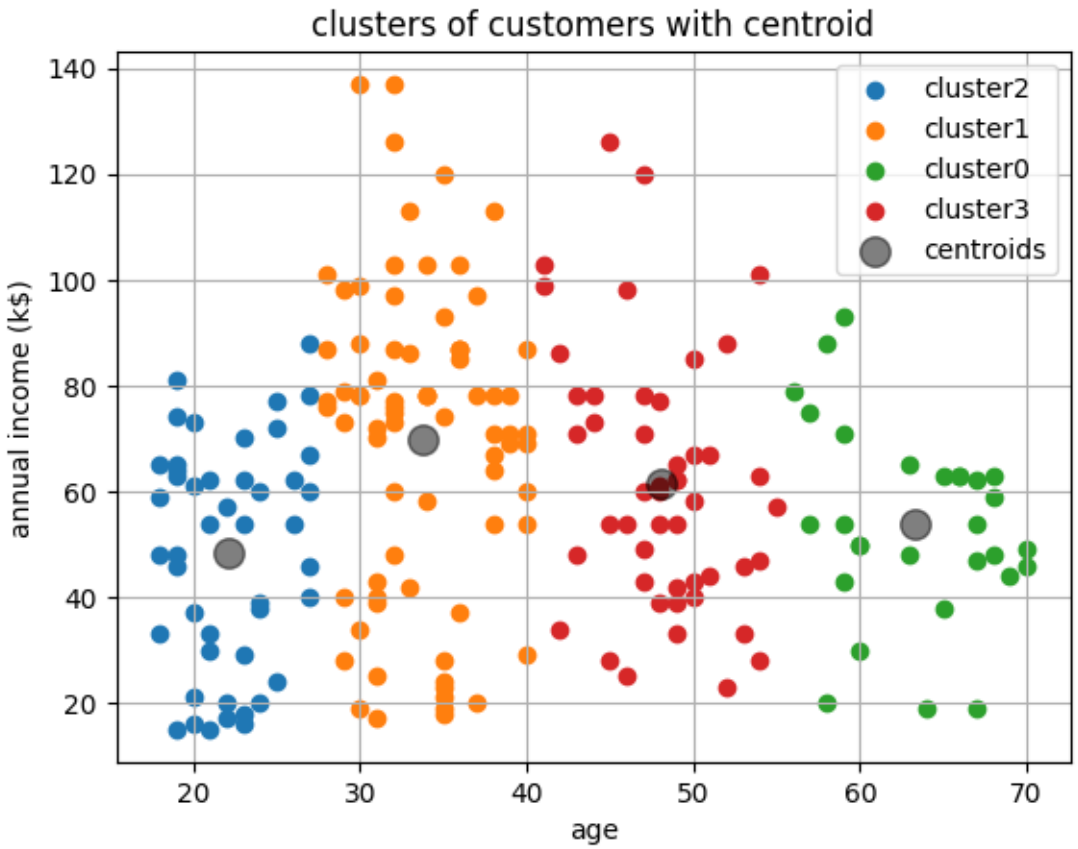
Ploting clusters along with Centroid

[22]:
```
for cluster in unique_clusters:
  cluster_data=df[df['lable']==cluster]
```

```
  plt.scatter(cluster_data['Age'],cluster_data['Annual Income␣
 ↪(k$)'],label=f'cluster{cluster}')

centroids=[]
for cluster in unique_clusters:
    cluster_data=df[df['lable']==cluster]
    centroid=[cluster_data['Age'].mean(),cluster_data['Annual Income (k$)'].
 ↪mean()]
    centroids.append(centroid)

centroids=np.array(centroids)
plt.scatter(centroids[:,0],centroids[:,1],c="black",s=500,alpha=0.5,marker='.
 ↪',label='centroids')
plt.title('clusters of customers with centroid')
plt.xlabel('age')
plt.ylabel('annual income (k$)')
plt.legend()
plt.grid()
plt.show()
```



Claculating Silhoutte score

[23]:
```
from sklearn.metrics import silhouette_score
score=silhouette_score(x,y)
print("silhouette score:",score)
```

