PREDICTIVE ANALYTICS

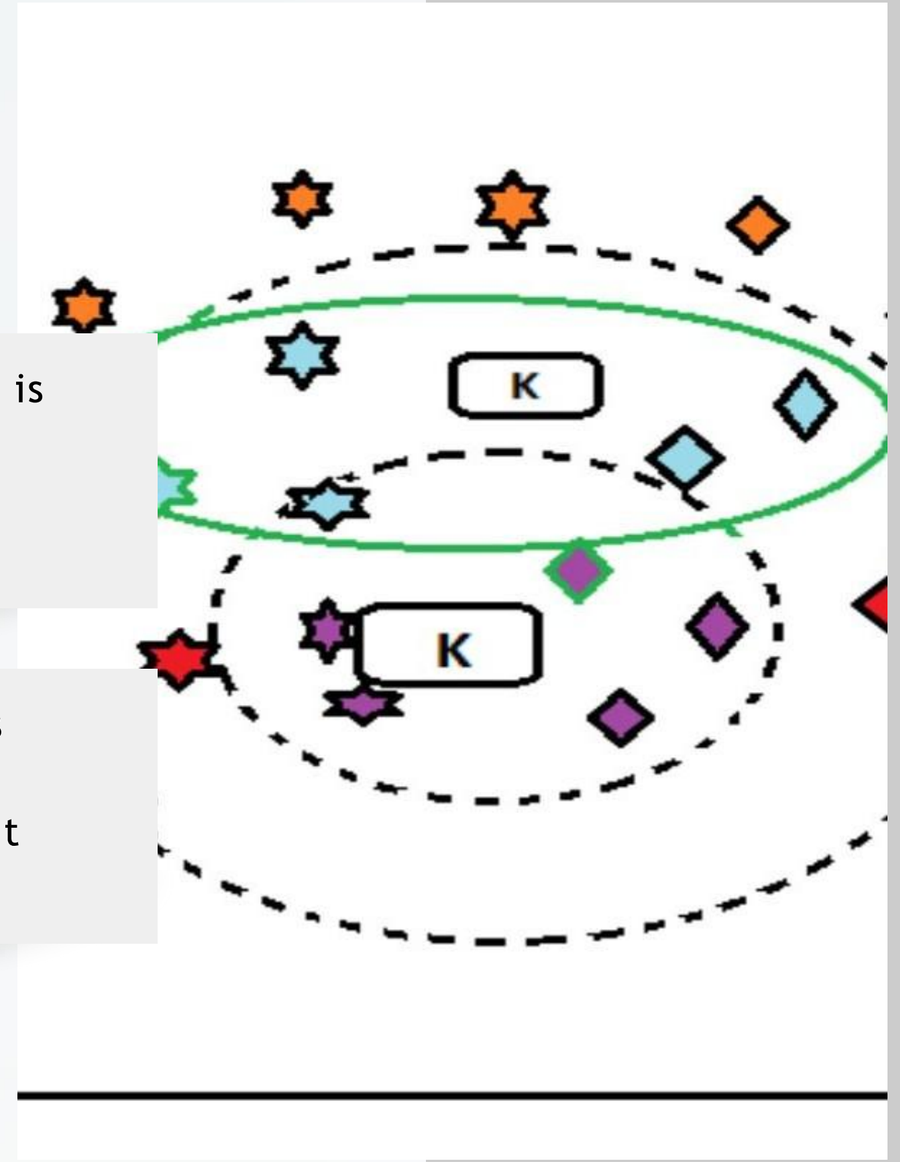
MITESH PANDA

CONTENT

KNN MODEL

The **K-Nearest Neighbors (KNN) algorithm** is a supervised machine learning method employed to tackle classification and regression problems.

It is widely disposable in real-life scenarios since it is non-parametric, meaning it does not make any underlying assumptions about the distribution of data



APPLICATIONS OF KNN MODEL

Recommendation Systems

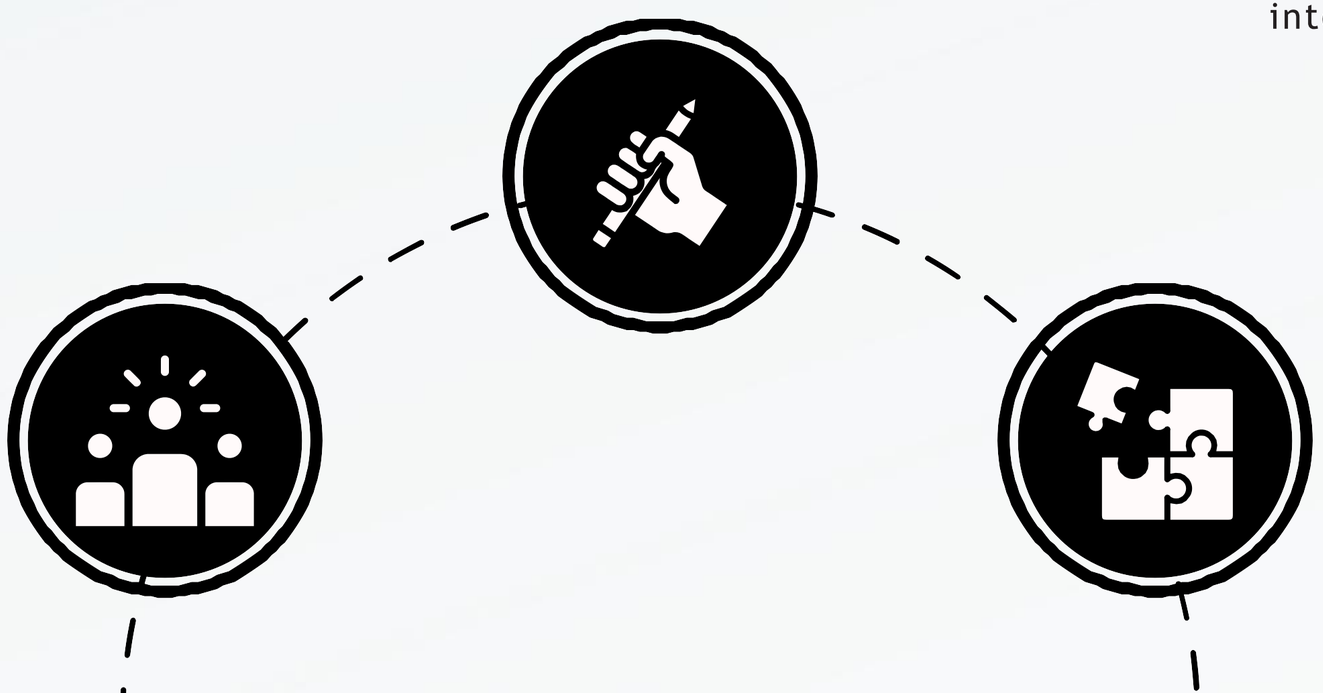
Predict user preferences for items.

