Hierarchical Clustering

ABSTRACT:

This project aims to use hierarchical clustering to analyze customer spending behaviour in a shopping mall. The dataset includes the annual income of customers in thousands of dollars and their spending score on a scale of 1 to 100. The objective is to group customers based on their spending patterns and income levels to help mall owners tailor their marketing strategies for specific customer segments. The project will also explore the relationship between income levels and spending behaviour to identify any patterns or trends that can inform marketing strategies. The results of this project can provide valuable insights into customer behaviour, help mall owners maximize profits, and improve customer satisfaction. The project has significant practical implications for businesses looking to optimize their marketing strategies and improve their bottom line.

OBJECTIVE:

The objective of this project is to analyze customer spending behaviour in a shopping mall using hierarchical clustering. The primary goal is to identify distinct customer segments based on their income level and spending patterns, which can help mall owners tailor their marketing strategies to specific customer groups. Additionally, the project aims to explore the relationship between income levels and spending behaviour to identify any trends or patterns that can inform marketing strategies. The results of this analysis can help mall owners better understand their customers' needs and preferences, and develop effective marketing strategies to improve customer satisfaction and increase profitability. Overall, the project's objective is to provide valuable insights into customer behaviour and help businesses optimize their marketing strategies to improve their bottom line.

Introduction:

Hierarchical clustering is a popular method for clustering data points based on their similarities or distances. There are two main types of hierarchical clustering: agglomerative clustering and divisive clustering.

Agglomerative clustering is a bottom-up approach, where each data point is initially considered as a separate cluster, and then clusters are successively merged based on their similarities or distances. In other words, the algorithm starts with as many clusters as there are data points and then iteratively merges the two most similar clusters until all points belong to a single cluster.

Divisive clustering, on the other hand, is a top-down approach, where all data points are initially considered as a single cluster, and then clusters are successively split into smaller clusters based on their dissimilarities or distances.

In other words, the algorithm starts with a single cluster containing all data points and then iteratively splits it into smaller and smaller clusters until each point belongs to its own cluster.

The main difference between these two types of hierarchical clustering is the direction in which the clusters are formed. Agglomerative clustering starts with individual data points and gradually builds up larger clusters, while divisive clustering starts with a single large cluster and gradually breaks it down into smaller clusters.

The main aim of this is a way to analyze customer behaviour through hierarchical clustering, a powerful data analysis technique that groups similar customers based on their spending patterns and income levels.

So we first need to collect the dataset containing customers' annual income in thousands of dollars and their spending score on a scale of 1 to 100.

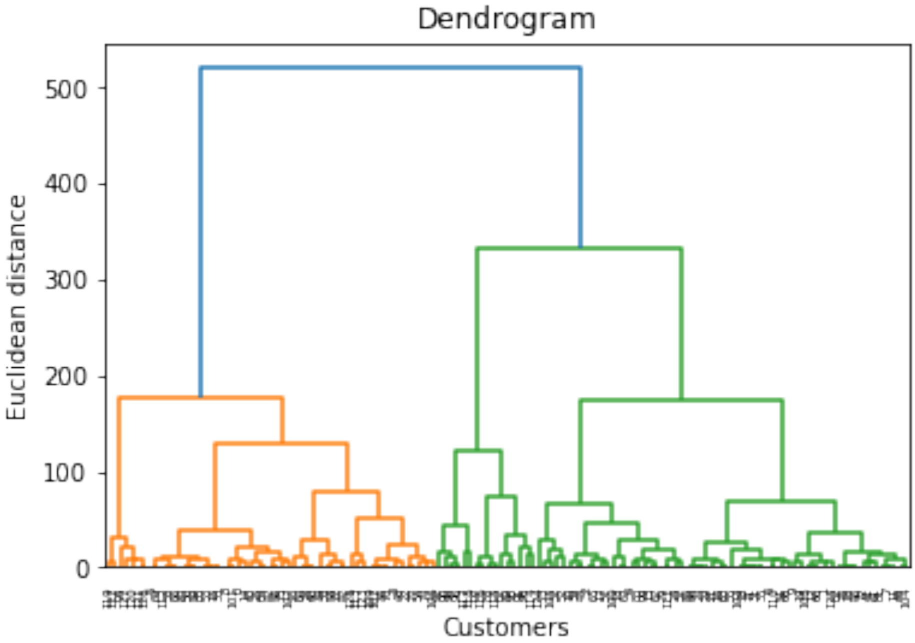
Then we make use of different libraries in python such as Pandas for the datasets containing customers' annual income and spending scores, the matplotlib for plotting purposes and also the scipy for methods for calculating the distance between the newly formed cluster

