```
# K-Means Clustering

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv(r"C:\Users\Vansh\OneDrive\Desktop\Mall_Customers.csv")
X = dataset.iloc[:, [3, 4]].values

# Using the elbow method to find the optimal number of clusters
from sklearn.cluster import KMeans
#we are going to findout the optimal number of cluster & we have to use the elbow method
wcss = []
#to plot the elbow metod we have to compute WCSS for 10 different number of cluster since we g
#we are going to write a for loop to create a list of 10 different wcss for the10 number of cl
#thats why we have to initialise wcss[] & we start our loop

#we choose 1-11 becuase the 11 bound is excluded & we want 10 wcss however the first bound is
#now in each iteration of loop we are going to do 2 things  1st we are going to fit the k-mean
#Now lets fit kmean to our data x
#now eare

for i in range(1, 11):
    kmeans = KMeans(n_clusters = i,init="k-means++",random_state = 0)
    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()
#wcss we have very good parameter called inertia_ credit goes to sklearn , that computes the s

# Training the K-Means model on the dataset
kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_state = 0)
y_kmeans = kmeans.fit_predict(X)

# Visualising the clusters
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2'
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4'
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100, c = 'magenta', label = 'Cluster
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'yellow
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```

Customer Segmentation with

Hierarchical Clustering

Upload CSV file

Drag and drop file here

Drag and drop file here
Limit 200MB per file • CSV
Browse files

Mall_Customers.csv 3.9KB

Mall_Customers.csv uploaded successfully!

Dataset Preview

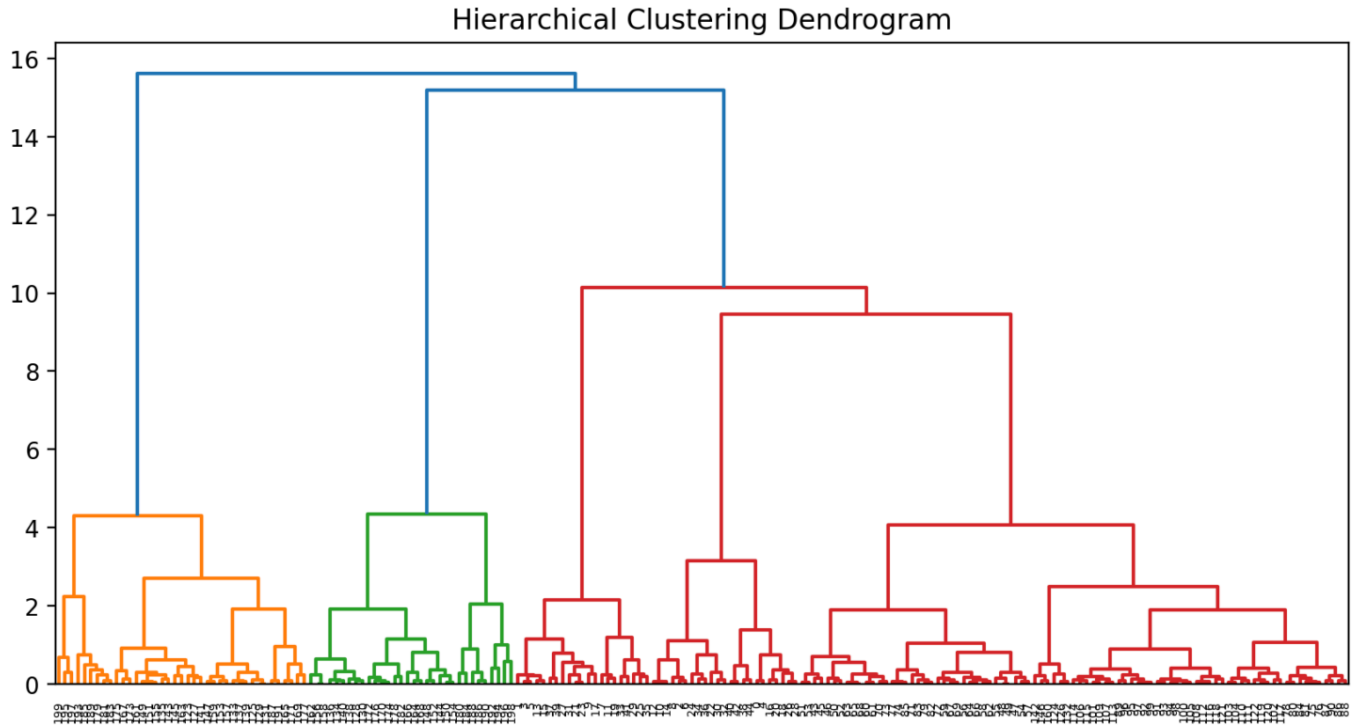
	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

Select Features for Clustering

Choose numerical columns

Annual Income (… × Spending Score (… ×

Dendrogram



Choose Number of Clusters

Number of clusters

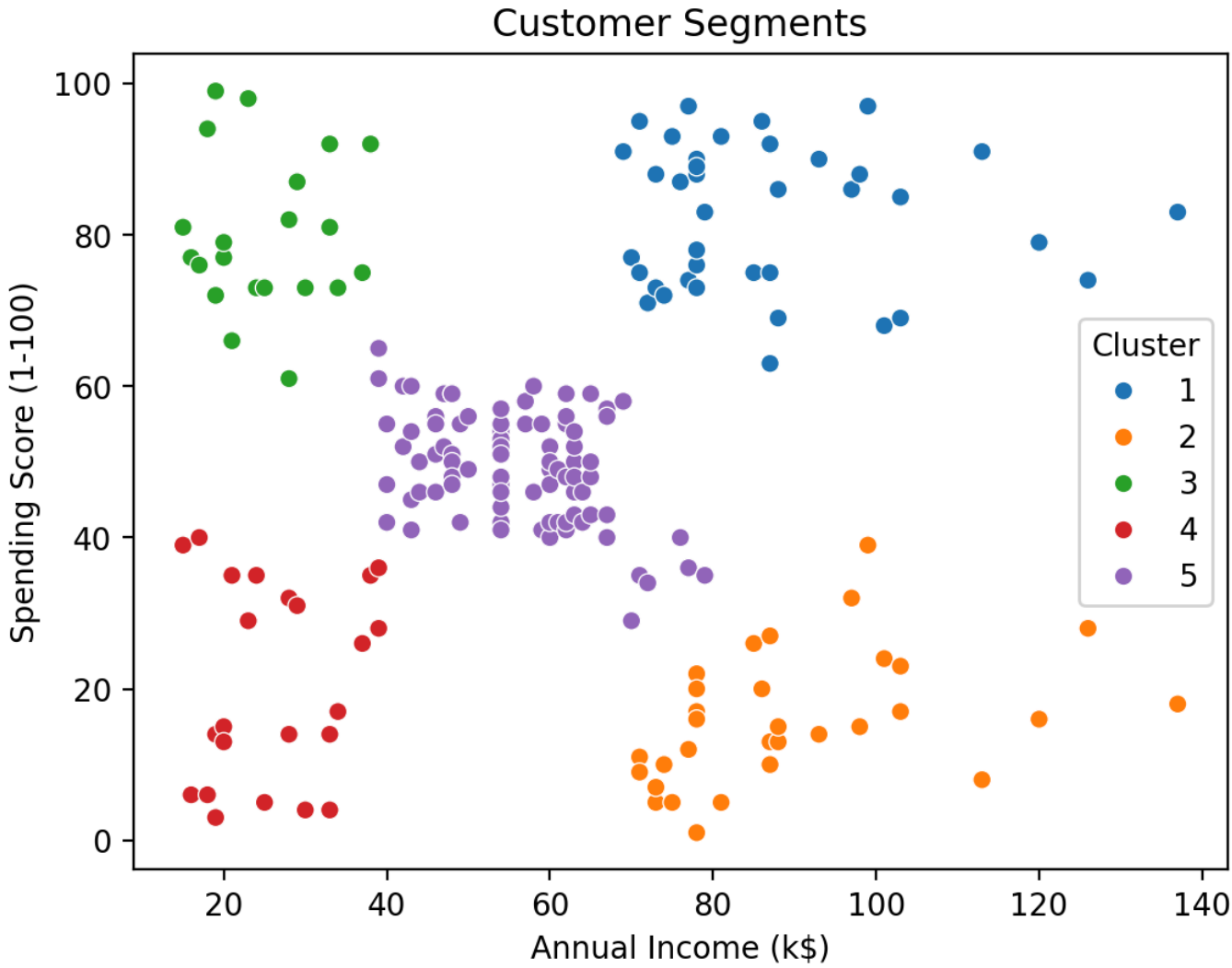
5

2 10

Clustered Data Preview

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

Cluster Visualization


Customer Segments
Spending Score (1-100)
Annual Income (k$)
Cluster
1
2
3
4
5
0
20
40
60
80
100
20
40
60
80
100
120
140