Pattern Recognition

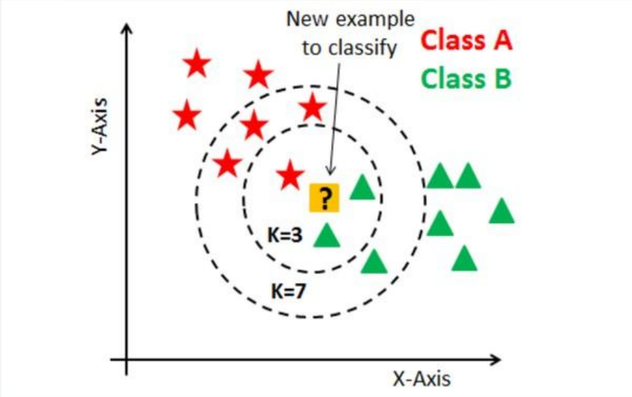
Identify patterns in image, audio or text

Healthcare and Medicine

Diagnose diseases by classifying symptoms into known conditions

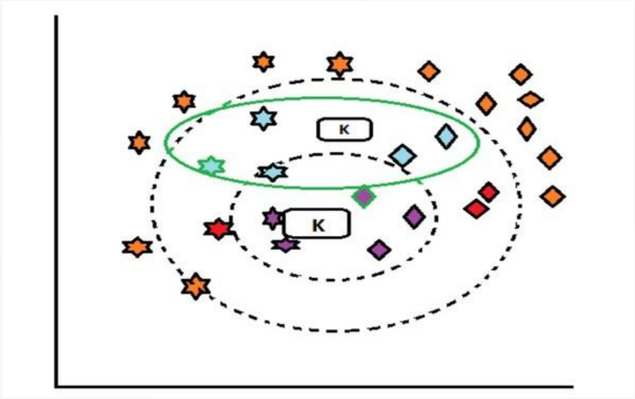


WORKING OF KNN



The **K-Nearest Neighbors (KNN)** algorithm is a simple, non-parametric machine learning model used for classification and regression tasks. It operates by storing the entire training dataset and making predictions based on the similarity between data points. For any new test instance, KNN calculates the distance (e.g., Euclidean distance) between the test point and all points in the training set. It then identifies the k-nearest neighbors—where k is a user-defined parameter—and bases its prediction on these neighbors. In classification tasks, the model assigns the majority class among the neighbors to the test instance.

For regression, it predicts the average value of the neighbors. KNN is intuitive and effective for smaller datasets but can become computationally expensive for large datasets, as it requires distance computation for all training points. Moreover, it is sensitive to irrelevant features and requires proper scaling of data for optimal performance.



NAIVE BAYES MODEL

The “Naive” part of the name indicates the simplifying assumption made by the Naïve Bayes classifier. The classifier assumes that the features used to describe an observation are conditionally independent, given the class label.

The “Bayes” part of the name refers to Reverend Thomas Bayes, an 18th-century statistician and theologian who formulated Bayes’ theorem.

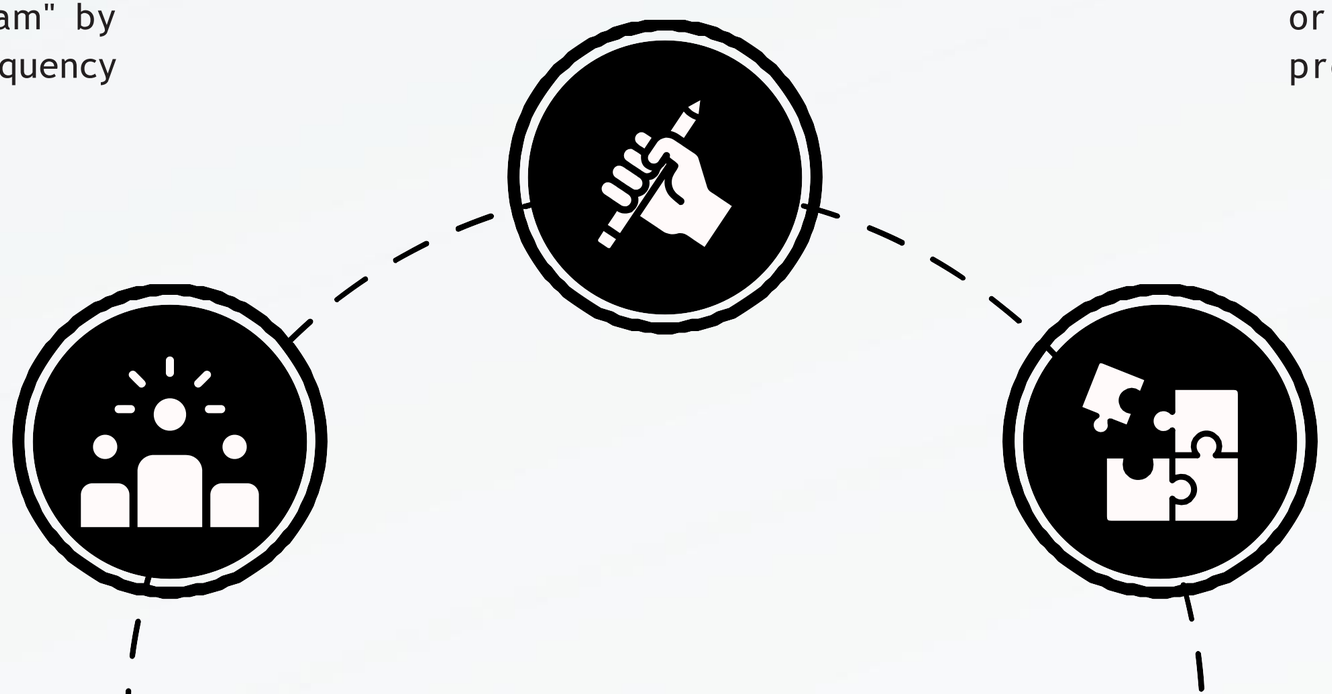
APPLICATIONS OF NB MODEL

Spam Email DEtection

Naive Bayes is widely used to classify emails as "spam" or "not spam" by analyzing word frequency and other features.

Sentiment Analysis

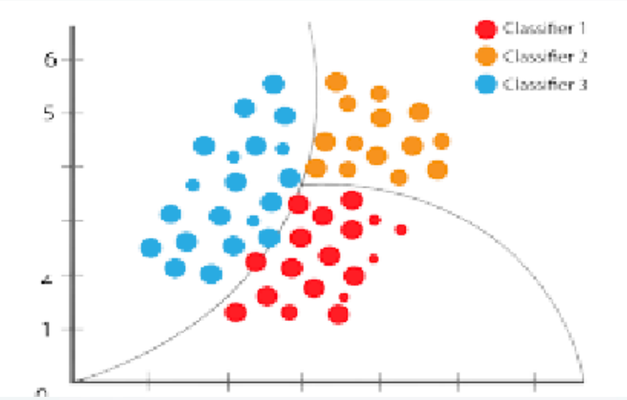
Classifies the sentiment of text (positive, negative, neutral) in reviews, tweets, or comments.



Document Categorization

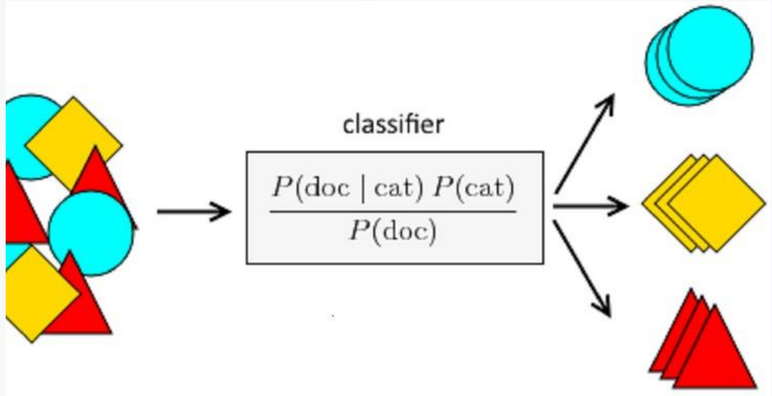
Automatically categorizes documents or news articles into predefined topics or genres.

WORKING OF NB



The Naive Bayes model is a probabilistic machine learning algorithm based on Bayes' Theorem, with the assumption that all features are independent (the "naive" assumption). It calculates the probability of a data point belonging to a particular class given its features. The model works by estimating the conditional probability of each feature given a class and combining these probabilities to predict the most likely class for the input.

Despite its simplicity, Naive Bayes performs well in real-world scenarios, especially for tasks like text classification, where feature independence often holds approximately true. It is efficient, easy to implement, and particularly effective for high-dimensional data.



CLUSTERING

Clustering groups data points with similar characteristics into clusters, enabling the identification of hidden patterns or natural groupings within data. These clusters can then serve as a foundation for predictive models by providing insights into distinct behaviors, such as customer segments or product categories.

