A logo on a black background

Description automatically generated

**Assignment for PMIM402**

|  |  |
| --- | --- |
| **Module number:** | PMIM402 |
| **Module name:** | Machine Learning |
| **Title of assignment:** | *Clustering Assignment* |
| **Student ID number:** | 2333193 |
| **Word count:** |  |
| **Declaration:** | I understand the following conditions which apply throughout this course:   1. I confirm that I am the sole author of this work. 2. I understand that proof reading by a third party is discouraged, but if used, records should be available as per guidelines. 3. I understand the need for academic integrity and that all my submitted work will adhere to its principles. 4. I understand that the teaching team will take measures to deter, detect and report any academic misconduct. 5. I agree to my work being submitted to the TurnItIn academic database. 6. I understand the importance of assignment deadlines and the need to seek help in good time where personal circumstances interrupt my work. |
| **Please copy and paste this declaration onto the front of the submission.** | |

**K-means clustering**

1. The attribute “num” represents the binary class attribute indicating the presence or absence of heart disease. Including this attribute in unsupervised learning, such as k-means clustering, would bias the creation of clusters towards the disease status, instead of allowing the algorithm to identify inherent groupings based on data features. The goal of clustering algorithms is to discover natural clusters in the data without using predefined labels.
2. The following snippet code illustrates different preprocessing steps, the k-means model, and determines the optimal number of clusters using the Elbow method.

library(tidyverse)

library(factoextra)

library(caret)

# Load the dataset

heart\_c <- read\_csv("C:/Users/USER/Desktop/HDS/machine learning/data/heart-c.csv")

str(heart\_c)

# Remove the 'num' attribute as it's the class label not to be included in clustering

# and the '...1' column.

heart\_c <- heart\_c %>% select(-num, -"...1")

# Remove missing values

heart\_c <- na.omit(heart\_c)

# Convert binary (logical) variables to numeric

heart\_c$fbs <- as.numeric(heart\_c$fbs)

# Convert 'sex' and 'exang' into binary numeric variables

heart\_c$sex <- ifelse(heart\_c$sex == "male", 1, 0)

heart\_c$exang <- ifelse(heart\_c$exang == "yes", 1, 0)

# Convert to factors

heart\_c$cp <- factor(heart\_c$cp, levels = c("typ\_angina", "atyp\_angina", "non\_anginal", "asympt"))

heart\_c$restecg <- factor(heart\_c$restecg, levels = c("normal", "st\_t\_wave\_abnormality", "left\_vent\_hyper"))

heart\_c$slope <- factor(heart\_c$slope, levels = c("up", "flat", "down"))

heart\_c$thal <- factor(heart\_c$thal, levels = c("normal", "fixed\_defect", "reversable\_defect"))

# One-hot encoding

dummy\_model <- dummyVars("~ . ", data = heart\_c)

heart\_c\_transformed <- predict(dummy\_model, newdata = heart\_c)

# Normalise the data

heart\_c\_scaled <- scale(heart\_c\_transformed)

# Initialize total within sum of squares error: WSS

wss <- 0

# For 1 to 10 cluster centers

for (i in 1:10) {

km\_out <- kmeans(heart\_c\_scaled, centers = i, nstart = 20)

# Save total within sum of squares to WSS variable

wss[i] <- km\_out$tot.withinss

}

# Plot total within sum of squares vs. number of clusters

plot(1:10, wss, type = "b",

xlab = "Number of Clusters",

ylab = "Within groups sum of squares")

# Perform K-means clustering

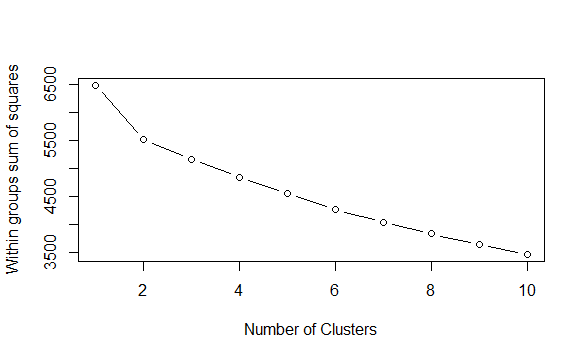
km\_out <- kmeans(heart\_c\_scaled, centers = 2, nstart = 20)

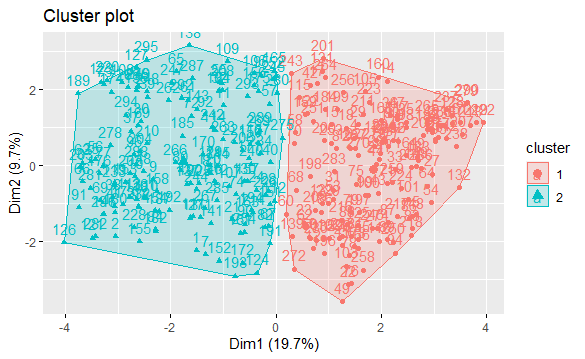
# Visualise clusters

fviz\_cluster(km\_out, data = heart\_c\_scaled)