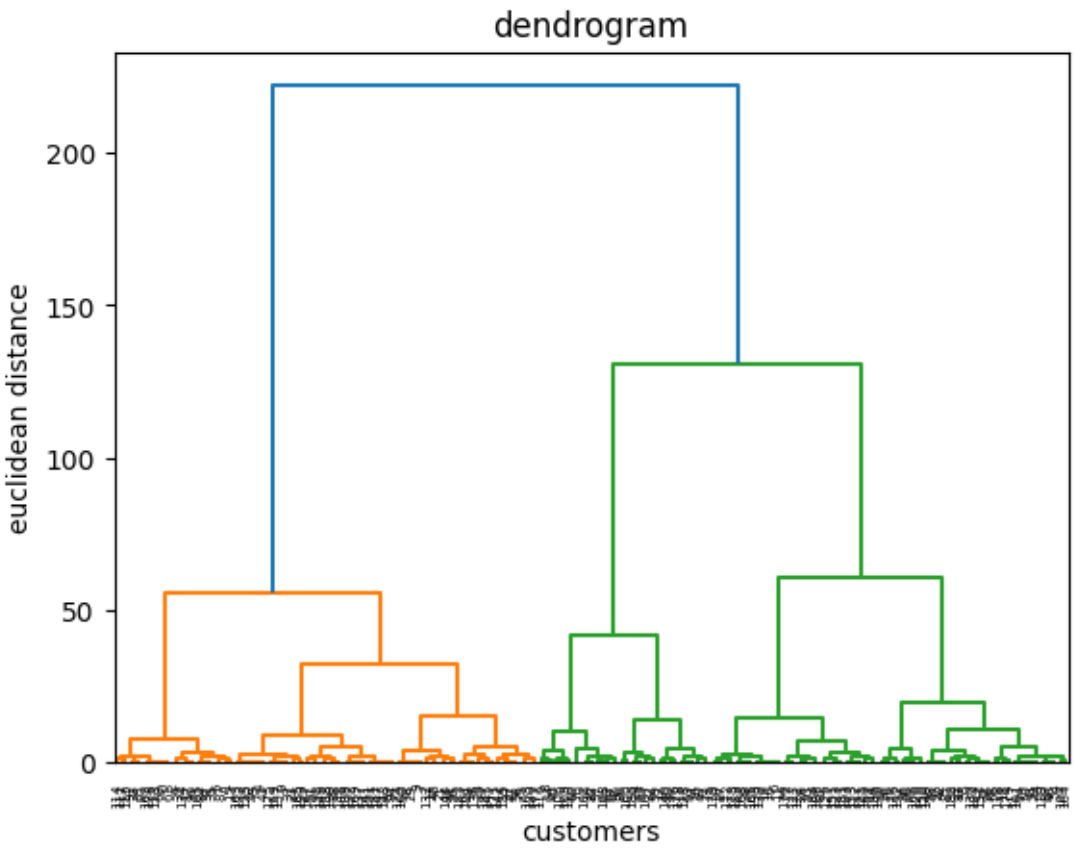
```
silhouette score: 0.5910585004019959
```

[24]:
```
ytrain_km=kmeans.fit_predict(x)
ytest_km=kmeans.predict(x)
```

[25]:
```
from sklearn.metrics import adjusted_rand_score
acc_train=adjusted_rand_score(y,ytrain_km)
acc_test=adjusted_rand_score(y,ytest_km)
print("kmean: accuracy on training data:",format(acc_train))
print("kmean: accuracy on testing data:",format(acc_test))
```

```
kmean: accuracy on training data: 1.0
kmean: accuracy on testing data: 1.0
```

[26]:
```
import scipy.cluster.hierarchy as sch
dendrogram=sch.dendrogram(sch.linkage(x,method='ward'))
plt.title('dendrogram')
plt.xlabel('customers')
plt.ylabel('euclidean distance')
plt.show()
```



[27]:
```
!pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-
packages (1.5.2)
Requirement already satisfied: numpy>=1.19.5 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn) (1.26.4)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
```

[28]:
```
silhouette_scores=[]
n_clusters_range=range(2,11)
for n_clusters in n_clusters_range:
  kmeans=KMeans(n_clusters=n_clusters)
  cluster_labels=kmeans.fit_predict(x)
  score=silhouette_score(x,cluster_labels)
  silhouette_scores.append(score)
```

Graph of Silhoutte score vs Number of clusters

[30]:
```
#plot the results
plt.plot(n_clusters_range,silhouette_scores,marker='o')
plt.title('silhouette score vs number of clusters')
plt.xlabel('number of clusters')
plt.ylabel('silhouette score')
plt.show()
```