By preprocessing data through clustering, predictive analytics models can focus on specific subgroups rather than treating the entire dataset as homogeneous. For example, separating customers into clusters (e.g., based on spending habits) allows personalized predictions, enhancing model performance and decision-making accuracy.

K-MEANS CLUSTERING

K means clustering, assigns data points to one of the K clusters depending on their distance from the center of the clusters. It starts by randomly assigning the clusters centroid in the space. Then each data point assign to one of the cluster based on its distance from centroid of the cluster. After assigning each point to one of the cluster, new cluster centroids are assigned. This process runs iteratively until it finds good cluster. In the analysis we assume that number of cluster is given in advanced and we have to put points in one of the group.

APPLICATIONS OF K-MEANS CLUSTERING

Customer Segmentation

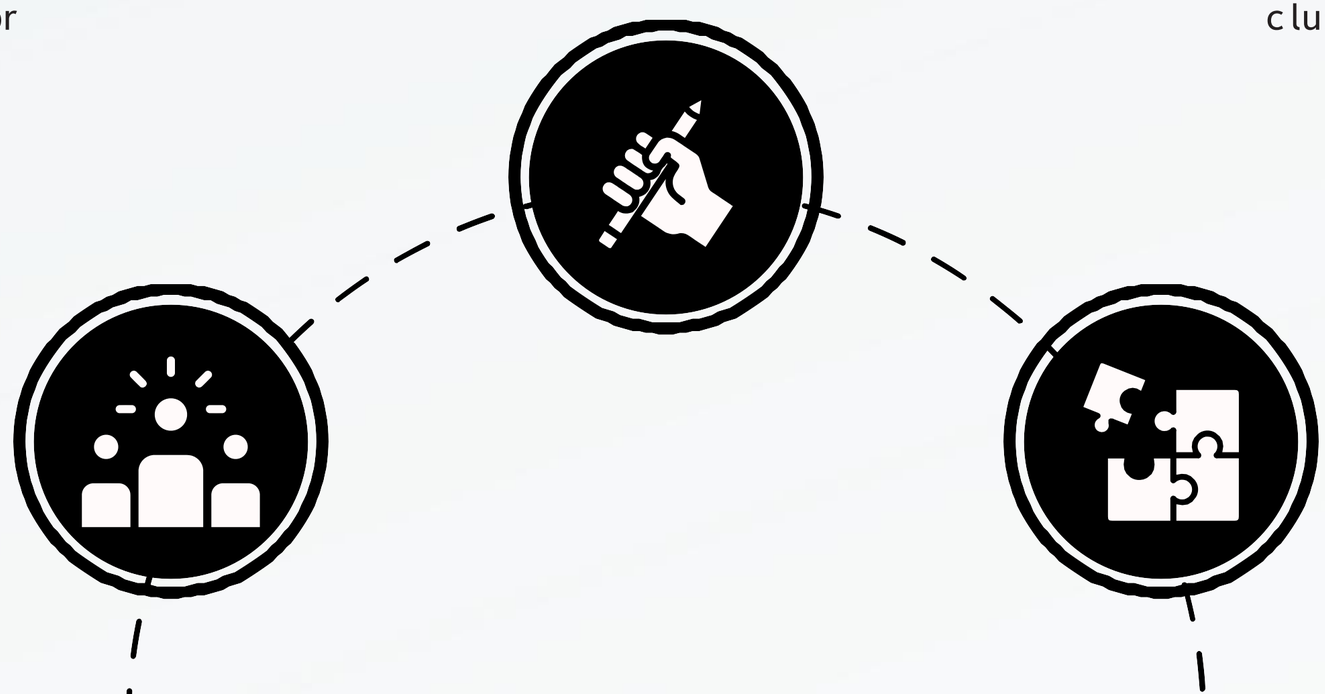
Groups customers based on purchasing behavior, demographics, or preferences.

Image Compression

Reduces the number of colors in an image by clustering pixels with similar colors and representing them with a single value.

Anomaly Detection

Identifies data points that do not fit into any cluster, flagging them as anomalies.



HIERARCHICAL CLUSTERING

Hierarchical clustering is a connectivity-based clustering model that groups the data points together that are close to each other based on the measure of similarity or distance. The assumption is that data points that are close to each other are more similar or related than data points that are farther apart. A dendrogram, a tree-like figure produced by hierarchical clustering, depicts the hierarchical relationships between groups. Individual data points are located at the bottom of the dendrogram, while the largest clusters, which include all the data points, are located at the top. In order to generate different numbers of clusters, the dendrogram can be sliced at various heights.

APPLICATIONS OF HIERARCHICAL CLUSTERING

Gene Expression Analysis

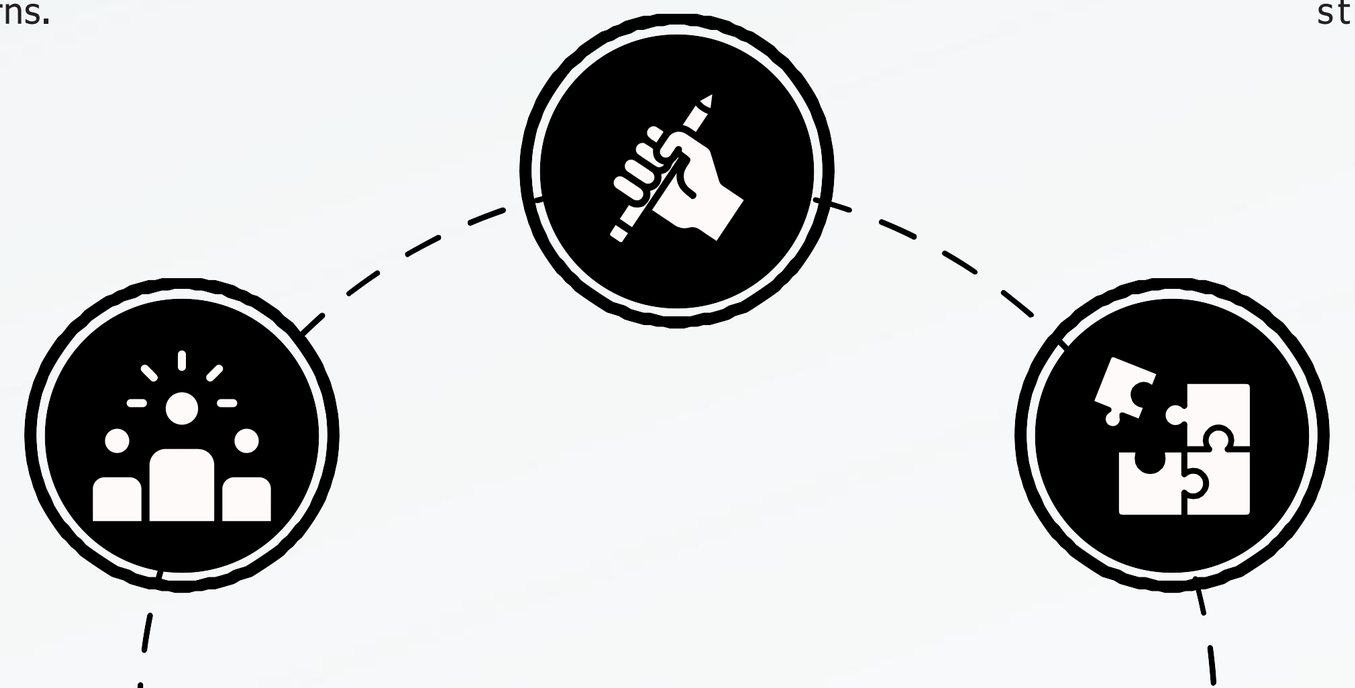
Used in bioinformatics to group genes with similar expression patterns.