The Elbow method graph for a range of k values (1 to 10) shows the Within-cluster Sum of Squares (WSS) as a function of the number of clusters. The optimum value of k is where the WSS begins to decrease at a slower rate, indicating that adding more clusters does not improve the model’s ability to fit the data. The graph suggests that k = 2 could be the optimum point. Considering the interpretability of the clusters and the clear separation with a significant reduction in WSS, K = 2 emerges as the most suitable choice.

Below are the Elbow graph and the clusters plot:





3- The numerical features with high variability\*, ‘age’, ‘trestbps’ (resting blood pressure), ‘chol’ (serum cholesterol), and ‘thalach’ (maximum heart rate achieved), are identified as **the most useful** for k-means clustering regarding heart disease risk factors. Their direct relevance and significant variability make them particularly valuable to identify clusters within the data. Binary features; ‘sex’ (male/female), ‘fbs’ (fasting blood sugar > 120 mg/dl) and ‘exang’ (exercise-induced angina) can be considered **less useful** for offering less variability.

\*Below is the code and the graph to visualise the distribution of the numeric features.

library(ggplot2)

# Boxplot for visualising distributions of numeric features

heart\_c %>%

select\_if(is.numeric) %>%

gather(key = "features", value = "value") %>%

ggplot(aes(x = features, y = value)) +

geom\_boxplot() +

theme(axis.text.x = element\_text(angle = 45, hjust = 1))



**Hierarchical clustering**

1. The following snippet code illustrates different steps to run hierarchical clustering and determine the optimal number of clusters by using the Dunn Index method.

library(tidyverse)

library(cluster)

library(factoextra)

library(caret)

library(clValid)

library(dendextend)

# Hierarchal clustering

# Compute the distance matrix

dist\_mat <- dist(heart\_c\_scaled)

# Perform hierarchical clustering with average linkage

hclust\_avg <- hclust(dist\_mat, method = "average")

# Plot the dendrogram

plot(hclust\_avg, main = "Hierarchical Clustering Dendrogram", sub= "", xlab = "")

# Initialize the dunns vector

dunns <- 0

# Loop from 1 to 10 to calculate Dunn index for each number of clusters

for (i in 1:10) {

cluster\_assignments <- cutree(hclust\_avg, i)

dunns[i] <- dunn(dist\_mat, cluster\_assignments)

}

# Plot dunns vector against the cluster numbers 1 to 10

plot(dunns, type = "b", xlab = "Number of clusters", ylab = "Dunn Index",

main = "Dunn Index for Different Numbers of Clusters")

# The dendrogram at k = 2 and color the clusters

dend <- as.dendrogram(hclust\_avg)

dend\_colored <- color\_branches(dend, k = 2)

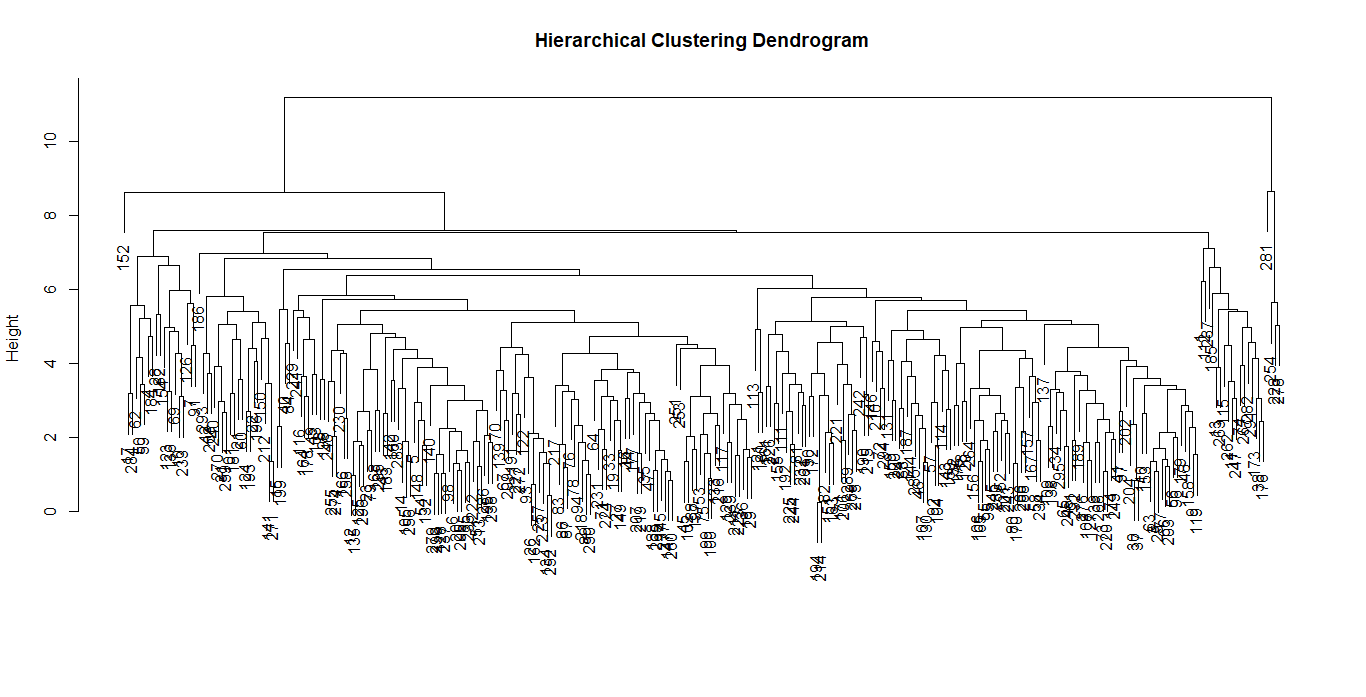
plot(dend\_colored, main = "Dendrogram with k = 2 Clusters")

