

# DATA SCIENCE AND MACHINE LEARNING (MSc)

## DAMA51: Foundations in Computer Science

Academic Year: 2022–2023

### #5 Written Assignment

Submission Deadline	<u>Wed, 24 May 2023, 11:59 PM</u>
---------------------	-----------------------------------

### Remarks

The deadline is definitive.

An indicative solution will be posted online along with the return of the graded assignments.

The assignment is due via the STUDY submission system. **You are expected to turn in a document (.DOC, .ODT, .PDF) and a compressed (.ZIP, .RAR) file containing all your work:**

- 1 document file (this document) with the answers to all the questions, along with the R code (where required) and the results of the execution of the code
- 1 compressed file with 3 R scripts with the code that answers each one of the problems to Topics 3 (3d), 4, and 5(5d).

**You should not make any changes in the written assignment file other than providing your own answers.** You should also type all of your answers into Word and not attach any handwritten notes as pictures into your work otherwise a 5% reduction of your final grade will be applied. Make sure to name all the files (ZIP file, DOC file and R script files) with **your last name first followed by a dash symbol and the names of each component at the end**. For example, for the student with the last name Aggelou the files should be named as follows: Aggelou-HW5.zip, Aggelou-HW5.doc , Aggelou-Topic3.R, Aggelou-Topic4.R, and Aggelou-Topic5.R. The R script files should automatically run with the **source** command and generate the correct results. Also, please include comments before each command to explain the functionality of the command that follows. In the computations, use **four decimal places**.

Topic	Points	Grades
1. Online Quiz	40	
2. Article review	5	
3. Bayes Classifier	20	
4. Linear Regression	20	
5. Decision Trees	20	
TOTAL	105 (max 100)	/100

## Topic 1: Online Quiz

**(40 points)** Complete the corresponding online quiz available at:

<https://study.eap.gr/mod/quiz/view.php?id=25614>

You have one effort and unlimited time to complete the quiz, up to the submission deadline.

## Topic 2: Article Review

The article "Round Robin Classification" (<https://www.jmlr.org/papers/v2/fuernkranz02a.html>) suggests that one could frame a multi-class classification problem as a voting-and-ranking problem. Briefly describe the proposed framing, including an example of a  $k$ -class problem.

Note: You should write up your answer to a maximum of 100 words. Any text in excess of 100 words will not be taken into consideration.

**(5 points)**

The paper "Round Robin Classification" proposes a novel approach to handle multi-class classification problems by reframing them as a voting-and-ranking challenge. It involves transforming the original problem into a series of binary classification tasks, where each pair of classes is treated as a separate problem. For instance, in a 3-class scenario with classes A, B, and C, three binary classifiers are trained: A vs. B, A vs. C, and B vs. C. The final prediction is obtained by aggregating the outputs of these classifiers using voting and ranking techniques. This methodology enhances accuracy while maintaining computational efficiency.

## Topic 3: Bayes Classifier

**(20 points)** We shall use the standard *weather* dataset shown in the table below, which has a total of 14 training examples of the Play Tennis task concept, where each day is described by features *Outlook*, *Temperature*, *Humidity*, *Wind*, and the class *PlayTennis*.

<i>Day</i>	<i>Outlook</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Wind</i>	<i>PlayTennis</i>
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No

D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Use the dataset and apply the Naïve Bayes classification to answer the following.

**(a) (2 points)** Calculate using pen and paper the following probabilities:

**Answer :**

$$P(\text{PlayTennis} = \text{Yes}) = 9/14 = 0.643$$

$$P(\text{PlayTennis} = \text{No}) = 5/14 = 0.357$$

**(b) (2 points)** Calculate using pen and paper the following conditional probabilities:

**Answer:**

$$P(\text{Wind} = \text{Strong} \mid \text{PlayTennis} = \text{Yes}) = 3/9 = 0.333$$

$$P(\text{Wind} = \text{Strong} \mid \text{PlayTennis} = \text{No}) = 3/5 = 0.6$$

**(c) (6 points)** Using pen and paper, calculate what a Naïve Bayes classifier would predict for the following test instance:

Outlook = Sunny, Temperature = Cool, Humidity = High, Wind = Strong

## Answer:

```
P(PlayTennis = Yes)*P(Outlook = Sunny | PlayTennis = Yes)*P(Temperature =
Cool | PlayTennis = Yes) * P(Humidity = High | PlayTennis = Yes) * P
(Wind = Strong | PlayTennis = Yes) = 0.643 * 2/9 * 3/4 * 1/3 * 1/9 =
0.0039691358
```

```
P(PlayTennis = No)*P(Outlook = Sunny | PlayTennis = No)*P(Temperature =
Cool | PlayTennis = No) * P(Humidity = High | PlayTennis = No) * P (Wind
= Strong | PlayTennis = No) = 0.357 * 3/5 * 1/2 * 1/2 * 2/5 =
0.02142
```

Since  $0.02142 > 0.0039691358$  the Naïve Bayes classifier predicts that  $PlayTennis = No$

**(d) (10 points)** By filling in the missing code in the R script below build a Naïve Bayes classifier using the **provided** file CLASS\_data.csv that will predict the outcome (*PlayTennis*) of the two following instances:

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D15	Overcast	Mild	Normal	Weak	? -> Yes
D16	Sunny	Mild	High	Strong	? -> No

Provide your R-script below:

```
install.packages('naivebayes')  
library(naivebayes)
```

```
class_data<-read.csv("CLASS_data.csv",head = TRUE, sep = ",")  
classifier <- naive_bayes(PlayTennis ~ ., data = class_data)
```

```
i. test1 <- <Fill in your code here>  
prediction <- predict(classifier, test1, type="prob")  
prediction
```

```
i. test2 <- <Fill in your code here>  
prediction <- predict(classifier, test2, type="prob")  
prediction
```

```
install.packages('naivebayes')  
library(naivebayes)
```

```
class_data<-read.csv("C:\\Users\\antonisk\\Desktop\\DAMA\\DAMA51\\DAMA51 - 5th  
Assignment\\CLASS_data.csv",head = TRUE, sep = ",")  
classifier <- naive_bayes(PlayTennis ~ ., data = class_data)
```

```
test1 <- data.frame(Outlook = "Overcast", Temperature = "Mild", Humidity = "Normal", Wind =  
"Weak")  
prediction1 <- predict(classifier, test1, type="prob")  
prediction1
```

```
test2 <- data.frame(Outlook = "Sunny", Temperature = "Mild", Humidity = "High", Wind =  
"Strong")  
prediction2 <- predict(classifier, test2, type="prob")  
prediction2
```

## Topic 4: Linear Regression

(20 total points)

(a) (6 total points) Write an R script to calculate analytically the parameters  $(a, b)$  of the linear regression model  $f(x) = a + b * x$  by **using the normal equations** given in the lecture (TM5). Use the `Auto` dataset from the `ISLR` package to perform linear regression with `mpg` as the response and `displacement` as the predictor. (hint: To solve the linear system use the `solve()` function).

Answer:

Normal equations

$$na + \left( \sum_{i=1}^n x_i \right) b = \sum_{i=1}^n y_i$$
$$\left( \sum_{i=1}^n x_i \right) a + \left( \sum_{i=1}^n x_i^2 \right) b = \left( \sum_{i=1}^n x_i y_i \right)$$

R script

```
install.packages('ISLR')  
library(ISLR)  
attach(Auto)
```

<Fill in the gap with your code here>

```
solve(A,b)
```

Hence  $a = \dots$  and  $b = \dots$

```
install.packages('ISLR')  
library(ISLR)  
data(Auto)  
df <- data.frame(Auto$displacement, Auto$mpg)  
names(df) <- c("x", "y")  
  
n <- nrow(df)  
x_sum <- sum(df$x)
```

```

y_sum <- sum(df$y)
xy_sum <- sum(df$x * df$y)
x_sq_sum <- sum(df$x^2)

A <- matrix(c(n, x_sum, x_sum, x_sq_sum), nrow = 2)
b <- matrix(c(y_sum, xy_sum), nrow = 2, ncol = 1)

param <- solve(A, b)
a <- param[1]
b <- param[2]

a
b

```

**(b) (5 total points)** Find the slope (parameter  $\beta$ ) of the regression line in the form  $y = \beta x$  by using the method of least squares which best fits to the points  $(x_i, y_i)$ ,  $i = 1, 2, \dots, 8$  that are given below.