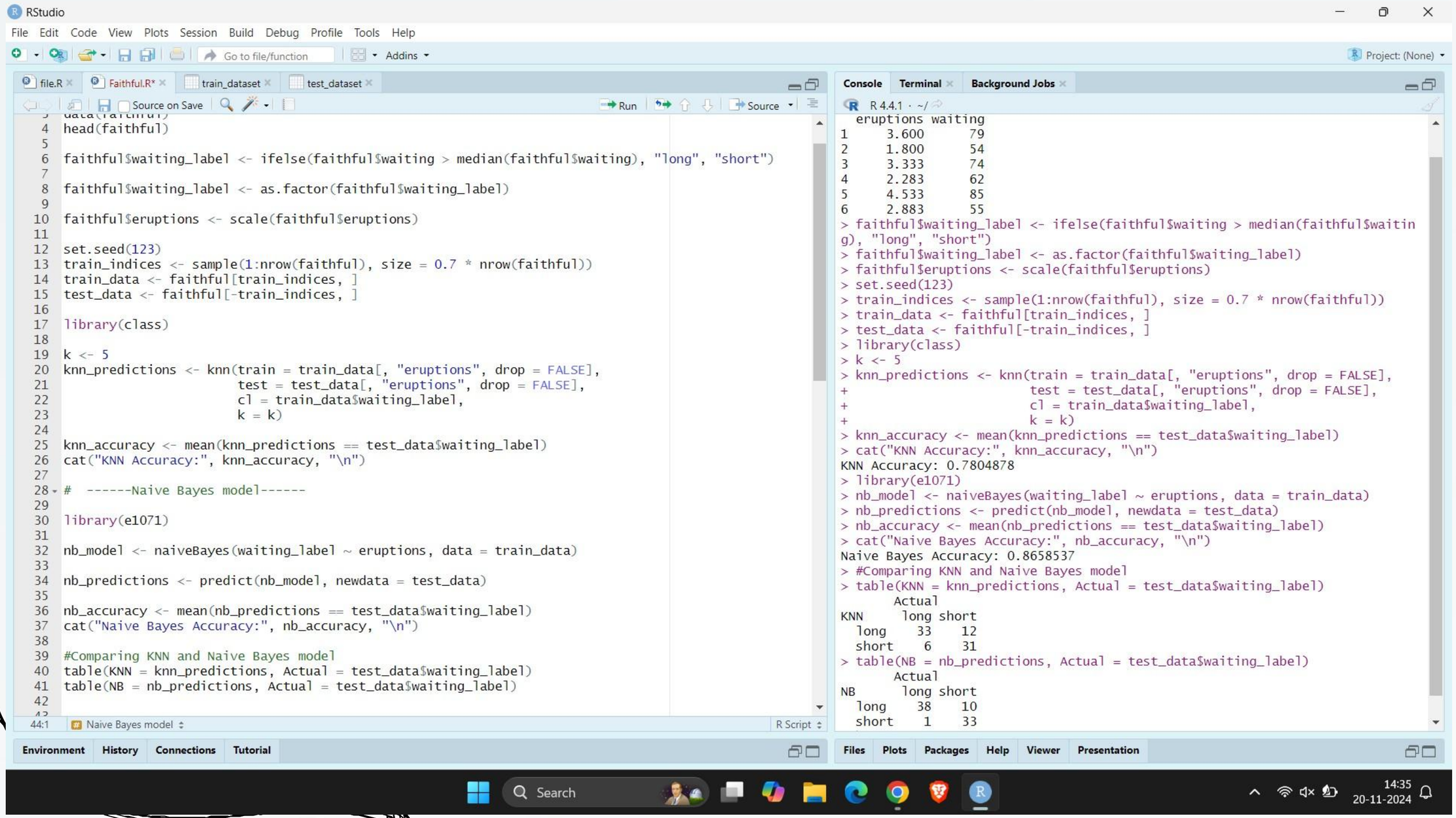
Market Research

Groups customers or products based on features like preferences, sales, or ratings.

Document Categorization

Organizes documents or texts into a hierarchical structure based on similarity.




RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
file.R
Faithful.R*
train_dataset
test_dataset
Source on Save
Run
Source
data(faithful)
head(faithful)
faithful$waiting_label <- ifelse(faithful$waiting > median(faithful$waiting), "long", "short")
faithful$waiting_label <- as.factor(faithful$waiting_label)
faithful$eruptions <- scale(faithful$eruptions)
set.seed(123)
train_indices <- sample(1:nrow(faithful), size = 0.7 * nrow(faithful))
train_data <- faithful[train_indices,]
test_data <- faithful[-train_indices,]
library(class)
k <- 5
knn_predictions <- knn(train = train_data[, "eruptions", drop = FALSE],
test = test_data[, "eruptions", drop = FALSE],
cl = train_data$waiting_label,
k = k)
knn_accuracy <- mean(knn_predictions == test_data$waiting_label)
cat("KNN Accuracy:", knn_accuracy, "\n")
# ------Naive Bayes model------
library(e1071)
nb_model <- naiveBayes(waiting_label ~ eruptions, data = train_data)
nb_predictions <- predict(nb_model, newdata = test_data)
nb_accuracy <- mean(nb_predictions == test_data$waiting_label)
cat("Naive Bayes Accuracy:", nb_accuracy, "\n")
#Comparing KNN and Naive Bayes model
table(KNN = knn_predictions, Actual = test_data$waiting_label)
table(NB = nb_predictions, Actual = test_data$waiting_label)
44:1
Naive Bayes model
R Script
Console
Terminal
Background Jobs
R 4.4.1 · ~/
eruptions waiting
1 3.600 79
2 1.800 54
3 3.333 74
4 2.283 62
5 4.533 85
6 2.883 55
> faithful$waiting_label <- ifelse(faithful$waiting > median(faithful$waiting), "long", "short")
> faithful$waiting_label <- as.factor(faithful$waiting_label)
> faithful$eruptions <- scale(faithful$eruptions)
> set.seed(123)
> train_indices <- sample(1:nrow(faithful), size = 0.7 * nrow(faithful))
> train_data <- faithful[train_indices,]
> test_data <- faithful[-train_indices,]
> library(class)
> k <- 5
> knn_predictions <- knn(train = train_data[, "eruptions", drop = FALSE],
+ test = test_data[, "eruptions", drop = FALSE],
+ cl = train_data$waiting_label,
+ k = k)
> knn_accuracy <- mean(knn_predictions == test_data$waiting_label)
> cat("KNN Accuracy:", knn_accuracy, "\n")
KNN Accuracy: 0.7804878
> library(e1071)
> nb_model <- naiveBayes(waiting_label ~ eruptions, data = train_data)
> nb_predictions <- predict(nb_model, newdata = test_data)
> nb_accuracy <- mean(nb_predictions == test_data$waiting_label)
> cat("Naive Bayes Accuracy:", nb_accuracy, "\n")
Naive Bayes Accuracy: 0.8658537
> #Comparing KNN and Naive Bayes model
> table(KNN = knn_predictions, Actual = test_data$waiting_label)
Actual
KNN long short
long 33 12
short 6 31
> table(NB = nb_predictions, Actual = test_data$waiting_label)
Actual
NB long short
long 38 10
short 1 33
Environment
History
Connections
Tutorial
Files
Plots
Packages
Help
Viewer
Presentation
Search
14:35
20-11-2024

