SC4020 DATA ANALYTICS & MINING | PROJECT 1

Group No. 21

2024

Name	**Matric No**	**Sign**
Hendy	U2122559J	
Goyal Ananya Surendrakumar	U2023124B	
Lee Bo Hua	U2122595D	
Mitra Ren Sachithananthan	U2020190D	

Table Of Contents

1. Introduction

Clustering is a fundamental technique in data analysis that involves grouping a set of objects in such a way that objects within the same group (or cluster) are more similar to each other than to those in other groups. This unsupervised learning approach is crucial in various applications, including market segmentation, social network analysis, organisation of computing clusters, and image compression. Clustering enables the identification of inherent structures within data, facilitating insights and enabling informed decision-making.

In this project, we examine and assess the effectiveness of many clustering algorithms on a range of datasets by utilising the information acquired in NTU's SC4020 Data Analytics and Mining course. The Wine dataset [1] from the UCI Machine Learning Repository and the Mall Customer Segmentation dataset [2] will be used in this project. These datasets include various features, allowing for a comprehensive evaluation of the clustering techniques.

Our team will use K-Mean Clustering and Hierarchical Clustering, introduced and discussed in the SC4020 course, to tackle the clustering task. By minimising the sum of squared distances between each point and the centroid of its cluster, the centroid-based clustering method K-Means divides the dataset into a predetermined number of clusters. In contrast, Hierarchical Clustering creates a dendrogram structure like a tree by dividing or merging clusters according to how related they are.

2. Problem Statement

This project's main goal is to assess and contrast the performance of various clustering methods on actual datasets. When it comes to making decisions about customer segmentation, product targeting, and data-driven marketing, among other things, clustering is essential for revealing hidden patterns and groupings in data. However, because different datasets have different properties and different clustering techniques have different strengths, choosing the right clustering algorithm and figuring out how many clusters are best can be difficult.

In this project, we focus on two datasets:

1. **Wine Dataset [1]**
   This dataset contains the chemical properties of wines derived from different cultivars, allowing us to explore how clustering can help distinguish between them based on their chemical composition.

2. **Mall Customer Segmentation Dataset [2]**
   This dataset provides information about customer spending habits and demographics, enabling us to cluster customers for potential marketing or segmentation strategies.

The challenge lies in understanding how different clustering algorithms **K-Means** and **Hierarchical Clustering** perform on these datasets. Each algorithm has distinct characteristics:

- **K-Means Clustering** requires specifying the number of clusters beforehand and is highly efficient for large datasets but may struggle with non-spherical or uneven cluster sizes.
- **Hierarchical Clustering**, which includes both agglomerative and divisive approaches, does not require prior knowledge of the number of clusters and produces a dendrogram, but may not scale well for very large datasets.

The goal of this project is to apply these algorithms to the Wine and Mall datasets, evaluate their performance, and determine which algorithm is better suited for each dataset.

Through this analysis, we aim to provide a comprehensive understanding of the strengths and limitations of K-Means and Hierarchical Clustering and offer insights into how these methods can be applied effectively to real-world data.

3. Dataset 1: Wine Chemical Data

3.1 Dataset Cleaning and Preprocessing

The dataset provided in the wine chemical data contains 178 samples, with 13 chemical attributes that analyse the chemical components in the wines grown in the Italy region. Here are the 13 attributes provided in the dataset. The dataset can be easily imported with Python code as shown below.

Variable Name	Role	Type	Description	Units	Missing Valu
class	Target	Categorical			no
Alcohol	Feature	Continuous			no
Malicacid	Feature	Continuous			no
Ash	Feature	Continuous			no
Alcalinity_of_ash	Feature	Continuous			no
Magnesium	Feature	Integer			no
Total_phenols	Feature	Continuous			no
Flavanoids	Feature	Continuous			no
Nonflavanoid_phenols	Feature	Continuous			no
Proanthocyanins	Feature	Continuous			no
Color_intensity	Feature	Continuous			no
Hue	Feature	Continuous			no
0D280_0D315_of_diluted_wines	Feature	Continuous			no
Proline	Feature	Integer			no

Install the ucimlrepo package

```
pip install ucimlrepo
```

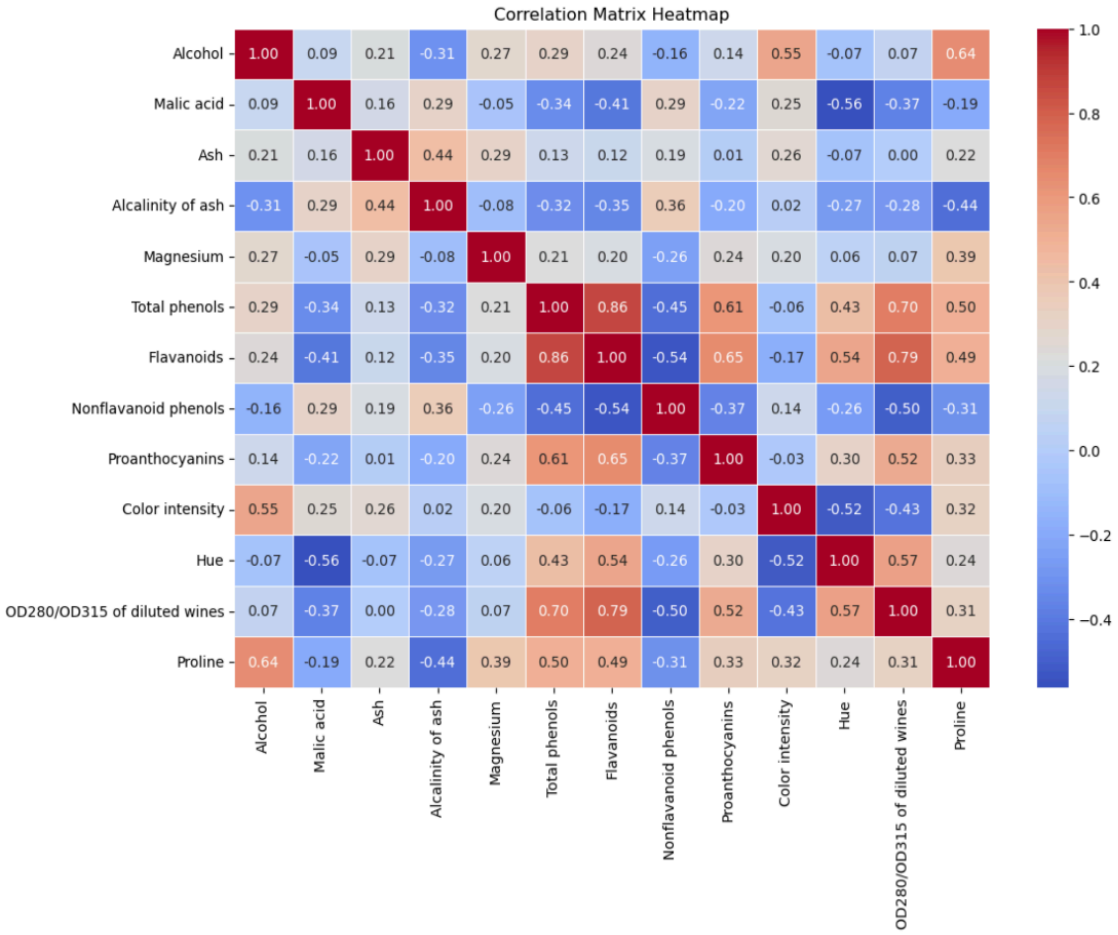
Import the dataset into your code

```
from ucimlrepo import fetch_ucirepo

# fetch dataset
wine = fetch_ucirepo(id=109)

# data (as pandas dataframes)
X = wine.data.features
y = wine.data.targets

# metadata
print(wine.metadata)

# variable information
print(wine.variables)
```

Data Preprocessing

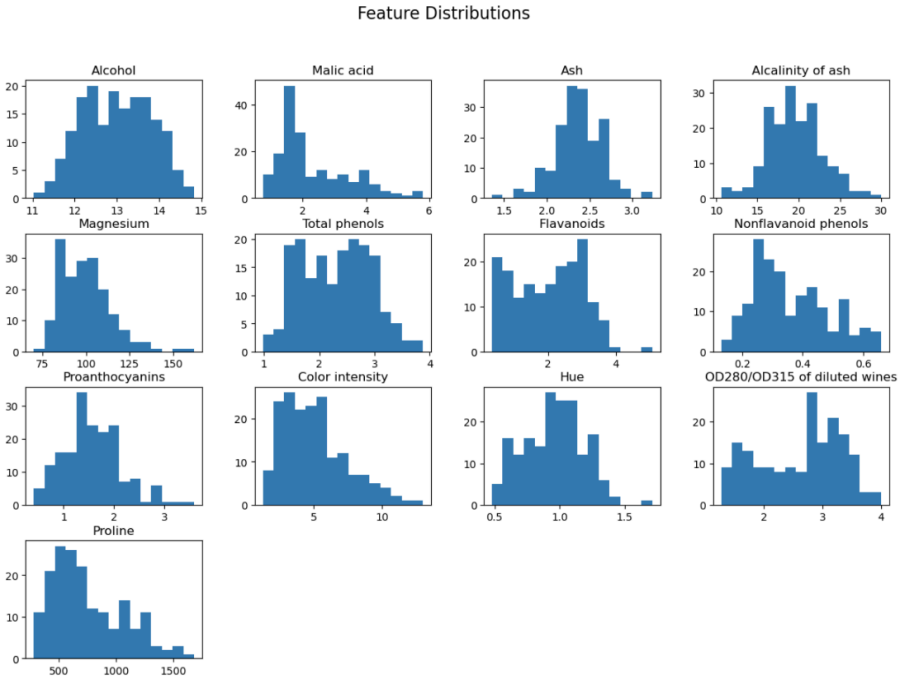
The initial step involves standardising data using Standardscaler() to ensure all features are on the same scale and contribute equally to the distance calculations. Without standardisation, attributes with larger numerical ranges would dominate the clustering, leading to biased results.

Exploratory Data Analysis



Correlation matrix for wine dataset

There are a variety of different correlation scores, where some variables have scores closer to 0 and some variables have scores closer to 1. Nevertheless, variables like Total phenols, Flavonoids, OD280 values and Proanthocyanidins seem to have relatively high correlation scores, indicating a stronger position correlation. On the other hand variables like Hue, Malic acid, Alkalinity of ash, and Nonflavonoid phenols have lower correlation scores, indicating a stronger negative correlation.



Feature distribution histograms for wine dataset

To understand each feature's behaviour, their distributions were explored. These distribution histograms helped reveal how data is spread and highlight statistical properties like mean, median and variance. These graphs also give insight into skewed distribution - suggesting potential outliers that could affect modelling - like malic acid, and colour intensity. Distributions that follow a bell-like curve indicate normal distribution, like with alcohol and alkalinity of ash. Moreover, these feature distributions help highlight the importance of scaling and normalising data, especially since we have various features that exhibit a varying range of values.

3.2 Method 1: K-Means Clustering

Background Information

K-Means clustering is an unsupervised machine learning algorithm used to partition data into distinct groups, or clusters, based on the similarity of features. The k-means clustering algorithm categorises data points into clusters by using a mathematical distance measure, usually Euclidean, from the cluster centre. The objective is to minimise the sum of distances between data points and their assigned clusters. Data points that are nearest to a centroid are grouped within the same category. The algorithms follow the following steps:

Step 1: Calculate the number of k (Clusters), using the elbow method
Step 2: Randomly select k data points as cluster centres.
Step 3: Using the Euclidean distance formula measure the distance between each data point and each cluster centre.

$$distance = \sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2}$$

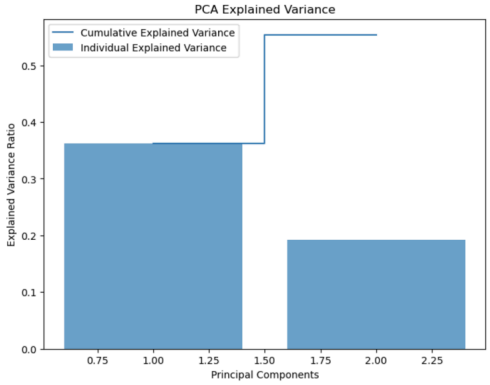
Step 4: Assign each data point to that cluster whose centre is nearest to that data point.
Step 5: Re-compute the centre of newly formed clusters. The centre of a cluster is computed by taking the mean of all the data points contained in that cluster.
Step 6: Keep repeating the procedure from Step 3 to Step 5 until any of the following stopping criteria is met-

- If data points fall in the same cluster
- Reached maximum iteration
- The newly formed cluster does not change in the centre points

Stage 1: Principal Component Analysis

A high correlation between features can result in redundancy in the data, meaning that different features can effectively communicate the same information. This not only makes the model unnecessarily more complex but also increases the risk of overfitting since variable variables community similar data. Therefore, Principal Component Analysis is recommended in such situations, as it reduces the dimensionality of the dataset while retaining as much information (variance) as possible. Here, the original features are reduced to smaller uncorrelated components, which comprise the most important variables.

The PCA explained variance graph below, helps to visually convey how much information is retained as dimensionality is reduced. The goal is to capture a significant amount of variance with fewer components, simplifying the data while minimising information loss. Since there is a steep drop from the first component to the next, it suggests most information has been captured by the first component.



