Chittagong University of Engineering and Technology

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Course Name: Data Communication (Sessional)

COURSE CODE: CSE- 314

Clustering of Mall Customers by K-Means & Hierarchical clustering

---

Submitted To:

- **Prof. Dr. Pranab Kumar Dhar**

  Professor, Department of CSE, CUET.

- **Dr. Mahfuzulhoq Chowdhury**

  Associate Professor, Department of CSE, CUET

Submitted By:

Sadia Islam Nova	**Salman Farsi**	**Abu Saiyed Mohammad Sadat**
ID: **1804091**	ID: **1804102**	ID: **1804105**
Section: B1	Section: B2	Section: B2

Date of Submission: 03 September, 2022

Contents

Abstract

On this project, we're going to make the clusters of a datasets from the shopping mall customers based on their different characteristics like age, gender, income, spending etc. by K-Means Clustering and Hierarchical Clustering. At the end of this study, I hope we could achieve the following understandings regarding our problem:

- What's a good way to segment our dataset on a small set of clusters?
- How can we achieve quick results using the **Pandas**, **Numpy**, **Matplotlib**, **Pyplot** and **SKLearn** modules?
- How the select the best hyperparameters for K-Means Clustering?
- How to get the **Hierarchical** Agglomerative Bottom-up approach of clustering?
- How to display and visualize data in the most honest and friendly way to our stakeholders?

Introduction

Clustering is an unsupervised machine learning task.Using a clustering algorithm means we're going to give the algorithm a lot of input data with no labels and let it find any groupings in the data it can. Those groupings are called *clusters*. A cluster is a group of data points that are similar to each other based on their relation to surrounding data points. Clustering is used for things like feature engineering or pattern discovery.

There are different types of clustering algorithms that handle all kinds of unique data.Centroid-based clustering is the one which is probably heard about the most. It's a little sensitive to the initial parameters we give it, but it's fast and efficient. These types of algorithms separate data points based on multiple centroids in the data. Each data point is assigned to a cluster based on its squared distance from the centroid. This is the most commonly used type of clustering.

Hierarchical-based clustering is typically used on hierarchical data, like we would get from a company database or taxonomies. It builds a tree of clusters so everything is organized from the top-down.This is more restrictive than the other clustering types, but it's perfect for specific kinds of data sets.

When we have a set of unlabeled data, it's very likely that we'll be using some kind of unsupervised learning algorithm.There are a lot of different unsupervised learning techniques, like neural networks, reinforcement learning, and clustering. The specific type of algorithm we want to use is going to depend on what your data looks like.We might want to use clustering when we're trying to do anomaly detection to try and find outliers in our data. It helps by finding those groups of clusters and showing the boundaries that would determine whether a data point is an outlier or not.If we aren't sure of what features to use for our machine learning model, clustering discovers patterns we can use to figure out what stands out in the data. Clustering is especially useful for exploring data you know nothing about. It might take some time to figure out which type of clustering algorithm works the best, but when we do, we'll get invaluable insight on our data. We might find connections we never would have thought of.

Some real world applications of clustering include fraud detection in insurance, categorizing books in a library, and customer segmentation in marketing. It can also be used in larger problems, like earthquake analysis or city planning.

Methodology

1.1. Data Collection

Before moving on to the action, let's have some grasp on the dataset acquired. The dataset used was downloaded from Mall Customer Segmentation a Kaggle fictional public available dataset. The dataset contains **200** clients information, comprising useful data as each client **Gender**, **Age**, **Annual Income** (in thousands of dollars) and a **Spending Score**, attributted from the consuming historics and potential. Here, We are going to show some of the dataset metrics, general visualization and some simple graphics regarding its data, to improve our comprehension about a future model to be implemented.

1.2. Data Preprocessing

In the process of data processing, redundant and null values are removed from the data set. We processed the data in the section through **python panda library** by reading the CSV file. Therefore we then do the following simple approach for data preprocessing,

- Select K random points
- Calculate centroid for all the points and assign points to closest centroid
- Repeat till converge

1.3. First Clustering By Age and Spending Score

At first, let's assume that these two columns could form clusters together, so let's just apply the fitting function at the class sklearn.Kmeans. The purpose here is to obtain as many values of cluster inertia as possible, selecting the closest value to the elbow of the inertia X cluster_number graph.

1.4. Second Clustering By Annual Income and Spending Score

In this section we did a second clustering after the first one. We for this purpose use Annual Income and Spending Score columns for the clustering.

1.5. Final Clustering By Age, Annual Income and Spending Score

Here we have an example of a multi-dimensional K-Means Clustering Analysis. The curve of inertia here for these section of three columns is harder than the previous ones to interpret, but a visual inspection at the plot is of help in these situations.After calculating the respective clusters, it's a good idea to append this information at the dataset.

1.6. Hierarchical Clustering of Whole Data Sets

In this part of customers segmentation analysis, I will use agglomerative hierarchical clustering (also known as bottom-up approach), a method used o group objects based on their similarity. At the beginning each observation starts in its own cluster, and step by step pairs of clusters are merged as we move up the hierarchy. Before I implement the clustering algorithm, I will compare the distances between the data points and in the next step the 'hclust' function will be used to perform the cluster analysis.