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

file.R | Faithful.R* | train_dataset | test_dataset

Source on Save | Run | Source

```
31
32 nb_model <- naiveBayes(waiting_label ~ eruptions, data = train_data)
33
34 nb_predictions <- predict(nb_model, newdata = test_data)
35
36 nb_accuracy <- mean(nb_predictions == test_data$waiting_label)
37 cat("Naive Bayes Accuracy:", nb_accuracy, "\n")
38
39 #Comparing KNN and Naive Bayes model
40 table(KNN = knn_predictions, Actual = test_data$waiting_label)
41 table(NB = nb_predictions, Actual = test_data$waiting_label)
42
43
44 #Clustering: K-means and Hierarchical
45
46 #  ------K-means Clustering------
47
48 data(faithful)
49
50 faithful_normalized <- scale(faithful)
51 head(faithful_normalized)
52
53 set.seed(123)
54 wss <- sapply(1:10, function(k) {
55   kmeans(faithful_normalized, centers = k, nstart = 10)$tot.withinss
56 })
57
58 plot(1:10, wss, type = "b", pch = 19, frame = FALSE,
59      xlab = "Number of Clusters (k)",
60      ylab = "Total Within-Cluster Sum of Squares",
61      main = "Elbow Method for Determining k")
62
63 kmeans_result <- kmeans(faithful_normalized, centers = 2, nstart = 10)
64
65 faithful$cluster <- as.factor(kmeans_result$cluster)
66
67 library(ggplot2)
68 ggplot(faithful, aes(x = eruptions, y = waiting, color = cluster)) +
69   geom_point(size = 3) +
70   labs(title = "K-Means Clustering of faithful Dataset",
```

63:1 | K-means Clustering | R Script

Environment | History | Connections | Tutorial

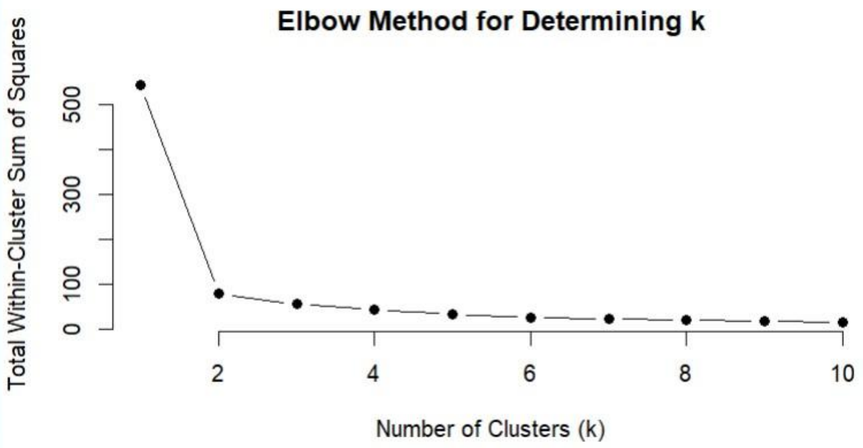
Console | Terminal | Background Jobs

R 4.4.1 · ~/

```
> head(faithful_normalized)
    eruptions    waiting
1  0.09831763  0.5960248
2 -1.47873278 -1.2428901
3 -0.13561152  0.2282418
4 -1.05555759 -0.6544374
5  0.91575542  1.0373644
6 -0.52987412 -1.1693335
> set.seed(123)
> wss <- sapply(1:10, function(k) {
+   kmeans(faithful_normalized, centers = k, nstart = 10)$tot.withinss
+ })
> plot(1:10, wss, type = "b", pch = 19, frame = FALSE,
+      xlab = "Number of Clusters (k)",
+      ylab = "Total Within-Cluster Sum of Squares",
+      main = "Elbow Method for Determining k")
>
```

Files | Plots | Packages | Help | Viewer | Presentation

Zoom | Export | Publish



Search

14:36
20-11-2024

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

file.R | Faithful.R* | train_dataset | test_dataset

Source on Save | Run | Source

```
49
50 faithful_normalized <- scale(faithful)
51 head(faithful_normalized)
52
53 set.seed(123)
54 wss <- sapply(1:10, function(k) {
55   kmeans(faithful_normalized, centers = k, nstart = 10)$tot.withinss
56 })
57
58 plot(1:10, wss, type = "b", pch = 19, frame = FALSE,
59      xlab = "Number of Clusters (k)",
60      ylab = "Total Within-Cluster Sum of Squares",
61      main = "Elbow Method for Determining k")
62
63 kmeans_result <- kmeans(faithful_normalized, centers = 2, nstart = 10)
64
65 faithful$cluster <- as.factor(kmeans_result$cluster)
66
67 library(ggplot2)
68 ggplot(faithful, aes(x = eruptions, y = waiting, color = cluster)) +
69   geom_point(size = 3) +
70   labs(title = "K-Means Clustering of faithful Dataset",
71        x = "Eruption Duration",
72        y = "Waiting Time")
73
74 #  ------Hierarchical Clustering------
75
76 distance_matrix <- dist(faithful_normalized)
77
78 hclust_result <- hclust(distance_matrix, method = "ward.D2")
79
80 plot(hclust_result, main = "Hierarchical Clustering Dendrogram",
81      xlab = "", sub = "", cex = 0.8)
82
83 hclust_clusters <- cutree(hclust_result, k = 2)
84
85 faithful$hcluster <- as.factor(hclust_clusters)
86
87 ggplot(faithful, aes(x = eruptions, y = waiting, color = hcluster)) +
88   geom_point(size = 3) +
```

74:1 | Hierarchical Clustering | R Script

Environment | History | Connections | Tutorial

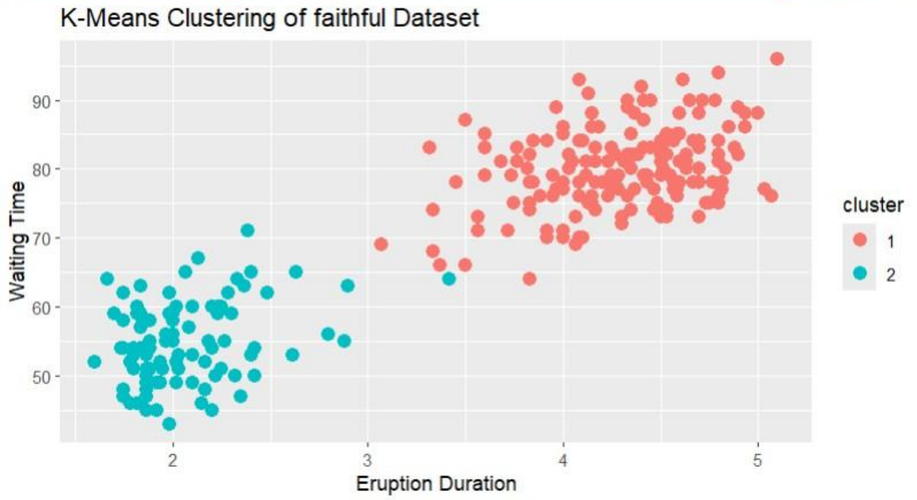
Console | Terminal | Background Jobs

R 4.4.1 · ~/

```
> set.seed(123)
> wss <- sapply(1:10, function(k) {
+   kmeans(faithful_normalized, centers = k, nstart = 10)$tot.withinss
+ })
> plot(1:10, wss, type = "b", pch = 19, frame = FALSE,
+      xlab = "Number of Clusters (k)",
+      ylab = "Total Within-Cluster Sum of Squares",
+      main = "Elbow Method for Determining k")
> kmeans_result <- kmeans(faithful_normalized, centers = 2, nstart = 10)
> faithful$cluster <- as.factor(kmeans_result$cluster)
> library(ggplot2)
> ggplot(faithful, aes(x = eruptions, y = waiting, color = cluster)) +
+   geom_point(size = 3) +
+   labs(title = "K-Means Clustering of faithful Dataset",
+        x = "Eruption Duration",
+        y = "Waiting Time")
>
```