$i$	1	2	3	4	5	6	7	8
$x_i$	30	20	60	80	40	50	70	90
$y_i$	75	52	120	170	86	110	153	194

**Answer :**

$$\beta = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2} = \frac{?}{?} = \frac{61200}{28400} = 2.15493$$

*R code:*

```

x <- c(30, 20, 60, 80, 40, 50, 70, 90)
y <- c(75, 52, 120, 170, 86, 110, 153, 194)

g <- sum(x*y)
k <- sum(x^2)
beta <- sum(x*y) / sum(x^2)

```

g  
k  
beta

**(c) (4 points)** A farmer is interested in determining how the amount  $X$  of fertilizer applied to a plot of land affects the yield  $Y$  of the farm. It is therefore experimented with  $n=10$  similar plots (of the same area, in areas with similar climatic conditions) so that any differences observed in the production of the fields are mainly due to the different amounts of fertilizer used. The below table gives the production  $Y$  (in thousands of kgr) for  $n=10$  identical plots as well as the quantity  $X$  of the fertilizer used in each one (in hundreds of kgr). Using pen and paper, calculate analytically the parameters  $(\beta_0, \beta_1)$  of the linear regression model  $f(x) = \beta_0 + \beta_1 * x$ .

$i$	$x_i$	$y_i$
1	20	706
2	10	550
3	26	790
4	8	517
5	20	694
6	16	634
7	20	715
8	12	571
9	8	529
10	24	754

**Answer :**

$$\beta_1 = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} = 400$$

$$\beta_0 = \frac{1}{n} \sum_{i=1}^n y_i - \beta_1 \cdot \frac{1}{n} \cdot \sum_{i=1}^n x_i = 15$$

```
x <- c(20, 10, 26, 8, 20, 16, 20, 12, 8, 24)
y <- c(706, 550, 790, 517, 694, 634, 715, 571, 529, 754)
n <- length(x)
```



```
beta1 <- (n * sum(x*y) - sum(x) * sum(y)) / (n * sum(x^2) - sum(x)^2)
beta0 <- mean(y) - beta1 * mean(x)
```

```
beta0
beta1
```

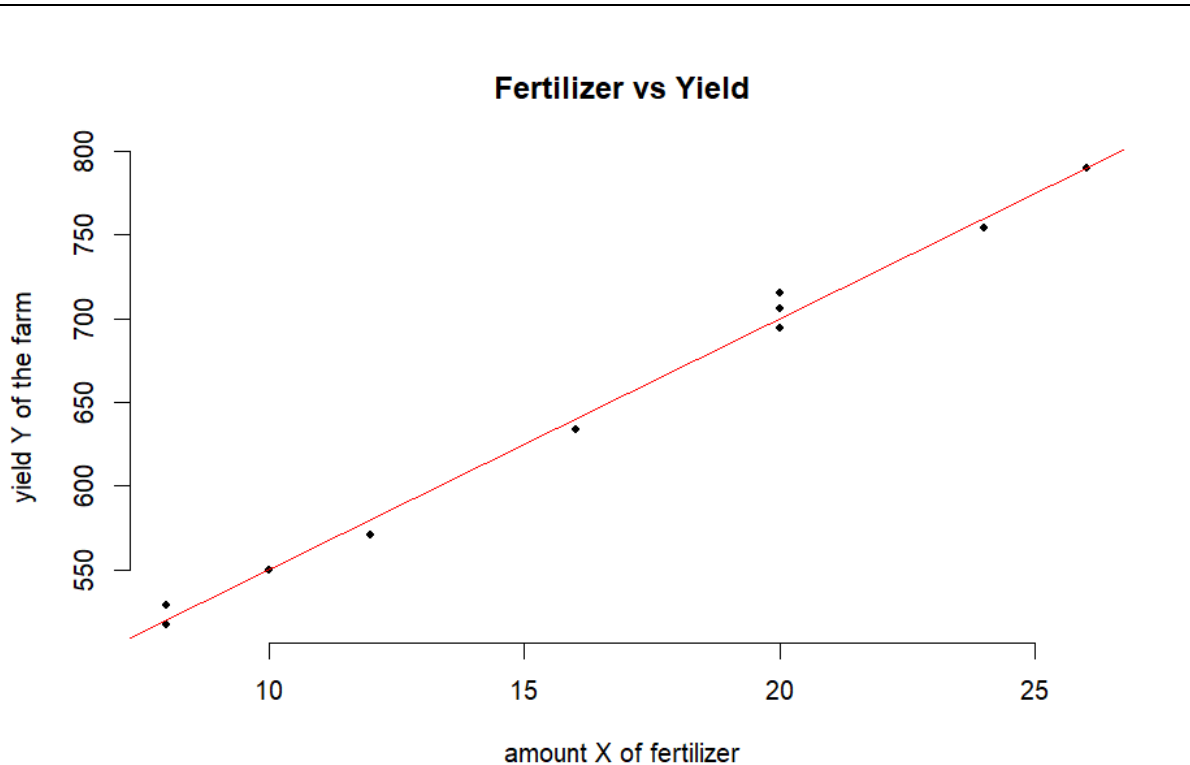
**(d) (5 points)** By filling in the missing code in the R script below provide the scatter plot of the dataset given in question (c) along with the corresponding regression line. Write a short conclusion about the predictor.