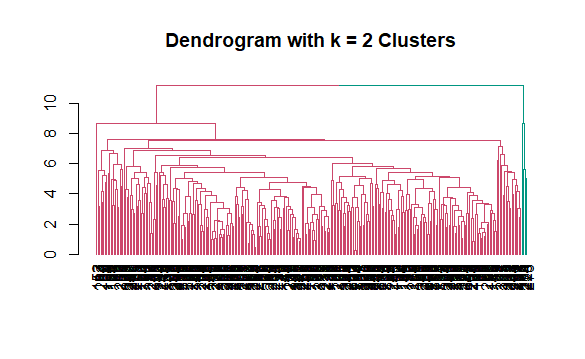
To find the optimal number of clusters, the Dunn Index is calculated for a range of cluster numbers (1 to 10) and plot these values. The number of clusters corresponding to the highest Dunn Index value is considered optimal. This is because the highest Dunn Index signifies the best balance between compactness (cohesion within clusters) and separation (distinction between clusters). By examining how the Dunn Index varies as the number of clusters changes, k = 2 is identified as an optimal value since increasing the number beyond this worsens the quality of clustering according to these criteria.

A graph with lines and dots

Description automatically generated

Below are the clustering results in tree structures:





1. The average linkage method, also referred to as Unweighted Pair Group Method with Arithmetic Mean (UPGMA), is a technique used in hierarchical clustering for measuring the distance between two clusters.

Below is a detailed description of how this method works (1–3):

1- Consider each data point as a single cluster.

2- For each pair of clusters, calculate the distances between each possible pair of data points, where one element is from the first cluster and the other is from a second cluster. The distance between two elements can be measured by the Euclidean distance method.

3- Compute the average distance between each pair of clusters.

4- Find the pair of clusters with the smallest average distance and merge them into a single cluster.

5- After merging, update the distance matrix to reflect the distances between the new cluster and all other clusters by calculating their average distances.

6- Repeat steps 3-5 until all data points are merged into a single cluster. This process results in a dendrogram that represents the hierarchical clustering of the data.

1. **Strength** of the average linkage method:

- It tends to create more **balanced and moderately compact clusters with clear separation**, compared to the single linkage method, which may produce chain-like clusters, or complete linkage, which can create tightly compact clusters (1–3).

- It is **less sensitive to outliers** than the single linkage method because it considers all pairwise distances between cluster elements rather than just the closest or farthest points (1–3).

**Limitations** of the average linkage method:

- The need to compute and update the average distances between clusters as they are merged can make the average linkage method **computationally intensive** compared to the single linkage method, especially for large datasets (1–3).

- Although less sensitive to outliers than the single linkage method, the average linkage method may still be **influenced by noise and outliers** (1–3)**.**

.

**References**

1. Yim O, Ramdeen KT. Hierarchical Cluster Analysis: Comparison of Three Linkage Measures and Application to Psychological Data. TQMP. 2015 Feb 1;11(1):8–21.

2. Jarman AM. Hierarchical Cluster Analysis: Comparison of Single linkage,Complete linkage, Average linkage and Centroid Linkage Method.

3. Wilkinson PR. 9.3 Hierarchical clustering methods | Multivariate Statistics [Internet]. [cited 2024 Mar 4]. Available from: https://rich-d-wilkinson.github.io/MATH3030/9.3-hierarchical-clustering-methods.html