1. From R:
    1. All code from your R script (Code should be presented single-spaced in a fixed-width font. Adjust the font size so that no lines of code extend to the next line in the document)

```
# YUEH-TING WU
# MIS 545 Section 02
# Lab13WuY.R
# In this R programming, import a csv file, and generate a model
to predict if
# a review is good or bad on the text.

# Install the tidyverse, Matrix, Rcpp, quanteda, caret, and
doSNOW packages
# install.packages("tidyverse")
# install.packages("Matrix")
# install.packages("Rcpp")
# install.packages("quanteda")
# install.packages("caret")
# install.packages("doSNOW")

# Load the tidyverse, Matrix, quanteda, caret, and doSNOW
libraries
library(tidyverse)
library(Matrix)
library(quanteda)
library(caret)
library(doSNOW)

# Set the working directory
setwd("~/MIS 545/Lab13")

# Read HotelReviews1000.csv into a tibble called hotelReviews
hotelReviews <- read_csv(file = "HotelReviews1000.csv",
                         col_types = "ci",
                         col_names = TRUE)

# Display the hotelReviews in the conole
print(hotelReviews)

# Display the structure of hotelReviews in the console
str(hotelReviews)

# Display the summary of hotelReviews in the console
summary(hotelReviews)
```

```
# Remove 3-star reviews from the dataset
hotelReviews <- hotelReviews %>% filter(Stars != 3)

# Show a count of reviews by star rating
print(hotelReviews %>% count(Stars))

# Create a new feature called Rating that has a value of "bad"
or "good."
# 1-2 star reviews are "bad" and 4-5 star reviews are "good."
hotelReviews <- hotelReviews %>%
  mutate(Rating = factor(ifelse(Stars == 1 | Stars == 2, "bad",
"good")))

# Drop the original Stars feature
hotelReviews <- hotelReviews %>%
  select(-Stars)

# Display the summary of hotelReivews in the console
summary(hotelReviews)

# Drop records with missing data in case there are any
hotelReviews <- drop_na(hotelReviews)

# Add a feature called ReviewLength for the number of characters
in the review
# text
hotelReviews <- hotelReviews %>%
  mutate(ReviewLength = nchar(Text))

# Display the average review length by rating
print(hotelReviews %>%
        group_by(Rating) %>%
        summarize(mean(ReviewLength)))

# Tokenize the reviews into an object called hotelReviewTokens
hotelReviewTokens <- tokens(x = hotelReviews$Text,
                            what = "word",
                            remove_numbers = TRUE,
                            remove_punct = TRUE,
                            remove_symbols = TRUE,
                            remove_url = TRUE,
                            split_hyphens = TRUE)

# Make the tokens all lowercase
hotelReviewTokens <- tokens_tolower(hotelReviewTokens)
```

```
# View review 506 both in hotelReviews and hotelReivewTokens
hotelReviews[506, ]
hotelReviewTokens[506, ]

# Remove stop words from hotelReviewTokens
hotelReviewTokens <- tokens_select(hotelReviewTokens,
                                   stopwords(),
                                   selection = "remove")

# View review 506 both in hotelReviews and hotelReivewTokens
hotelReviews[506, ]
hotelReviewTokens[506, ]

# Combine stemmed words in tokens
hotelReviewTokens <- tokens_wordstem(hotelReviewTokens,
                                     language = "english")

# View review 506 both in hotelReviews and hotelReivewTokens
hotelReviews[506, ]
hotelReviewTokens[506, ]

# Generate a DFM into an object called hotelReviewTokensDFM
hotelReviewTokensDFM <- dfm(hotelReviewTokens)

# Convert hotelReviewTokensDFM into an R matrix called
hotelReviewTokensMatrix
hotelReviewTokensMatrix <- as.matrix(hotelReviewTokensDFM)

# Display the dimensions of hotelReviewTokensMatrix on the
console
print(dim(hotelReviewTokensMatrix))

# View a subset of the matrix (the first 20 rows and the first
100 columns)
View(hotelReviewTokensMatrix[1:20, 1:100])

# Generate a feature data frame called
hotelReviewTokensDataFrame with labels
hotelReviewTokensDataFrame <-cbind(Label = hotelReviews$Rating,

data.frame(hotelReviewTokensDFM))

# Clean the column names using the names() function
names(hotelReviewTokensDataFrame) <-
  make.names(names(hotelReviewTokensDataFrame))
```

```
# Set the random seed to 5904
set.seed(5904)

# Set up the stratified cross-validation parameters in an object
called
# crossValidationFolds
crossValidationFolds <- createMultiFolds(y =
hotelReviews$Rating,
                                         k =10,
                                         times = 3)

# Set up the training process in an object called
crossValidationControl
crossValidationControl <- trainControl(method = "repeatedcv",
                                       number = 10,
                                       repeats = 3,
                                       index =
crossValidationFolds)

# Create a cluster called cluster to work on 3 logical cores
# Register the cluster using registerDoSNOW() function
cluster <- makeCluster(3, type = "SOCK")
registerDoSNOW(cluster)

# Develop a single decision tree algorithm called
hotelReviewDecisionTree
hotelReviewDecisionTree <- train(Label ~ .,
                                 data =
hotelReviewTokensDataFrame,
                                 method = "rpart",
                                 trControl =
crossValidationControl,
                                 tuneLength = 7)

# Stop the cluster when processing is finished
stopCluster(cluster)

# Display hotelReviewDecisionTree on the console
print(hotelReviewDecisionTree)
```

2. Answer the following question in a sentence: What complexity parameter yielded the highest accuracy?

Complexity parameter is 0.03298611 yielded the highest accuracy.

3. Answer the following question in a sentence: In this assignment, due to time and scope, we did not split the data into a training and a testing dataset. What are the downsides to using the same data for training and testing a model?

Using the same data for training and testing a model may make it more complex because the goal is to predict a review based on the text and split the text into training and testing dataset may cause less accuracy

4. Answer the following question in a sentence: How could a hotel chain use this model to help bolster their reputation?

A hotel chain can use this model to find out the rating that their customers gave them, and improve their shortcut.