**Data Science and Machine Learning (MSc)**

DAMA51: Foundations in Computer Science

Academic Year: 2022–2023

|  |  |
| --- | --- |
| #5 Written Assignment | |
| Submission Deadline | Wed, 24 May 2023, 11:59 PM  TRANTEROU VASILEIA |

# 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 |  |
| 1. **Article review** | 5 |  |
| 1. **Bayes Classifier** | 20 |  |
| 1. **Linear Regression** | 20 |  |
| 1. **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](about:blank)

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](about:blank)) 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)**

|  |
| --- |
|  |

# 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,* andthe class *PlayTennis*.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Day* | *Outlook* | *Temperature* | *Humidity* | *Wind* | *PlayTennis* |
| 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*(*PIayTennis* = *Yes*) = 9/14 = 0.642  *P*(*PIayTennis* = *No*) = 5/14 = 0.357 |

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

|  |  |  |
| --- | --- | --- |
| **Answer:**   |  | | --- | | P(Wind = Strong | PlayTennis = Yes) = [(3/14) \* 0.642]/ (6/14) =  = (0.214 \* 0.642) / 0.428 = 0.137/ 0.428 = 0.321 |  |  | | --- | | P (Wind = Strong | PIayTennis = No) = (0.214 \* 0.357)/ 0.428 =  = 0.076 / 0.428 = 0.1785 | |

**(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(*PIayTennis* = Yes)\*P(Outlook = Sunny | PIayTennis = Yes)\*P(Temperature = Cool | *PIayTennis* = Yes) \* P(Humidity = High | *PIayTennis* = Yes) \* P (Wind = Strong | *PIayTennis* = Yes) **= 0.642\* 0.255\* 0.482\* 0.274 \*0.321=**  **=0.006** |  |  | | --- | | P(PIayTennis = No)\*P(Outlook = Sunny | PIayTennis = No)\*P(Temperature = Cool | PIayTennis = No) \* P(Humidity = High | PIayTennis = No) \* P (Wind = Strong | PIayTennis = No) = **0.357\* 0.214\* 0.089\*0.203\*0.1785 = =0.0002** | |

**(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 (0.956) |
| D16 | Sunny | Mild | High | Strong | No (0.755) |

|  |
| --- |
| **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)**   1. **test1 <-** data.frame(Outlook = "Overcast", Temperature = "Mild", Humidity = "Normal", Wind = "Weak")   **prediction <- predict(classifier, test1, type="prob")**  **prediction**   1. **test2 <-** data.frame(Outlook = "Sunny", Temperature = "Mild", Humidity = "High", Wind = "Strong")   **prediction <- predict(classifier, test2, type="prob")**  **prediction** |

# Topic 4: Linear Regression

**(20 total points)**

**(a) (6 total points)** Write an R script to calculate analytically the parameters of the linear regression model 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**  **R script**  **install.packages('ISLR')**  **library(ISLR)**  **attach(Auto)**  **predictor1 <- Auto$displacement**  **predictor2 <- Auto$mpg**  **n <- length(predictor1)**  **A <- cbind(rep(1, n), predictor1)**  **params <- solve(t(A) %\*% A) %\*% t(A) %\*% predictor2**    **a <- params[1]**  **b <- params[2]**  **cat("a =", a, "\n")**  **cat("b =", b, "\n")**  **Hence**  35.12064 **and**  -0.06005143 |

**(b) (5 total points)** Find the slope (parameter β) of the regression line in the form by using the method of least squares which best fits to the points that are given below.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***i*** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| ***xi*** | 30 | 20 | 60 | 80 | 40 | 50 | 70 | 90 |
| ***yi*** | 75 | 52 | 120 | 170 | 86 | 110 | 153 | 194 |

|  |
| --- |
| **Answer:** |

**(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 of the linear regression model .

|  |  |  |
| --- | --- | --- |
| ***i*** | ***xi*** | ***yi*** |
| 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:** | |
| |  | | --- | |  |  |  | | --- | | *β0*  = | |  |
|  |  |

**(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 <- c(20,10,26,8,20,16,20,12,8,24)  y <- c(706,550,790,517,694,634,715,571,529,754)  # Plot with main and axis titles  plot(**x**, **y**, main = "Fertilizer vs Yield ",  xlab = "amount X of fertilizer ", ylab = "yield Y of the farm",  pch = 20, frame = FALSE)  # Add regression line  plot(**x**,**y**, main = "Fertilizer vs Yield ",  xlab = "amount X of fertilizer", ylab = "yield Y of the farm",  pch = 15, frame = FALSE)  model<-lm(y ~ x)  abline(model, col = "red")  conclusion (1 point): It appears that there is a positive linear relationship between the amount of fertilizer used and the yield of the farm. As the amount of fertilizer increases, the yield tends to increase as well. |

# 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 13 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(Play,Outlook)= 0.841**

**Calculations**

**Rainy**

**Entropy1 = -2/7\*(log2 2/7) – 5/7\*(log2 5/7) = 0.86**

**w1 = 7/11**

**Sunny**

**Entropy2 = -1/4\*(log2 1/4) – 3/4\*(log2 3/4)= 0.811**

**w2 = 4/11**

**H(Play,Outlook)= 7/11\*0.86+4/11\*0.81=0.841**

**(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**

**Entropybefore= 3/11\*1.874+8/11\*0.459=0.841**

**Entropyafter= 7/11\*0.86+4/11\*0.811=0.841**

**Inf. Gain (Outlook)= 0.841-0.841=0**

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

**Results:**

**For Outlook**

**Inf. Gain = 0**

**For Temperature**

**Inf. Gain = 0.117**

**For Humidity,**

**Inf. Gain = 0.401**

**Justification:** Splitting on Humidity provides the highest information gain, so the root node of the decision tree should be split on Humidity. So the decision tree algorithm will first evaluate the value of Humidity for each instance and split the dataset into subsets accordingly.

Topic 3: B

**(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$AddChild(names(childObs)[i])  #the algorithm recursively on the child and the subset will be called  TrID3(child, childObs[[i]])  }  }  }  tree <- **Node$new("Root")**  TrID3 (tree, p\_dec)  print(tree, "feature", "obsCount")  Output:  levelName feature obsCount  1 Root Humidity 11  2 ¦--high Play 6  3 ¦ °--no 6  4 °--normal Temperature 5  5 ¦--cool Play 2  6 ¦ °--no 2  7 ¦--hot Play 1  8 ¦ °--yes 1  9 °--mild Play 2  10 °--yes 2 |