BU510.650 – Data Analytics

Assignment # 4

(Updated on 11/22)

Please submit two documents: Your answers to each part of every question in .pdf or .doc format, and your R script, in .R format. In your document with answers, please do **not** respond with R output only. While it is okay to include R output in that document, please make sure you spell out the response to the question asked and also include all the plots. Please submit your assignment through Blackboard and name your files using the convention LastName_FirstName_AssignmentNumber. For example, Yazdi_Mohammad_4.pdf and Yazdi_Mohammad_4.R.

Programming Part

1. In this question, you will estimate a decision tree for the AutoLoss data. The data file for this question, AutoLoss-DT.csv, is slightly different from the data file in Assignment 2. In particular, instead of the actual loss amount for each vehicle, it has a column called HighLoss, which indicates whether the loss is high ("Yes") or low ("No") for each vehicle. Our goal is to create a decision tree that predicts whether the loss for a vehicle will be high or low. To begin your work on this question, run the following two lines of code: The first one replaces ?s with NA while reading the data from the .csv file, and the second one removes all the observations with any NA.

```
AutoLoss <- read.csv("AutoLoss-DT.csv", na.strings = "?", stringsAsFactors = TRUE)
AutoLoss <- na.omit(AutoLoss)
```

Please include set.seed(5) once at the beginning of your code, so we all get the same results.

a) Fit a decision tree to the entire data, with HighLoss as the response and all other variables as predictors. Plot the tree (including the names of predictors in the plot) and answer the following questions: Which predictors are used at the nodes of the tree? How many terminal nodes (leaves) does the tree have?

b) Determine the best tree size, using cross-validation and pruning. Plot the tree you obtained (including the names of predictors in the plot).

c) Use the best tree to answer the following question (you do not need to use R for this): Suppose my car fits the description shown below. Will this car incur a high loss or not?

FuelType	Aspiration	NumDoors	BodyStyle	DriveWheels	Length	Width	Height
gas	std	two	wagon	4wd	160	70	60

Weight	EngineSize	Horsepower	PeakRPM	Citympg	Price
3423	122	241	5000	26	23000

2. In this question, you will use the K-Nearest Neighbors (KNN) algorithm to predict whether a passenger will survive or not. To begin your work on this question, first read the data from the file "TitanicforKNN.csv" to a data frame named Titanic. Note: Please review the data before proceeding. You will notice that I already converted all the categorical variables (Gender, Fare, Class) into 0-1 columns. I did so, because KNN does not work well with non-numeric variables.

Next, split the data into training data and test data, using random selection. Include half of the records in the training data and the rest in the test data. You learned how to do this using sample function in Task 3 in Carseats-DecisionTree.R for a related example. (Remember to include set.seed(1) before the random selection in your code, so we all end up making the same split.)

(a) Run the KNN algorithm to predict the response variable Survived for each passenger in the test data. Do this for K = 2, 4, and 6. According to these predictions for K = 2, 4, and 6, what is the proportion of passengers in the test data that will survive?

R Hints: To run the function knn(), recall that you need four inputs:

(i) a matrix that contains the values of predictors in the training data,
(ii) a matrix that contains the values of predictors in the test data,
(iii) a vector containing the values of the response (Survived) in the training data,
(iv) a value for K.

To obtain (i), remove the Survived column from the training data. To obtain (ii), remove the Survived column from the test data. To obtain (iii), create a vector that stores the values of Survived column in the training data. See the Smarket-KNN.R for a related example.

(b) For each K, compute the accuracy of predictions for the test data. Which K works best in this case?

Conceptual Part

For this conceptual section, there is no requirement to use any R coding.

Question 1 KNN:

Consider the table below, which presents a training dataset with six observations and three predictors.

Obs.	X_1	X_2	X_3
1	7	0	8
2	10	8	4
3	6	0	4
4	5	5	1
5	5	3	3
6	5	6	4

Now, let's assume we want to predict Y when X1 = X2 = X3 = 5 using K-nearest neighbors with K = 3.

a) Begin by computing the Euclidean distance between each observation and the test point.

b) **KNN-Classification:** The qualitative response variable is listed below. Determine our classification prediction based on the 3 nearest neighbors.

Obs.	Y
1	High
2	High
3	High
4	Medium
5	Medium
6	Low

c) **KNN-Regression:** The quantitative response variable is listed below. Calculate our regression prediction. Hint: The regression prediction is the average of Y values for the 3 nearest neighbors.