**Answer:**

Code(4 points)

```
x <- <Fill in your code here>
y <- <Fill in your code here>

# Plot with main and axis titles
plot(?, ?, main = "Fertilizer vs Yield ",
      xlab = "amount X of fertilizer ", ylab = "yield Y of the
farm",
      pch = 20, frame = FALSE)
# Add regression line
plot(?, ?, main = "Fertilizer vs Yield ",
      xlab = "amount X of fertilizer", ylab = "yield Y of the
farm",
      pch = 15, frame = FALSE)
abline(<Fill in your code here>)
```



conclusion (1 point):

There seems to be a positive linear relationship between the amount of fertilizer applied and the yield of the farm. The yield seems to increase with the amount of fertilizer applied. That is not definite because the data is not perfectly linear so other factors might affect the yield.

## Topic 5: Decision Trees

**(20 total points)** In this topic, you will implement a Decision Tree algorithm based on the `Tennis.csv` dataset. This dataset consists of a header row, followed by 11 rows of training data as shown below:

	Outlook	Temperature	Humidity	Play
1	rainy	cool	normal	no
2	rainy	cool	high	no
3	sunny	hot	high	no
4	rainy	mild	high	no
5	sunny	mild	high	no
6	rainy	cool	normal	no
7	rainy	mild	normal	yes
8	rainy	hot	high	no
9	rainy	hot	normal	yes
10	sunny	mild	normal	yes
11	sunny	cool	high	no

The `.csv` file contains four categorical attributes: *Play*, *Outlook*, *Temperature*, and *Humidity*. *Play* would be the output variable (or the predicted class), and *Outlook*, *Temperature* and *Humidity* would be the input variables.

(a) (5 points) Calculate by hand the entropy of the class after the dataset has been split according to the values of `Outlook` and provide the result and the calculations in the following spaces, respectively.

$$H(\text{Play}, \text{Outlook}) = (w_1 * \text{Entropy}_1) + (w_2 * \text{Entropy}_2) = (0.545 * 0.9183) + (0.455 * 0.9709) = 0.5005 + 0.4417 = 0.9422$$

### Calculations

#### Rainy

$$\text{Entropy}_1 = -(2/6) * \log_2(2/6) - (4/6) * \log_2(4/6) = -0.333 * \log_2(0.333) - 0.667 * \log_2(0.667) = 0.9183$$

$$w_1 = 6/11 = 0.545$$

#### Sunny

$$\text{Entropy}_2 = -(3/5) * \log_2(3/5) - (2/5) * \log_2(2/5) = -0.6 * \log_2(0.6) - 0.4 * \log_2(0.4) = 0.9709$$

$$w_2 = 5/11 = 0.455$$

(b) (3 points) Calculate by hand the Information Gain if the split is done on `Outlook` and provide the results in the following space.

### Results

$$\text{Entropy}_{\text{before}} = -(5/11) * \log_2(5/11) - (6/11) * \log_2(6/11) = -0.45 * \log_2(0.45) - 0.55 * \log_2(0.55) = 0.994$$

$$\text{Entropy}_{\text{after}} = (6/11) * 0.9183 + (5/11) * 0.9709 = 0.9422$$

$$\text{Inf. Gain (Outlook)} = \text{Entropy before} - \text{Entropy after} = 0.994 - 0.9422 = 0.0518$$

(c) (3 points) How does the root node splits for the given dataset? Justify your answer.