Methodology:

1. Data collection and preprocessing: Collect the dataset containing customers' annual income and spending scores, and preprocess the data as needed (e.g., removing missing values, and scaling the data). Thus for all this, we make use of Pandas which are used for working with data, including data cleaning, manipulation, and analysis. It provides a flexible and powerful toolset for working with structured data, such as databases, spreadsheets, and CSV files, making it an excellent choice for analyzing customer spending behaviour in a shopping mall, We can also use pandas to create a dendrogram to visualize the results of the clustering analysis and gain insights into the relationship between different customer segments.
2. Distance calculation: Calculate the distance matrix that represents the distance between each pair of data points. There are several distance metrics that can be used for this step like the single, complete, average, weighted, centroid, median and ward methods in which ward will be more accurate and best to use compared to all others. Thus we make use of the ward method for a better understanding.We find all these in the python library of scipy - scipy.cluster.hierarchy.linkage (y,method='single', metric='euclidean', optimal_ordering=False)
3. Linkage calculation: Calculate the linkage matrix that represents the hierarchical clustering of the data. There are several linkage methods that can be used for this step, such as single linkage, complete linkage, or average linkage.
4. Dendrogram creation: Create a dendrogram using Matplotlib to visualize the results of the hierarchical clustering analysis. The dendrogram can be used to identify the number of clusters and the relationships between different clusters.
5. Cluster identification: Use the dendrogram and other methods to identify distinct customer segments based on their spending behaviour.
6. Cluster analysis: Analyze the characteristics of each customer segment using pandas and other tools. For example, we can create scatter plots or histograms to visualize the distribution of annual income and spending scores for each segment.

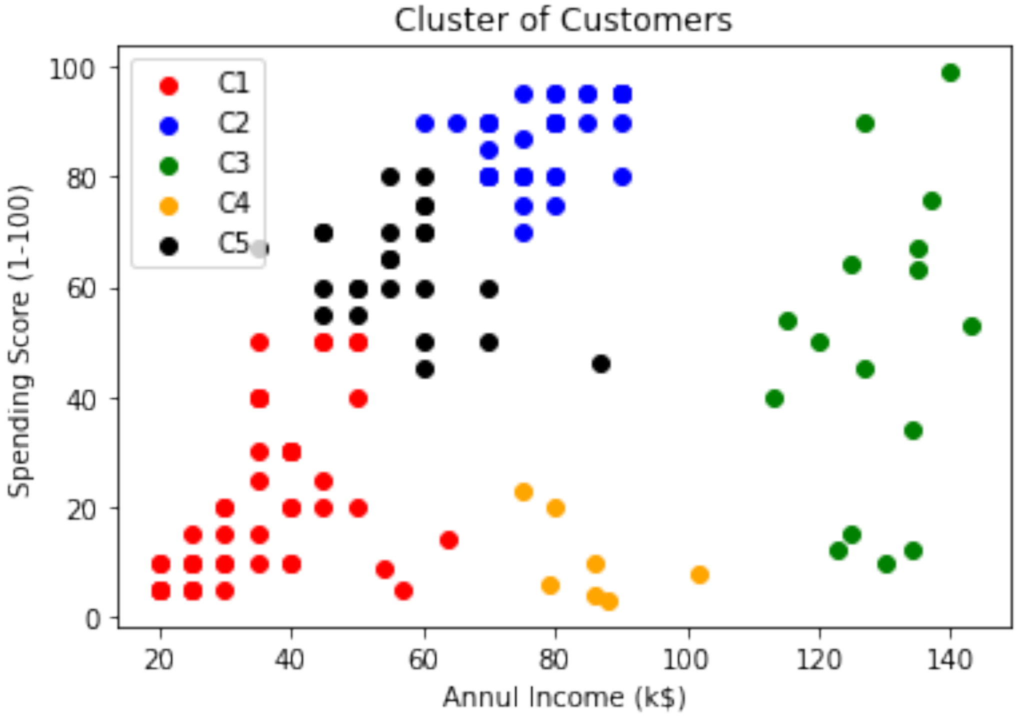
Code:

```
import pandas as pd
import matplotlib.pyplot as plt
dataset = pd.read_csv("Mall_Customers.csv")
X = dataset.iloc[:, :].values
X
#Dendrogram
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()
```