1.7. K-Means Clustering Analysis

From a visual perspective of the graph above, we can identify some clusters of clients, from a raw behavioural perspective.

- *Cluster 0* (**Dark Blue**): Clients of all ages, with a low annual income and low spending score.
- *Cluster 1* (**Purple**): Young clients (< 40 years old) with low average annual income and high spending score.
- *Cluster 2* (**Magenta**): Clients of all ages, with a an average to high annual income and low spending score.
- *Cluster 3* (**Red**): Clients with age greather than 50 years, with an average annual income and average spending score.
- *Cluster 4* (**Orange**): Young clients (< 40 years old), with high annual income and high spending score.
- *Cluster 5* (**Yellow**): Clients with age greather than 50 years, with an average annual income and average spending score.

Results and Discussion

Initialization

Module Imports and Style Definition

```
In [11]: import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as sns
         import plotly.express as px
         from sklearn.cluster import KMeans
         plt.style.use('Solarize_Light2')
```

Read the CSV File

In [12]:
```
# Get dataframe from CSV file
df = pd.read_csv('Mall_Customers.csv')
df.head()
```

Out[12]:

	CustomerID	Gender	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

In [13]:
```
df.shape
```

Out[13]: (200, 5)

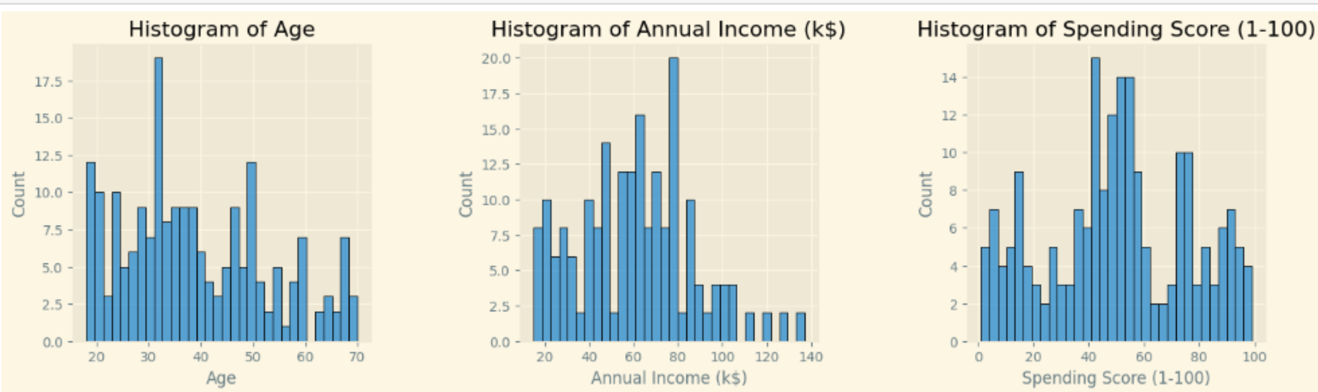
In [14]:
```
df.describe()
```

Out[14]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

Draw The Histogram of Data Sets

In [18]:
```
plt.figure(1, figsize=(16,4))
n = 0
for i in ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']:
    n += 1
    plt.subplot(1 , 3 , n)
    plt.subplots_adjust(hspace =0.5 , wspace = 0.5)
    sns.histplot(df[i] , bins = 32)
    plt.title(f'Histogram of {i}')
plt.show()
```



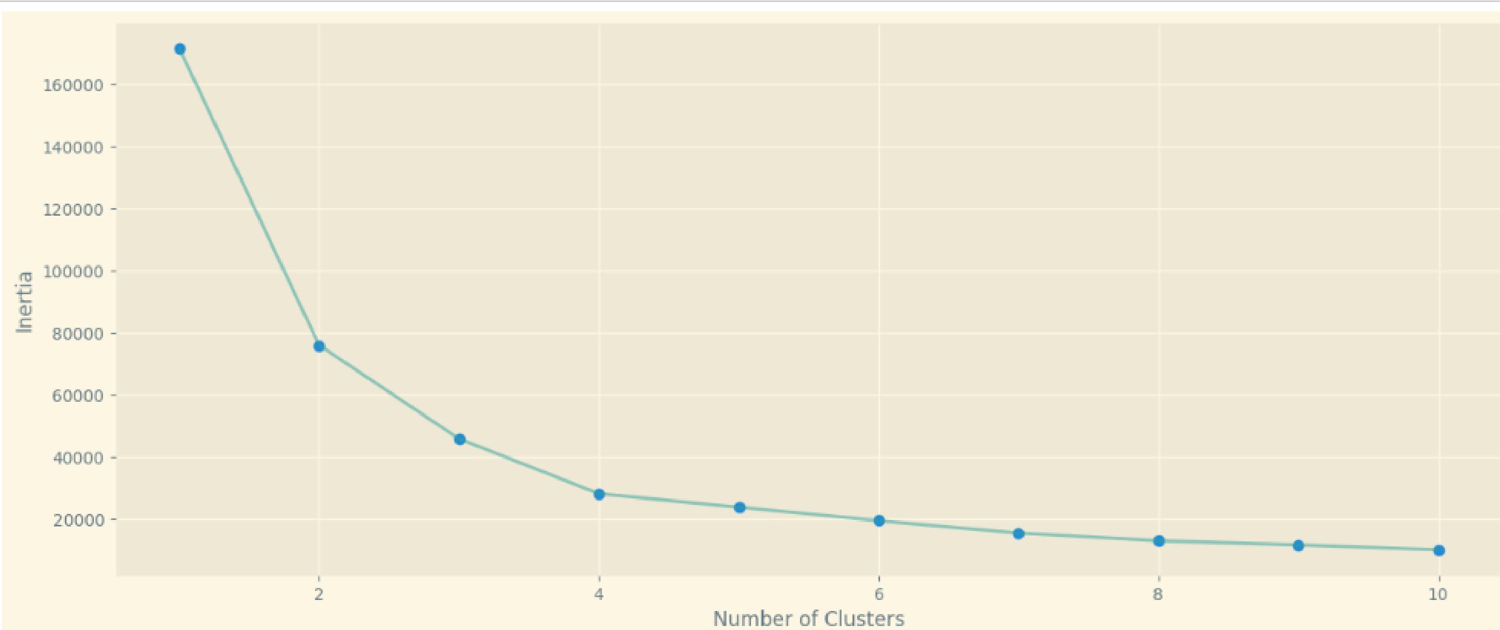
First Clustering By Age and Spending Score

