

**COMSATS UNIVERSITY ISLAMABAD, ABBOTTABAD**

Introduction to data science

Assignment # 02

***Submitted by:***

Laiba Binta Tahir FA21-BSE-019

***Submitted to:***

Dr. Ghulam Mujtaba

Table of Contents

[**Question – 01** 3](#_Toc186024378)

[Introduction 3](#_Toc186024379)

[Code In R 3](#_Toc186024380)

[Decision Tree Classifier (A) 3](#_Toc186024381)

[Naive Bayes Classifier (B) 4](#_Toc186024382)

[KNN (K-Nearest Neighbors) Classifier (C) 6](#_Toc186024383)

[Discussion 7](#_Toc186024384)

[Solution Recommendations 8](#_Toc186024385)

[Conclusion 8](#_Toc186024386)

[**Question-02** 9](#_Toc186024387)

[Explanation: 9](#_Toc186024388)

[Code in R: 9](#_Toc186024389)

[Result: 10](#_Toc186024390)

# **Question – 01**

For the Drug200 dataset, build the following classification models: A. Decision Tree classifier B. NaiveBayes Classifier C. KNN Classifier For each classifier, evaluate the model on 30% test instances and 70 % training instances. You need to submit the code and confusion matrix of the results for each classifier and the accuracy.

## Introduction

In this task, we are required to build and evaluate three different classification models on the **Drug200 dataset**. The goal is to predict the type of drug (drugA, drugB, drugC, drugX, or drugY) prescribed to patients based on various attributes, such as age, blood pressure (BP), cholesterol level, and sodium-to-potassium ratio (Na\_to\_K).

Each classifier will be evaluated on a split dataset with 30% used for testing and 70% used for training. The models will be evaluated using a confusion matrix to assess performance, and accuracy will be the primary metric to judge how well the model performs.

## Code In R

### Decision Tree Classifier (A)

# Load necessary libraries for Decision Tree

library(rpart)

library(rpart.plot)

library(caret)

# Load the dataset

data <- read.csv("C:/Users/Laiba Binta Tahir/Desktop/drug.csv")

data$Sex <- as.factor(data$Sex)

data$BP <- as.factor(data$BP)

data$Cholesterol <- as.factor(data$Cholesterol)

data$Drug <- as.factor(data$Drug)

# Split the dataset (70% training, 30% testing)

set.seed(42)

trainIndex <- createDataPartition(data$Drug, p = 0.7, list = FALSE)

trainData <- data[trainIndex, ]

testData <- data[-trainIndex, ]

# Train Decision Tree model

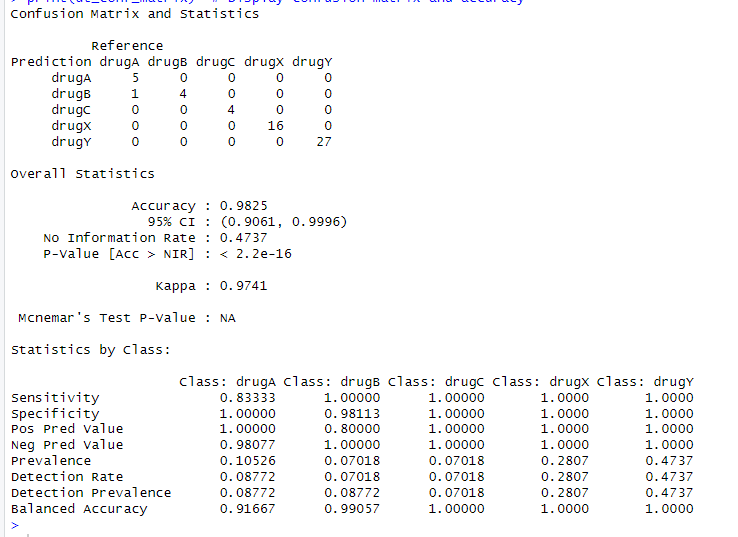
dt\_model <- rpart(Drug ~ ., data = trainData, method = "class")

rpart.plot(dt\_model) # Visualize the decision tree

# Predictions and evaluation

dt\_predictions <- predict(dt\_model, testData, type = "class")

dt\_conf\_matrix <- confusionMatrix(dt\_predictions, testData$Drug)

print(dt\_conf\_matrix) # Display confusion matrix and accuracy

### Naive Bayes Classifier (B)

# Load necessary libraries for Naive Bayes

library(e1071)

library(caret)