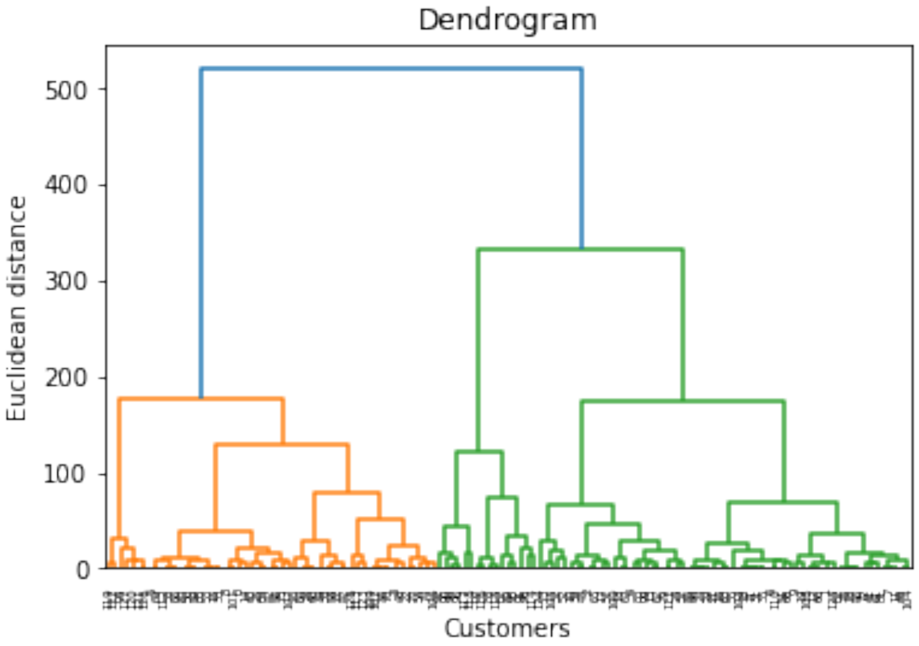


```
from sklearn.cluster import AgglomerativeClustering
clustering = AgglomerativeClustering(n_clusters = 5)
y_hc = clustering.fit_predict(X)
y_hc
#visualization of clusters
plt.scatter(X[y_hc ==0 , 0] , X[y_hc == 0 , 1] , c = "red" , label = "C1")
plt.scatter(X[y_hc ==1 , 0] , X[y_hc == 1 , 1] , c = "blue" , label = "C2")
plt.scatter(X[y_hc ==2 , 0] , X[y_hc == 2 , 1] , c = "green" , label = "C3")
plt.scatter(X[y_hc ==3 , 0] , X[y_hc == 3 , 1] , c = "orange" , label = "C4")
plt.scatter(X[y_hc ==4 , 0] , X[y_hc == 4 , 1] , c = "black" , label = "C5")
plt.title("Cluster of Customers")
plt.xlabel("Annul Income (k$)")
plt.ylabel("Spending Score (1-100)")
plt.legend()
```