(a) PCA Explained Variance graph

```
PCA Feature Loadings:
    Alcohol  Malic acid       Ash  Alcalinity of ash  Magnesium  \
0  0.144329   -0.245188 -0.002051          -0.239320   0.141992
1 -0.483652   -0.224931 -0.316069           0.010591  -0.299634

   Total phenols  Flavanoids  Nonflavanoid phenols  Proanthocyanins  \
0       0.394661    0.422934             -0.298533         0.313429
1      -0.065040    0.003360             -0.028779        -0.039302

   Color intensity       Hue  OD280/OD315 of diluted wines   Proline
0        -0.088617  0.296715                      0.376167  0.286752
1        -0.529996  0.279235                      0.164496 -0.364903

Explained Variance Ratio (how much variance each component explains):
[0.36198848 0.1920749 ]

Top 5 most influential features for each principal component:
{'PC1': Index(['Flavanoids', 'Total phenols', 'OD280/OD315 of diluted wines',
       'Proanthocyanins', 'Nonflavanoid phenols'],
      dtype='object'), 'PC2': Index(['Color intensity', 'Alcohol', 'Proline', 'Ash', 'Magnesium'], dtype='object')}
```

(b) Identifying most influential features in each component

The compositions of each component were further analysed to identify the 5 most influential features. This analysis gives insights to wine manufacturers on which chemical properties affect overall wine characteristics the most. The findings revealed:

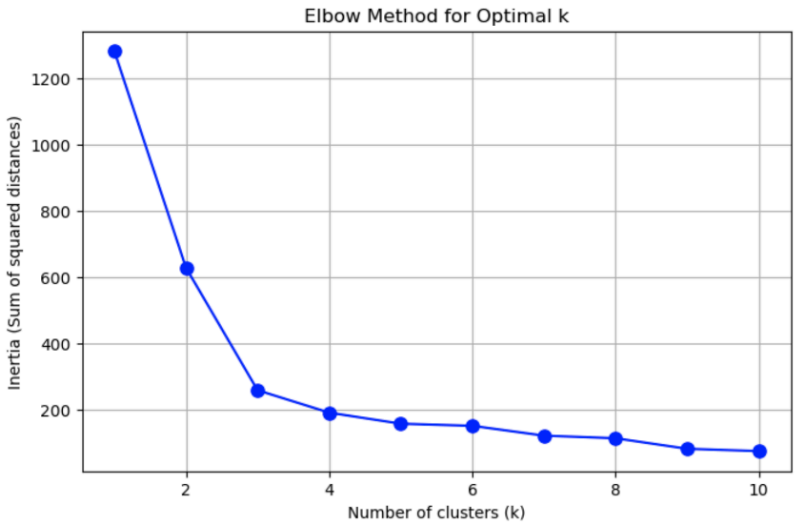
Component 1	• Flavonoids	attributes associated with **taste** and **bitterness** of the wine
	• Total phenols	
	• OD280/OD315 of diluted wines	measures the concentration of chemical compounds which contributes to **taste, colour** and **mouthfeel**
	• Proanthocyanidins	Compounds that contribute to **astringency, colour,** and **mouthfeel**
	• Non Flavonoid phenols	
	Therefore, it can be inferred that Component 1 is largely made of **taste attributes** (taste, astringency and mouthfeel)	
Component 2	• Colour intensity	Wines with high colour intensity are usually rich in pigments, often associated with red wines or **more robust wines**
	• Alcohol	High alcohol content usually refers to more **bold and stronger** flavours and makes for more **body**
	• Proline	Helps comment on grape health during the **winemaking process**
	• Ash	Refer to the **nutritional content,** mineral balance and **vineyard soil quality**. And gives manufacturers insight on which soils may yield better results
	• Magnesium	
	Therefore, it can be inferred Component 2 relates to wine robustness and **winemaking process characteristics** (body, nutritional content, vineyard soil quality)	

Stage 2: Perform K-Means Clustering on the components

Elbow Method - Compute Optimal Cluster Number

Before the components can be clustered, it is imperative to compute the optimum number of clusters for the dataset. To facilitate this, the Elbow method was used. It involves plotting the sum of squared distances between data points and their assigned cluster centres (called the inertia) as a function of the number of clusters. The “elbow” point indicates the number of clusters after which there is no significant inertria loss. So the elbow point helps indicate the

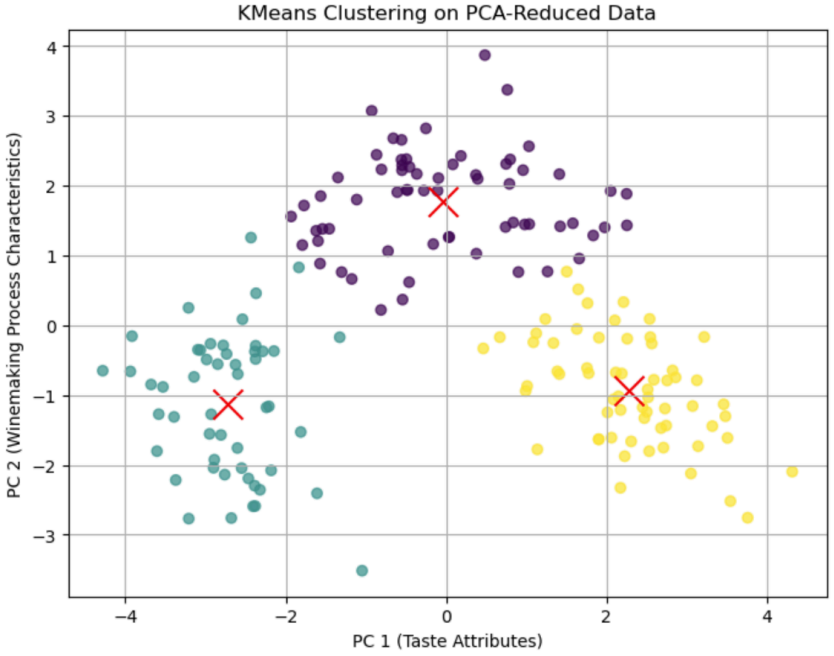
optimum number of clusters for K Means clustering. As can be seen below, the optimal number of clusters for the wine dataset is 3 clusters.



Graphing Elbow method to find optimal k value

K Mean Clustering Implementation

Therefore K means clustering with 3 clusters was implemented on the PCA components generated from the wine dataset.



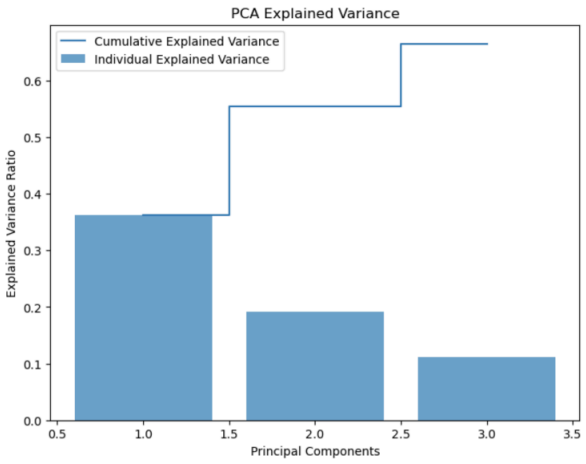
Final K means clustering graph with optimal k (k = 3)

Stage 3: Experimental Results & Evaluation

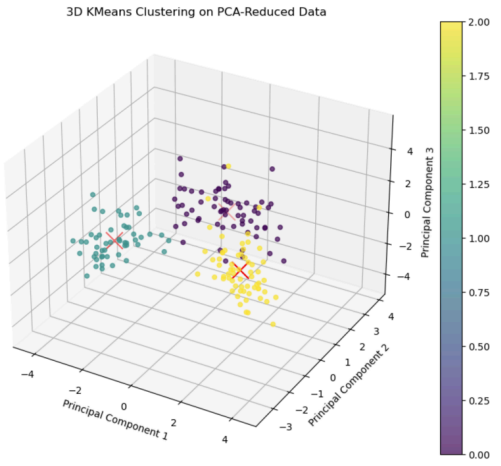
For comparison, the process was repeated but with 3 components generated from the PCA method, rather than 2 components as before. This gave the below PCA explained graph and a 3D graph to effectively map our clusters across the 3 dimensions of the 3 components.

```
Top 5 most influential features for each principal component:
{'PC1': Index(['Flavanoids', 'Total phenols', 'OD280/OD315 of diluted wines',
       'Proanthocyanins', 'Nonflavanoid phenols'],
      dtype='object'), 'PC2': Index(['Color intensity', 'Alcohol', 'Proline', 'Ash', 'Magnesium'], dtype='object'), 'PC3': Index(['Ash', 'Alcalinity of a
sh', 'Alcohol', 'Nonflavanoid phenols',
       'OD280/OD315 of diluted wines'],
      dtype='object')}
```

(a) Most influential features for component analysis for 3 components



(b) PCA explained variance for 3 components with optimal k for 3 components



(c) Final K means clustering graph

To evaluate the accuracy and quality of clusters obtained from 2-dimensional PCA versus 3-dimensional PCA, the following metrics were used:

- Silhouette Score: Values closer to 1 indicate well-defined clusters, while values closer to -1 suggest poor clustering.
- Davies-Bouldin Index: Lower values indicate better clustering quality, with clusters being well separated.
- Calinski-Harabasz Index: Higher values indicate better-defined clusters with distinct separation

And here are the computed scores:

Metrics	2D PCA	3D PCA
Silhouette Score	0.5602	0.4525
Davies-Bouldin Index	0.5977	0.8405
Calinski-Harabasz Index	343.9492	174.6207

As can be seen from the table above, the 2 dimensional PCA clustering yielded better results across all three parameters. This suggests that for the wine dataset, increasing dimensionality may reduce the amount of informative data being captured. Therefore, the 2 component clustering was selected for further analysis and insights.

Stage 4: Analysis and Insights

Many insights can be extracted from the PCA and K means clustering analysis.

Product Formulation: With knowledge of the most influential wine properties for both taste and winemaking, wine manufacturers can effectively adjust their product formulation, in a data-driven approach. For example, since 'Flavonoids' most influence component 1 (taste attribute), then manufacturers may consider increasing their flavonoid content for their wine production.

Targeted Improvements: Wine Manufacturers can note clusters that have high-quality winemaking processes in place in comparison to their taste attributes.

- For example, in the purple cluster - premium grapes, high-quality vineyard soil and high alcohol may have been used to produce wine, suggested by higher component 2 (winemaking characteristics).
- However, the purple cluster fairs less than the yellow cluster in terms of component 1 (taste attributes). This may suggest that brands in the yellow cluster are doing better off since they do not need to invest heavily in the winemaking process and are still able to produce wine of better taste.

Segmenting Wine Varieties: Manufacturers can use such clustering analysis to identify distinct clusters of wine with similar characteristics which may appeal to different customer segments.

- Purple cluster - has a higher component 1 - suggesting high-quality wines made from premium vineyard soil and with relatively higher alcohol contents. This should relate to full-bodied and more **robust red wines,** suggesting that the purple cluster wine labels may appeal to a red wine-drinking demographic.
- Green cluster Clusters with lower alcohol content (low component 2) and lighter colour (low component 1), may relate to a **crisp white wine-drinking** demographic.

3.3 Method 2: Agglomerative Hierarchical Clustering

This section describes the methodology and analyses the wine dataset using agglomerative hierarchical clustering. For comparison, we will use silhouette score and elbow method.

The silhouette score measures how well-separated clusters are, with values ranging from -1 to 1, where higher values indicate that clusters are well-defined and distinct from one another. It calculates the similarity of each point to its own cluster compared to others, offering a comprehensive assessment of clustering performance. Rousseeuw (1987) highlights the utility of the silhouette score for identifying whether clusters are meaningful and sufficiently compact. It is particularly valuable when comparing different clustering methods or evaluating different numbers of clusters.

The elbow method is used to determine the optimal number of clusters by plotting the sum of squared distances (or within-cluster variance) against the number of clusters. The point where the decrease in variance slows down significantly, forming an “elbow,” suggests the ideal number of clusters. Thorndike (1953) first introduced the elbow method as an intuitive way to balance the trade-off between reducing within-cluster variance and overfitting by increasing the number of clusters. This method helps identify a clustering solution that captures the underlying data structure without unnecessary complexity.

Background Information

Hierarchical clustering is a powerful unsupervised learning technique widely used in exploratory data analysis. It aims to build a hierarchy of clusters, allowing for a more intuitive understanding of the data's structure.

There are 2 different types of hierarchical clustering, mainly agglomerative hierarchical (bottom-up) which involves merging closet data points into a bigger cluster repetitively, and divisive (top-down) which involves repeatedly splitting clusters into smaller ones.

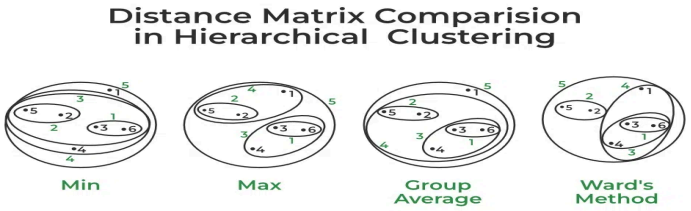
This section focuses on using agglomerative clustering, this is because agglomerative methods are often preferred due to their simplicity and effectiveness. They start with individual data points and progressively merge them into larger clusters based on proximity, making the process intuitive and easy to interpret (Hastie et al., 2009). Furthermore, agglomerative clustering requires fewer parameters and is generally computationally more efficient, particularly when applied to large datasets (Müllner, 2013).

The wine dataset, sourced from the UCI Machine Learning Repository, contains chemical properties of different wine cultivars from the same region in Italy. These features can be grouped using hierarchical clustering to discover patterns or hidden structures.

Stage 1: Perform Hierarchical Clustering on the dataset

After preprocessing, we applied agglomerative hierarchical clustering to the standardised dataset. We explored different linkage methods: Ward, Single, Complete, and Average linkage. Each method uses a different strategy to calculate the distance between clusters:

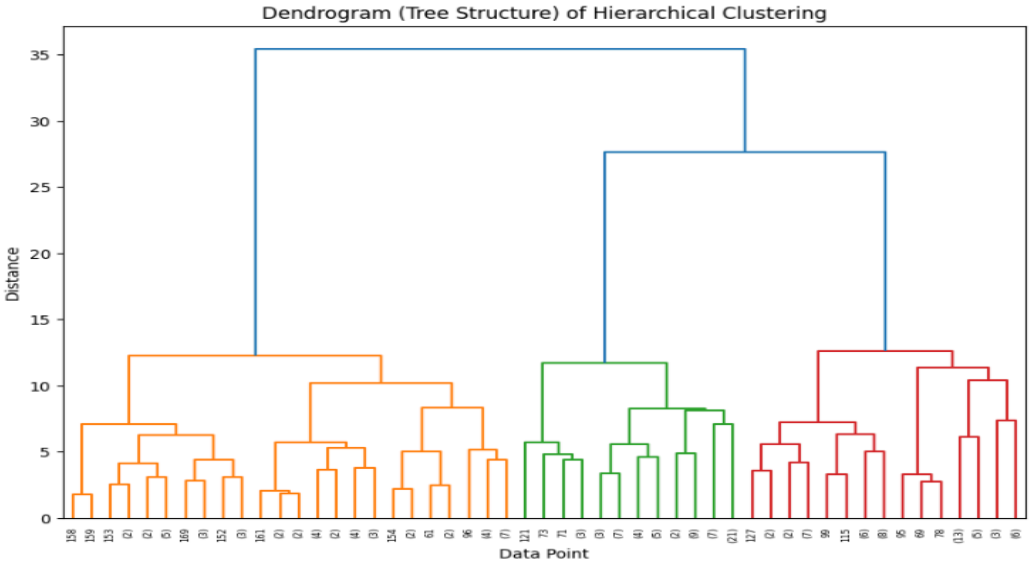
- Ward's Method / Centroid Linkage: Minimises the total variance within each cluster.
- Single Linkage: Merges clusters based on the minimum distance between data points in different clusters.
- Complete Linkage: Uses the maximum distance between data points in different clusters.
- Average Linkage: Averages the distances between points in different clusters.