In [ ]:

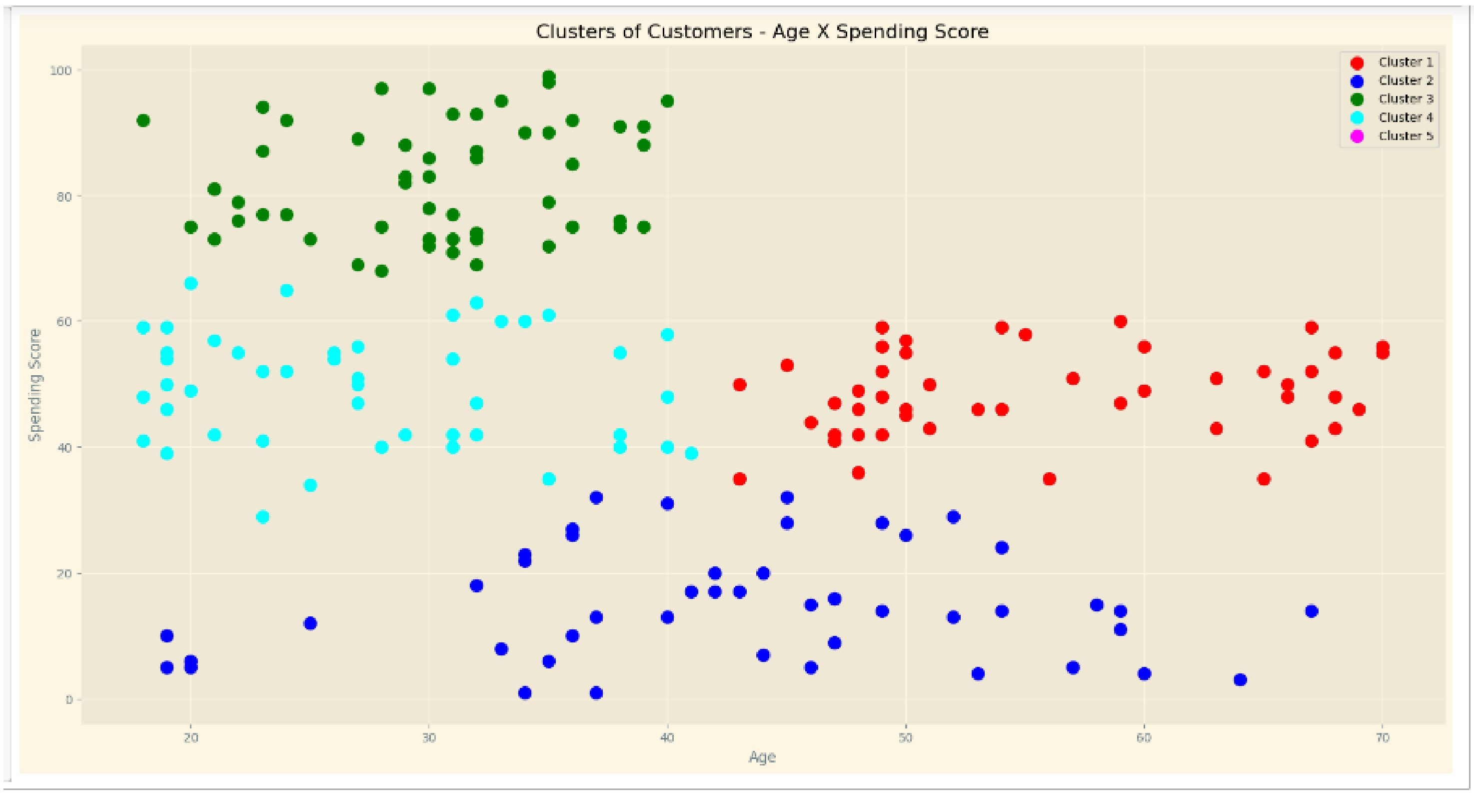
```
# Assignment Stage

X1 = df.loc[:, ['Age', 'Spending Score (1-100)']].values
inertia = []
for n in range(1 , 11):
    model = KMeans(n_clusters = n,
                   init='k-means++',
                   max_iter=500,
                   random_state=42)
    model.fit(X1)
    inertia.append(model.inertia_)
```