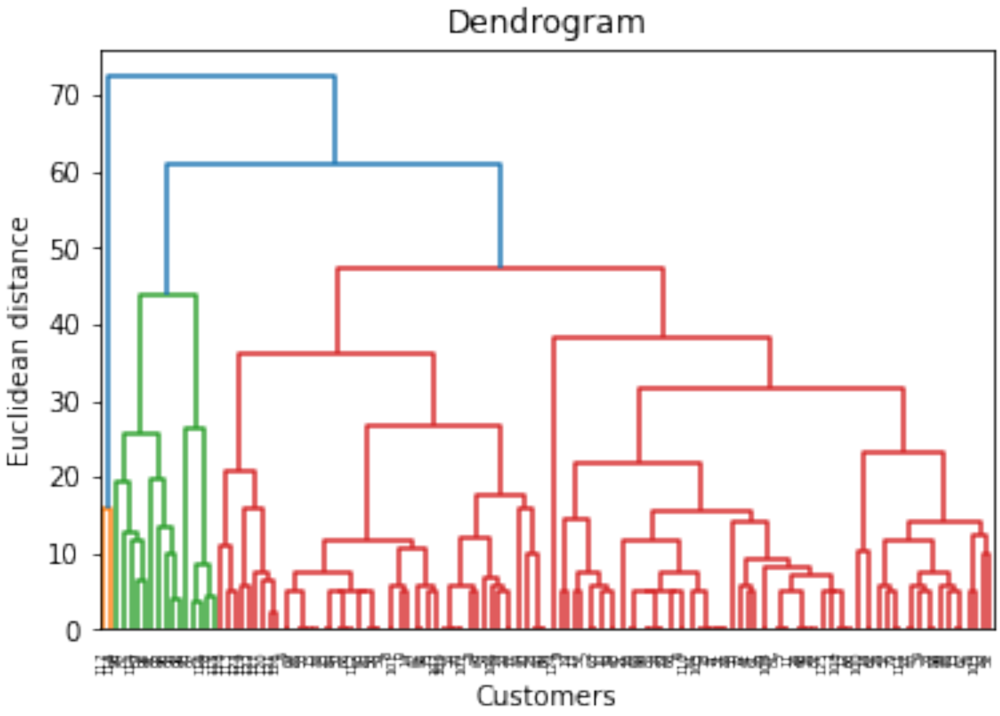
plt.show()



```
dendrogram = sch.dendrogram(sch.linkage(X, method = 'ward'))
plt.title("Dendrogram")
plt.xlabel("Customers")
plt.ylabel("Euclidean distance")plt.show()
```

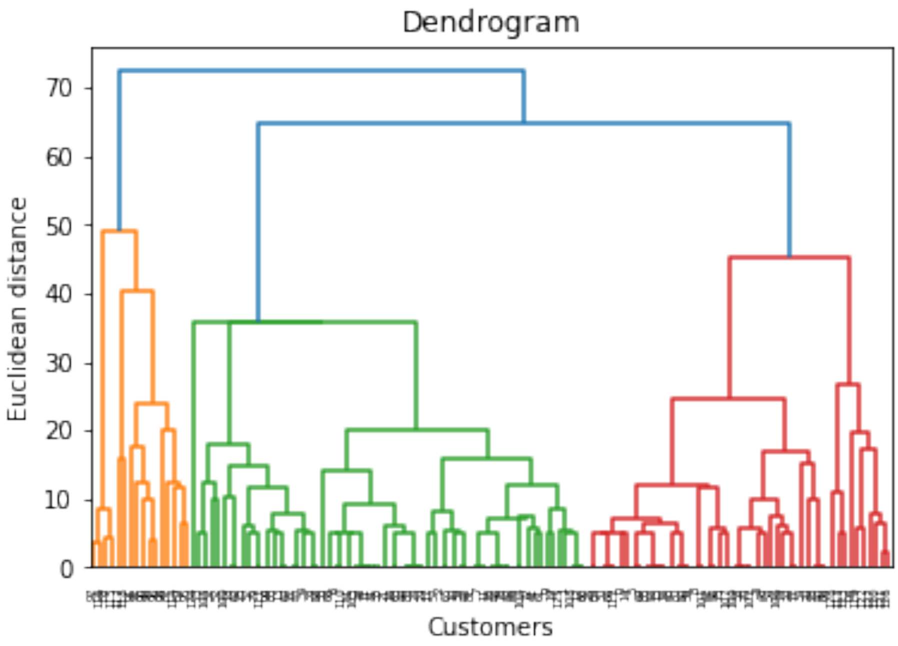


```
dendrogram = sch.dendrogram(sch.linkage(X, method = 'median'))
plt.title("Dendrogram")
plt.xlabel("Customers")
plt.ylabel("Euclidean distance")
```



```
plt.show()
```

```
dendrogram = sch.dendrogram(sch.linkage(X, method = 'centroid'))
plt.title("Dendrogram")
plt.xlabel("Customers")
plt.ylabel("Euclidean distance")
```



```
plt.show()
```

```
In [27]: import pandas as pd
         import matplotlib.pyplot as plt
```