Files | Plots | Packages | Help | Viewer | Presentation

Zoom | Export | Publish



Search

14:38
20-11-2024

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

file.R | Faithful.R* | train_dataset | test_dataset

Source on Save | Run | Source

```
57
58 plot(1:10, wss, type = "b", pch = 19, frame = FALSE,
59      xlab = "Number of Clusters (k)",
60      ylab = "Total Within-Cluster Sum of Squares",
61      main = "Elbow Method for Determining k")
62
63 kmeans_result <- kmeans(faithful_normalized, centers = 2, nstart = 10)
64
65 faithful$cluster <- as.factor(kmeans_result$cluster)
66
67 library(ggplot2)
68 ggplot(faithful, aes(x = eruptions, y = waiting, color = cluster)) +
69   geom_point(size = 3) +
70   labs(title = "K-Means Clustering of faithful Dataset",
71        x = "Eruption Duration",
72        y = "Waiting Time")
73
74 #  ------Hierarchical Clustering------
75
76 distance_matrix <- dist(faithful_normalized)
77
78 hclust_result <- hclust(distance_matrix, method = "ward.D2")
79
80 plot(hclust_result, main = "Hierarchical Clustering Dendrogram",
81      xlab = "", sub = "", cex = 0.8)
82
83 hclust_clusters <- cutree(hclust_result, k = 2)
84
85 faithful$hcluster <- as.factor(hclust_clusters)
86
87 ggplot(faithful, aes(x = eruptions, y = waiting, color = hcluster)) +
88   geom_point(size = 3) +
89   labs(title = "Hierarchical Clustering of faithful Dataset",
90        x = "Eruption Duration",
91        y = "Waiting Time")
92
93 # Comparing the cluster assignments from K-Means and Hierarchical Clustering
94 table(KMeans = faithful$cluster, Hierarchical = faithful$hcluster)
95
96
```

85:1 | Hierarchical Clustering | R Script

Environment | History | Connections | Tutorial

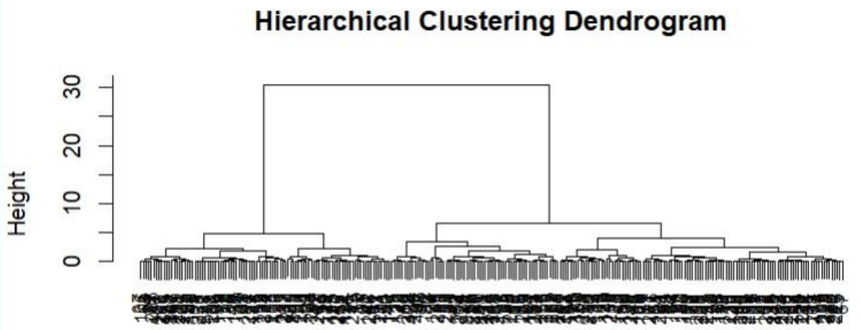
Console | Terminal | Background Jobs

R 4.4.1 · ~/

```
+       xlab = "Number of Clusters (k)",
+       ylab = "Total Within-Cluster Sum of Squares",
+       main = "Elbow Method for Determining k")
> kmeans_result <- kmeans(faithful_normalized, centers = 2, nstart = 10)
> faithful$cluster <- as.factor(kmeans_result$cluster)
> library(ggplot2)
> ggplot(faithful, aes(x = eruptions, y = waiting, color = cluster)) +
+   geom_point(size = 3) +
+   labs(title = "K-Means Clustering of faithful Dataset",
+        x = "Eruption Duration",
+        y = "Waiting Time")
> distance_matrix <- dist(faithful_normalized)
> hclust_result <- hclust(distance_matrix, method = "ward.D2")
> plot(hclust_result, main = "Hierarchical Clustering Dendrogram",
+      xlab = "", sub = "", cex = 0.8)
> hclust_clusters <- cutree(hclust_result, k = 2)
>
```

Files | Plots | Packages | Help | Viewer | Presentation

Zoom | Export | Publish



Search | 14:39 20-11-2024

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

file.R | Faithful.R* | train_dataset | test_dataset

Source on Save | Run | Source

```
plot(1:10, wss, type = "b", pch = 19, frame = FALSE,
     xlab = "Number of Clusters (k)",
     ylab = "Total Within-Cluster Sum of Squares",
     main = "Elbow Method for Determining k")

kmeans_result <- kmeans(faithful_normalized, centers = 2, nstart = 10)

faithful$cluster <- as.factor(kmeans_result$cluster)

library(ggplot2)
ggplot(faithful, aes(x = eruptions, y = waiting, color = cluster)) +
  geom_point(size = 3) +
  labs(title = "K-Means Clustering of faithful Dataset",
       x = "Eruption Duration",
       y = "Waiting Time")

#  ------Hierarchical Clustering------

distance_matrix <- dist(faithful_normalized)

hclust_result <- hclust(distance_matrix, method = "ward.D2")

plot(hclust_result, main = "Hierarchical Clustering Dendrogram",
     xlab = "", sub = "", cex = 0.8)

hclust_clusters <- cutree(hclust_result, k = 2)

faithful$hcluster <- as.factor(hclust_clusters)

ggplot(faithful, aes(x = eruptions, y = waiting, color = hcluster)) +
  geom_point(size = 3) +
  labs(title = "Hierarchical Clustering of faithful Dataset",
       x = "Eruption Duration",
       y = "Waiting Time")

# Comparing the cluster assignments from K-Means and Hierarchical Clustering
table(KMeans = faithful$cluster, Hierarchical = faithful$hcluster)
```

95:1 | Hierarchical Clustering | R Script

Environment | History | Connections | Tutorial

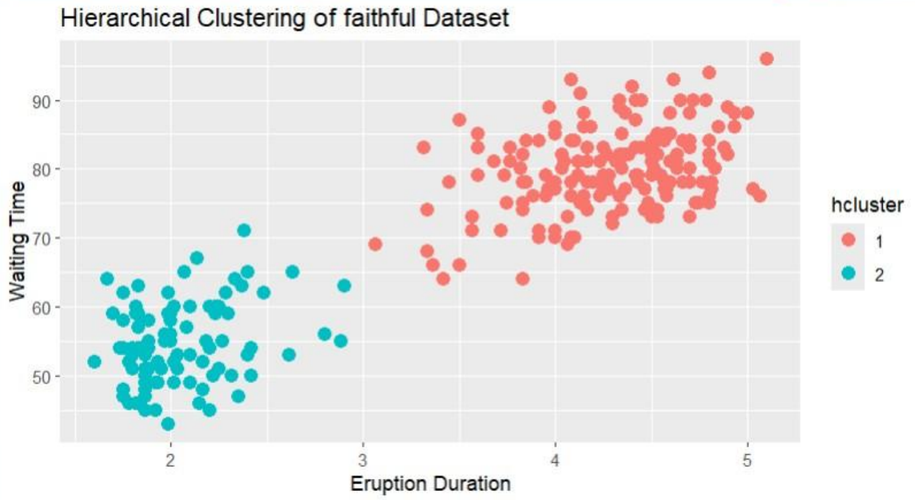
Console | Terminal | Background Jobs

R 4.4.1 · ~/

```
> plot(hclust_result, main = "Hierarchical Clustering Dendrogram",
+      xlab = "", sub = "", cex = 0.8)
> hclust_clusters <- cutree(hclust_result, k = 2)
> faithful$hcluster <- as.factor(hclust_clusters)
> ggplot(faithful, aes(x = eruptions, y = waiting, color = hcluster)) +
+   geom_point(size = 3) +
+   labs(title = "Hierarchical Clustering of faithful Dataset",
+        x = "Eruption Duration",
+        y = "Waiting Time")
> # Comparing the cluster assignments from K-Means and Hierarchical Clusterin
g
> table(KMeans = faithful$cluster, Hierarchical = faithful$hcluster)
      Hierarchical
KMeans   1   2
     1 174   0
     2   1  97
>
```