In [ ]:

```
plt.figure(1 , figsize = (15 ,6))
plt.plot(np.arange(1 , 11) , inertia , 'o')
plt.plot(np.arange(1 , 11) , inertia , '-' , alpha = 0.5)
plt.xlabel('Number of Clusters') , plt.ylabel('Inertia')
plt.show()
```



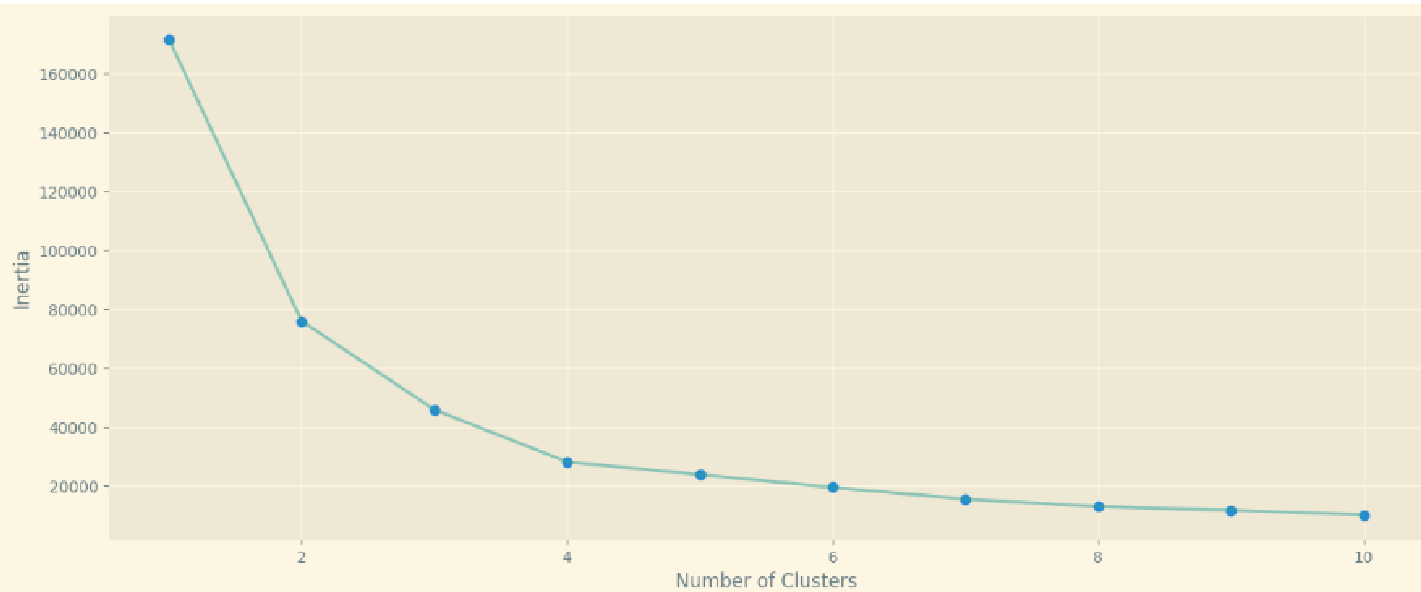
In [21]:
```
model = KMeans(n_clusters = 4,
            init='k-means++',
            max_iter=500,
            random_state=42)
model.fit(X1)
labels = model.labels_
centroids = model.cluster_centers_
y_kmeans = model.fit_predict(X1)

plt.figure(figsize=(20,10))
plt.scatter(X1[y_kmeans == 0, 0], X1[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X1[y_kmeans == 1, 0], X1[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X1[y_kmeans == 2, 0], X1[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X1[y_kmeans == 3, 0], X1[y_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X1[y_kmeans == 4, 0], X1[y_kmeans == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
plt.title('Clusters of Customers - Age X Spending Score')
plt.xlabel('Age')
plt.ylabel('Spending Score')
plt.legend()
plt.show()
```