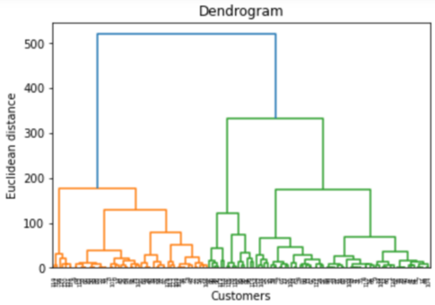
```
In [37]: dataset = pd.read_csv("Mall_Customers.csv") #reading the dataset using pandas
```

```
In [38]: X = dataset.iloc[:, :].values #selecting all the values in the DataFrame and converting them to a NumPy array.
```

```
In [39]: X
```

```
       [ 50,  50],
       [ 35,  40],
       [ 80,  90],
       [ 25,  10],
       [ 90,  95],
       [ 70,  60],
       [ 20,   5],
       [ 50,  40],
       [ 90,  95],
       [ 40,  30],
       [ 55,  60],
       [ 45,  50],
       [ 60,  70],
       [ 35,  40],
       [ 75,  75],
       [ 40,  10],
       [ 60,  70],
       [ 80,  90],
       [ 30,   5],
```

```
In [40]: #Dendrogram
         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()
```



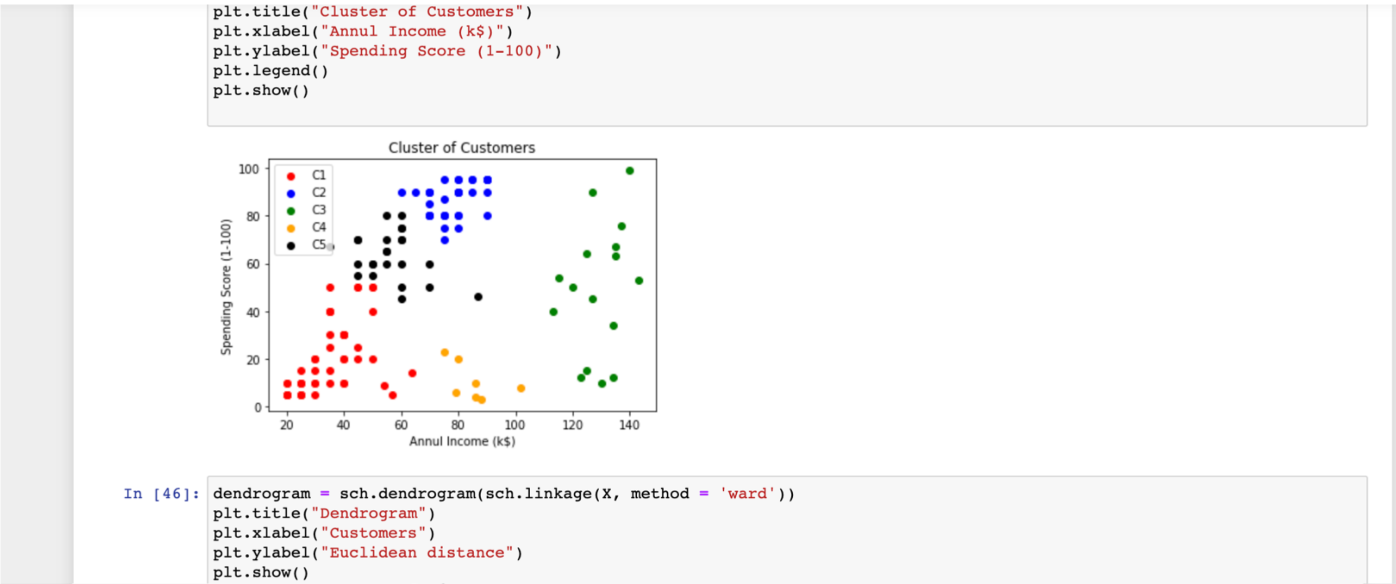
```
In [41]: from sklearn.cluster import AgglomerativeClustering
```

```
In [42]: clustering = AgglomerativeClustering(n_clusters = 5)
         y_hc = clustering.fit_predict(X)
```

```
In [43]: y_hc
```

```
Out[43]: array([0, 1, 4, 0, 4, 0, 1, 1, 0, 1, 4, 1, 1, 4, 0, 0, 1, 4, 4, 1, 0, 1,
                0, 4, 1, 0, 1, 0, 0, 1, 0, 1, 4, 0, 0, 1, 0, 4, 0, 4, 0, 1, 0, 4,
                1, 0, 1, 0, 0, 1, 0, 1, 4, 0, 0, 1, 0, 4, 0, 0, 1, 1, 4, 1, 0, 4,
                0, 4, 1, 0, 0, 1, 1, 4, 0, 0, 1, 0, 1, 4, 0, 1, 1, 4, 0, 0, 1, 2,
                2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 4, 4, 0, 0, 1, 1, 4, 0, 0, 1, 0,
                1, 4, 0, 0, 2, 2, 2, 2, 2, 3, 3, 3, 3, 0, 4, 1, 3, 3, 0])
```