Different linkage methods

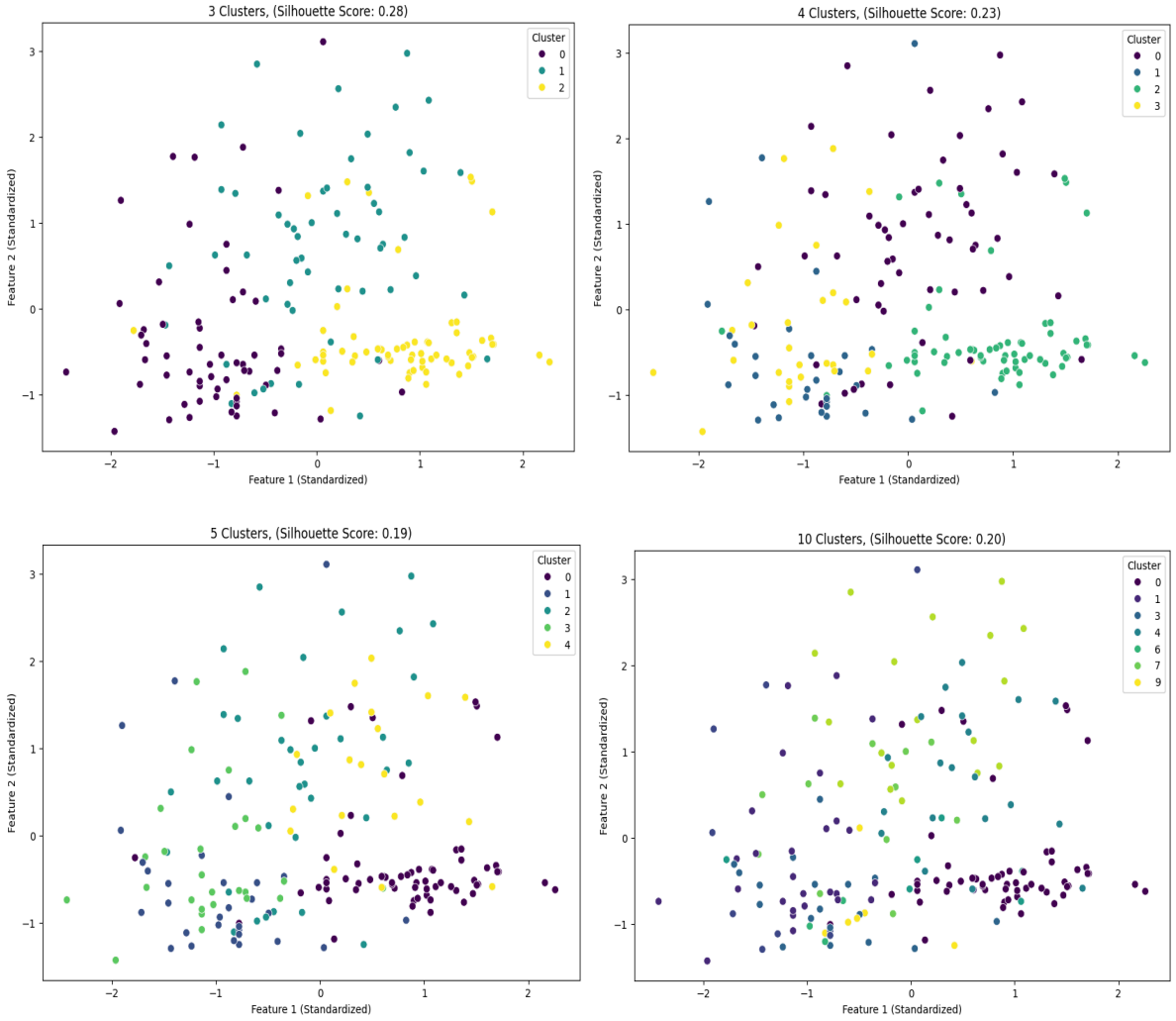
In this project, centroid linkage (Ward's method) is preferred in agglomerative hierarchical clustering because it minimises within-cluster variance, leading to compact and well-separated clusters. According to Murtagh and Contreras (2012), this approach prevents the formation of elongated or poorly defined clusters, which can occur with methods like single linkage. Ward's method also consistently produces higher silhouette scores, indicating superior cluster quality compared to other techniques, as highlighted by Jain (2010). Its ability to create clear, interpretable clusters makes it a strong choice for hierarchical clustering tasks.

Next, we generated the dendrogram using the Scipy library to visualise the agglomerative hierarchical structure to observe the cluster formation at different process steps.



Dendrogram of the hierarchical clustering

We used the existing AgglomerativeClustering function from the Sklearn library for the implementation of hierarchical clustering and fit the model to visualise the different numbers of clustering groups.



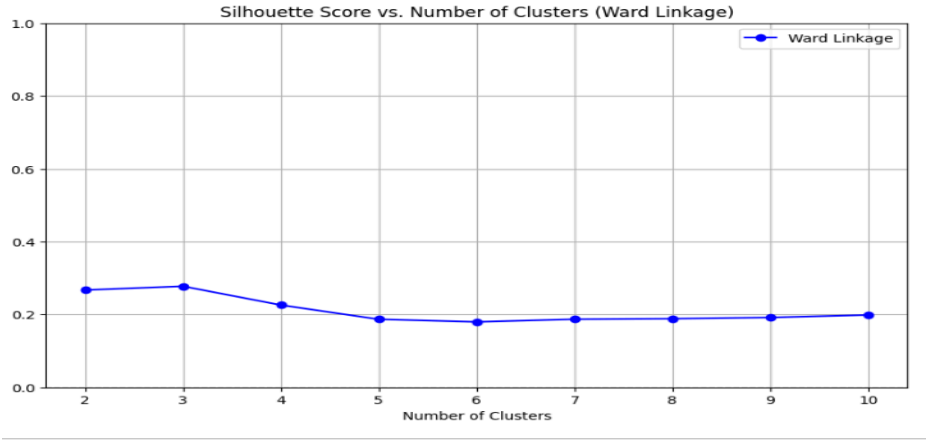
Different numbers of clustering visualisation

The visualisation from the Figure above helps us get a good visualisation of how the clustering group form will end up after fitting it to the AgglomerativeClustering model.

Stage 2: Experimental Results and Analysis

This section of the report will discuss the result of agglomerative hierarchical clustering and compare the result with silhouette scores and elbow methods to discuss the suitable numbers of clusters for the wine dataset.

The figure below shows different silhouette scores of different assigned numbers of clusters using centroid linkage (Ward's method).

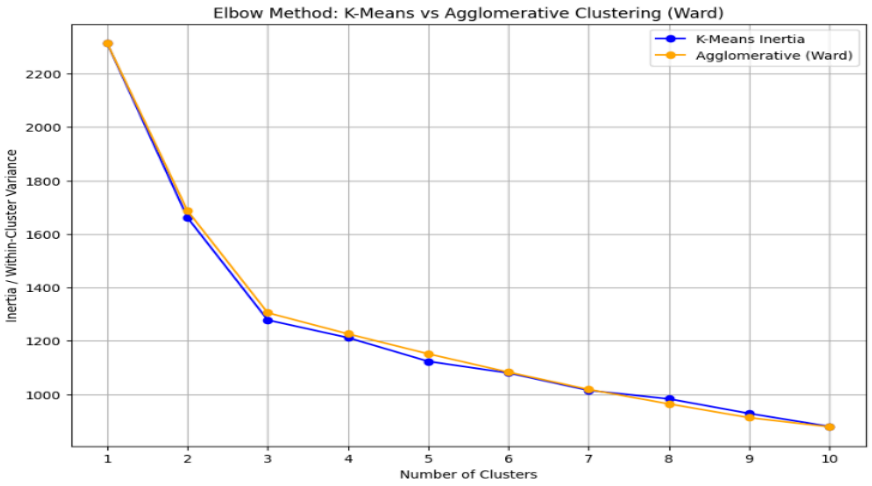


Different numbers of clustering and its silhouette scores

From the graph above we observed that the number of clusters with the highest score is 3 with a silhouette score of 0.28, this indicates that 3 clusters are the best grouping in terms of cohesion and separation. Meanwhile, The scores for 5, 7, and 10 clusters are all around 0.19-0.20, which suggests that increasing the number of clusters beyond 3 does not improve the clustering quality much. These numbers might indicate some level of over-partitioning, where the data is being split into clusters that are not as well-separated or meaningful.

Next, we also compare the result using elbow method analysis to determine the optimal number of clusters by calculating:

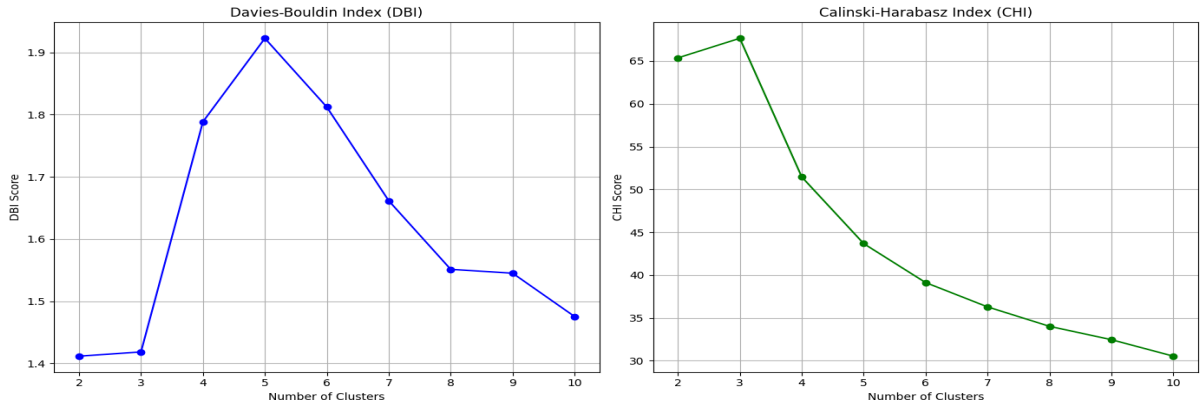
- K-Means Inertia: Inertia is the sum of squared distances between data points and their respective centroids.
- Agglomerative Clustering (Ward/Centroid method): The average distances for different numbers of clusters.



Elbow point analysis for different numbers of clustering

From the graph above we can observe that 3 clustering and adding more clusters do not significantly improve its average squared centroid distances.

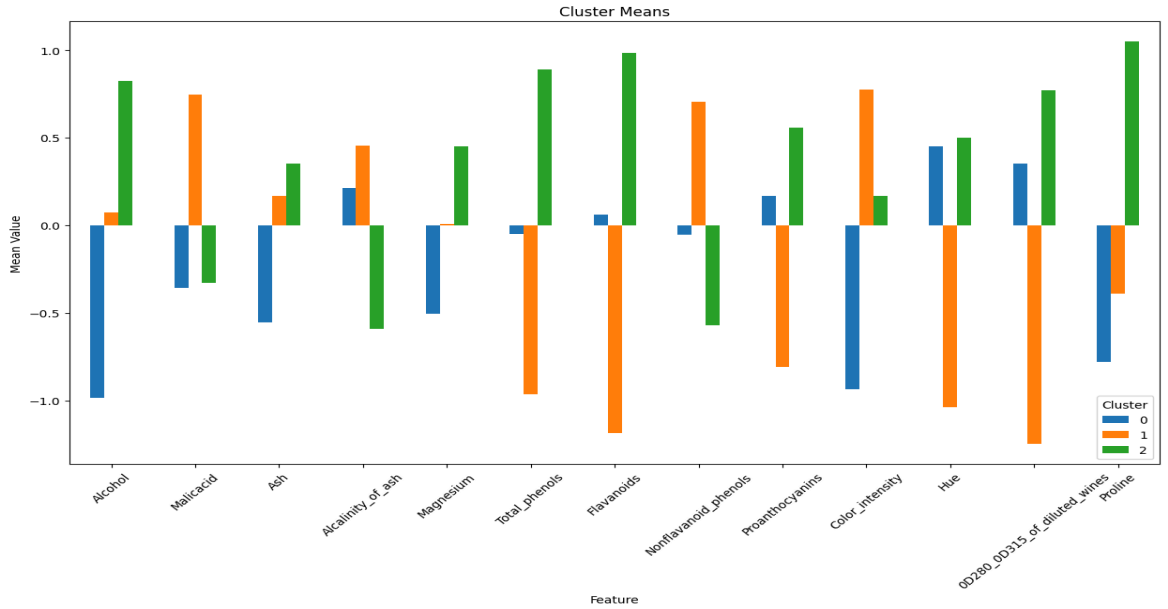
We also further support this analysis with the use of the Davies-Bouldin Index (DBI) and the Calinski-Harabasz Index (CHI). A low DBI and high CHI scores indicate that the clusters are well-separated and compact, indicating better clustering quality. Meanwhile, high DBI and low CHI scores indicate that the clusters are overlapping or poorly separated.