# Load the dataset (same preprocessing as above)

data <- read.csv("C:/Users/Laiba Binta Tahir/Desktop/drug.csv")

data$Sex <- as.factor(data$Sex)

data$BP <- as.factor(data$BP)

data$Cholesterol <- as.factor(data$Cholesterol)

data$Drug <- as.factor(data$Drug)

# Split the dataset (70% training, 30% testing)

set.seed(42)

trainIndex <- createDataPartition(data$Drug, p = 0.7, list = FALSE)

trainData <- data[trainIndex, ]

testData <- data[-trainIndex, ]

# Train Naive Bayes model

nb\_model <- naiveBayes(Drug ~ ., data = trainData)

# Predictions and evaluation

nb\_predictions <- predict(nb\_model, testData)

nb\_conf\_matrix <- confusionMatrix(nb\_predictions, testData$Drug)

print(nb\_conf\_matrix) # Display confusion matrix and accuracyA screenshot of a computer

Description automatically generated

### KNN (K-Nearest Neighbors) Classifier (C)

# Load necessary libraries for KNN

library(class)

library(caret)

# Load the dataset (same preprocessing as above)

data <- read.csv("C:/Users/Laiba Binta Tahir/Desktop/drug200.csv")

data$Sex <- as.factor(data$Sex)

data$BP <- as.factor(data$BP)

data$Cholesterol <- as.factor(data$Cholesterol)

data$Drug <- as.factor(data$Drug)

# Split the dataset (70% training, 30% testing)

set.seed(42)

trainIndex <- createDataPartition(data$Drug, p = 0.7, list = FALSE)

trainData <- data[trainIndex, ]

testData <- data[-trainIndex, ]

# Normalize numerical columns (Age and Na\_to\_K)

cols\_to\_normalize <- c("Age", "Na\_to\_K")

trainData[cols\_to\_normalize] <- scale(trainData[cols\_to\_normalize])

testData[cols\_to\_normalize] <- scale(testData[cols\_to\_normalize],

center = attr(scale(trainData[cols\_to\_normalize]), "scaled:center"),

scale = attr(scale(trainData[cols\_to\_normalize]), "scaled:scale"))

# Ensure data is numeric for KNN

trainData\_knn <- trainData[, c("Age", "Na\_to\_K")]

testData\_knn <- testData[, c("Age", "Na\_to\_K")]

cl <- trainData$Drug

# Train and evaluate KNN (k = 5)

knn\_predictions <- knn(train = trainData\_knn, test = testData\_knn, cl = cl, k = 5)

knn\_conf\_matrix <- confusionMatrix(knn\_predictions, testData$Drug)

print(knn\_conf\_matrix) # Display confusion matrix and accuracy

A screenshot of a computer

Description automatically generated

## Discussion

**Accuracy Comparison**: The accuracy across the models may vary depending on the nature of the dataset. **Decision Tree** model tends to perform well on structured data with clear decision boundaries, the **Naive Bayes** model may struggle if the assumption of feature independence does not hold. **KNN**, on the other hand, requires well-normalized data, and its performance can drop if the dataset has noisy data or irrelevant features.

**Confusion Matrix Insights**: Each model generated a confusion matrix, which shows the true positives, false positives, true negatives, and false negatives for each class (drug).