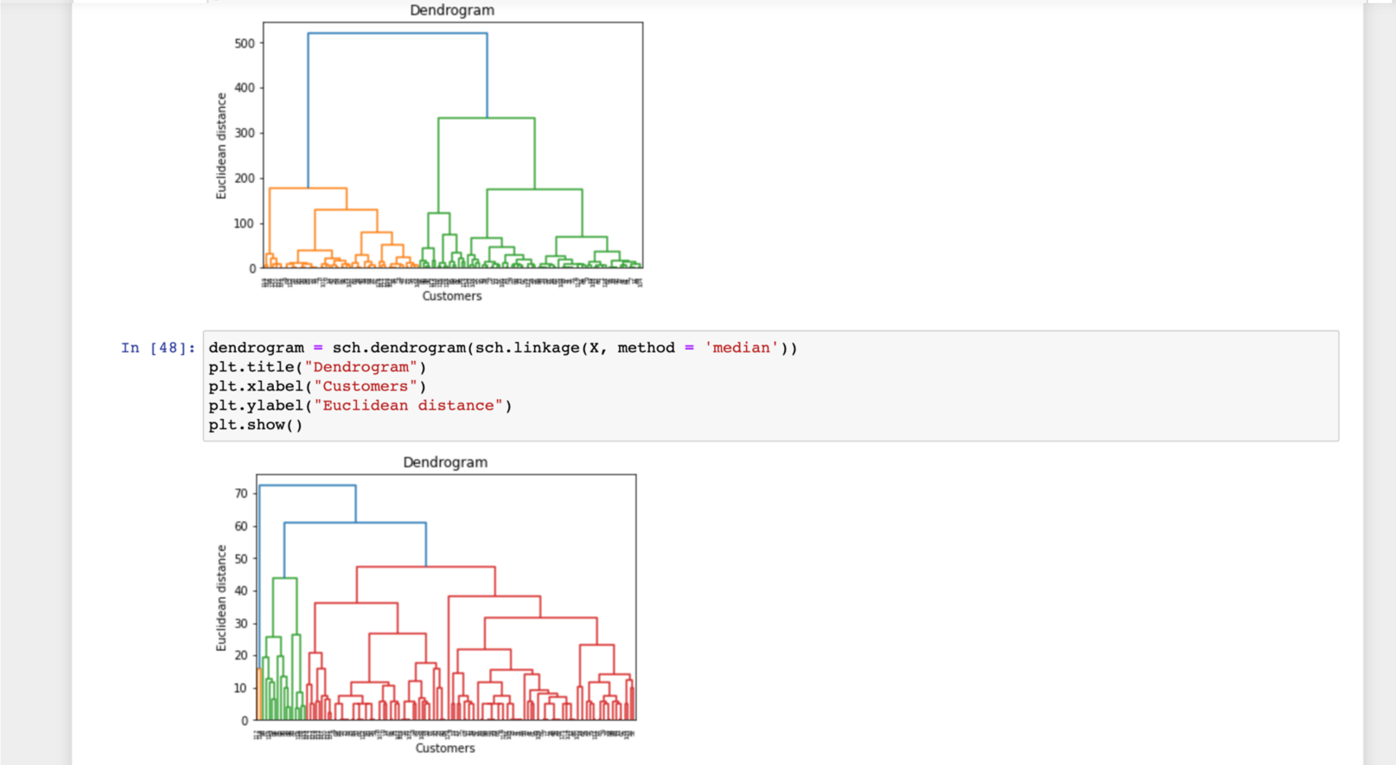
```
In [44]: #visualization of clusters
         plt.scatter(X[y_hc ==0 , 0] , X[y_hc == 0 , 1] , c = "red" , label = "C1")
         plt.scatter(X[y_hc ==1 , 0] , X[y_hc == 1 , 1] , c = "blue" , label = "C2")
         plt.scatter(X[y_hc ==2 , 0] , X[y_hc == 2 , 1] , c = "green" , label = "C3")
         plt.scatter(X[y_hc ==3 , 0] , X[y_hc == 3 , 1] , c = "orange" , label = "C4")
         plt.scatter(X[y_hc ==4 , 0] , X[y_hc == 4 , 1] , c = "black" , label = "C5")
```



```
plt.title("Cluster of Customers")
plt.xlabel("Annul Income (k$)")
plt.ylabel("Spending Score (1-100)")
plt.legend()
plt.show()
```