DBI and CHI Scores

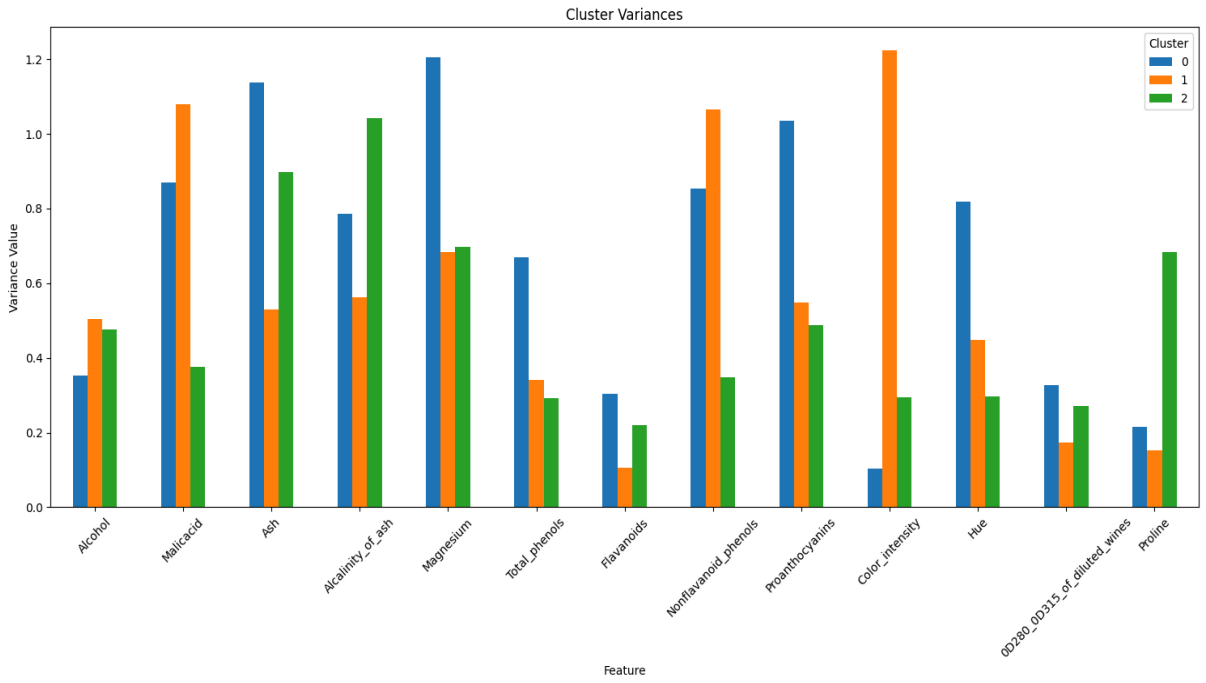
The figure above shows that 3 is the optimal number of clusters to be assigned. With a Davies-Bouldin Index score of 1.419 and a Calinski-Harabasz Index score of 67.647. Solidify 3 as the best optimal number indicating the clusters are well-separated and compact, indicating better clustering quality.

We then assigned the number of clusters 3 to calculate the cluster means and variances across all the features in the wine datasets to describe the central tendency for the data points within the clusters and identify feature important. By calculating the means and variances, you can gain insights into what defines each cluster and how well-separated or distinct the clusters are from each other.



Cluster means across features

The figures above show that Cluster 2 for Feature 'Proline' has the highest mean value and Cluster 1 for Feature '0D280_0D315_of_diluted_wines' has the lowest mean value, indicating that Feature 'Proline' is important because it strongly influences Cluster 2, which has the highest mean value. It helps distinguish this cluster from others. Although Feature '0D280_0D315_of_diluted_wines' has the lowest mean in Cluster 1, understanding its characteristics is still important for defining this cluster's profile.



Cluster variances across features

The figure above shows that Cluster 0 for Feature 'Color_intensity' has the lowest variance and Cluster 1 for Feature 'Color_intensity' has the highest variance, indicating that Feature 10 is also important, but its impact varies across clusters:

- In **Cluster 0**, it is a stable feature with low variance, meaning it is a key characteristic of this homogenous cluster.
- In **Cluster 1**, it has the highest variance, indicating that it represents a more diverse aspect of the members of this cluster.

Stage 3: Evaluation of Hierarchical Clustering implementation

In Conclusion for the agglomerative hierarchical clustering implementation on the wine dataset, we can conclude that the number of clustering to achieve the best grouping is 3 based on the conclusion and analysis using silhouette scores and elbow method analysis. Based on the silhouette scores and elbow method, we concluded that the number of optimal clusters is 3.

Metrics	Scores
Silhouette Score	**0.28**
Davies-Bouldin Index	1.419
Calinski-Harabasz Index	67.647

We also analyze the feature's importance by calculating the cluster means and variances and conclude that Feature 'Proline' and Feature 'Color_intensity' should be prioritized for further analysis or when making decisions based on the clustered wine dataset. These insights highlight the need for targeted analyses of specific features for better understanding and segmentation in wine studies, which can inform marketing strategies and consumer preferences in the wine industry.

4. Dataset 2: Customer Mall Data

In this project, we use Mall customer segmentation (Choudhary, 2019) and wine datasets (Aeberhard, Coomans, & de Vel, 1991).

The dataset used in this analysis is the Mall_Customers.csv file, which contains demographic and spending behaviour information for 200 customers. The dataset consists of five columns: CustomerID, Gender, Age, Annual Income (k$), and Spending Score (1-100).