Second Clustering By Annual Income and Spending Score

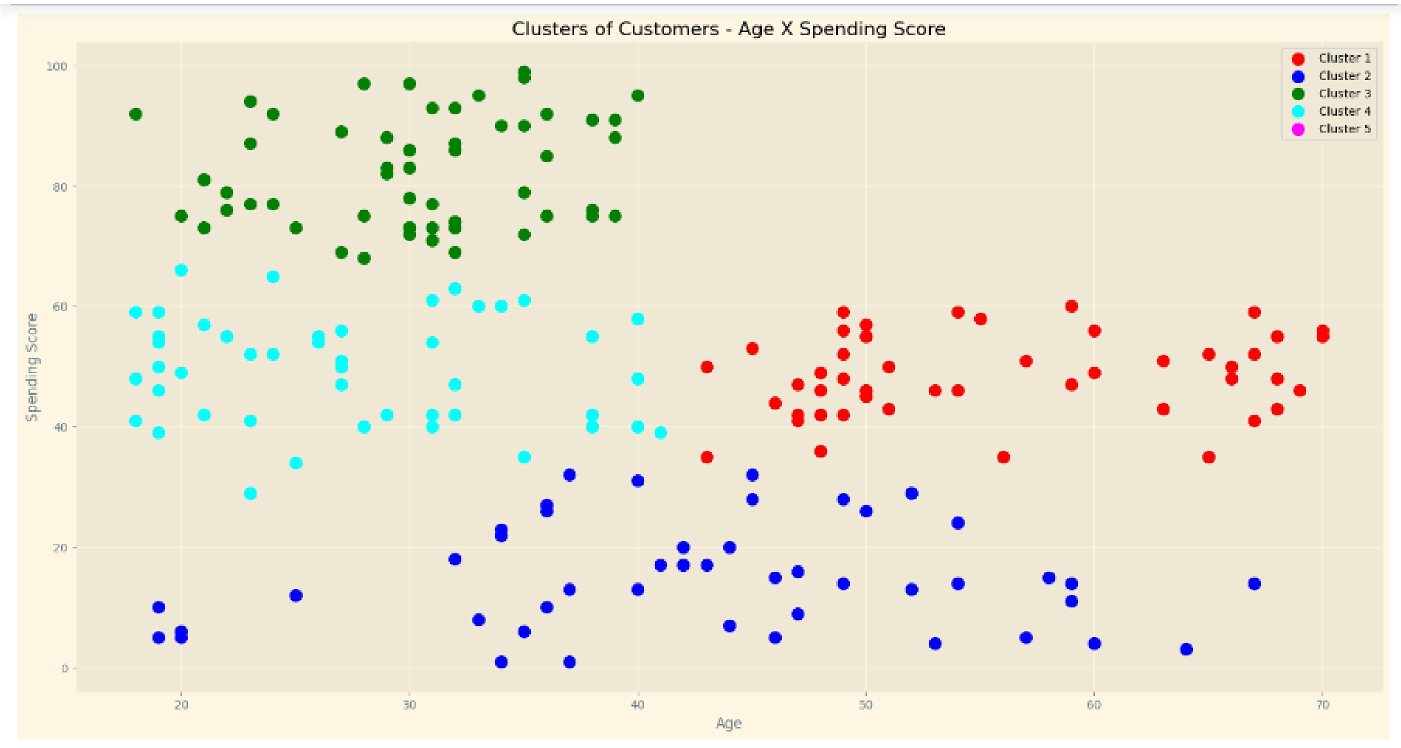
In [ ]:

```
# Assignment Stage

X2 = df.loc[:, ['Annual Income (k$)', 'Spending Score (1-100)']].values
inertia = []
for n in range(1 , 11):
    model = KMeans(n_clusters = n,
                   init='k-means++',
                   max_iter=500,
                   random_state=42)
    model.fit(X2)
    inertia.append(model.inertia_)

plt.figure(1 , figsize = (20, 10))
plt.plot(np.arange(1 , 11) , inertia , 'o')
plt.plot(np.arange(1 , 11) , inertia , '-' , alpha = 0.5)
plt.xlabel('Number of Clusters') , plt.ylabel('Inertia')
plt.show()
```



