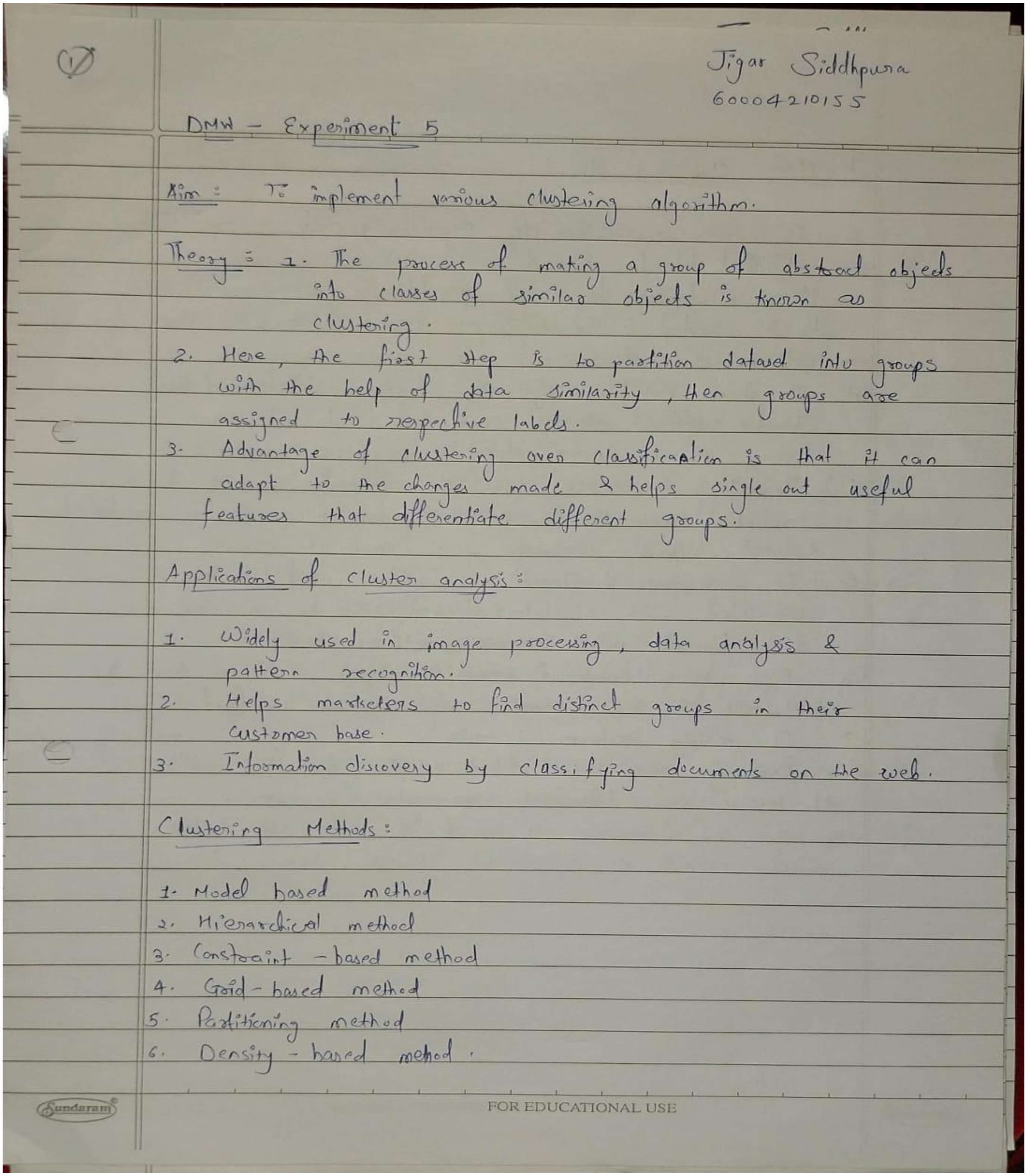
Name: Jigar Siddhpura **SAPID:** 60004200155

DIV: C/C2 **Branch:** Computer Engineering

DMW EXPERIMENT 5



Jigar Siddhpura
60004210155

DMW - Experiment 5

Aim : To implement various clustering algorithm.

Theory : 1. The process of making a group of abstract objects into classes of similar objects is known as clustering.

2. Here, the first step is to partition dataset into groups with the help of data similarity, then groups are assigned to respective labels.

3. Advantage of clustering over classification is that it can adapt to the changes made & helps single out useful features that differentiate different groups.

Applications of cluster analysis :

1. Widely used in image processing, data analysis & pattern recognition.
2. Helps marketers to find distinct groups in their customer base.
3. Information discovery by classifying documents on the web.

Clustering Methods :

1. Model based method
2. Hierarchical method
3. Constraint -based method
4. Grid-based method
5. Partitioning method
6. Density -based method.