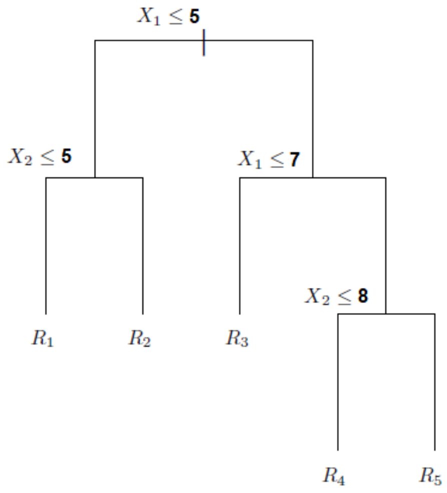
Obs.	Y
1	200
2	180
3	150
4	120
5	110
6	70

Question 2 Decision Tree-Classification:

Refer to the table below, which contains a training dataset with 15 observations, 2 predictors, and one qualitative variable.

Obs.	X1	X2	Y
1	0	7.5	High
2	1.5	8.5	Low
3	0	3	Low
4	6	4.5	Low
5	7.5	5.5	High
6	1	7	High
7	6.5	3.5	High
8	3	2	Low
9	9	9	Low
10	0	3.5	Low
11	6	9	Low
12	1	6.5	High
13	10	6	Low
14	8.5	6.5	High
15	4.5	4	High

a) The provided decision tree and regions are displayed in the figure below. Given the training dataset, identify which regions (R1 to R5) are labeled as "Low" and which regions are labeled as "High."



b) Following part a, compute the confusion matrix and accuracy for the training dataset.

c) Now, consider a training dataset with 5 observations. Following part a, compute the confusion matrix and accuracy.

Obs.	X1	X2	Y
1	6	2	High
2	9.5	2.5	Low
3	3.5	8	High
4	7.5	9	Low
5	1.5	0.5	Low

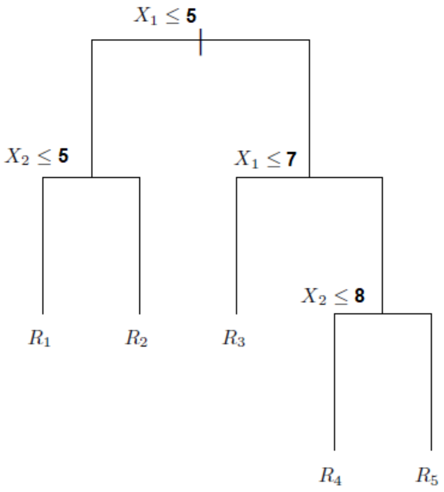
Question 3: Decision Tree – Regression

Refer to the table below, which contains a training dataset with 15 observations, 2 predictors, and one quantitative variable.

Obs.	X1	X2	Y
1	0	7.5	19
2	1.5	8.5	6
3	0	3	1
4	6	4.5	5
5	7.5	5.5	19
6	1	7	17
7	6.5	3.5	13
8	3	2	9

Obs.	X1	X2	Y
9	9	9	8
10	0	3.5	1
11	6	9	6
12	1	6.5	18
13	10	6	6
14	8.5	6.5	11
15	4.5	4	13

a) The decision tree and corresponding regions are illustrated in the figure below. To determine the values of R1 to R5 using the training dataset, compute the averages of the Y values within each respective region.



b) Now, consider a training dataset with 5 observations. Following part a, compute the Residual Sum of Squares (RSS) and Mean Squared Error (MSE) for the test dataset.

Obs.	X1	X2	Y
1	6	2	11
2	9.5	2.5	9
3	3.5	8	15
4	7.5	9	7
5	1.5	0.5	7

RSS is the sum of the squared differences between the actual observed values (Y) and the values predicted by the regression model (Ŷ). Mathematically, it is represented as:

$$RSS = \sum_{i=1}^{n} \left(Y_i - \widehat{Y_\imath}\right)^2$$

where n is the number of observations, Yi is the actual observed value, and Ŷi is the predicted value.

MSE is a normalized version of RSS, obtained by dividing RSS by the number of observations. It is essentially the average of the squared differences between the actual and predicted values. Mathematically, MSE is represented as:

$$MSE = \frac{RSS}{n}$$

Hint: Ŷi for each observation in the test dataset is equal to the value of the region (R1-R5) to which it belongs, based on the figure in part a.