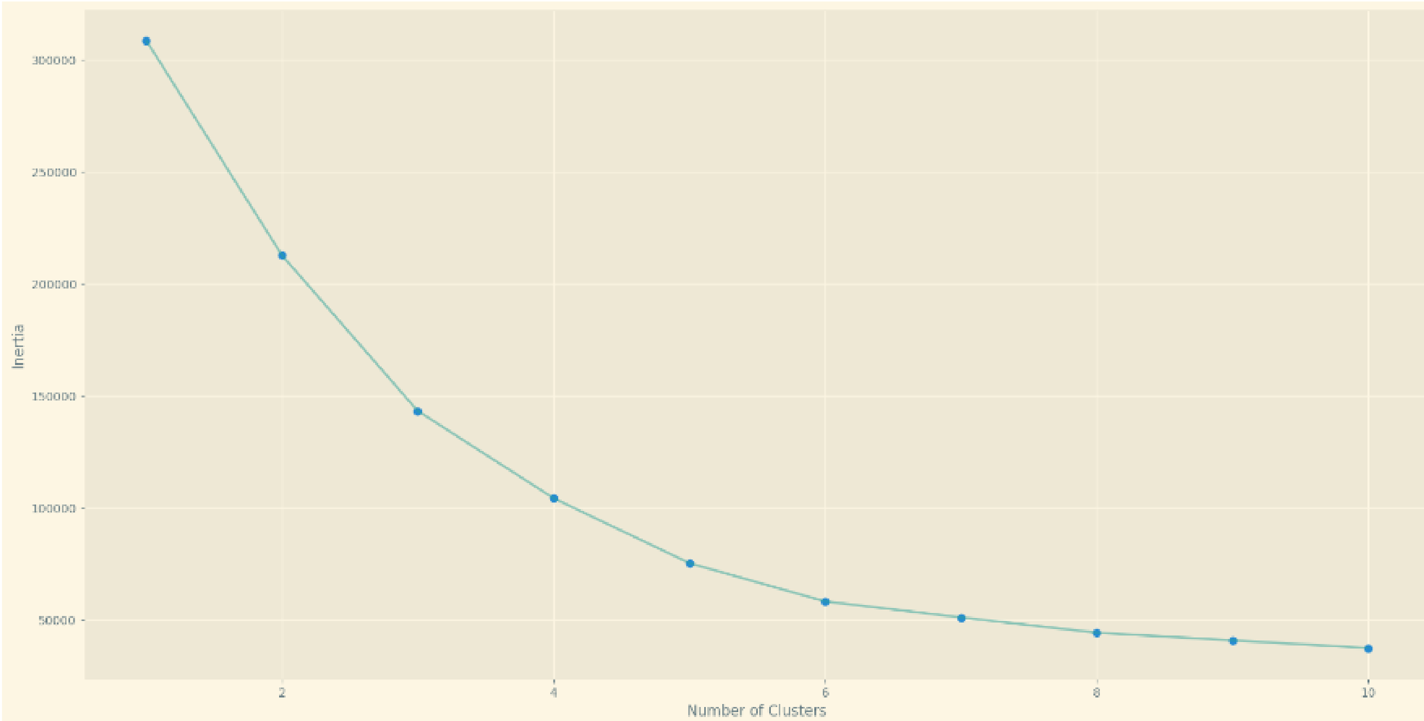
In [21]:

```
model = KMeans(n_clusters = 4,
            init='k-means++',
            max_iter=500,
            random_state=42)
model.fit(X1)
labels = model.labels_
centroids = model.cluster_centers_
y_kmeans = model.fit_predict(X1)

plt.figure(figsize=(20,10))
plt.scatter(X1[y_kmeans == 0, 0], X1[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X1[y_kmeans == 1, 0], X1[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X1[y_kmeans == 2, 0], X1[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X1[y_kmeans == 3, 0], X1[y_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X1[y_kmeans == 4, 0], X1[y_kmeans == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
plt.title('Clusters of Customers - Age X Spending Score')
plt.xlabel('Age')
plt.ylabel('Spending Score')
plt.legend()
plt.show()
```



Final Clustering By Age, Annual Income and Spending Score

In [24]:
```
# Assignment Stage

from sklearn.cluster import KMeans

X3 = df.loc[:, ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']].values
inertia = []
for n in range(1 , 11):
    model = KMeans(n_clusters = n,
                init='k-means++',
                max_iter=500,
                random_state=42)
    model.fit(X3)
    inertia.append(model.inertia_)

plt.figure(1 , figsize = (20, 10))
plt.plot(np.arange(1 , 11) , inertia , 'o')
plt.plot(np.arange(1 , 11) , inertia , '-' , alpha = 0.5)
plt.xlabel('Number of Clusters') , plt.ylabel('Inertia')
plt.show()
```



In [25]:
```
model = KMeans(n_clusters = 6,
               init='k-means++',
               max_iter=500,
               random_state=42)
model.fit(X3)
labels = model.labels_
#centroids = model.cluster_centers_

df['cluster'] =  labels
df
```