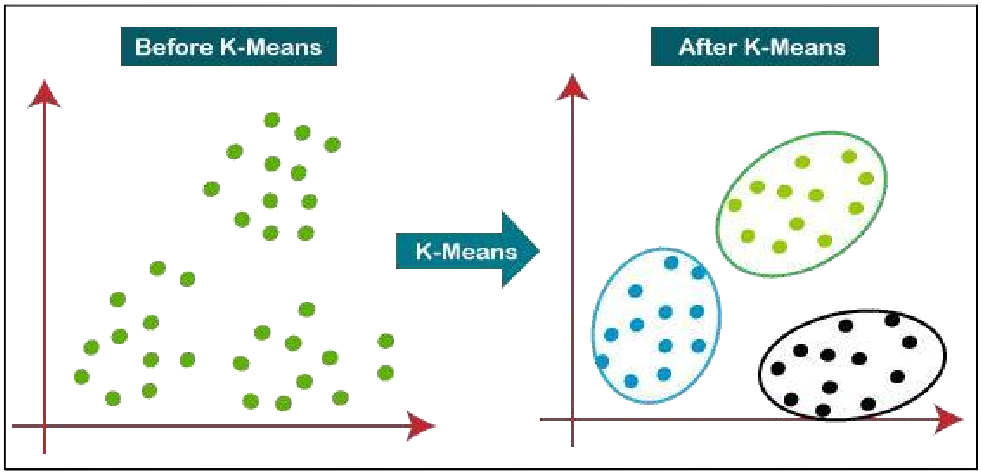
FOR EDUCATIONAL USE

K - Means Clustering algo :

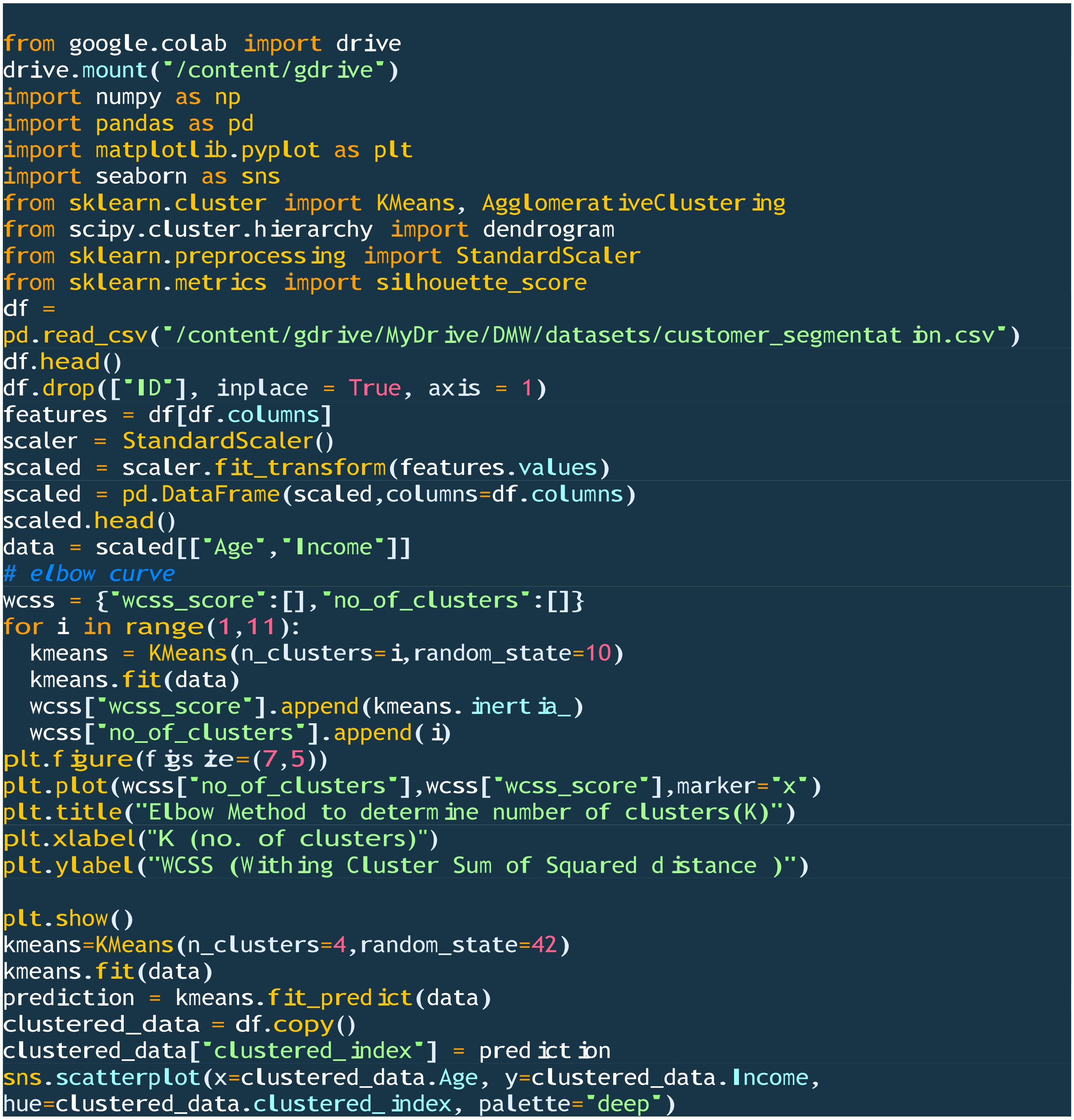
1. It is an unseveriFed ML Algorithm, which groups the
~~2~~ unlabelled dataset into different clusters.
2. Here, K defines the no. of predefined clusters, that need to be created in the neighbors. Eg : If K = 2, there will be 2 clusters.
3. It is an iterative algo that divides unlabeled dataset into K different clusters such that each data item belong to 1 cluster that has similar properties.
4. It allows to cluster data into different groups & a convenient way to discover categories of groups in unlabeled dataset without training.
5. It is a centroid based algo where each cluster is associated with a centroid. Aim is to minimize the dist. btw data point & corresponding clusters.
6. Algo takes unlabeled dataset as input, divides into K-clusters & repeats until best clusters are found.
7. It mainly performs 2 tasks :
   a) Determine best value of K centroids.
   b) Assign each data point to closest K-center.

Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

The below diagram explains the working of the K-means Clustering Algorithm:

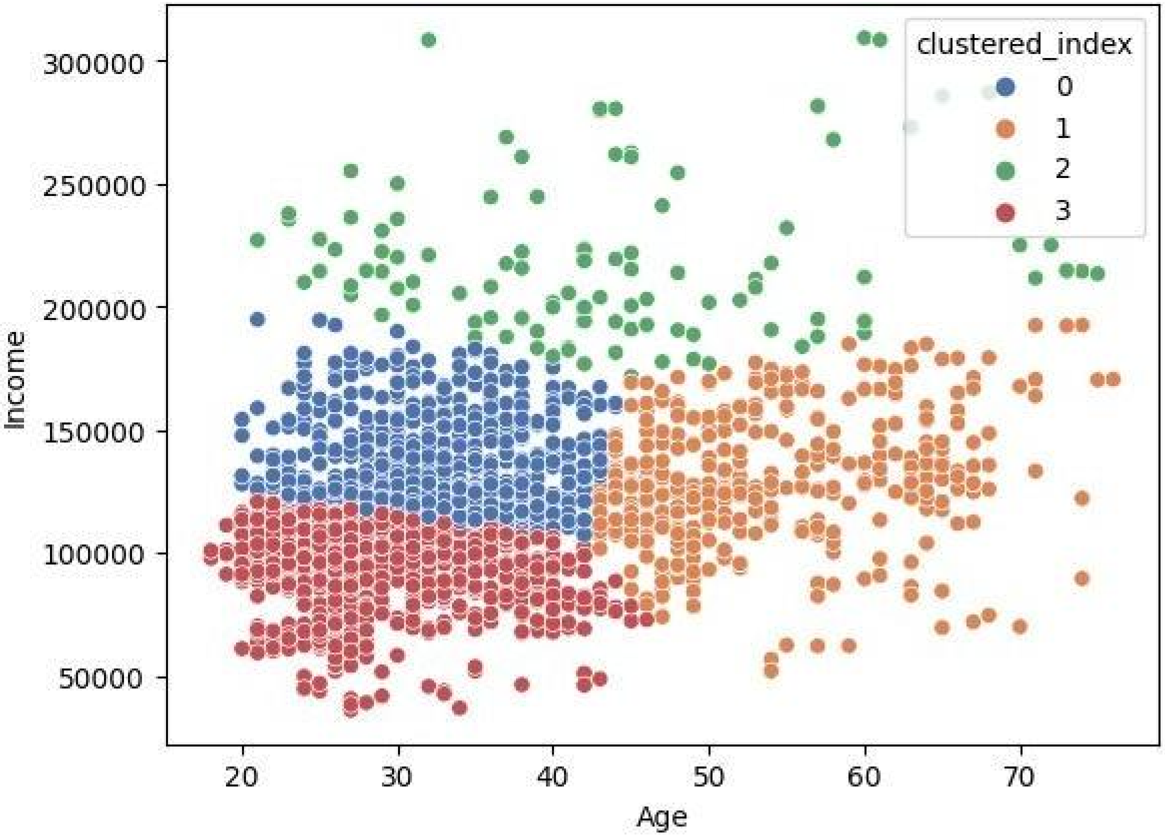


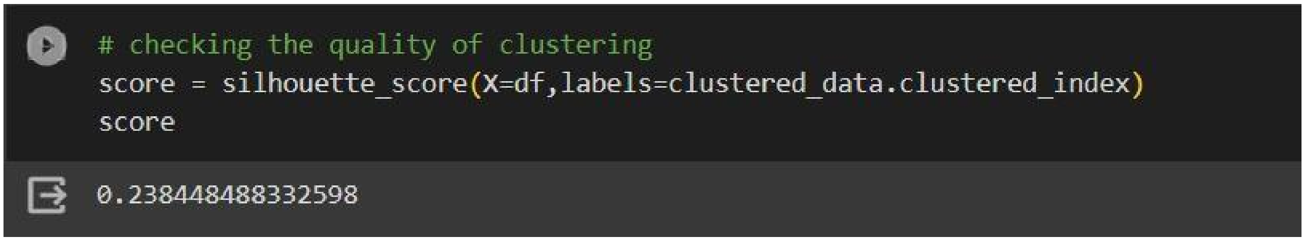
Program:



```
from google.colab import drive
drive.mount("/content/gdrive")
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans, AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score
df =
pd.read_csv("/content/gdrive/MyDrive/DMW/datasets/customer_segmentation.csv")
df.head()
df.drop(["ID"], inplace = True, axis = 1)
features = df[df.columns]
scaler = StandardScaler()
scaled = scaler.fit_transform(features.values)
scaled = pd.DataFrame(scaled,columns=df.columns)
scaled.head()
data = scaled[["Age","Income"]]
# elbow curve
wcss = {"wcss_score":[],"no_of_clusters":[]}
for i in range(1,11):
  kmeans = KMeans(n_clusters=i,random_state=10)
  kmeans.fit(data)
  wcss["wcss_score"].append(kmeans.inertia_)
  wcss["no_of_clusters"].append(i)
plt.figure(figsize=(7,5))
plt.plot(wcss["no_of_clusters"],wcss["wcss_score"],marker="x")
plt.title("Elbow Method to determine number of clusters(K)")
plt.xlabel("K (no. of clusters)")
plt.ylabel("WCSS (Withing Cluster Sum of Squared distance )")

plt.show()
kmeans=KMeans(n_clusters=4,random_state=42)
kmeans.fit(data)
prediction = kmeans.fit_predict(data)
clustered_data = df.copy()
clustered_data["clustered_index"] = prediction
sns.scatterplot(x=clustered_data.Age, y=clustered_data.Income,
hue=clustered_data.clustered_index, palette="deep")
```