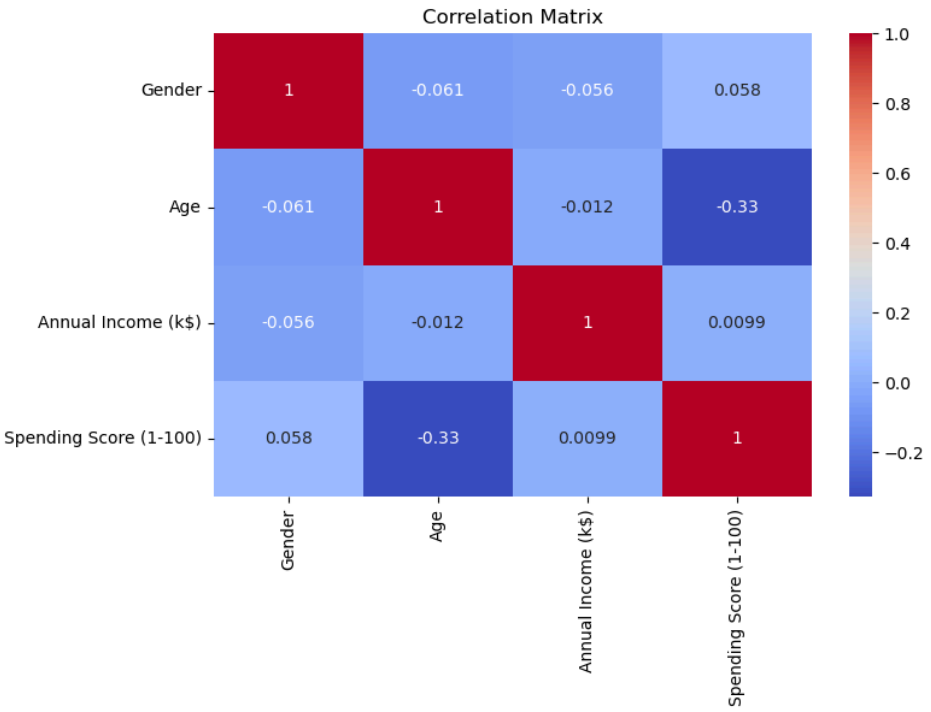
4.1 Dataset Cleaning and Preprocessing

Dataset summary

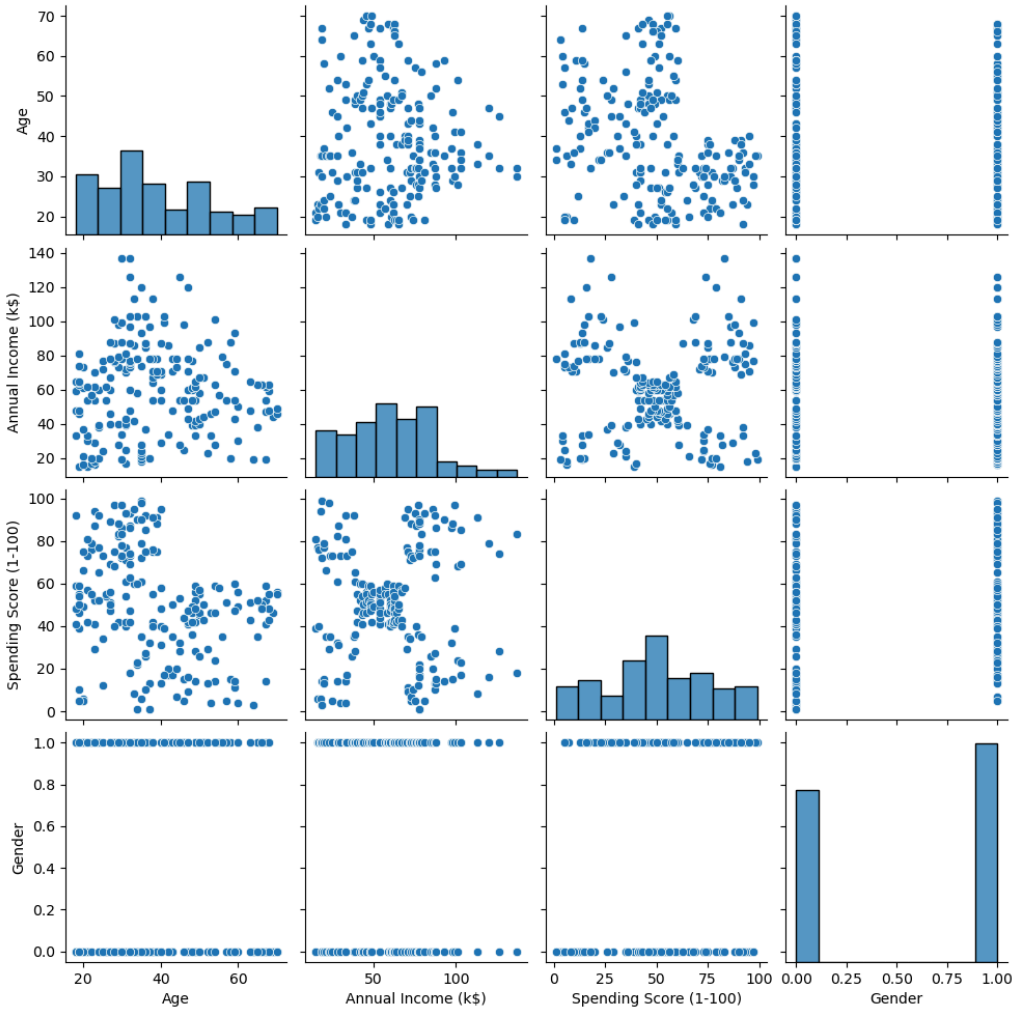
The dataset contains **200 rows and 5 columns**, and there are **no missing values** across any of the features, as shown in the output below. Hence, no further data cleaning is required

```
Shape of the DataFrame: (200, 5)
Missing values: CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

Exploratory Data Analysis



All the correlation values are close to zero, which indicates weak or **no linear correlation** between most features. The most prominent correlated features are between Gender and Spending Score at 0.058 and between Age and Spending Score (-0.33) is moderately negative, suggesting that older customers may tend to have lower spending scores.



Annual Income vs. Spending Score:

There appear to be distinct groupings:

- One group has high income and low spending.
- Another group has low income and high spending.
- There are also groups in the middle with moderate spending.
- This indicates potential clusters that K-means could capture effectively

Based on the EDA, we created another dataset using only the annual income and spending score as these seem to be the most related features. This, along with the original dataset, will be used to determine if there is any difference in terms of the quality of the clusters generated.

Data Pre-Processing: Handling non-numerical data

In this dataset, the column “Gender” is a categorical variable, with two possible values: "Male" and "Female." To allow for clustering, this was converted to 1 for "Male" and 0 for "Female".

Data Pre-Processing: Standardisation of values

For some parts, we standardised the dataset because clustering algorithms rely on distance metrics like Euclidean distance. The features in the dataset have very different ranges, as shown in our EDA and without standardisation, the features with larger ranges would disproportionately affect the clustering results. Using **StandardScaler**, the values are scaled to have a mean of **0** and a SD of **1**, ensuring that all features contribute equally to the clustering process.

4.2 Method 1: K-Means Clustering

Background Information

The KMeans algorithm clusters data by trying to separate samples in n groups of equal variance, minimising a criterion known as the inertia or within-cluster sum-of-squares (see below). This algorithm requires the number of clusters to be specified. It scales well to large numbers of samples and has been used across a large range of application areas in many different fields.

The k-means algorithm divides a set of N samples X into K disjoint clusters C, each described by the mean μ_j of the samples in the cluster. The means are commonly called the cluster “centroids”; note that they are not, in general, points from X, although they live in the same space.

- X: represents the dataset you want to cluster
- N: the number of samples/data points within X
- K: the number of clusters you want to divide X into. For each cluster j (from 1 to K), there is a centroid/mean point denoted by μ_j
- Cluster centroid (μ_j): all samples are assigned to the nearest centroid based on their distance (usually Euclidean distance)

The K-means algorithm aims to choose centroids that minimise the inertia, or within-cluster sum-of-squares criterion:

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$$

Inertia measures how internally coherent the clusters are / how tight the points in each cluster are around their respective centroids. The formula computes the total sum of squared distances from each point to its nearest centroid. K-means tries to minimise this value, which makes the clusters as compact and distinct as possible.

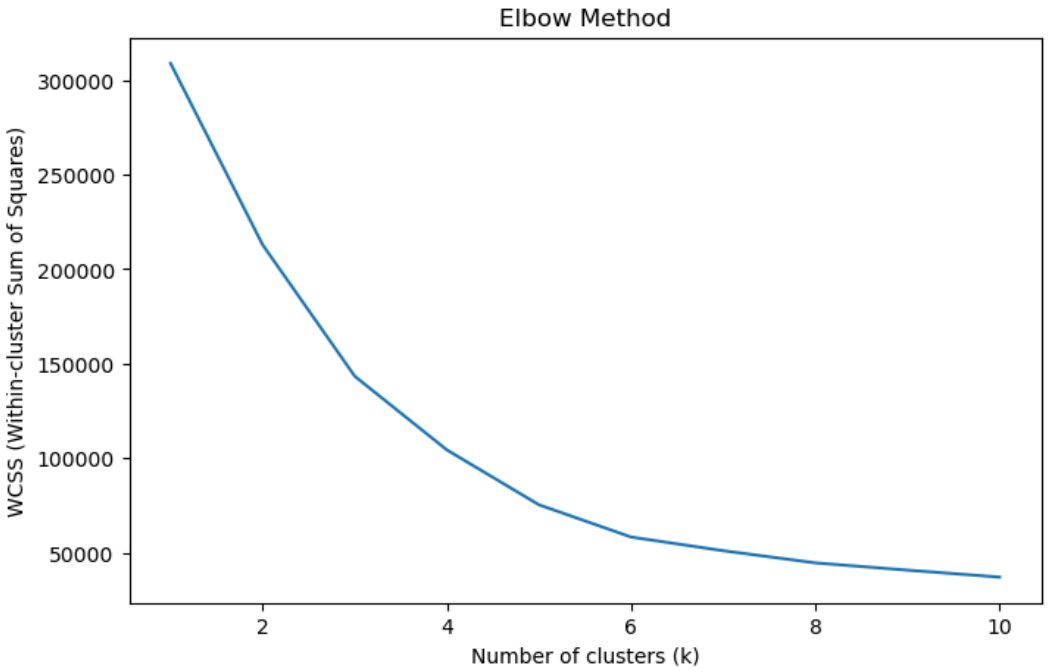
Stage 2: Perform K-Means Clustering on the dataset

We conducted K-means experiments in the following way:

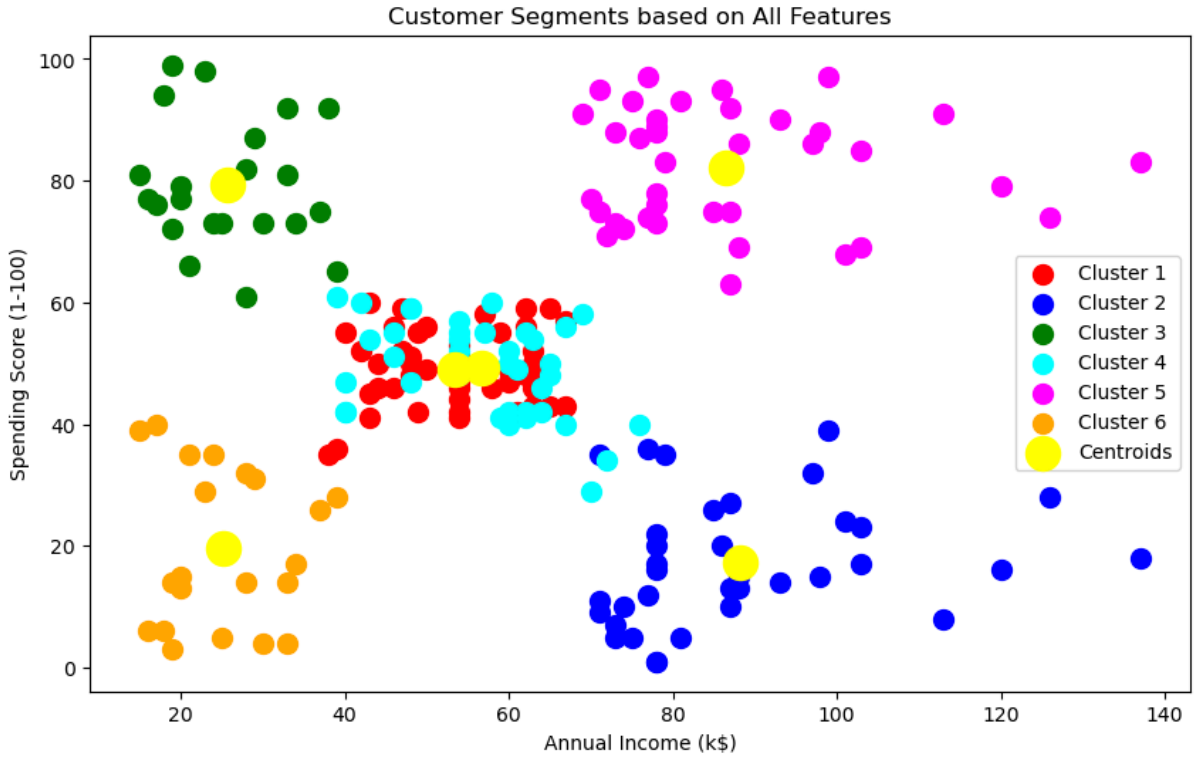
1. Find the Optimal Number of Clusters (k): We've use the Elbow Method to find the ideal number of clusters.
2. Fit K-means Model: Once we determine the optimal k, we'll apply K-means to the dataset.
3. Visualise the Clusters
4. Apply feature reduction

A) Elbow Method to Determine Optimal k

The **Elbow Method** helps determine the optimal number of clusters by plotting the sum of squared distances from each point to its assigned cluster centre (inertia) for different values of k. The n_cluster hyperparameter is set to this to begin with.



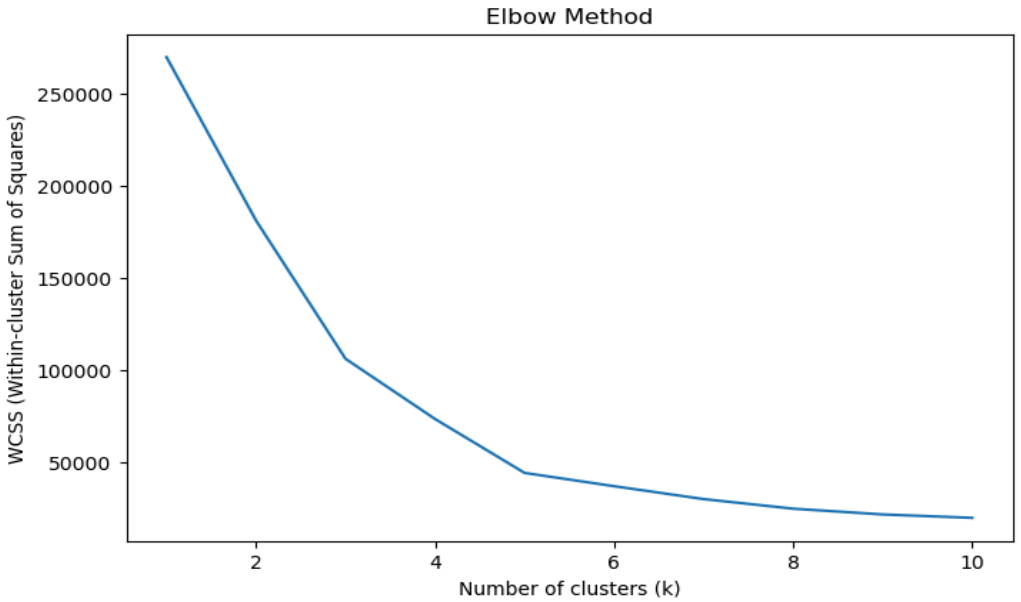
For the all_df, the "elbow" point, where the Within-Cluster Sum of Squares (WCSS) starts flattening significantly, is around **6 clusters**. This indicates that using k=6 is a good choice for this dataset.



To evaluate the quality of the clusters, the Silhouette Score can be used to measure how close each point in one cluster is to the points in neighbouring clusters. Values range from -1 to 1, where 1 is the best. The silhouette score of 0.445 shows that the clusters are less

distinct when all four features are used. Davies-Bouldin Score is 0.82 and Calinski-Harabasz Score is 150.76. Gender and age are plausibly not key in adding separability in the clusters, given their low correlation score. Clusters 1 and 4 are largely overlapping.

I created another dataset called money_df that drops the gender and age columns from the all_df. Let's use the money_df and observe if there is a difference.



The "elbow" point is around **5 clusters**.

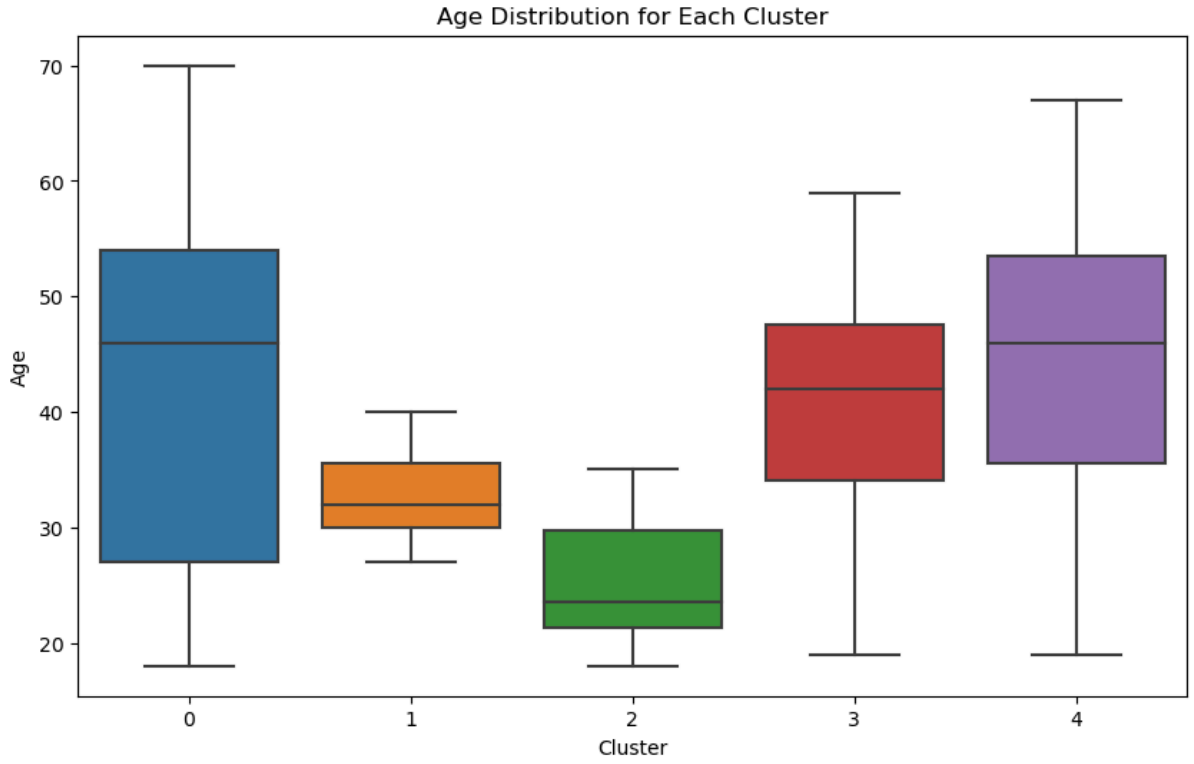
The improved Silhouette Score of 0.554, Davies-Bouldin Score of 0.571 and Calinski-Harabasz Score of 247.82 confirm our hypothesis that age and gender are not so helpful in adding separability.

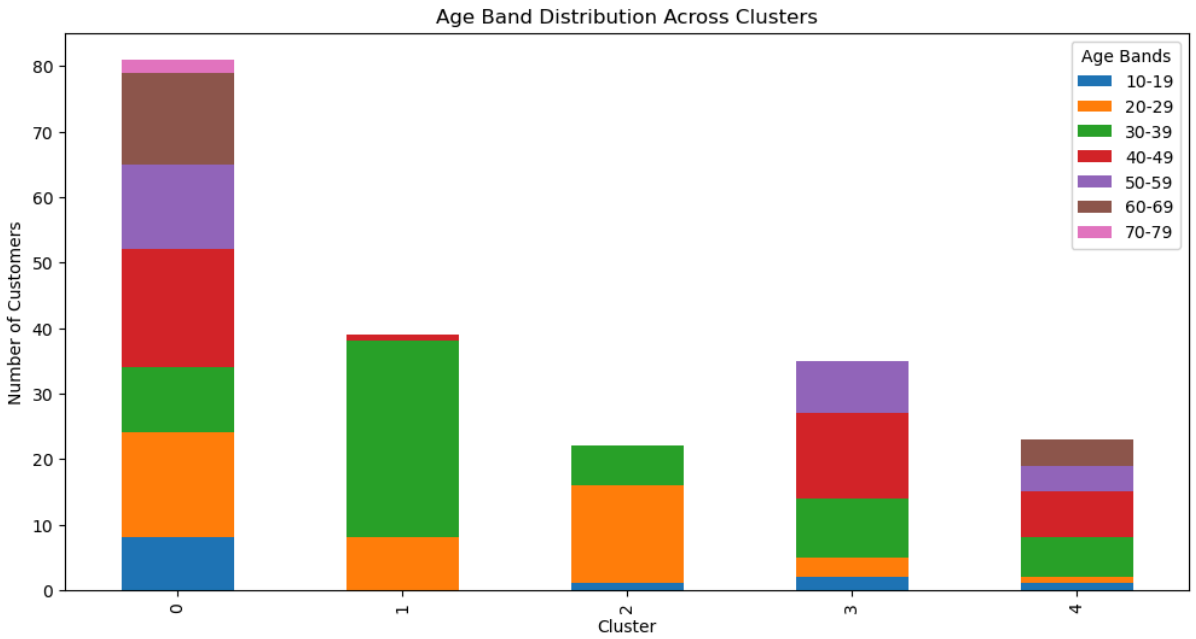
Stage 3: Analysis and Insights

Money_df

- Cluster 1 (red) - represents average spenders who have a balanced spending pattern
- Cluster 2 (dark blue) - wealthy and high-spending customers. Top-tier customers are more likely to spend on premium products or services.
- Cluster 3 (green) - lower income, have a high spending score. This indicates that these may be the category of individuals belonging to the phenomenon of “Daddy’s Money”.
- Cluster 4 (light blue) - earn a lot but spend cautiously. They might not be swayed easily by offers or might be saving for future investments.
- Cluster 5 (pink) - low-income, low-spending customers. They are less likely to spend on non-essential items

In order to look into the clusters and gain more insights, we will observe the demographics of each cluster to see the age/gender distribution.




Age Band Distribution Across Clusters
Number of Customers
Cluster
Age Bands
10-19
20-29
30-39
40-49
50-59
60-69
70-79
0
1
2
3
4
0
10
20
30
40
50
60
70
80

4.3 Method 2: Hierarchical Clustering

Background Information

Hierarchical clustering is a powerful unsupervised learning technique widely used in exploratory data analysis. It aims to build a hierarchy of clusters, allowing for a more intuitive understanding of the data's structure.

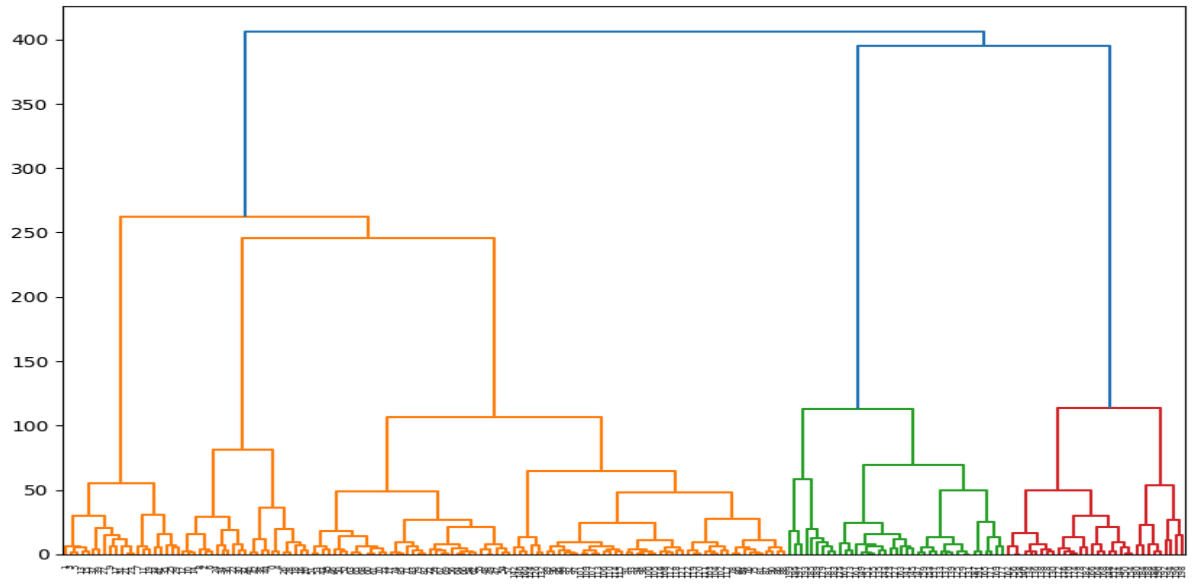
There are 2 different types of hierarchical clustering, mainly agglomerative hierarchical (bottom-up) which involves merging closet data points into a bigger cluster repetitively, and divisive (top-down) which involves repeatedly splitting clusters into smaller ones.

This section focuses on using agglomerative clustering, this is because agglomerative methods are often preferred due to their simplicity and effectiveness. They start with individual data points and progressively merge them into larger clusters based on proximity, making the process intuitive and easy to interpret (Hastie et al., 2009). Furthermore, agglomerative clustering requires fewer parameters and is generally computationally more efficient, particularly when applied to large datasets (Müllner, 2013).

The wine dataset, sourced from the UCI Machine Learning Repository, contains chemical properties of different wine cultivars from the same region in Italy. These features can be grouped using hierarchical clustering to discover patterns or hidden structures.

Stage 1: Perform Hierarchical Clustering on the dataset

We used the Ward linkage method for hierarchical clustering, which minimises the variance within each cluster when merging. The hierarchical clustering process was visualised using a dendrogram (shown below), which shows how clusters are merged at different distance thresholds.

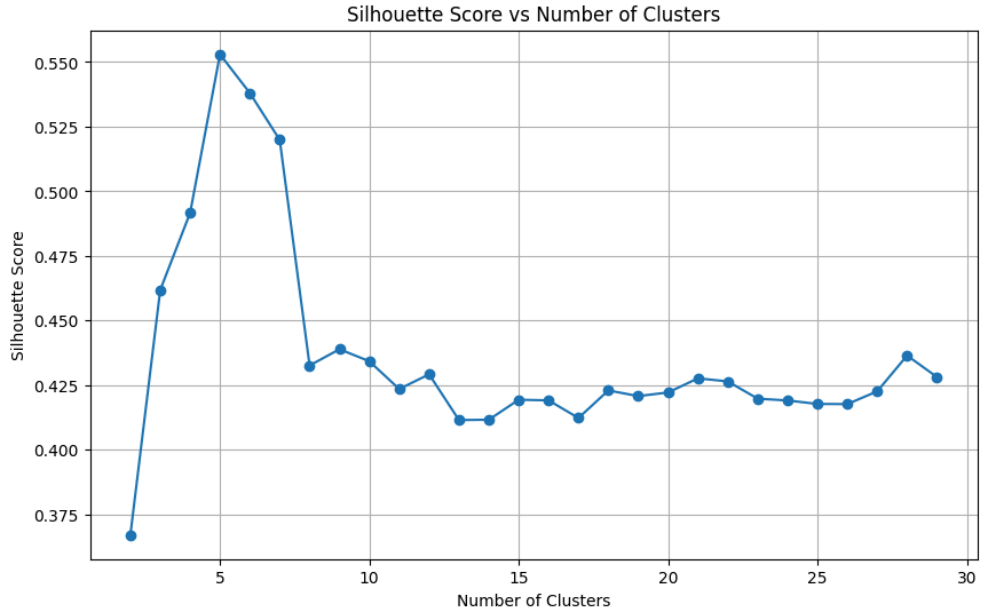


To determine the optimal number of clusters, we used 3 different metrics to evaluate: the **Silhouette Score**, **Davies-Bloudin score and Calinski-Harabasz Index** for different cluster numbers ranging from 2 to 30.

A. Silhouette Score

The **Silhouette Score** measures **how well-separated and cohesive the clusters are**. It ranges from -1 to 1, with **higher values indicating better-defined clusters**. A higher silhouette score means that the data points are close to other points within the same cluster and far from points in other clusters.

As shown in the diagram below, **n=5** gives the highest silhouette score, which indicates data points are closest to other points within the same clusters, while also being furthest from points in other clusters when we have 5 clusters.

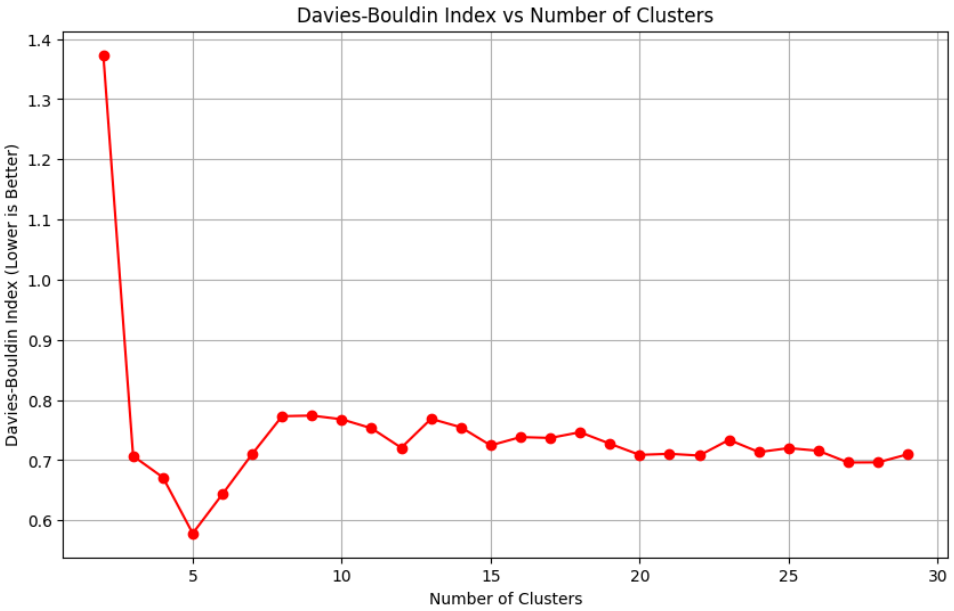