* + **For Decision Trees**: Misclassifications typically occur when the decision boundaries are not clear, which could be the case if a drug is influenced by multiple features.
  + **For Naive Bayes**: Misclassifications may occur due to the assumption of conditional independence between features.
  + **For KNN**: Misclassifications may arise from noisy features or if the wrong value of k is chosen.

## Recommendations

1. **Decision Tree**: Applying pruning can avoid overfitting and tune hyperparameters for better performance.
2. **Naive Bayes**: Feature selection or transformation techniques can improve performance when the independence assumption isn’t valid.
3. **KNN**: Normalize features and experiment with different values of k to find the optimal one.

## Conclusion

In summary, each model (Decision Tree, Naive Bayes, KNN) has its strengths and weaknesses. The key to improving the performance of these models lies in proper data preprocessing (normalization, handling missing data), hyperparameter tuning, and careful feature engineering. By understanding the unique characteristics of each model, we can optimize their performance and gain better insights into the classification task for the Drug200 dataset.

# **Question-02**

Write R functions to answer these questions about the dataset. Please write the sequence of functions in a pipeline with your answer: 1. Calculate the average age of males who used drug C. 2. Females with HIGH BP and HIGH cholesterol are treated with what drugs? 3. Find the average Na-K of all High BP patients , and average Na-K of all low BP patients and average Na-K of all Normal BP patients.

## Explanation:

1. Average Age of Males Using Drug C:
   * I have used filter(Sex == "M", Drug == "drugC") to select males who used drug C.
   * Then, used summarise(Average\_Age = mean(Age, na.rm = TRUE)) to calculate the average age.
   * The result is extracted using pull(Average\_Age).
2. Drugs for Females with High BP and High Cholesterol:
   * The filter(Sex == "F", BP == "HIGH", Cholesterol == "HIGH") selects females with high BP and cholesterol.
   * select(Drug) selects the drug column, and distinct() ensures that we get unique drug names.
3. Average Na-K by BP Categories (High, Low, and Normal):
   * group by BP using group\_by(BP) to separate the data by BP category.
   * summarise(Average\_Na\_to\_K = mean(Na\_to\_K, na.rm = TRUE)) calculates the average Na-K for each BP group.
   * Finally, arrange(BP) sorts the results by BP categories.

## Code in R:

# Load necessary libraries

library(dplyr)

# Read the dataset

data <- read.csv("C:/Users/Laiba Binta Tahir/Desktop/drug.csv")

**# 1. Calculate the average age of males who used drug C**

average\_age\_male\_drugC <- data %>%

filter(Sex == "M", Drug == "drugC") %>% # Filter for males who used drug C

summarise(Average\_Age = mean(Age, na.rm = TRUE)) %>% # Calculate average age

pull(Average\_Age) # Extract the average age value

# Print the result for question 1

cat("\nAverage age of males who used drug C: ", average\_age\_male\_drugC, "\n")

**# 2. Find drugs for females with HIGH BP and HIGH cholesterol**

drugs\_for\_females\_high\_bp\_cholesterol <- data %>%

filter(Sex == "F", BP == "HIGH", Cholesterol == "HIGH") %>% # Filter for females with high BP and cholesterol

select(Drug) %>% # Select the 'Drug' column

distinct() # Get unique drug names

# Print the result for question 2

cat("\nDrugs for females with HIGH BP and HIGH cholesterol:\n")

print(drugs\_for\_females\_high\_bp\_cholesterol)

**# 3. Find the average Na-K of all High BP, Low BP, and Normal BP patients**

average\_na\_k\_by\_bp <- data %>%

group\_by(BP) %>% # Group by BP category

summarise(Average\_Na\_to\_K = mean(Na\_to\_K, na.rm = TRUE)) %>% # Calculate the average Na-K for each group

arrange(BP) # Arrange results in BP order

# Print the result for question 3

cat("\nAverage Na-K levels by BP:\n")

print(average\_na\_k\_by\_bp)

## Result:

Each question is processed using the dplyr functions in a pipeline, making the code concise and efficient.

A screenshot of a computer

Description automatically generated