Assignment2

STAT380

```
###
library(tree)
###
library(ISLR)
#attach(Carseats)
library(rattle)

library(rpart.plot)

## Loading required package: rpart

library(RColorBrewer)
library(partykit)
```

The data set `Carseats' is a simulated data set containing sales of child car seats at 400 different stores of a specific departmental store over a period of a few months. For the different activities in this assignment, we consider a categorical binary variable, that we call 'High_Sales'. We consider the sales amount is high, i.e., High_Sales=1, if the number of car sets that are sold is greater than 8 in the particular store.

Problem A

A1. (3 points) Create a new binary categorical variable called High_Sales' which is defined as follows: $ \text{High_Sales}='High' \text{ if }Sales'>8 $ and $ =0'Low' `Sales' $

A2. (2 points) Add the new variable to the dataset `Carseats'.

```
data(Carseats)
names(Carseats)

##  [1] "Sales"       "CompPrice"   "Income"      "Advertising" "Population"
##  [6] "Price"       "ShelveLoc"   "Age"         "Education"   "Urban"
## [11] "US"

### A1.



### A2.
```

A3. (5 points) Build a classification tree where the response is 'High_Sales' and the predictors are all the other variables except 'Sales' and 'High_Sales'.

```
### A3.
```

A4 (5 points) Plot the fitted tree

```
# A4 Plot the regression tree

#you may use fancyRpartPlot(fit_object, caption = NULL) # Nicer plot but need
libraries `RColorBrewer', `rattle' and `rpart.plot'
```

Training and Testing Set

A5.1. (1+4+2 =7points) Now, set a seed. Create a training set and a testing set , use a training set (70% in the training Set and 30% in test set). Print the dimension of the Training and Testing set

```
#### A5.1:
```

A5.2. (5+3=8 points) Fit the Classification Tree in the Training set and Plot the tree

```
### A5.2.



# plot the fitted tree you may use: fancyRpartPlot(fit_object, caption =
NULL)
```

A6.1(5+2=7 points) Print the summary, cross-validated cp values and plot the cp values. Comment on the Error Rate in the validation part.

A6.2 (3 points) Identify an optimal value for complexity parameter `cp'.

Note: cp = "value" is an assigned numeric value that will determine how tall a tree is to be grown the smaller value (closer to 0) leads to the larger the trees. The default value is 0.01.

```
### A6.1


### A6.2
#bestcp <-fit_train$cptable[which.min(fit_train$cptable[,"xerror"]),"CP"]
```

A7. (5 points) Obtain a Pruned tree based on to the optimal value of `cp' that you have obtained in A6.

```
### A7.

## Pruned tree
```

A8.1 (3 points) Predict on the Testing set with the pruned tree

A8.2 (3 points) Predict on the Testing set with the entire tree fitted using the training set

```
### A8.1
```

```
### A8.2
```

A9.1 (2 points) **Create the classification tables of the errors using the Predicted values from the pruned tree**

A9.2 (2 points) **Create A classification Tables of the errors using the Predicted values from entire tree fitted using the training set**

```
#A9.1

#A9.2

## Write a Conclusion of your finding
```

A10. (5 points) **Compare the classification performance of the tree and the pruned tree.**

#Problem B (Fitting Regression Trees)

We will use the regression trees for the Boston Housing data

Load the data from the github course page using: BostonH<- read.csv(url("https://raw.githubusercontent.com/subhadippal2019/STAT380UAEU/main/BostonHousing.csv"))

```
BostonH<-
read.csv(url("https://raw.githubusercontent.com/subhadippal2019/STAT380UAEU/main/BostonHousing.csv"))
names(BostonH)
```

```
##  [1] "CRIM"      "ZN"        "INDUS"     "CHAS"      "NOX"       "RM"
##  [7] "AGE"       "DIS"       "RAD"       "TAX"       "PTRATIO"   "LSTAT"
## [13] "MEDV"      "CAT..MEDV"
```

B1 (4+1=5 points) **Split the data in Training and Testing Set. Use a 60%/40% split for the Training and Testing Set. Print the dimension of the Testing and the Training set.**

```
#### A5.1:
set.seed(1234)
 #
```

B2.(5 points)

Fit a regression tree on the Training Set using the `MEDV', the median price of houses in a region, as the response variable while all the other variables EXCEPT` theCAT..MEDV' as the covariates. Display/plot the fitted tree.

```
### B2.

# plot fitted tree# You may use fancyRpartPlot(fitted_object, caption = NULL)
```

B3.(3 points) **Print the summary and the tables containig the crossvalidated `cp' and plot the `crossvalidatedcp'. Comment on the Error Rate in the validation part. (summary, printcp, plotcp)**

(2 points) Identify an optimal value for the complexity parameter `cp'.

```
### B3.
```

B4.(2+3=5 points) **Find the optimal value of `cp' and Prune the regression tree.**

```
## B4.
#bestcp <-fit_train$cptable[which.min(fit_train$cptable[,"xerror"]),"CP"]
```

B5.1 (3+2=5 points) **Predict on the Testing set with the pruned tree. Plot the predicted values vs the response values in the test set.**

B5.2 (3+2=5 points) **Predict on the Testing set with the Entire tree fitted to the training set. Plot the predicted values vs the response values in the test set.**

```
### B5.1
##Predict:


##Plot



### B5.2
##Predict:


##Plot
```

B6 (4+4+2=10 points) **Calculate the MSE for prediction using both the trees, the pruned tree, and the entire tree based on the Training set. compare their MSE. Comment on your findings.**

```
### B6
#mean((Predicted_response  - Response_in_Test )^2)
#Calculate the MSE for prediction using both the trees


 #Compare their MSE. Comment on your findinsg.
```