CS 422: Data Mining

Department of Computer Science
Illinois Institute of Technology
Vijay K. Gurbani, Ph.D.

Spring 2018: Homework 1 (10 points)

Due date: Wed, Feb 07, 2018 11:59:59 PM Chicago Time

Please read all of the parts of the homework carefully before attempting any question. If you detect any ambiguities in the instructions, please let me know right away instead of waiting until after the homework has been graded.

Additional parts to this homework will be added as new material is covered.

1 **Recitation problems** (total points 3)

1.1 Tan, Chapter 1 (1 point divided evenly among the questions)

Besides the lecture, make sure you read Chapter 1. After doing so, answer the following questions at the end of the chapter: 1, 3.

1.2 Tan, Chapter 2 (1 point divided evenly among the questions)

Besides the lecture, make sure you read Chapter 2, sections 2.1 – 2.3. After doing so, answer the following questions at the end of the chapter: 2, 3, 7, 12.

1.3 ISLR 7e (Gareth James, et al.) (1 point divided evenly among the questions)

Section 3.7 (Exercises), page 120: Exercises 1, 3, 4-a.

2 **Practicum problems** (total points 3, divided as indicated below)

2.1 Problem 1

This exercise relates to the College data set, which can be found in the file College.csv. It contains a number of variables for 777 different universities and colleges in the US. The variables are

- Private : Public/private indicator
- Apps : Number of applications received
- Accept : Number of applicants accepted
- Enroll : Number of new students enrolled
- Top10perc : New students from top 10 % of high school class
- Top25perc : New students from top 25 % of high school class
- F.Undergrad : Number of full-time undergraduates

- P.Undergrad : Number of part-time undergraduates
- Outstate : Out-of-state tuition
- Room.Board : Room and board costs
- Books : Estimated book costs
- Personal : Estimated personal spending
- PhD : Percent of faculty with Ph.D.'s
- Terminal : Percent of faculty with terminal degree
- S.F.Ratio : Student/faculty ratio
- perc.alumni : Percent of alumni who donate
- Expend : Instructional expenditure per student
- Grad.Rate : Graduation rate

Before reading the data into R , it can be viewed in Excel or a text editor.

(a) (0.5 pts) Use the read.csv() function to read the data into R . Call the loaded data 'college' . Make sure that you have the directory set to the correct location for the data. You can use the setwd() function to set the current working directory. For example, assume that College.csv is located in /home/vkg/CS422/Homework-1. Then, setwd("/home/vkg/CS422/Homework-1") will set the current working directory appropriately. Once you do this, you can simply invoke read.csv("College.csv", …) instead of prefxing the path name.

(b) (0.5 pts) Look at the data using the fix() function. You should notice that the first column is just the name of each university. We don't really want R to treat this as data. However, it may be handy to have these names for later. Try the following commands:

```
     > rownames(college) <- college[,1]
     > fix(college)
```

You should see that there is now a row.names column with the name of each university recorded. This means that R has given each row a name corresponding to the appropriate university. R will not try to perform calculations on the row names. However, we still need to eliminate the first column in the data where the names are stored. Try:

```
     > college <- college [ , -1]
     > fix(college)
```

Now you should see that the first data column is Private . Note that another column labeled row.names now appears before the Private column. However, this is not a data column but rather the name that R is giving to each row.

(c) (2 pts)

i. Use the summary() function to produce a numerical summaryof the variables in the data set.

ii. Use the pairs() function to produce a scatterplot matrix of the first ten columns or variables of the data. Recall that you can reference the first ten columns of a matrix A using A[,1:10] .

iii. Use the boxplot() function to produce side-by-side boxplots that help answers the following question: Which alumni donate more to their colleges --- those who go to public schools or those who go to private schools?

To do this, you will use the boxplot() function, but you will group the attribute you are interested in by a control group. The control group here is the attribute Private (which is ‘Yes’ or ‘No’), and the attribute of interest here is perc.alumni. The format is boxplot(x, data=), where x is a formula and data= denotes the data frame providing the data. An example of a formula is y~group where a separate boxplot for numeric variable y is generated for each value of group. Label the X- and Y-axes appropriately and provide a main= parameter to the boxplot() command for a graph title. You should see two boxplots if all works.

