**IST 687 PREP EXERCISE 10**

**Student Name: Bhavish Kumar**

**Prep Exercise No: 10**

**Date Due: 6th November 2019**

1. **Getting Ready: Loading and Verifying the Diamonds Dataset.**
   1. This week’s “diamonds” dataset comes from the ggplot2 package while the SVM function are located within a package called kernlab. Therefore, you will need to install and library ggplot2 and kernlab.

*library(ggplot2)*

*install.packages("kernlab")*

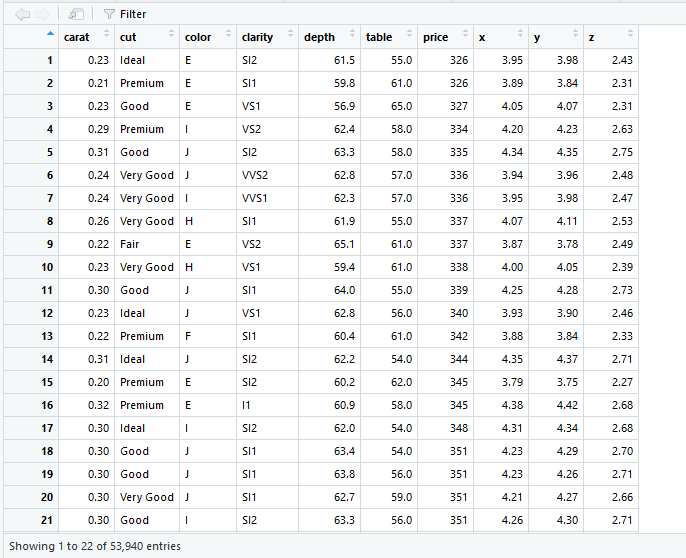
*library(kernlab)*

* 1. Assign the “diamonds” dataset to a dataframe and view the dataset. You will notice that there are five cuts of diamonds: Fair, Good, Very Good, Premium and Ideal.

*diamonds\_df <-diamonds*

* 1. View the dataframe and place a screenshot below. It is not important to fit the entire dataset in the screenshot, just enough to verify that you have completed the above steps.

*View(diamonds\_df)*



1. **Cleaning the data.**
   1. To simplify our classification task, we will focus only on Premium and Ideal cuts of diamonds. Below, there are two different ways to create the subset of the dataframe that fits the two “cut” categories. Create a new dataframe, called ‘goodDiamonds’ that only has the “Premium” and “Ideal” cut of diamonds.

*goodDiamonds <- diamonds\_df[(diamonds\_df$cut== "Premium" | diamonds\_df$cut == "Ideal"),]*

* 1. The clarity and color variables in the dataframe are “ordered factors.” This means that for analytical purposes such as this, you can convert the factor level directly into a number and it will make sense. Convert the clarity and color variables into numbers.

(**Hint:** use the *as.numeric()* command to accomplish this)

*goodDiamonds$color <- as.numeric(goodDiamonds$color)*

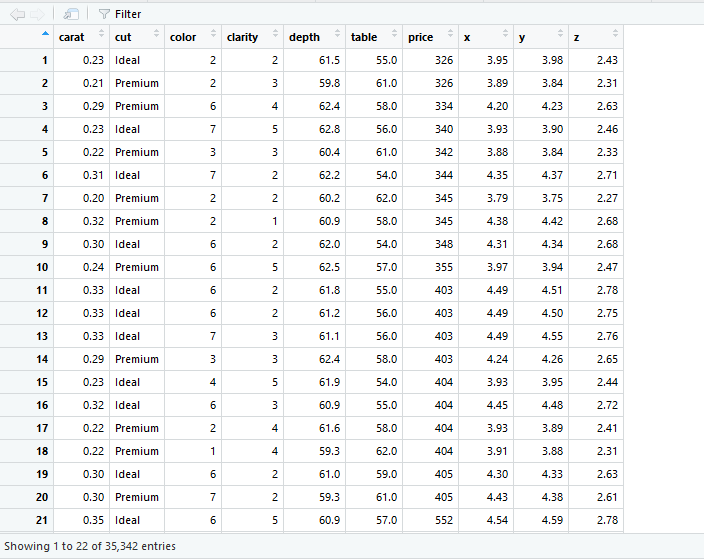
*goodDiamonds$clarity <- as.numeric(goodDiamonds$clarity)*

* 1. The cut attribute now has two level (just premium and ideal), but the dataframe still has the initial five factors. You can fix this by using the following code:

*as.factor(as.character(goodDiamonds$cut))*

* 1. View the adjusted dataframe and place a screenshot below.

*View(goodDiamonds)*



* 1. In a block comment, describe the meaning of each variable within the dataframe.

*The diamonds dataframe contains 53940 rows and 10 columns (variables). The 10 variables are:*

*Price: price in US dollars*

*Carat: weight of the diamond*

*Cut: quality of the cut (Fair, Good, Very Good, Premium, Ideal)*

*Color: Diamond color from D (best) to J (worst)*

*Clarity: a measurement of how clear the diamond is*

*X: length in mm*

*Y: width in mm*

*Z: depth in mm*

*Depth: total depth percentage = z/mean(x,y)*

1. **Understanding Terminology that will be used in this PE and HW.**
   1. In a paragraph or two explain the concept of a confusion matrix and the theoretical process behind creating one in RStudio.

*Confusion Matrix is used to measure the accuracy and error rate of the prediction model. If the dependent variable to be classified contains 2 values, for example Spam & Not Spam, then the confusion matrix creates a 2x2 matrix with the values for the following cases:*

* + 1. *Number of times the email was spam and was classified correctly as spam by the model*
    2. *Number of times the email was not spam and was classified correctly as not spam by the model*
    3. *Number of times the email was spam but was incorrectly tagged as not spam by the model*
    4. *Number of times the email was not spam but was incorrectly tagged as spam by the model*

*Increase in cases i. & ii. will improve the accuracy of the model whereas increase in iii. & iv. will result in an increase in error rate of the model*

*The process of creating a confusion matrix in R is to create a Dataframe with 2 columns which are the actual value of the email type and the value classified by the model. For example if the actual value contains Spam & Non Spam, then the column with the value predicted by the SVM model contains 1s for non-spam predictions and 0 for spam predictions. Hence by creating a dataframe containing these 2 columns and then a contingency table using the table() function on the dataframe will produce the confusion matrix.*