```
Maximum Silhouette Score =  0.5529945955148897
Number of clusters =  5
```

B. Davies-Bloudin score

The **DBI** evaluates the average **similarity ratio between each cluster and the cluster most similar to it**. Unlike the silhouette score, **lower DBI values are better** since they indicate well-separated clusters with less dispersion.
Higher DB index values correspond to poorer clustering solutions. This is because a higher DBI value indicates that the clusters are not well-separated and/or that the clusters are not compact. However, a lower DB index value is desirable. It indicates that the clusters are well-separated and compact, which is often a good indication of a successful clustering solution.

As shown in the diagram below, **n=5** gives lowest DBI score, which indicates the highest similarity ratio between each cluster and the cluster most similar to it and increasing number of clusters beyond n=5 does not yield any improvement to cluster seperation



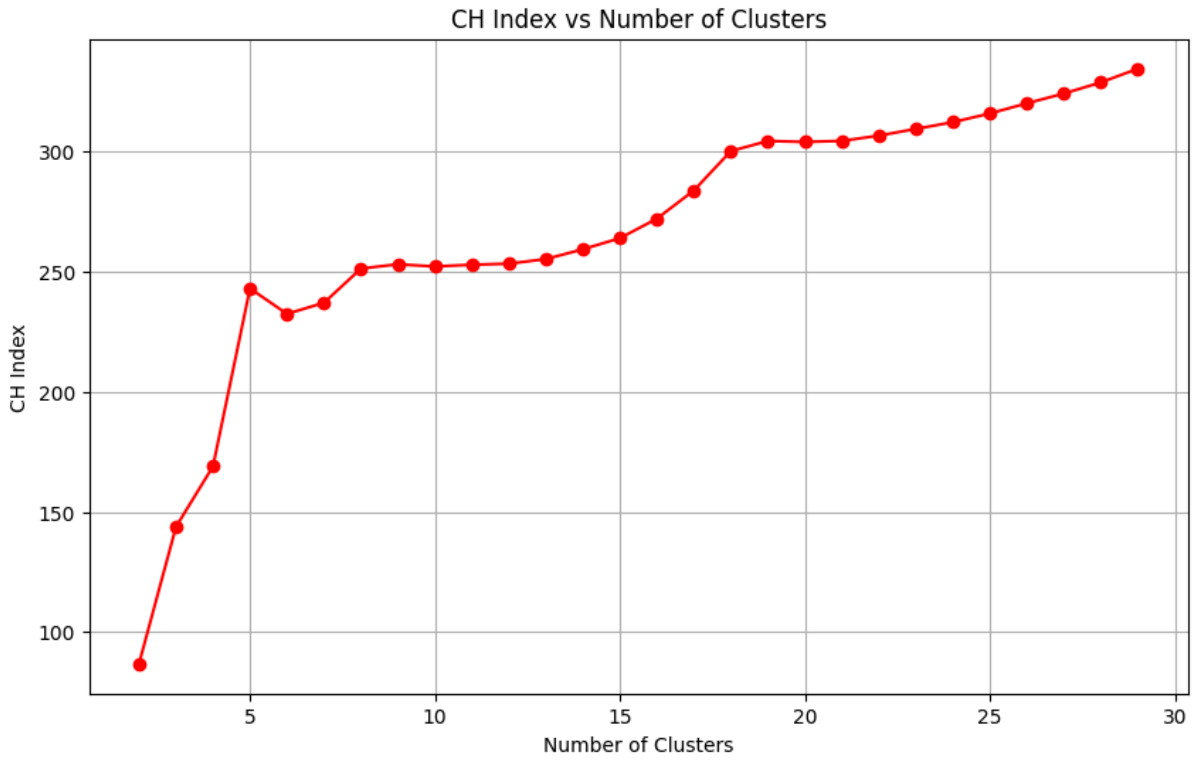