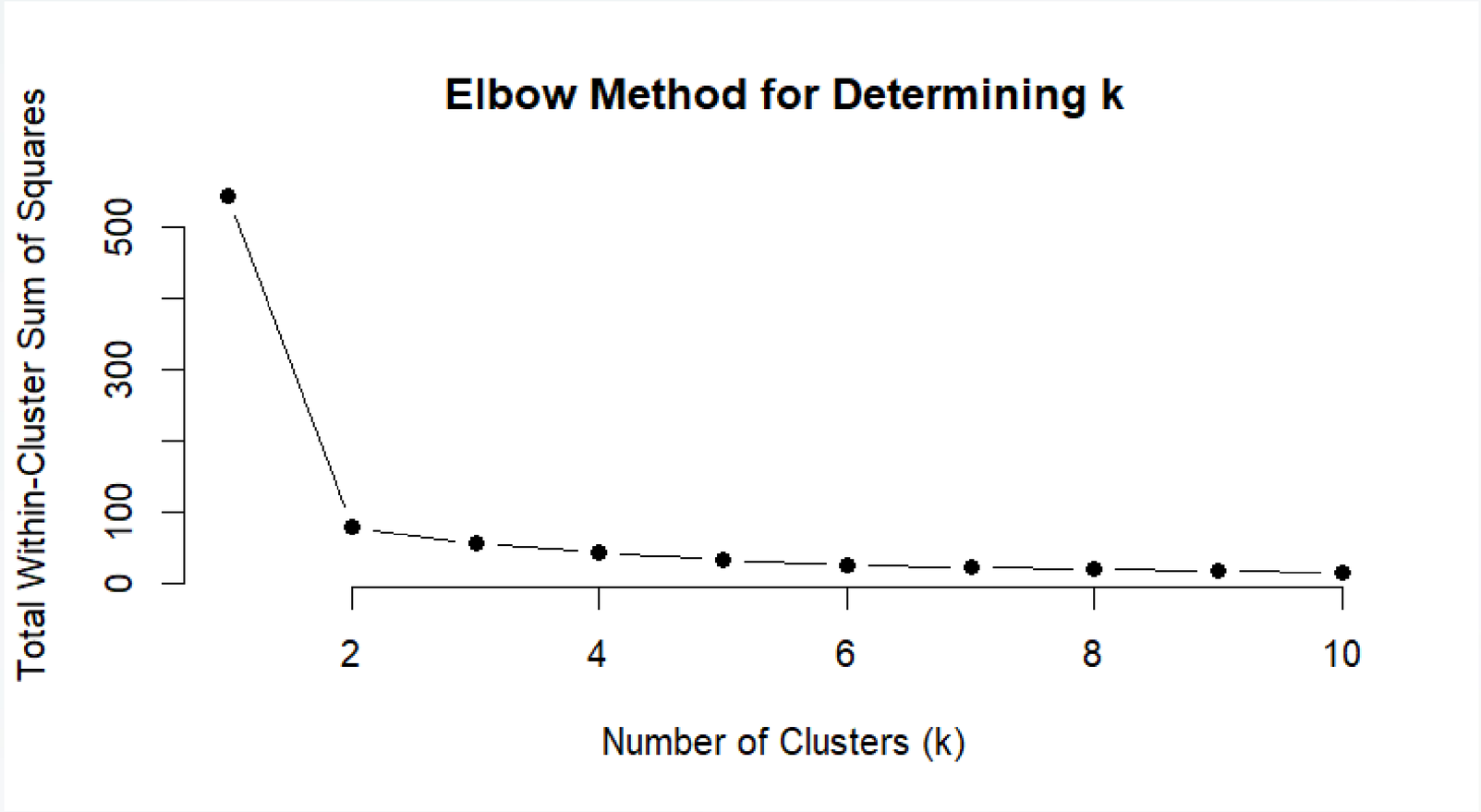
Files | Plots | Packages | Help | Viewer | Presentation