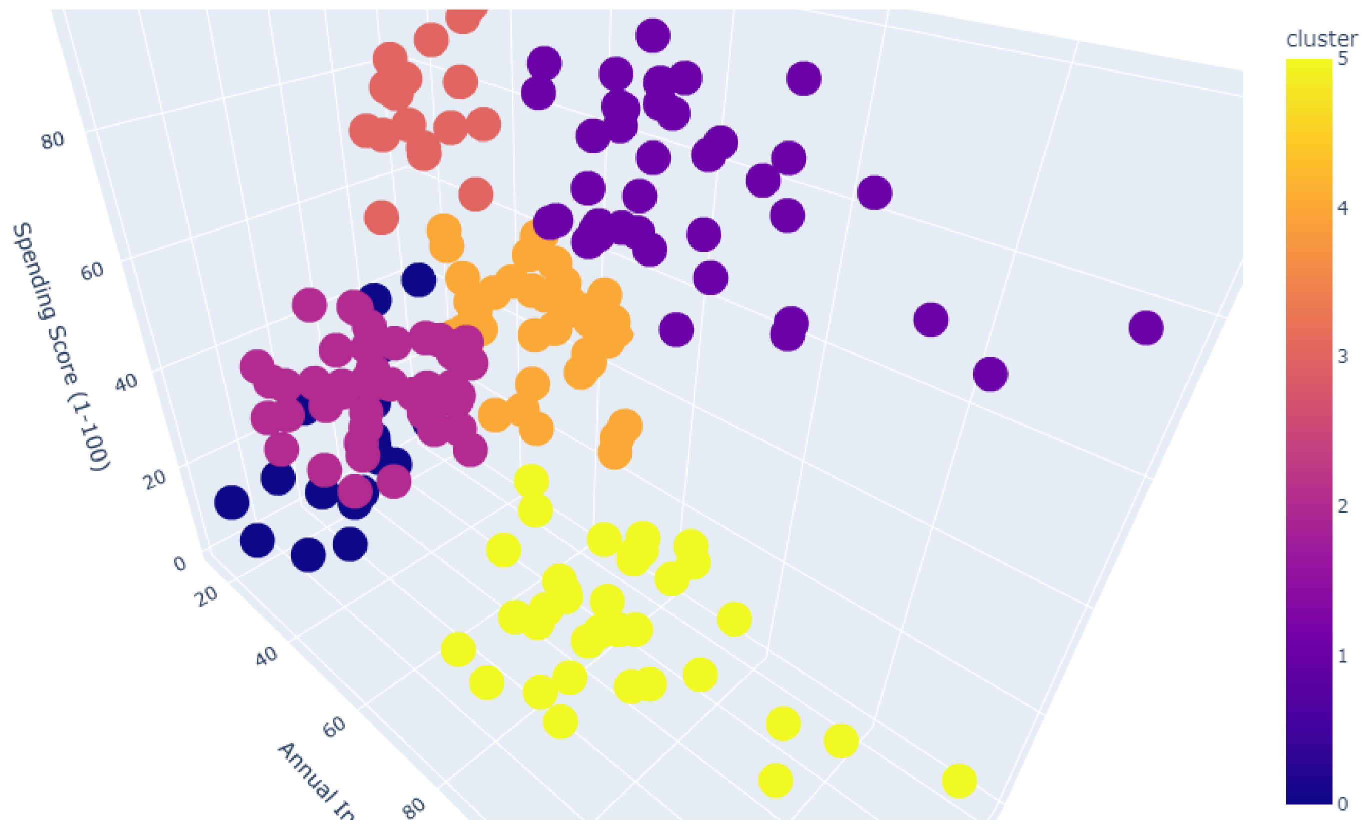
Out[25]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	cluster
0	1	Male	19	15	39	0
1	2	Male	21	15	81	3
2	3	Female	20	16	6	0
3	4	Female	23	16	77	3
4	5	Female	31	17	40	0
...	...	...	...	...	...	...
195	196	Female	35	120	79	1
196	197	Female	45	126	28	5
197	198	Male	32	126	74	1
198	199	Male	32	137	18	5
199	200	Male	30	137	83	1

200 rows × 6 columns

3-D view of clusters

In [26]:
```
fig = px.scatter_3d(df,
                    x="Age",
                    y="Annual Income (k$)",
                    z="Spending Score (1-100)",
                    color='cluster',
                    hover_data=["Age",
                                "Annual Income (k$)",
                                "Spending Score (1-100)"],
                    category_orders = {"cluster": range(0, 5)},
                    )

fig.update_layout(margin=dict(l=0, r=0, b=0, t=0))
fig.show()
```

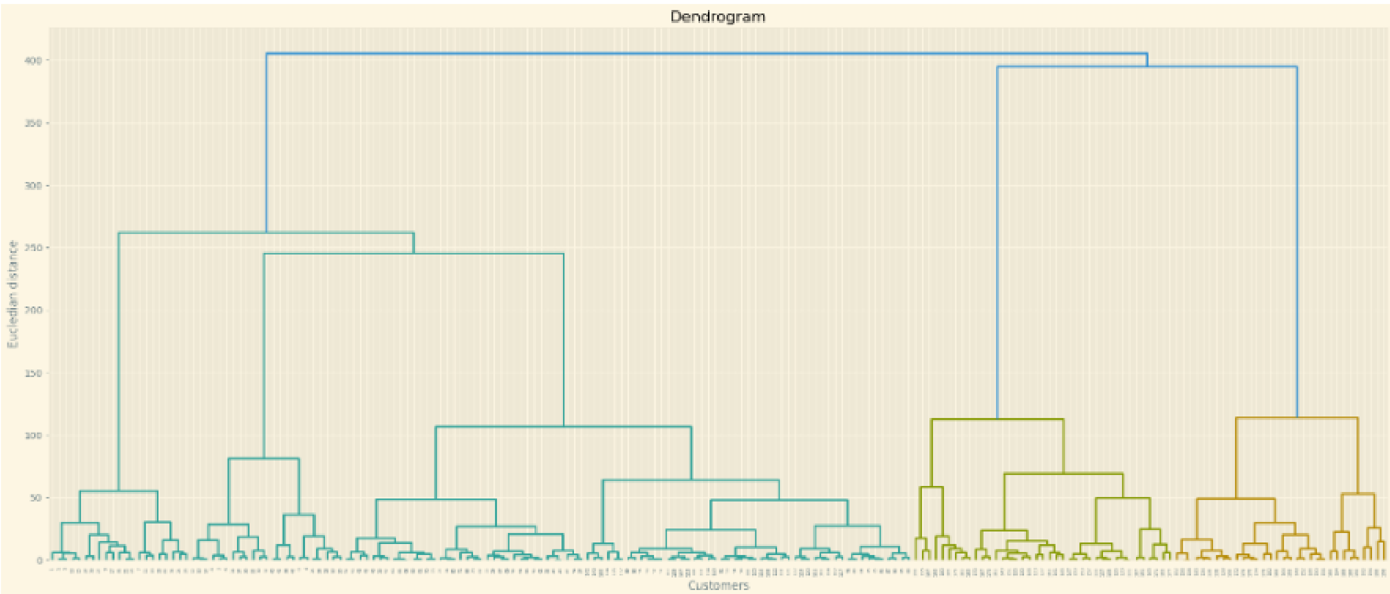


Hierarchical Clustring

In [31]:
```
import scipy.cluster.hierarchy as sch
```

In [32]:
```
# Visualising the dendrogram
fig = plt.figure(figsize=(25, 10))
dendrogram=sch.dendrogram(sch.linkage(X2,method='ward'))
plt.title("Dendrogram")
plt.xlabel("Customers")
plt.ylabel("Eucledian distance")
plt.show()
```