Output:





```
# checking the quality of clustering
score = silhouette_score(X=df,labels=clustered_data.clustered_index)
score
```

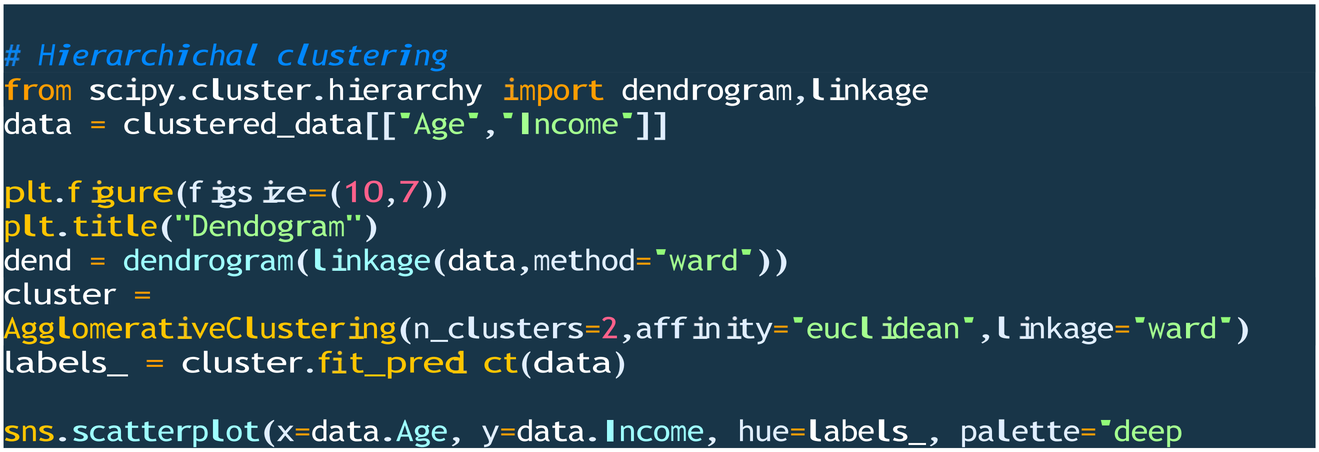
```
0.238448488332598
```

Hierarchical Clustering :

1. It is an unsupervised learning algo which is used to group unlabeled dataset into a cluster.
2. Here, we develop hierarchy of clusters in the form of a tree, & this tree shaped structure is Dendrogram.
3. Sometimes results of K-means & hierarchical may be Similar, but they differ on how they work as there is no need to determine 'k' here beforehand.
4. It has 2 approaches :
   a) Agglomerative – It is a bottom-up approach that where initially all data points form as individual cluster to form one overall cluster.
   b) Divisive : It is a top-down approach & is reverse of agglomerative.
5. ~~Ch~~ Challenges in k-means that it solves :
   ① To determine value of k beforehand
   ② k-means tries to make clusters of same size.

Elbow Method : A method to determine the value of 'k'.
To determine optimal value of k, we have to select the value of k at the 'elbow' i.e. the point after which inertia start decreasing in linear fashion.

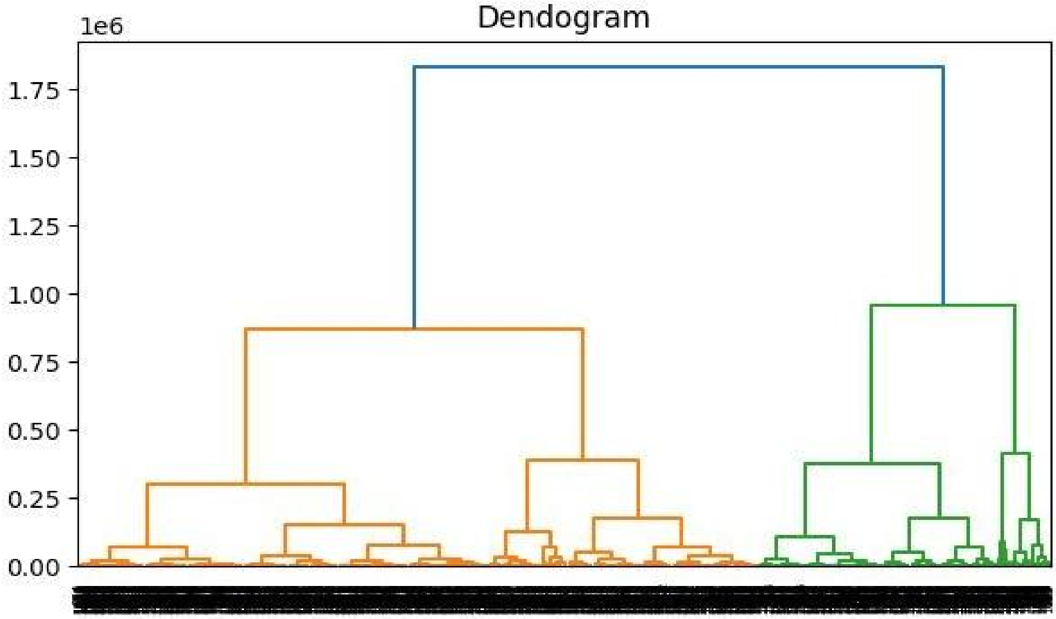
Program :