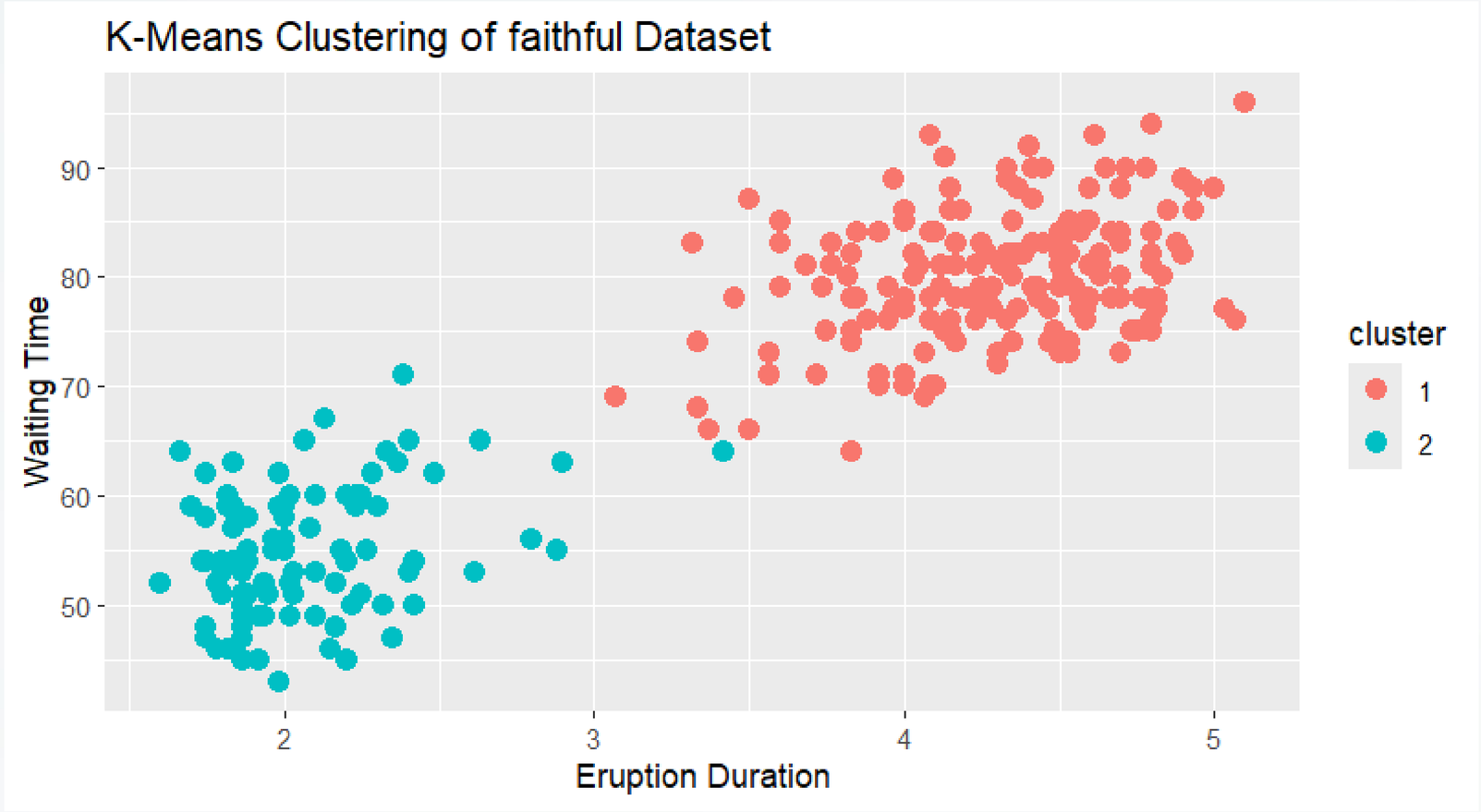
Zoom | Export | Publish

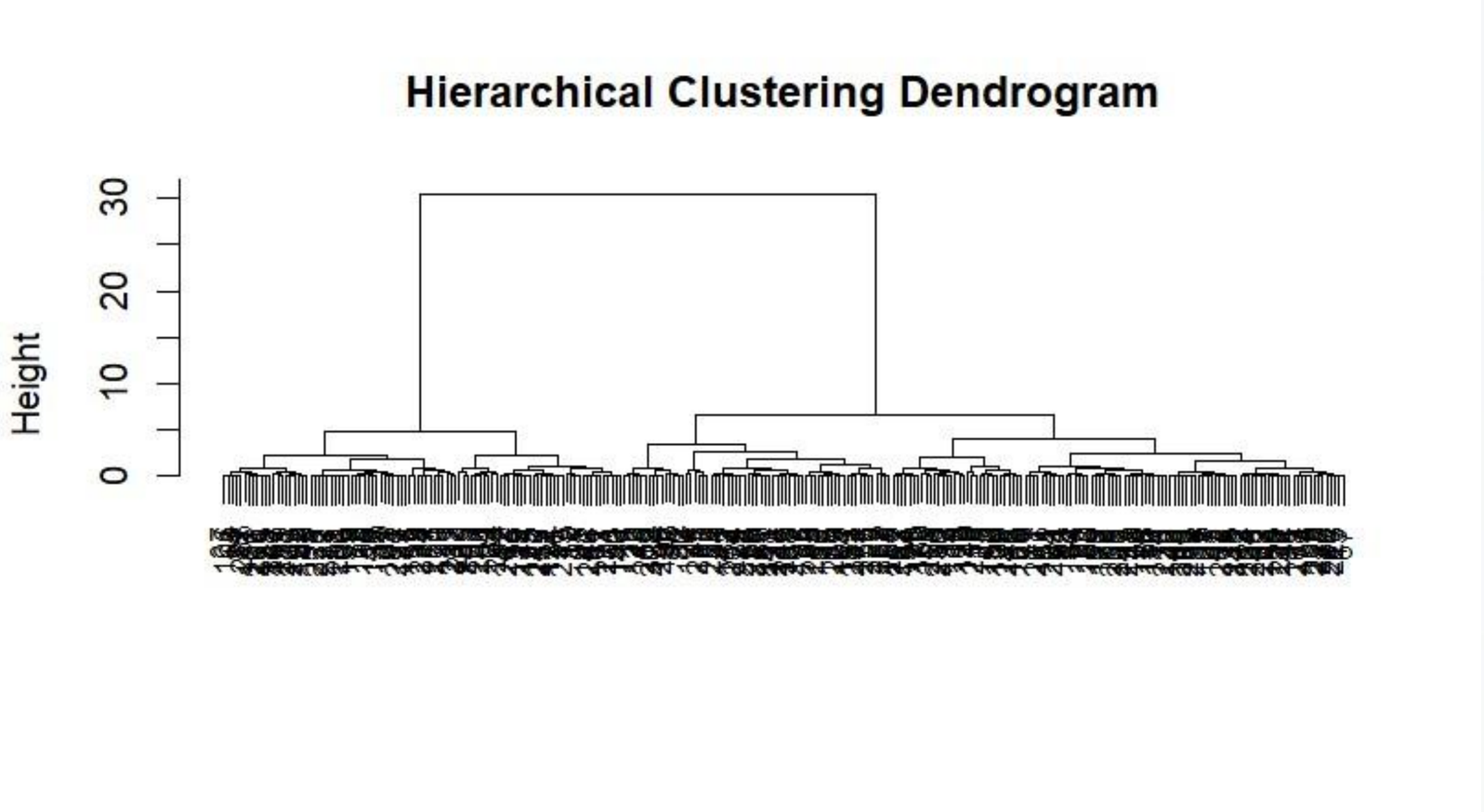


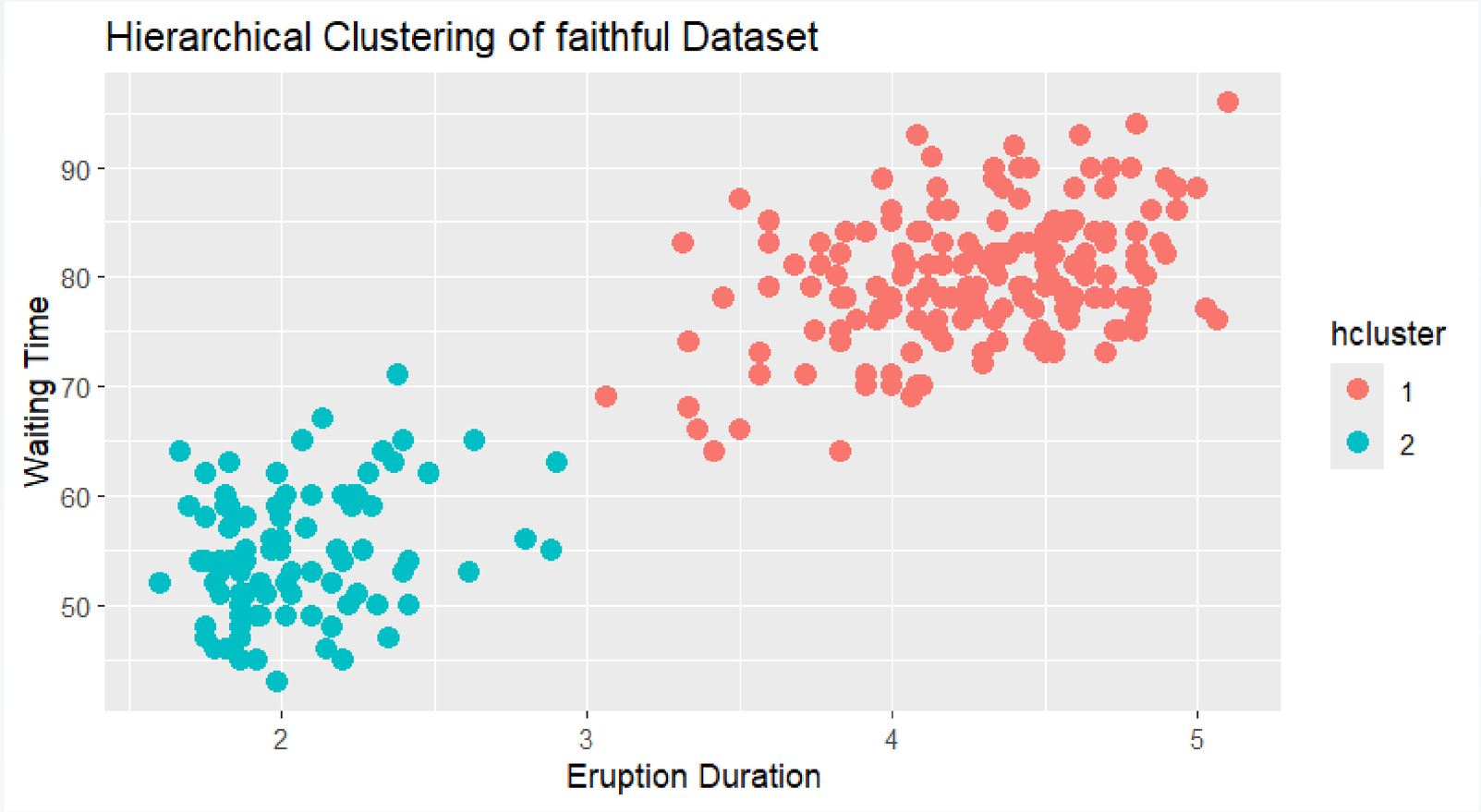
Search

14:40
20-11-2024


Elbow Method for Determining k
Total Within-Cluster Sum of Squares
0
100
300
500
2
4
6
8
10
Number of Clusters (k)


K-Means Clustering of faithful Dataset
Waiting Time
90
80
70
60
50
2
3
4
5
Eruption Duration
cluster
1
2


Hierarchical Clustering Dendrogram
Height
0
10
20
30


Hierarchical Clustering of faithful Dataset
Waiting Time
90
80
70
60
50
2
3
4
5
Eruption Duration
hcluster
1
2

SUBMITTED BY

MITESH
PANDA

Everest
Cantu

Ceo Of Ingoude
Company