Dendogram of Hierarchical Clustering




```
In [33]: from sklearn.cluster import AgglomerativeClustering
```

```
In [34]: hc=AgglomerativeClustering(n_clusters=5,affinity="euclidean",linkage="ward")
```

```
In [35]: y_hc=hc.fit_predict(X2)
```

```
In [36]: y_hc
```

```
Out[36]: array([4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3,
                4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 1,
                4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2, 0, 2, 0, 2,
                1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 1, 2, 0, 2, 1, 2, 0, 2, 0, 2, 0, 2,
                0, 2, 0, 2, 0, 2, 1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
                0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
                0, 2], dtype=int64)
```

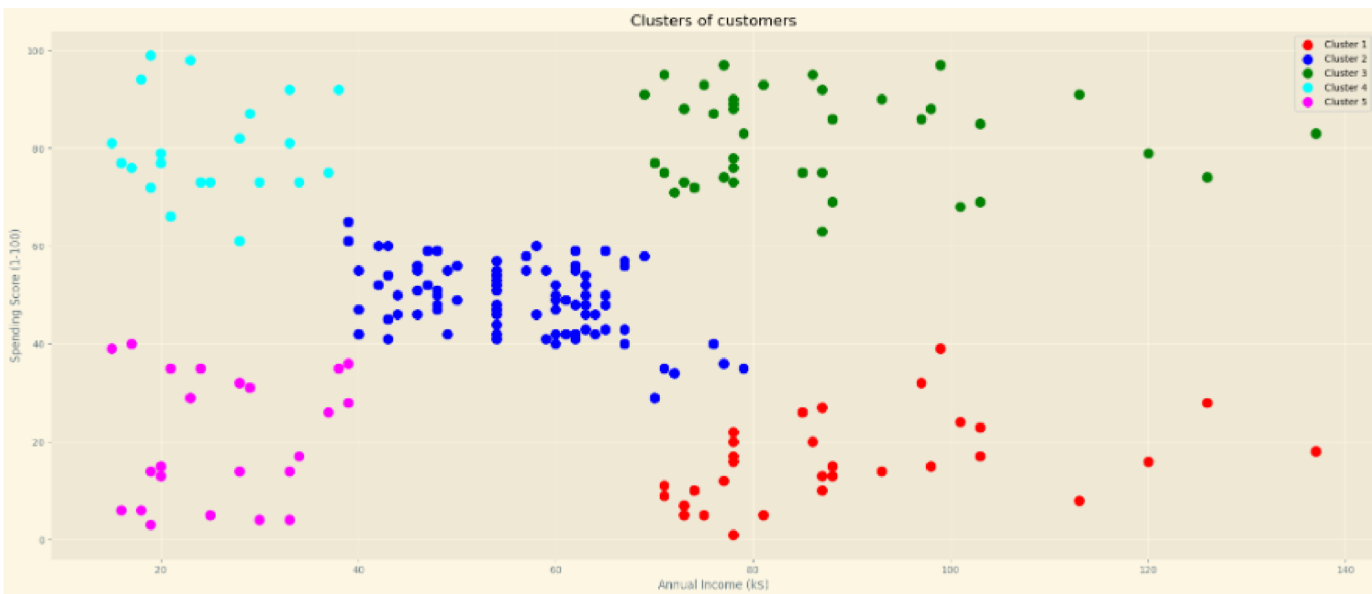
```
In [37]: y_hc.astype
```

```
Out[37]: <function ndarray.astype>
```

Final Clustring Results of Hierarchical Method

In [39]:

```
# Visualising the clusters
fig = plt.figure(figsize=(25, 10))
plt.scatter(X2[y_hc == 0, 0], X2[y_hc == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X2[y_hc == 1, 0], X2[y_hc == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X2[y_hc == 2, 0], X2[y_hc == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X2[y_hc == 3, 0], X2[y_hc == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X2[y_hc == 4, 0], X2[y_hc == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```



Conclusion

After all the data processing ad visualization, it's safe to assume that this particular dataset was clustered in some pretty efficient ways. Regarding the age X spending score, we see that it's a bit unclear how the data could be clustered, so the algorithm help us a lot. The other graphs are way easier to understand, despite the fact that the last one give us a multidimensional analysis and let's us split the customers in more personalizaed groups.

This have some interesting pratical applications. For example, customers from *cluster 2* might be biased to spent more of their income on a particular business service, while customers from *cluster 0* might be not. In another way, *cluster 4* represents young customers, with high acquisitive power, individuals who the business would like to preserve as much as possible on its customer base.