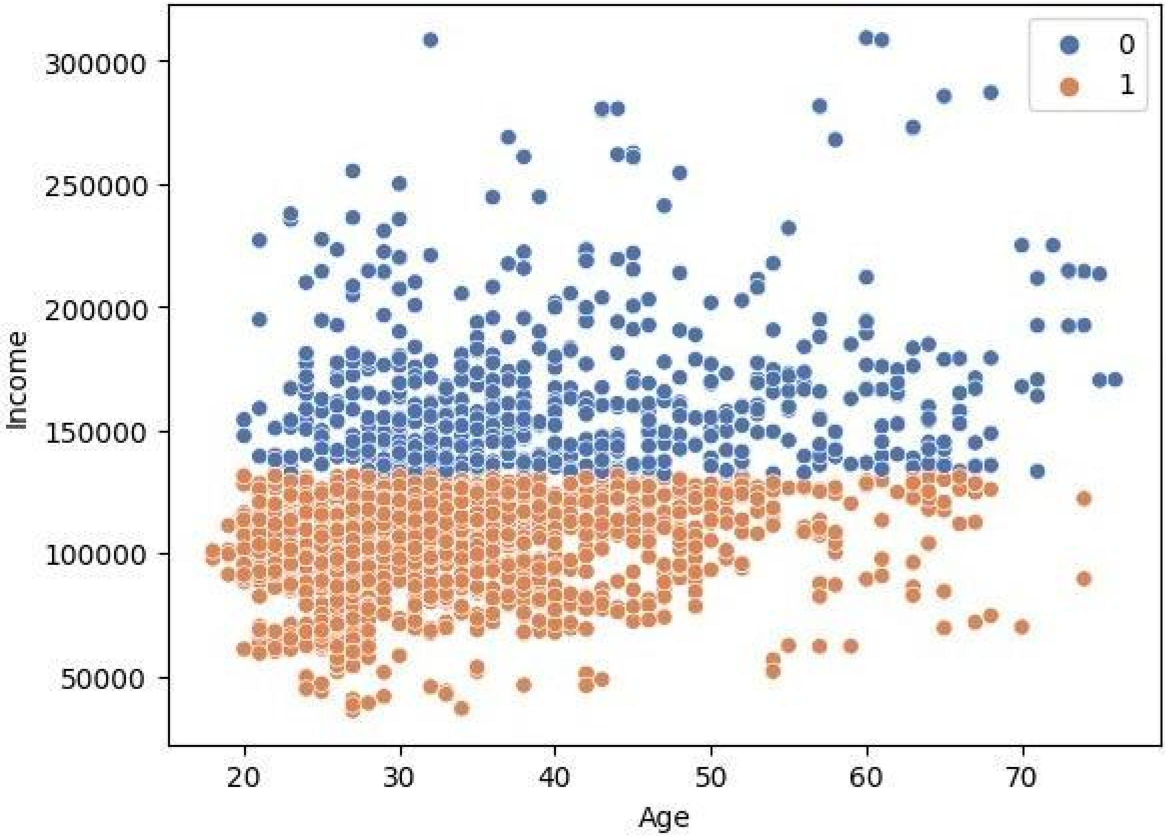


```
# Hierarchichal clustering
from scipy.cluster.hierarchy import dendrogram,linkage
data = clustered_data[["Age","Income"]]

plt.figure(figsize=(10,7))
plt.title("Dendogram")
dend = dendrogram(linkage(data,method="ward"))
cluster =
AgglomerativeClustering(n_clusters=2,affinity="euclidean",linkage="ward")
labels_ = cluster.fit_predict(data)

sns.scatterplot(x=data.Age, y=data.Income, hue=labels_, palette="deep
```

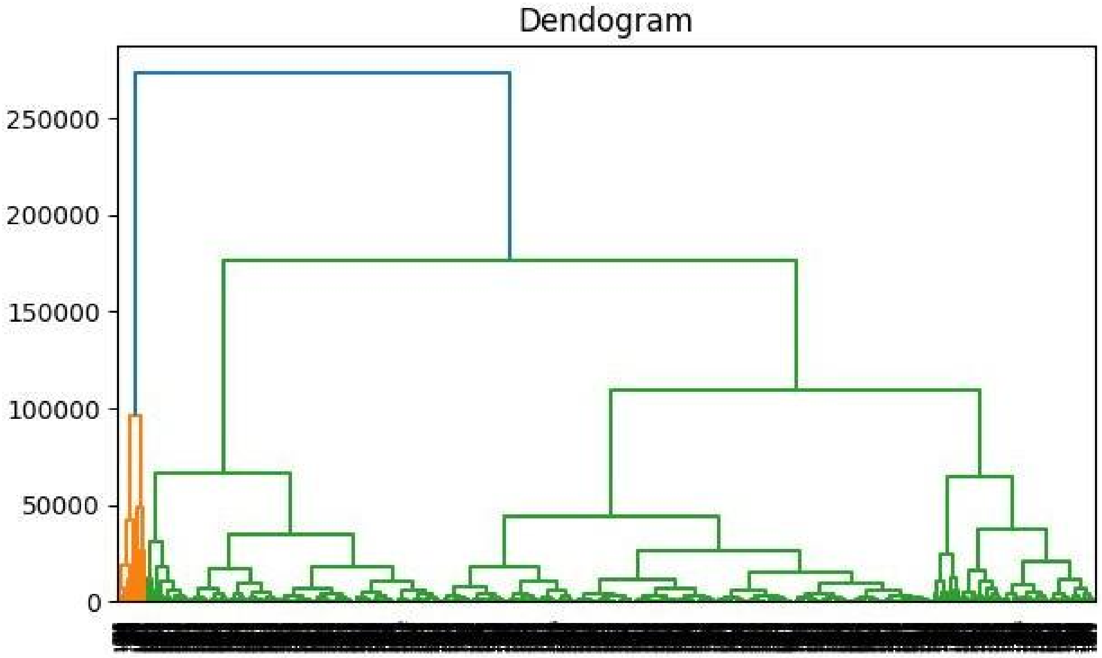
Output:

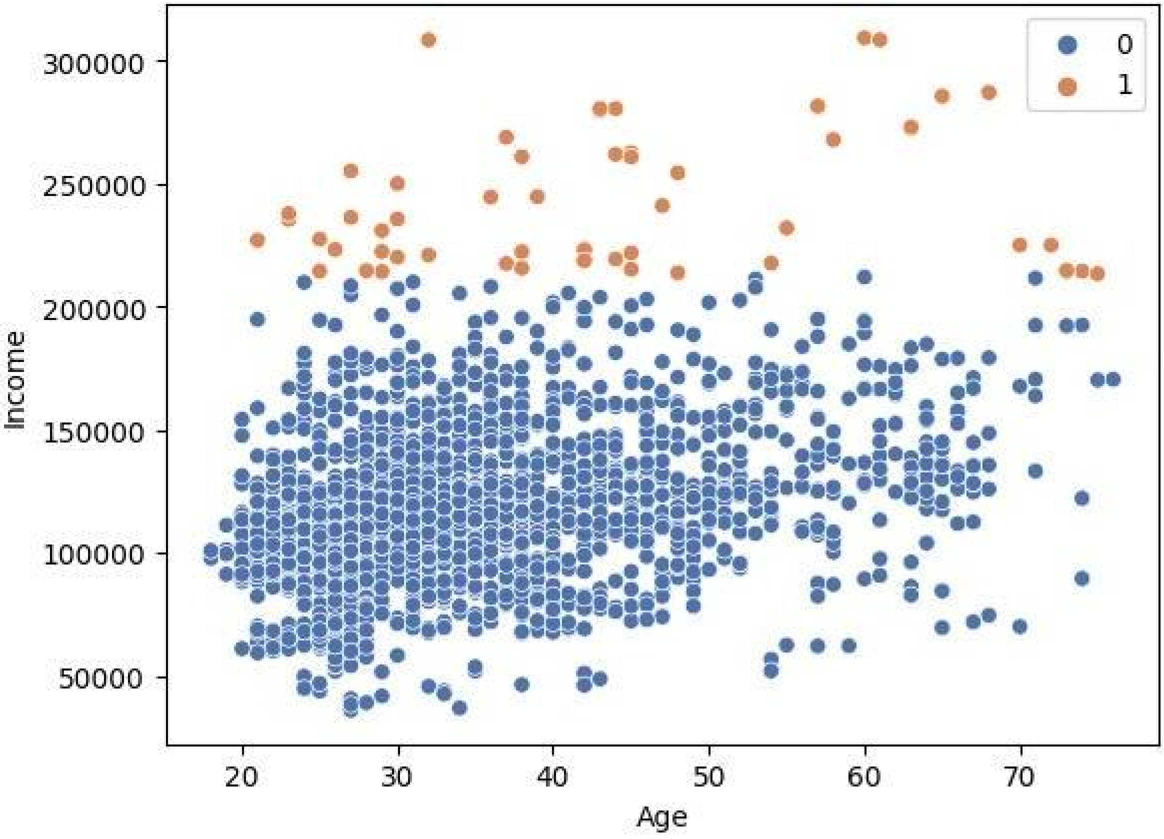
- ***Ward Hierarchical Clustering***



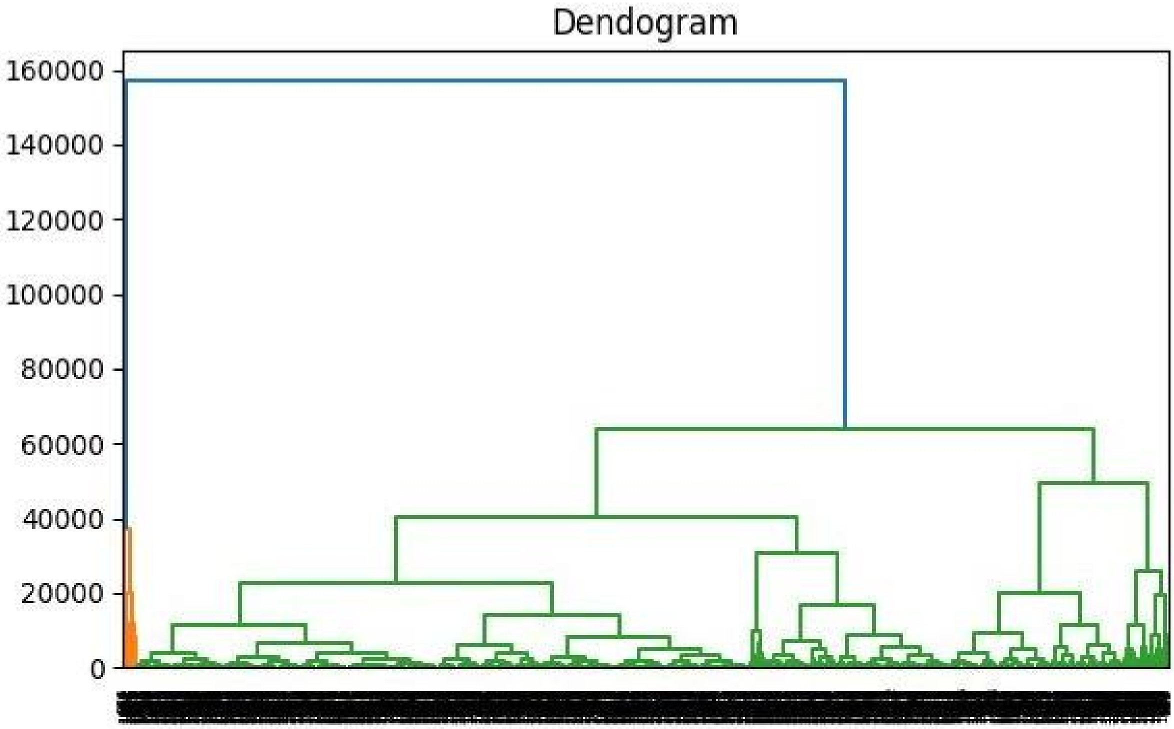


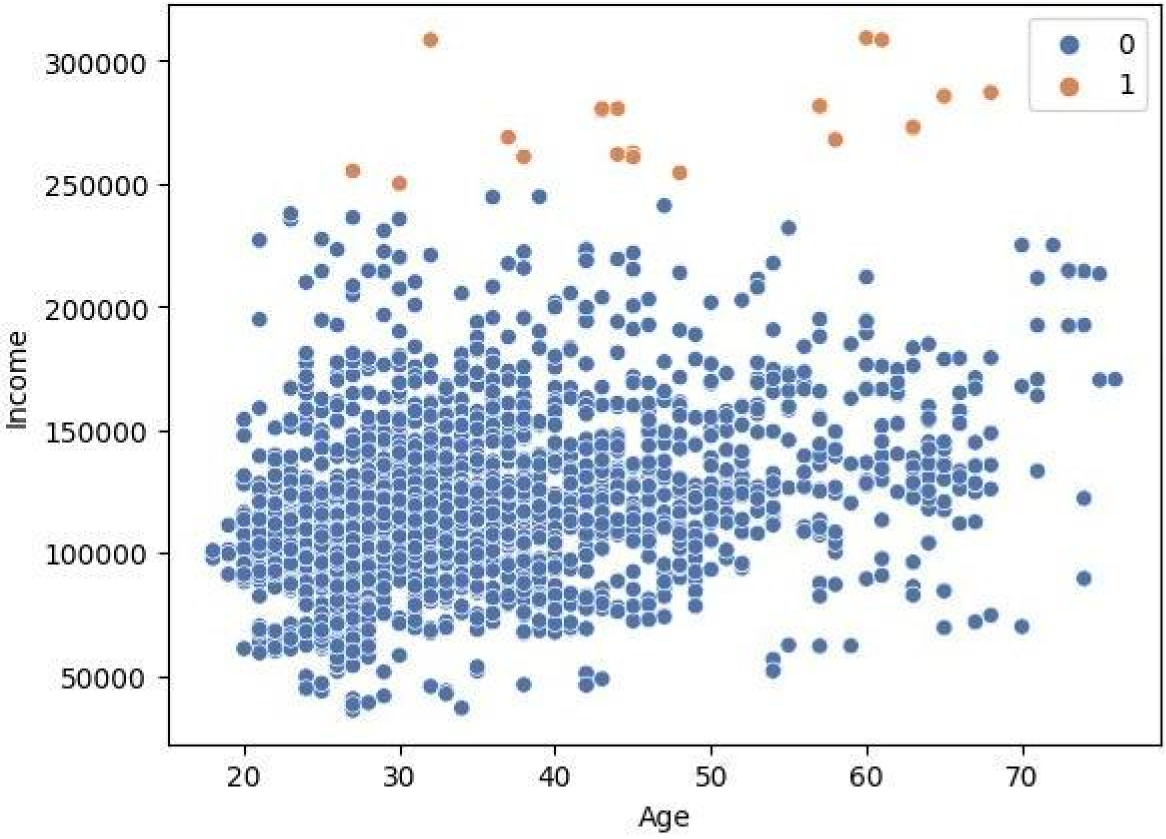