Results:

For Outlook

Entropy before = 0.994

Entropy1 (Rainy) = 0.9183,  $w_1 = 6/11 = 0.545$

Entropy2 (Sunny) = 0.9709,  $w_2 = 5/11 = 0.455$

Entropy after =  $(w_1 * \text{Entropy1}) + (w_2 * \text{Entropy2}) = (0.545 * 0.9183) + (0.455 * 0.9709) = 0.5005 + 0.4417 = 0.9422$

Inf. Gain = Entropy before - Entropy after =  $0.994 - 0.9422 = 0.0518$

For Temperature

Entropy before = 0.994

Entropy1 (Cool) = 0.8113,  $w_1 = 4/11 = 0.364$

Entropy2 (Mild) = 1.0,  $w_2 = 4/11 = 0.364$

Entropy3 (Hot) = 0.0,  $w_3 = 3/11 = 0.273$

Entropy after =  $(w_1 * \text{Entropy1}) + (w_2 * \text{Entropy2}) + (w_3 * \text{Entropy3}) = (0.364 * 0.8113) + (0.364 * 1.0) + (0.273 * 0.0) = 0.2953 + 0.364 + 0 = 0.6593$

Inf. Gain = Entropy before - Entropy after =  $0.994 - 0.6593 = 0.3347$

For Humidity,

Entropy before = 0.994

Entropy1 (Normal) = 0.9852,  $w_1 = 7/11 = 0.636$

Entropy2 (High) = 0.5922,  $w_2 = 4/11 = 0.364$

Entropy after =  $(w_1 * \text{Entropy1}) + (w_2 * \text{Entropy2}) = (0.636 * 0.9852) + (0.364 * 0.5922) = 0.6266 + 0.2157 = 0.8423$

Inf. Gain = Entropy before - Entropy after =  $0.994 - 0.8423 = 0.1517$

Justification:

Based on the information gains calculated above, we can see that Outlook has the highest information gain (0.0518), followed by Humidity (0.1517), and Temperature (0.3347). Therefore, the root node of the decision tree would be Outlook as it provides the most significant reduction in entropy.

**(d) (9 points)** Filling in the missing code of the R script below which implements the ID3 algorithm in the `Tennis.csv` file and provide the deduced decision tree. (The ID3 decision tree algorithm is based on the Information Gain metric in accordance with the previous questions of Topic 5)

**Answer:**

```
p_dec <- read.table("Tennis.csv",header=TRUE,sep=",")
p_dec
install.packages('data.tree')
library('data.tree') # load library

IsPure <- function(data) {
  length(unique(data[,ncol(data)])) == 1
}

Entropy <- function(vls) {
  res <- vls/sum(vls) * log2(vls/sum(vls))
  res[vls == 0] <- 0
  -sum(res)
}

InformationGain <- function(tble) {
  tble <- as.data.frame.matrix(tble)
  entropyBefore <- Entropy(colSums(tble))
  s <- rowSums(tble)
  entropyAfter <- sum(s / sum(s) * apply(tble, MARGIN = 1, FUN = Entropy))
  informationGain <- entropyBefore - entropyAfter
  return(informationGain)
}

TrID3 <- function(node, data) {
  node$obsCount <- nrow(data)
  #if the data-set is pure (e.g. all no), then
  if(IsPure(data)) {
    #a leaf having the name of the pure feature (e.g. 'no')will be constructed
    child <- node$AddChild(unique(data[,ncol(data)]))
    node$feature <- tail(names(data), 1)
    child$obsCount <- nrow(data)
    child$feature <- ''
  } else {
    #the feature with the highest information gain (e.g. 'outlook') will be
    chosen
    ig <- sapply(colnames(data)[-ncol(data)],
```

```
function(x) InformationGain(  
  table(data[,x], data[,ncol(data)])  
)  
)  
  
feature <- names(ig)[ig == max(ig)][1]  
node$feature <- feature  
  
#the subset of the data-set having that feature value will be taken  
childObs <- split(data[,!(names(data) %in% feature)], data[,feature], drop  
= TRUE)  
  
for(i in 1:length(childObs)) {  
  #a child having the name of that feature value (e.g. 'sunny') will be  
  constructed  
  child <- node$AddChild(names(childObs)[i])  
  #the algorithm recursively on the child and the subset will be called  
  TrID3(child, childObs[[i]])  
}  
}
```

tree <- <Fill in your code here>

TrID3(<Fill in your code here>)

print(tree, "feature", "obsCount")

