Assignment on Clustering Techniques

LP Lab Assignment 4

33323 - Aditya Kangune

Problem Statment

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

K-Means Clustering

Importing the libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

Importing the dataset

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

```
df = pd.read_csv('/content/drive/MyDrive/Datasets/Mall_Customer
```

```
print(df)
```

```
     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]
```

```
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

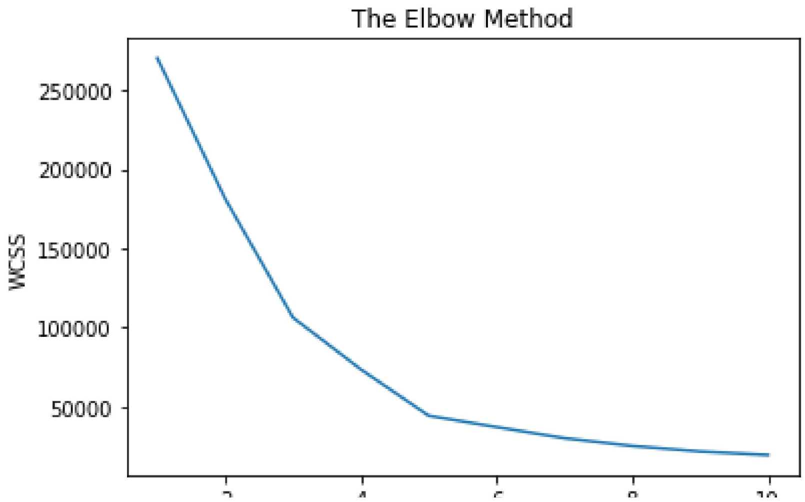
```
X = df.iloc[:, [3, 4]].values
print(X)
```

```
[[ 15  39]
 [ 15  81]
 [ 16   6]
 [ 16  77]
 [ 17  40]
 [ 17  76]
 [ 18   6]
 [ 18  94]
 [ 19   3]
 [ 19  72]
 [ 19  14]
 [ 19  99]
 [ 20  15]
 [ 20  77]
 [ 20  13]
 [ 20  79]
 [ 21  35]
 [ 21  66]
 [ 23  29]
 [ 23  98]
 [ 24  35]
 [ 24  73]
```

```
 [ 25   5]
 [ 25  73]
 [ 28  14]
 [ 28  82]
 [ 28  32]
 [ 28  61]
 [ 29  31]
 [ 29  87]
 [ 30   4]
 [ 30  73]
 [ 33   4]
 [ 33  92]
 [ 33  14]
 [ 33  81]
 [ 34  17]
 [ 34  73]
 [ 37  26]
 [ 37  75]
 [ 38  35]
 [ 38  92]
 [ 39  36]
 [ 39  61]
 [ 39  28]
 [ 39  65]
 [ 40  55]
 [ 40  47]
 [ 40  42]
 [ 40  42]
 [ 42  52]
 [ 42  60]
 [ 43  54]
 [ 43  60]
 [ 43  45]
 [ 43  41]
 [ 44  50]
```

Using the elbow method to find the optimal number of clusters

```
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



Training the K-Means model on the dataset

```
kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_stat
y_kmeans = kmeans.fit_predict(X)
```

Visualising the clusters

```
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100,
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100,
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100,
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 100,
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100,
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_cente
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```

Hierarchical Clustering

Hierarchical clustering starts by treating each observation as a separate cluster. Then, it repeatedly executes the following two steps:

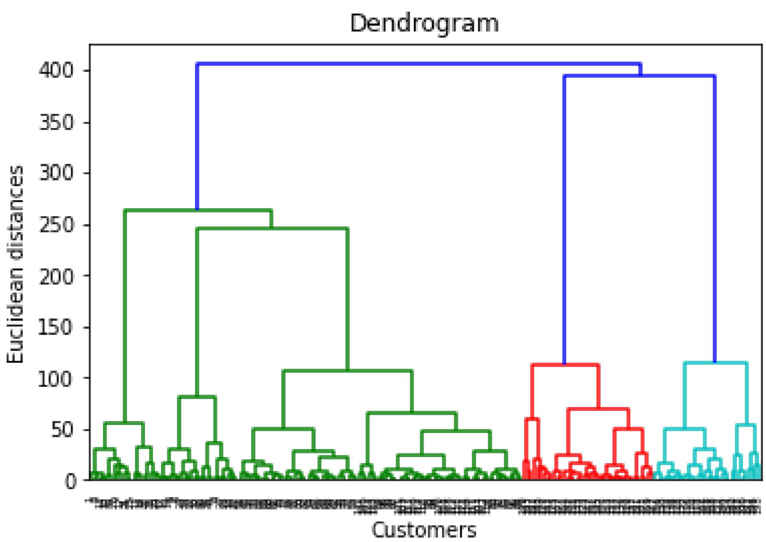
(1) identify the two clusters that are closest together.

(2) merge the two most similar clusters. This iterative process continues until all the clusters are merged together.

Using the dendrogram to find the optimal number of clusters

A Dendrogram is a tree-like diagram that records the sequences of merges or splits.

```
import scipy.cluster.hierarchy as sch
dendrogram = sch.dendrogram(sch.linkage(X, method = 'ward'))
plt.title('Dendrogram')
plt.xlabel('Customers')
plt.ylabel('Euclidean distances')
plt.show()
```



Training the Hierarchical Clustering model on the dataset

Agglomerative Hierarchical clustering Technique: In this technique, initially each data point is considered as an individual cluster. At each iteration, the similar clusters merge with other clusters until one cluster or K clusters are formed.

Ward's Linkage: The linkage function specifying the distance between two clusters is computed as the increase in the "error sum of squares" (ESS) after fusing two clusters into a single cluster.

```
from sklearn.cluster import AgglomerativeClustering
hc = AgglomerativeClustering(n_clusters = 5, affinity = 'euclid
y_hc = hc.fit_predict(X)
```

Visualising the clusters

```
plt.scatter(X[y_hc == 0, 0], X[y_hc == 0, 1], s = 100, c = 'red
plt.scatter(X[y_hc == 1, 0], X[y_hc == 1, 1], s = 100, c = 'blu
plt.scatter(X[y_hc == 2, 0], X[y_hc == 2, 1], s = 100, c = 'gre
plt.scatter(X[y_hc == 3, 0], X[y_hc == 3, 1], s = 100, c = 'cya
plt.scatter(X[y_hc == 4, 0], X[y_hc == 4, 1], s = 100, c = 'mag
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```