```
Min Davies-Bouldin Score =  0.5782207452369451
Number of clusters =  5
```

C. Calinski-Harabasz Index

The Calinski-Harabasz Index (CHI) (also known as the Variance Ratio Criterion) measures the ratio between the variance of clusters and the variance within clusters, giving an indication of how well-separated and compact the clusters are. A higher score on the CHI suggests that the clusters are dense (compact) and well-separated, indicating better clustering.

How It Works:

- Between-cluster dispersion (separation): This measures how far the clusters' centroids are from each other. The larger the distance between the centroids of the clusters, the better the clusters are separated.
- Within-cluster dispersion (compactness): This measures how tightly the points in a cluster are packed around the centroid. Lower intra-cluster dispersion means points in the same cluster are close to each other, indicating good compactness.



Interestingly, the CH score kept increasing with the number of clusters, implying that the optimal number of clusters may be better at a much higher number of clusters than n=5 as agreed by the previous 2 evaluation metrics. There is also a peak at n=5, for clusters lower than n=8.

However, this behaviour can be attributed to how the CH method works. The CH index evaluates cluster compactness and separation, favouring solutions with more clusters as they tend to reduce within-cluster variance. However, this may lead to overfitting, where increasing the number of clusters captures noise instead of meaningful patterns

On the other hand, both the Davies-Bouldin and Silhouette scores indicated that 5 clusters provide an optimal balance between cohesion and separation. These metrics are more sensitive to the trade-off between compactness and the separation of clusters, which aligns with the goal of creating meaningful, interpretable groupings.

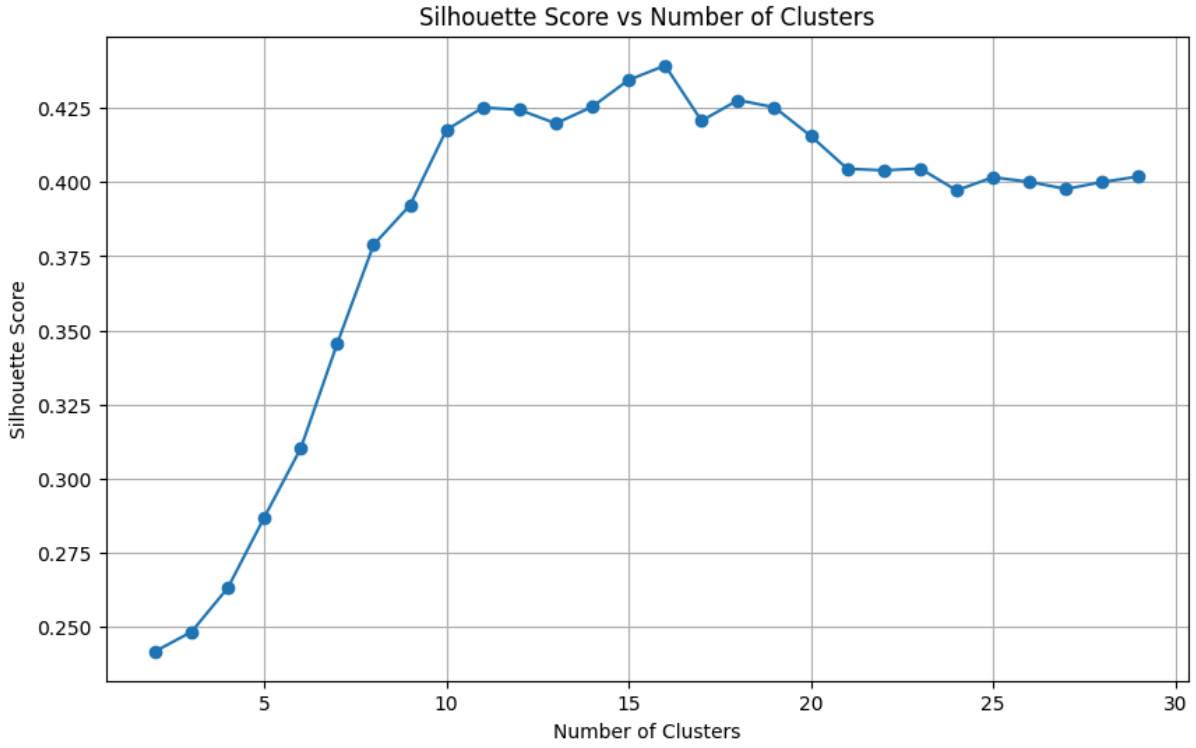
Conclusion based on the 3 evaluation metrics

Given the risk of overfitting with higher cluster numbers in the CH method and the consistency of results from both the Davies-Bouldin and Silhouette scores, and the fact that CH method peaked at n=5 for lower number of clusters, we can safely conclude that

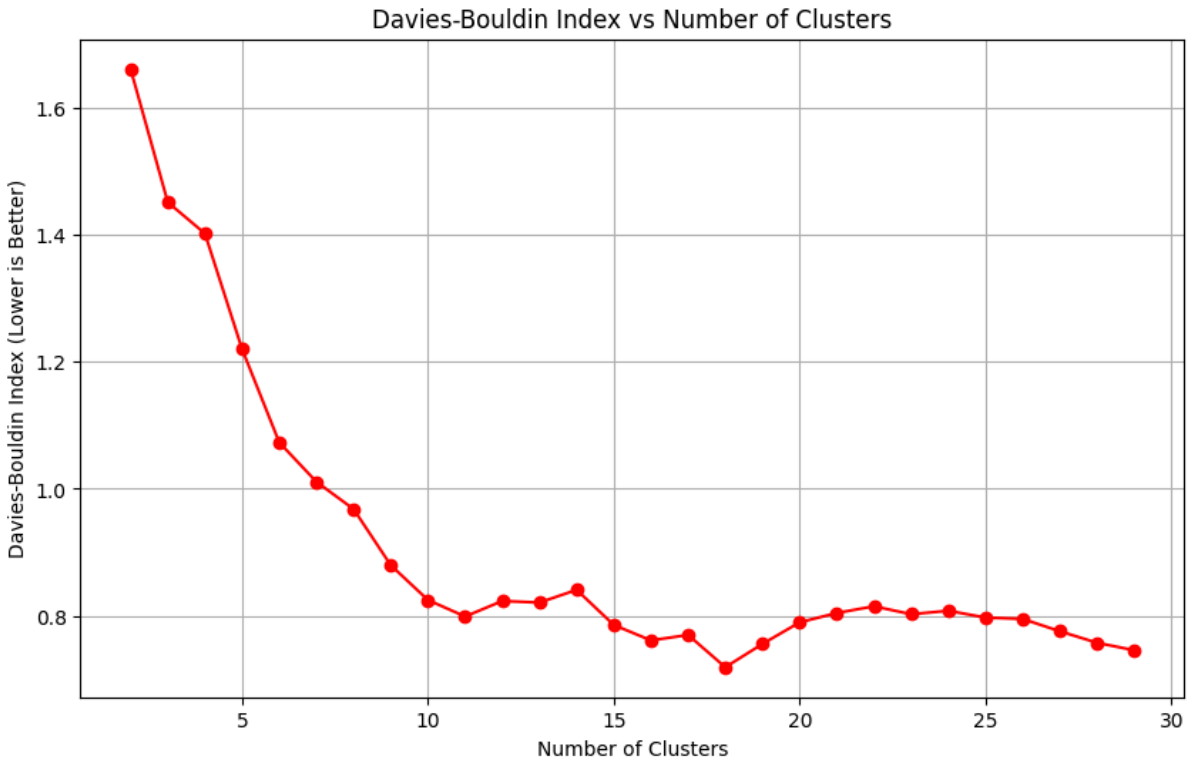
Agglomerative Clustering works well with this dataset with around **5 clusters** and hence, we shall use that for further analysis

Clustering with standardisation and all features

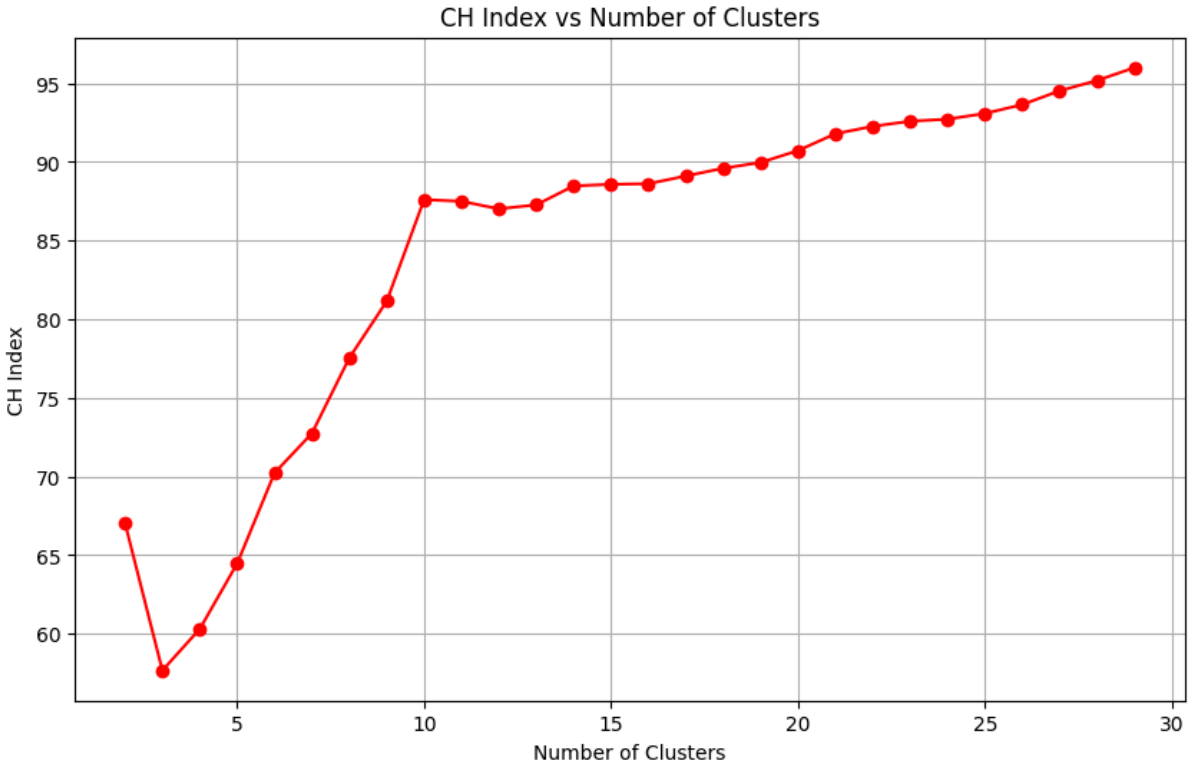
We also attempted to perform clustering with all features and standardisation. We will show a quick summary of the evaluation metrics as shown above