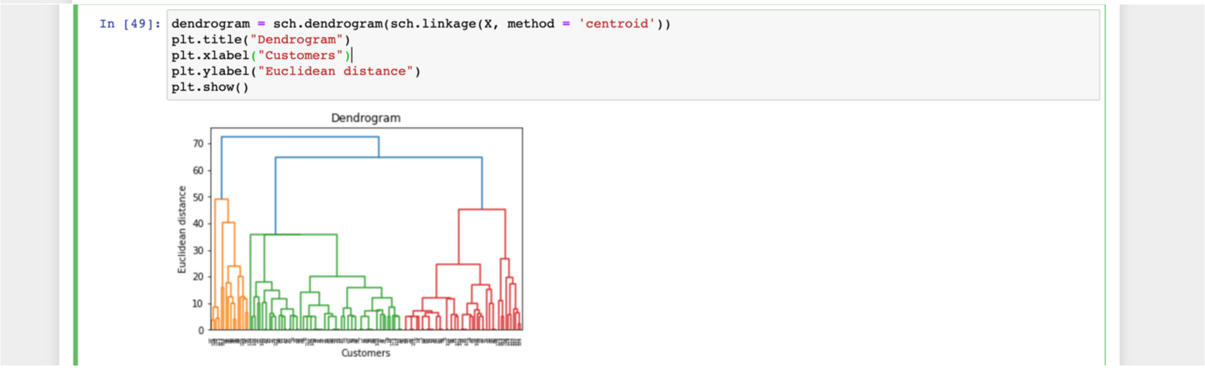
In [46]:

```
dendrogram = sch.dendrogram(sch.linkage(X, method = 'ward'))
plt.title("Dendrogram")
plt.xlabel("Customers")
plt.ylabel("Euclidean distance")
plt.show()
```



In [48]:

```
dendrogram = sch.dendrogram(sch.linkage(X, method = 'median'))
plt.title("Dendrogram")
plt.xlabel("Customers")
plt.ylabel("Euclidean distance")
plt.show()
```



In [49]:

```
dendrogram = sch.dendrogram(sch.linkage(X, method = 'centroid'))
plt.title("Dendrogram")
plt.xlabel("Customers")
plt.ylabel("Euclidean distance")
plt.show()
```

Conclusion:

In conclusion, hierarchical clustering is a powerful method for analyzing and clustering data points based on their similarities or distances. In this project, we applied hierarchical clustering to a dataset of mall customers' annual income and spending scores to identify distinct customer segments and analyze their characteristics.

We first collected and preprocessed the dataset using pandas and other tools, then calculated the distance and linkage matrices to perform hierarchical clustering. We then used Matplotlib to create a dendrogram to visualize the clustering results and identify distinct clusters.

Finally, we analyzed the characteristics of each customer segment using pandas and other tools, creating scatter plots, histograms, and other visualizations to gain insights into the spending behaviours of different customer groups.

Overall, this project demonstrates the power and versatility of hierarchical clustering as a method for analyzing and clustering data, and provides a useful example of how to apply this method to real-world datasets in the context of customer segmentation in the retail industry.