As an added resource, take a look at https://www.statmethods.net/graphs/boxplot.html

iv Use the boxplot() function to produce side-by-side boxplots that help answers the following question: Which colleges --- public or private --- employ more Ph.D.’s?

v. Create a new qualitative variable, called Elite by binning the Top10perc variable. We are going to divide universities into two groups based on whether or not the proportion of students coming from the top 10 % of their high school classes exceeds 50 %.

```
      > Elite <- rep(“No”, nrow(college))
      > Elite[college$Top10perc > 50] <- “Yes”
      > Elite <- as.factor(Elite)
      > college <- data.frame(college, Elite)
```

Use the summary() function to see how many elite universities there are.

vi. Use the hist() function to produce some histograms with differing numbers of bins for a few of the quantitative variables. You may find the command par(mfrow=c(2,2)) useful: it will divide the print window into four regions so that four plots can be made simultaneously. Modifying the arguments to this function will divide the screen in other ways.

vii. Continue exploring the data, and provide a brief summary of what you discover.

2.2 **Problem 2: Linear Regression (**total points 4, divided evenly by each question below)

You are given 269 observations of NBA players in the file “nba.csv”. The attribute description is as follows:

SEASON: NBA Season	DATE: Date of grame
PLAYER.FULLNAME: Full name of player	POSITION: Player position
OWNTEAM: Player’s team	OPPTEAM: Opposing team
VENUE: R(oad) game or H(ome) game	MIN: Minutes played
FG: Field goals	FGA: Field goals attempted
X3P: 3 pointers made	X3PA: 3 pointers attempted
FT: Free throws made	FTA: Free throws attempted
OR: Offensive rebounds	DR: Defensive rebounds
TOT: OR + DR	A: Assists
PF: Personal fouls	ST: Steals
TO: Turnovers	BL: Blocks
PTS: Points scored (response variable)	

You are to run linear regression models on the NBA dataset to predict the points scored (PTS).

Please follow the directions below carefully.

(a) **Simple regression** (univariate): Study the attributes of the NBA dataset. Pick one attribute that you think will be strongly correlated with the response variable (PTS). (You may run pairwise correlations to see which predictor is positively or negatively correlated with the response.) Print the variable chosen and the correlation plot.

Perform a simple regression on PTS using the predictor. Print the summary of the regression model and comment on how well the model fits the data.

(b) For the model in (a), plot the X-Y (or scatterplot) of the predictor and the regressor. On that plot, overlay the the regression line obtained from the model. Label the X and Y axes appropriately and provide a main title for the graph.

For the remaining questions below, set the seed as follow and divide the dataset in two parts: a training part and a test part. The training part will be used to build your regression model, and the testing part will be used to test your regression model to see how effective it is. The following commands are to be entered exactly as shown.

```
> set.seed(1122)
> index <- sample(1:nrow(df), 250)
> train <- df[index, ]
> test <- df[-index, ]
```

Above, the first statement sets the seed for the pseudo-random number generator. The second statement samples (without replacement) 250 numbers between 1 and the total rows of the data frame that you read the NBA dataset in. The third statement creates the training subset from the NBA dataset, and the final statement creates the test subset from the NBA dataset (note the use of -index as the row variable to gather all the rows that were not in the sampled index). At the end of these statements, you have two new dataframes: train and test.

(c) You are to pick 3-4 attributes that will act as regressors (predictors, or the independent variable) in your regression model. Using the psych package, study the correlation of the attributes versus the response variable to narrow down the list of regressors you will use. Once you have the list of regressors, plot the correlation between the response variable and the regressors.

(d) **Multiple regression (multi-variate)**: Run your regression model using the regressors picked in (c). Print a summary of your model. Determine which predictors are statistically significant and which are not. Eliminate those that are not statistically significant by going back to (c) and examining other options. Note that if you end up with an R^2 of 1.0, you may be overfitting (this means that the model is picking up patterns in the training set that may not be in the test set); if that happens, you will need to change the combination of regressors to give you a R^2 that is less than 1.0. (Please do not ask what a good value of R^2 should be; you should have notes from class that guide you on choosing R^2.) Provide comments on the fit of your model; i.e., evaluate it based on the R^2 value, the F-statistic and how statistically sound are the predictors.

(e) Plot the residuals of the model. Comment on the shape of the graph of the residuals.

(f) Plot a histogram of the residuals. Does the histogram follow a Gaussian distribution?

(g) Using the predict() method, fit the **test** dataset to the model. Get the predictions and put them in a new dataframe as a column vector. Put the test response variable as the second column vector in the new dataframe. Use R commands to find out how many of the fitted values matched (exactly) the PTS in the **test** dataset.

(h) Use R to calculate the residual vector for the predictions on the **test** dataset. Then, using the residual vector, calculate and print the following statistics:

1. RSS (Residual Sum of Errors).
2. TSS (Total Sum of Errors).
3. The F-statistic.
4. The RSE (Residual Standard Error).