- ***Complete Hierarchical Clustering***





- ***Average Hierarchical Clustering***



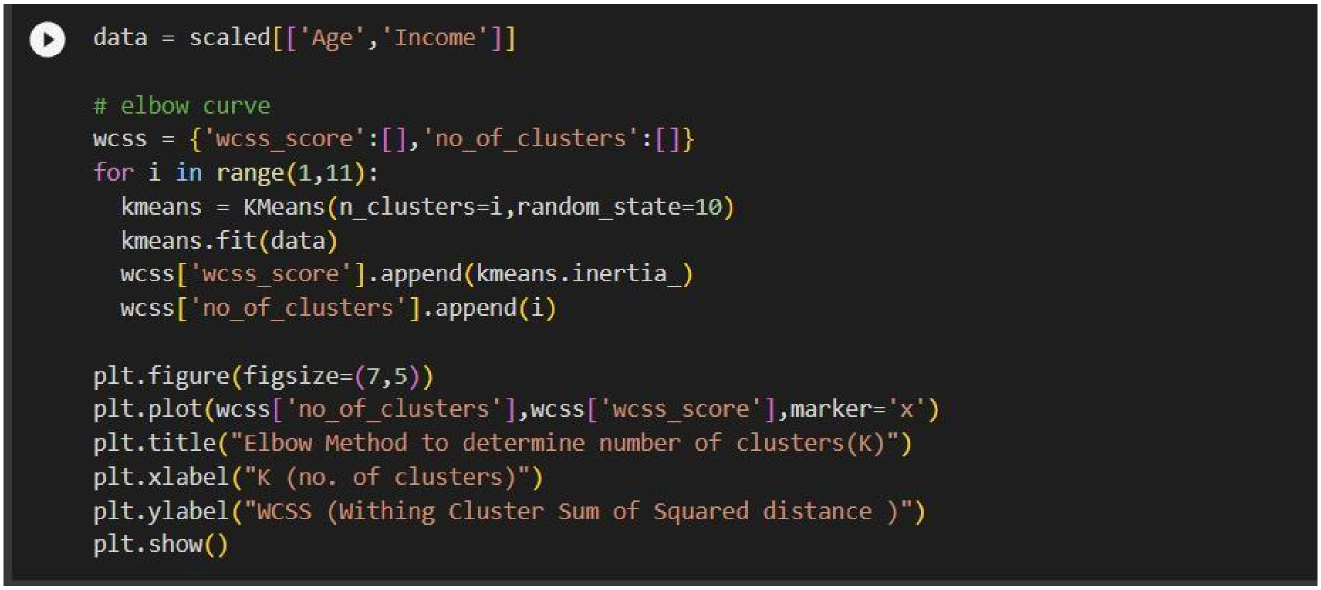


Part B:

1. Plot Elbow Method and suggest optimal number of clusters

A fundamental step for any unsupervised algorithm is to determine the optimal number of clusters into which the data may be clustered. The **Elbow Method** is one of the most popular methods to determine this optimal value of k.

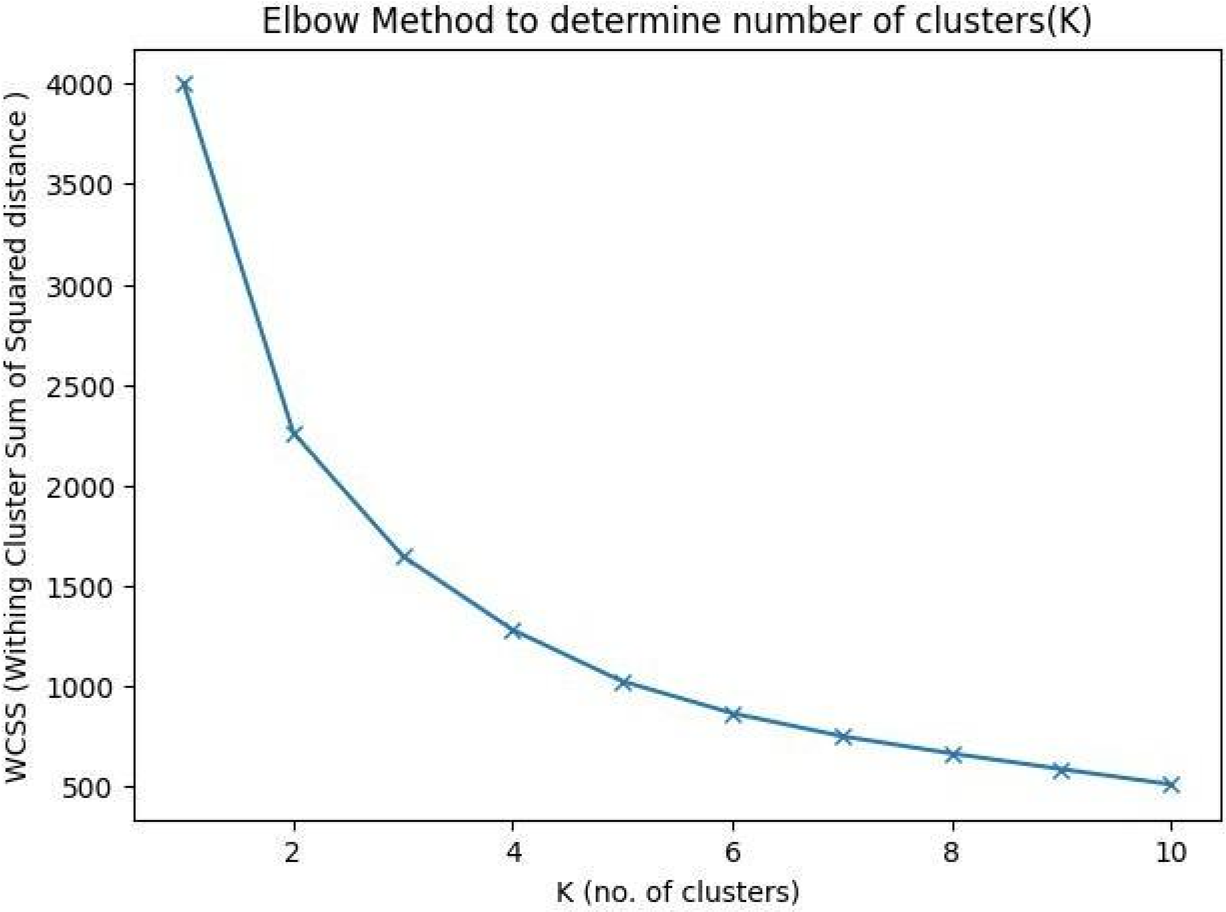
Program:



```
data = scaled[['Age','Income']]

# elbow curve
wcss = {'wcss_score':[],'no_of_clusters':[]}
for i in range(1,11):
  kmeans = KMeans(n_clusters=i,random_state=10)
  kmeans.fit(data)
  wcss['wcss_score'].append(kmeans.inertia_)
  wcss['no_of_clusters'].append(i)

plt.figure(figsize=(7,5))
plt.plot(wcss['no_of_clusters'],wcss['wcss_score'],marker='x')
plt.title("Elbow Method to determine number of clusters(K)")
plt.xlabel("K (no. of clusters)")
plt.ylabel("WCSS (Withing Cluster Sum of Squared distance )")
plt.show()
```

To determine the optimal number of clusters, we have to select the value of k at the “elbow” i.e. the point after which the distortion/inertia start decreasing in a linear fashion. Thus, for the given data, we conclude that the optimal number of clusters for the data is 3.

Output :



Conclusion: Thus, we have successfully implemented Clustering Algorithm Using

1. k-means 2. Hierarchical(ward/complete/average)