```
Maximum Silhouette Score =  0.43932320264453567
Number of clusters =  16
```




```
Min Davies-Bouldin Score =  0.7188492873154483
Number of clusters =  18
```



We shall stick into n=17, by taking average of n=16 and n=18 as suggested by the first 2 evaluation metrics as the CH index is not very helpful

Conclusion based on the 3 evaluation metrics

From the clustering above with n=17,there is an overwhelming evidence of overfitting. In addition, n=17 is relatively high for a dataset with only 200 datasets, suggesting the model may have over segmented the data. This happens when the number of clusters exceeds the intrinsic complexity of the data, which is very likely what happened with n=17 clusters.

One main reason could be because we included features like Gender and Age into the clustering, which are non-essential features compared to Annual Income and Spending Score Hence, we decided to not go ahead with the standardising, and to not use all the features in the clustering. For further analysis, we will focus on core features like Annual income and Spending Score.

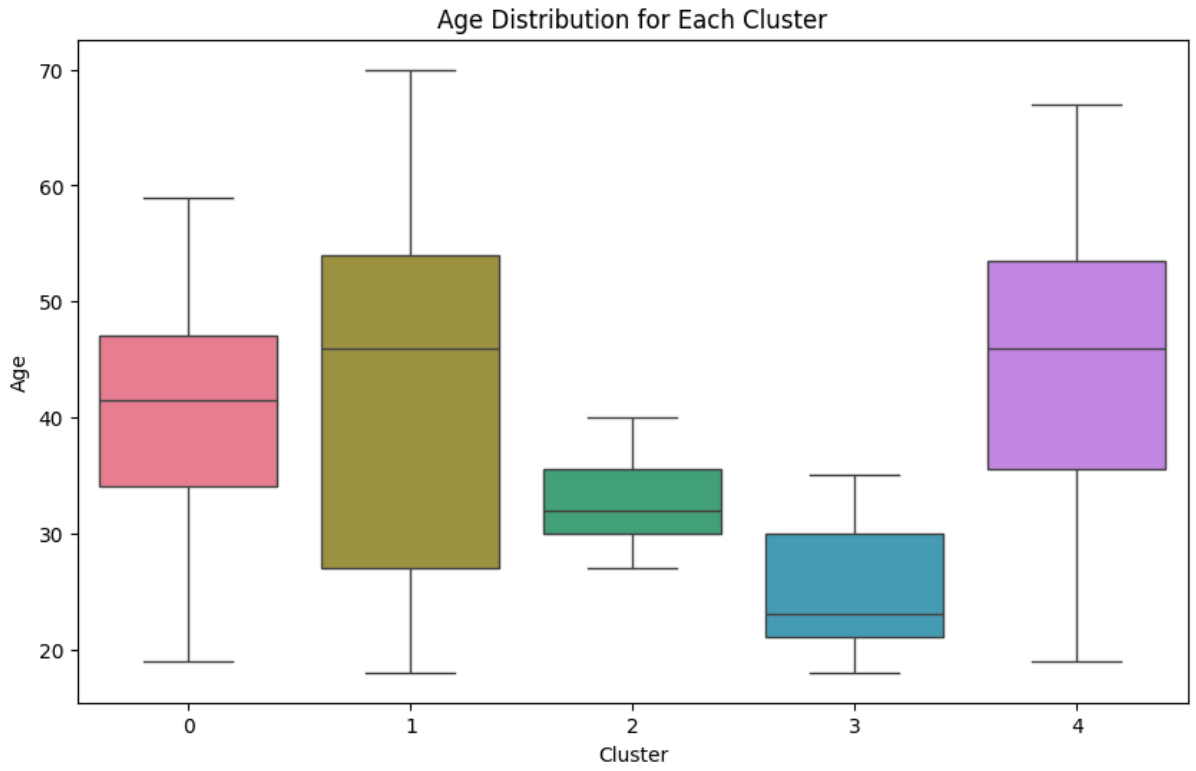
Stage 3: Experimental Results and Analysis

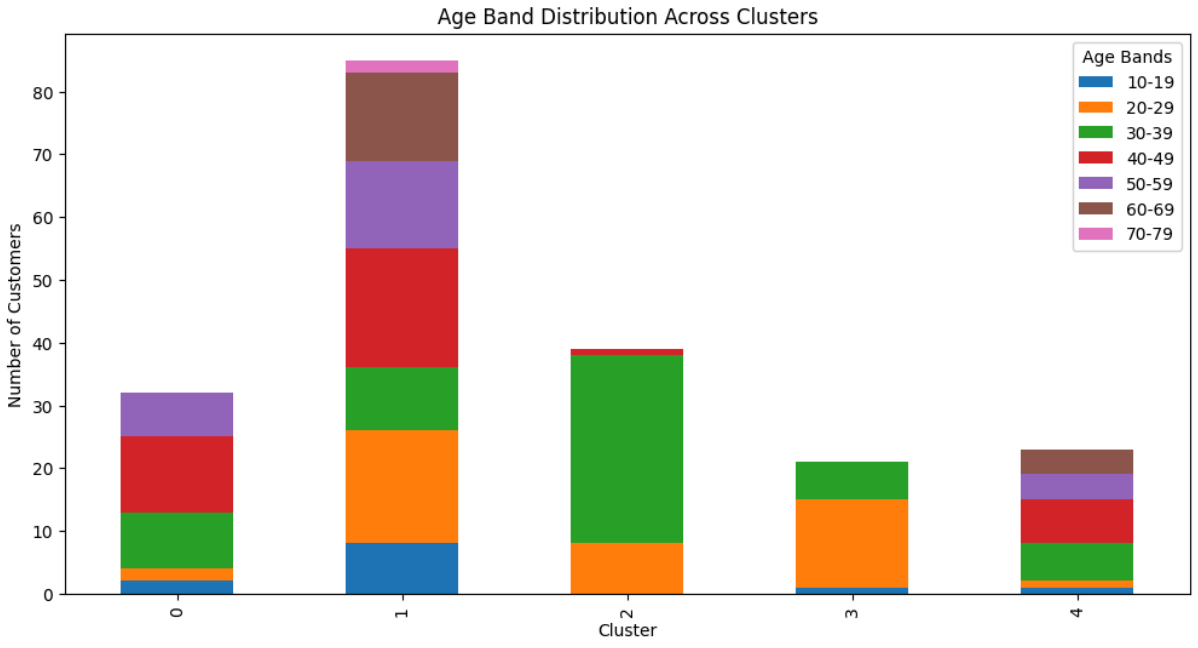
Here are the results of the clustering using n=5

Insights into cluster

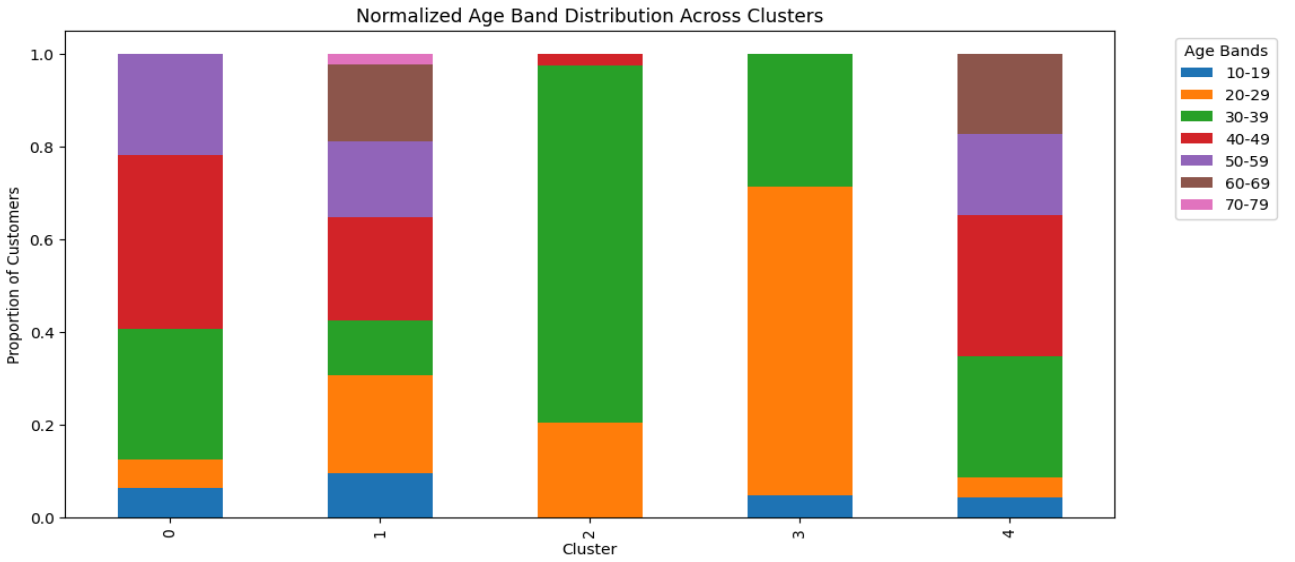
- Cluster 0 (bottom right) - high income, low spending score
- Cluster 1 (middle) - likely spenders who have an average income and average spending score
- Cluster 2 (top right) - high income, high spending scores
- Cluster 3 (top left) - lower income, but with high spending scores.
- Cluster 4 (bottom left) - low-income, low-spending scores

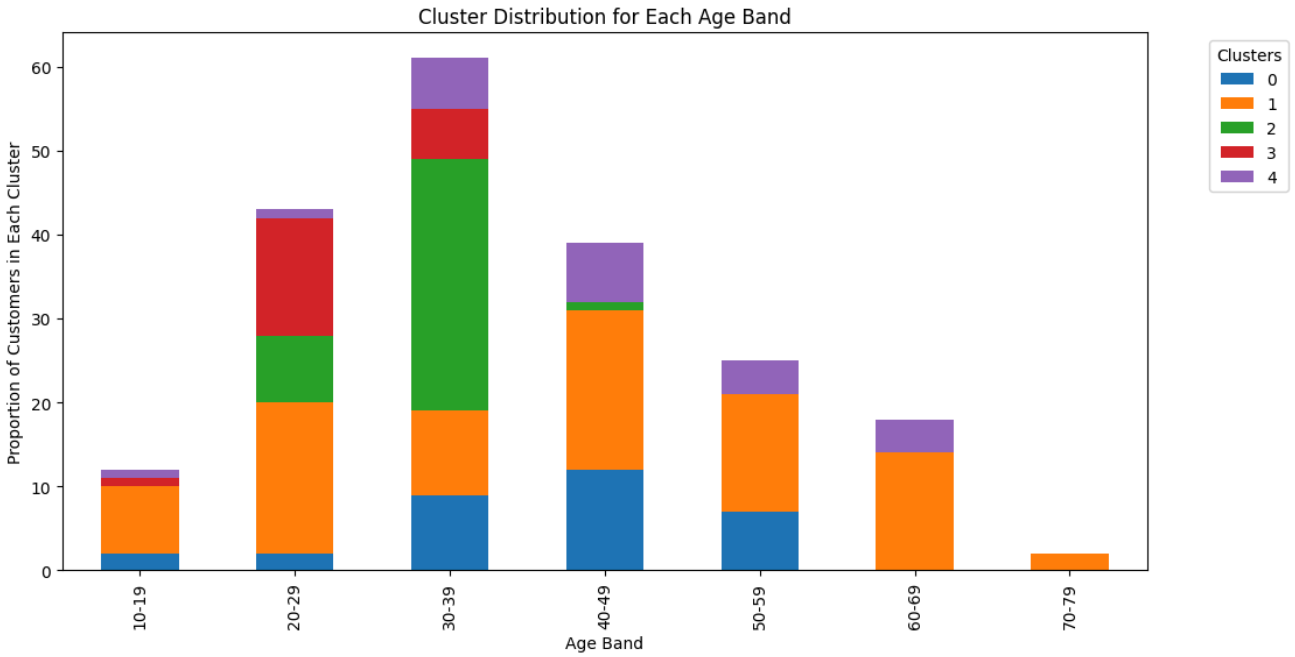
Let's look further into the statistics of each cluster to gain more insights, particularly the age and gender distribution





You can notice that certain clusters have disproportionately more customers than others. Hence, we feel that a better way to represent would be the portion of each age of each cluster





For reference, this was the observation we made previously.

- Cluster 0 (bottom right) - high income, low spending score
- Cluster 1 (middle) - average income and average spending score
- Cluster 2 (top right) - high income, high spending scores
- Cluster 3 (top left) - lower income, but with high spending scores.
- Cluster 4 (bottom left) - low-income, low-spending scores

Stage 4: Insights and Analysis from Clustering Implementation

What stood out was Cluster 3, which consisted of individuals with lower income but higher spending scores. These are people who, despite earning less, have higher consumption habits. Interestingly, this cluster is made up entirely of younger individuals, with the majority falling in the 20-29 age band, and the rest from 10-39. It makes sense, as younger people, particularly those in their 20s, are often influenced by trends driven by social media. They're likely to spend more liberally, possibly because they don't yet have major financial responsibilities—like kids or elderly parents to support—so they can use most of their income on themselves. There's also the possibility that some of them are spending beyond their means, either with "daddy's money" or by accumulating credit card debt.

Cluster 2 was also noteworthy. This group represents high-income individuals with high spending scores. They are mostly from the 20-39 age range, likely professionals or high earners who spend a significant portion of their income. Similar to Cluster 3, the youthfulness of this group suggests that their spending habits are influenced by social media trends, but because they have higher incomes, they can afford to maintain a high level of consumption. Unlike older high-income earners who might save more, these younger individuals probably have fewer financial obligations and therefore are willing to spend a larger portion of their income.

Cluster 1, on the other hand, includes middle-income earners with moderate spending scores. This cluster has a rather balanced representation across all age groups, which makes sense. These individuals earn an average income and spend within their means, without being overly conservative, which explains why every age group has a relatively equal share in this cluster.

Stage 5: Evaluation of Hierarchical Clustering implementation

Agglomerative clustering worked very well for this dataset, as all 5 clusters produced gave very meaningful insights. The resulting clusters are well-separated and reflect different customer segments. This provides distinct insights, allowing for efficient marketing and business strategies

5. Conclusion

Dataset	K means Clustering	Hierarchical Clustering
Wine dataset	Silhouette = 0.56 DBI = 0.59 CHI = 343.95	Silhouette = 0.28 DBI = 1.42 CHI = 67.65
Mall dataset	Silhouette = 0.55 DBI = 0.57 CHI = 247.82	Silhouette = 0.55 DBI = 0.58 CHI = 243

Conclusions for Wine Dataset

For the wine dataset, both K-mean and agglomerative hierarchical clustering concluded that the **optimal number of clusters is 3**. The elbow method also indicates that adding more clusters does not significantly improve its average squared centroid distances. In fact, if clustering is done with more than 3 clusters there is a drop in silhouette score, indicating poorer quality of clusters generated. The optimal k value (3) is also consistent with additional analysis done using the Davies-Bouldin Index (DBI) and Calinski-Harabasz Index (CHI) indicating that the clusters are well-separated and compact, indicating better clustering quality.

From the table above it is evident that **K means clustering produced better quality clusters**, across all three metrics. Owing to the high dimensional nature of the dataset, hierarchical clustering may have struggled to differentiate between points, while K-Means, especially after dimensionality reduction techniques like PCA, can still identify meaningful clusters based on centroids. Moreover from the cluster graphs produced, it was apparent that the data was more spherical in nature. Since Hierarchical clustering does not depend on centroids, it tends to recognise clusters with arbitrary shapes better. But in our dataset's case, with more spherical clusters, K means performed better.

We further analyse the feature's importance by comparing its cluster means and variance we also conclude that features Color_intensity' and 'Proline' of the wine dataset are more impactful to be used for targeted analyses of specific features for better understanding and segmentation in wine studies.

Conclusions for Mall Customer Dataset

For the customer mall dataset, we focused on segmenting mall customers based on demographic and spending behaviour using two clustering methods: K-Means and Hierarchical Clustering. K-Means clustering revealed an optimal number of clusters is 6 for the full range of features in the dataset and 5 for a focused analysis of the Annual Income and Spending Score (after dropping gender), leading to an improved silhouette score of from 0.445 to 0.554, indicating better cluster separability. The DBI also dropped from 0.82 to 0.57 while CHI went from 150.76 to 247.82. All three of these evaluation metrics mean that the clusters are highly distinct, well-compacted and well-defined. This is expected as the gender

is a qualitative binary categorical feature and the k-means algorithm works best with continuous numerical data due to the algorithm using Euclidean distance to compute similarity in the clusters.

For Hierarchical Clustering, we started with a focused analysis of the Annual Income and Spending Score, and got a silhouette score of 0.552, DBI is 0.58 while CHI is 243, which is very close to that for K-Means.

In addition, we did an analysis with the Standardised, full range of features and revealed that the optimal number of clusters is around n=17, with silhouette score of 0.439, DBI of 0.72 and CHI of around 90, of which is significantly worse. As discussed in the section, the large number of clusters (17) with respect to the number of datapoints (200) is likely over segmented, and this happens when the number of clusters exceeds the intrinsic complexity of the data.

The silhouette score is a good evaluation metric to suggest that the clusters were well separated. The identified clusters highlighted distinct customer segments, such as average spenders and wealthy customers. Upon looking into the demographics of the clusters, the distribution of age and gender makes sense when cross-referenced with domain knowledge on customer behaviour norms.

Hierarchical clustering reinforced these findings, suggesting 5 clusters as optimal. Overall, the study demonstrated the significance of Annual Income and Spending Score in defining customer segments while indicating that Gender and Age contribute less to cluster separation.

Conclusions for K Means Clustering Algorithm against the 2 datasets

Across both datasets, K Means clustering yielded positive results. Both datasets had robust data design with relationships that were spherical in nature. Since K means clustering generates similarity scores based on Euclidean distances from a centroid point, it leveraged the data's spherical nature to its advantage.

Since K means clustering aims to minimise within-cluster variance and relies on centroids - it is most effective for compact and equally sized clusters. From the cluster diagrams for both datasets, it is evident that the datasets chosen generated such circular and compact clusters. This further highlights how Kmeans was an effective algorithm for clustering.

Moreover, from the analysis of the wine dataset, it is apparent that K means clustering is particularly strong in clustering high-dimensional datasets. Since it has a time complexity of O(n), it is more efficient for large datasets.

Conclusions for Hierarchical Algorithm against the 2 datasets

We observed that Hierarchical clustering works better on lower-dimensional datasets. This is because the high-dimensional datasets are often more sensitive to noises as hierarchical

merge clusters based on distances while in lower-dimensional datasets, there is a lesser chance of having noisy features.

This is backed by our experimental results shown below, with the Mall Dataset (2 features used in clustering) having significantly better evaluation score across all 3 metrics than the Wine Dataset (13 features used in clustering)

Dataset	Hierarchical Clustering
Wine dataset	Silhouette = 0.28 DBI = 1.42 CHI = 67.65
Mall dataset	Silhouette = 0.55 DBI = 0.58 CHI = 243

Hierarchical clustering has a higher time complexity compared to K-Means. The computational cost becomes significant as the number of dimensions and data points increases. Lower-dimensional datasets are more computationally manageable and distance calculations are more meaningful.

In addition, the Wine dataset with more features led to fewer distinct hierarchical groupings in the dendrogram, compared to the mall customer dataset that offered insights into the relationship of different customer segments.

All in all, we can safely conclude that for datasets with higher dimensionality, we should avoid Hierarchical clustering, while we can expect good clustering from datasets with lower dimensionality.

Contribution Form

Name	**Matriculation Number**	**Detailed Individual Contribution**
Hendy	U2122559J	● Hierarchical Clustering on Wine Dataset ● Report Writing : ○ Introduction ○ Problem statement ○ Dataset 1: Wine Chemical Data ○ Hierarchical Clustering on Wine Dataset ○ Conclusion
Goyal Ananya Surendrakumar	U2023124B	● K-Means on Wine Dataset ● Report Writing : ○ Exploratory Data Analysis for Dataset 1 ○ Dataset 1: Wine Chemical Data ○ K-Means on Wine Dataset ○ Conclusion
Lee Bo Hua	U2122595D	● Hierarchical Clustering on Mall Customer Dataset ● Report Writing : ○ Dataset 2: Customer Mall Data ○ Hierarchical Clustering on Mall Customer Dataset ○ Conclusion
Mitra Ren Sachithananthan	U2020190D	● K-Means on Mall Customer Dataset ● Report Writing : ○ Dataset 2: Customer Mall Data ○ K-Means on Mall Customer Dataset ○ Conclusion

References

1. Choudhary, V. (2019). *Customer Segmentation Tutorial in Python* [Data set]. Kaggle. https://www.kaggle.com/datasets/vjchoudhary7/customer-segmentation-tutorial-in-python/data

2. Aeberhard, S., Coomans, D., & de Vel, O. (1991). *Wine* [Data set]. UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/109/wine

3. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.

4. Müllner, D. (2013). "fastcluster: Fast Hierarchical, Agglomerative Clustering Routines for R and Python." *Journal of Statistical Software*, 53(9), 1-29.

5. Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53-65

6. Thorndike, R. L. (1953). Who belongs in the family? *Psychometrika*, 18(4), 267-276.

7. Murtagh, F., & Contreras, P. (2012). Methods of Hierarchical Clustering. *Computing Surveys*, 44(3), 1-36.

8. Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8), 651-666.

9. Kavlakoglu, E., & Winland, V. (2024, August 27). *What is K-means clustering?*. IBM. https://www.ibm.com/topics/k-means-clustering#:~:text=K%2Dmeans%20clustering%20is%20an%20iterative%20process%20to%20minimize%20the,euclidean%2C%20from%20the%20cluster%20center