Output:

Plot the tree here

## ANSWER:

```
p_dec <- read.table("C:\\Users\\antonisk\\Desktop\\DAMA\\DAMA51\\DAMA51 - 5th  
Assignment\\Tennis.csv", header = TRUE, sep = ",")
```

```
p_dec
```

```
install.packages('data.tree')
```

```
library('data.tree')
```

```
# Define the ID3 algorithm
```

```
IsPure <- function(data) {
```

```
  length(unique(data[, ncol(data)])) == 1
```

```
}
```

```
Entropy <- function(vls) {
```

```
res <- vls / sum(vls) * log2(vls / sum(vls))
res[vls == 0] <- 0
-sum(res)
}

InformationGain <- function(tble) {
  tble <- as.data.frame.matrix(tble)
  entropyBefore <- Entropy(colSums(tble))
  s <- rowSums(tble)
  entropyAfter <- sum(s / sum(s) * apply(tble, MARGIN = 1, FUN = Entropy))
  informationGain <- entropyBefore - entropyAfter
  return(informationGain)
}

TrID3 <- function(node, data) {
  node$obsCount <- nrow(data)

  if (IsPure(data)) {
    child <- node$AddChild(unique(data[, ncol(data)]))
    node$feature <- tail(names(data), 1)
    child$obsCount <- nrow(data)
    child$feature <- ''
  } else {
    ig <- sapply(
      colnames(data)[-ncol(data)],
      function(x) InformationGain(table(data[, x], data[, ncol(data)]))
    )
    feature <- names(ig)[ig == max(ig)][1]
    node$feature <- feature
    childObs <- split(data[, !(names(data) %in% feature)], data[, feature], drop =
TRUE)
    for (i in 1:length(childObs)) {
      child <- node$AddChild(names(childObs)[i])
      TrID3(child, childObs[[i]])
    }
  }
}

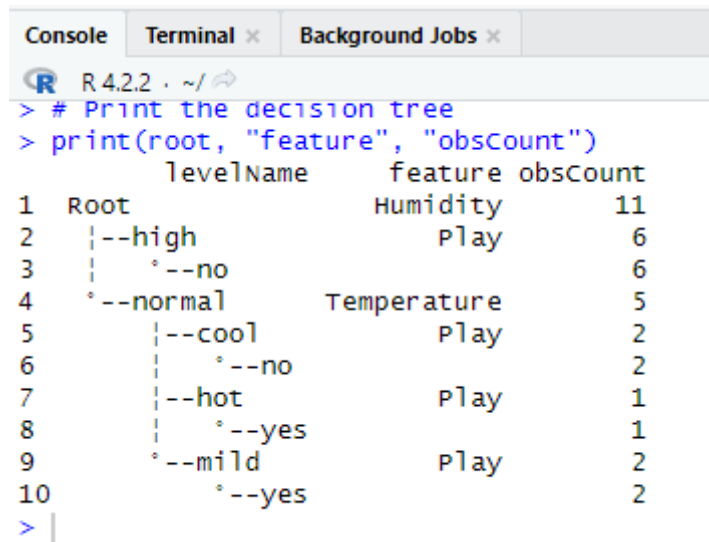
# Create the root node and build the decision tree
root <- Node$new("Root")
```




```
TrID3(root, p_dec)
```

```
# Print the decision tree
```

```
print(root, "feature", "obsCount")
```



The screenshot shows an R console window with the following content:

```
R 4.2.2 . ~/ 
> # Print the decision tree
> print(root, "feature", "obsCount")
      levelName      feature obsCount
1  Root                Humidity      11
2  |--high                Play        6
3  |  |--no                Play        6
4  |  |--normal            Temperature  5
5  |    |--cool            Play        2
6  |      |--no            Play        2
7  |      |--hot            Play        1
8  |      |--yes            Play        1
9  |      |--mild            Play        2
10 |      |--yes            Play        2
> |
```