**DSBA/MBAD 6211 Assignment 3**

Due Date: 11:59 pm @ 11/25/2020

**Data description**

A fuel company has 250+ gas stations in the US. It captures customers’ comments via phone, which are merged with numeric variables by matching them with the company’s royalty card number. All data were provided in the Gas\_text\_numeric\_data file. Some of the text comments, variable names, and descriptions were disguised to protect the identity of the client company.

* The target variable is identified by the column name.
* ***Cust\_ID,*** and ***Loyal\_Status*** are nominal variables, and all other variables are binary.
* ***Comment*** column contains the text information.

**Variable and model naming requirements:**

* + Please include your ***name initials*** to the data frame names as well as model names in your R coding.
  + Please instance, in my coding, I would name the data frames as ***dfKZ, dfKZ.train***, and ***dfKZ.valid.*** I would also name the models as ***treeKZ***, etc.

**Questions**

1. Provide the word cloud after all necessary pre-processing.

Pre-processing:

![Timeline

Description automatically generated]()

Post-processing:

![Text, timeline

Description automatically generated]()

1. What are the top 5 terms that are most related to “price”? Please specify your similarity measurement method and detailed results.
2. What are the top 5 terms that are most related to “service”? Please specify your similarity measurement method and detailed results. ![Text

   Description automatically generated]()

In both my similarity measurement methods, I show how closely these five different words are closely correlated to the words chosen in the assignment.

In the first method, the chosen variable is price, and we are comparing the features in the correlation. Our variables chosen are high, good, access, gas, and experi and they are ranked from highest to lowest based on how correlated they are to price. Based on these variables, they are correlated to price because gas stations have high or low prices, good or bad locations, access to stations, and usually a quick experience.

In the second method, the chosen variable is servic and we are comparing the featured based on their correlation. Again, they rank highest to lowest. All services can be good or bad. Usually services have a program that goes over a period of tie. Some of them can be custom based on the consumers desires. Each service has a specific location and there is always a membership when it comes to services. As you can see, most consumers base their opinion on if the service is good or bad.

1. Perform topic modeling with 4 topics
   * Further remove some common words, such as “shower” & “point”
   * You might encounter the issue with all zero rows, and you need to remove those all zero rows. Here are some sample codes for your reference

myDfm **<-** dfm\_remove**(**myDfm, c**(**'shower','point'**))**

myDfm **<-** as.matrix**(**myDfm**)**

myDfm **<-**myDfm**[**which**(**rowSums**(**myDfm**)>**0**)**,**]**

myDfm **<-** as.dfm**(**myDfm**)**

* + Provide the term/beat plots for four topics.
  + Try your best to summarize those four topics

![Chart, bar chart

Description automatically generated]()

![Table

Description automatically generated]()

Topic 1: Topic one has to do with the service of the gas stations. It has all the features of a gas station from the gas itself to the food and drinks inside. I also believe if the station has a car wash.

Topic 2: Topic two has to do with the overall experience within the gas station. What are customers doing? Buying gas? Buying food? How long do people spend at gas stations?

Topic 3: Topic three has to do with the products given by the gas station. It will offer food and drink products. It will give different levels of gas ranging from regular to diesel. It will also give the prices of each product and service.

Topic 4: Topic four has to do with the services and products reviews. It will give reviews on everything and whether the services, and products are good and bad. Are they a respectable price to buy?

1. Please run two decision tree models
   * Model 1 only uses non-text information (i.e., does not use the ***Comment*** column)

![Diagram

Description automatically generated]()

![Text, letter

Description automatically generated]()

![A screenshot of text

Description automatically generated]()

* + Model 2 combines both non-text and text information
    - Text mine the ***Comment*** column
    - Apply SVD to extract text information from the ***Comment*** column
    - Keep the number of SVD as 8

![Diagram

Description automatically generated]()

![Text, letter

Description automatically generated]()

![A screenshot of text

Description automatically generated]()

* + Please compare the model performance of two models based on the confusion matrix of the validation dataset

As you can see in model 1, it has a higher accuracy at 0.5891 than model 2 at 0.5647; therefore, model 1 has a better accuracy then model 2. In addition, model 1 has a higher final value of cp at 0.054 while model 2 has a final value cp of 0.028. However, the first model has a lot of false predictions based on the confusion matrix meaning that the accuracy could not be reliable. Model 2 also has a higher p-value of the overall model. Based on those predictions, using the comments section and model 2 would be more beneficial due to the feedback from the customers can be proven.

1. Please copy and paste your R codes in your WORD submission.

install.packages('quanteda')

install.packages('rpart')

install.packages('rpart.plot')

install.packages("caret")

library(quanteda)

library(pROC)

library(rpart)

library(rpart.plot)

library(caret)

setwd('C:/Users/grays/Desktop/BusinessAnalytics')

textRC <- read.csv('gastext.csv',stringsAsFactors = F)

str(textRC)

#change to factor

textRC$Target <- factor(textRC$Target)

textRC$Service\_flag <- factor(textRC$Service\_flag)

textRC$CustType\_flag <- factor(textRC$CustType\_flag)

textRC$Contact\_flag <- factor(textRC$Contact\_flag)

textRC$new\_flag <- factor(textRC$new\_flag)

textRC$Choice\_flag <- factor(textRC$Choice\_flag)

textRC$Comp\_card\_flag <- factor(textRC$Comp\_card\_flag)

textRC$AcctType\_flag <- factor(textRC$AcctType\_flag)

textRC$Contact\_Flag2 <- factor(textRC$Contact\_Flag2)

textRC$HQ\_flag <- factor(textRC$HQ\_flag)

textRC$Multi\_flag <- factor(textRC$Multi\_flag)

textRC$NewCust\_Flag <- factor(textRC$NewCust\_Flag)

cols.dont.want <- c("Cust\_ID")

textRC <- textRC[, ! names(textRC) %in% cols.dont.want, drop = F]

#OR - Sample code to convert multiple columns into factors: df[,3:13]<-lapply(df[,3:13],factor)

str(textRC)

#create corpus

myCorpusRC <- corpus(textRC$Comment)

summary(myCorpusRC)

# Create a dfm (UNITGRAM) data future matrix

myDfmRC <- dfm(myCorpusRC)

topfeatures(myDfmRC)

#INITIAL ANALYSIS

#frequency analysis

tstat\_freqRC <- textstat\_frequency(myDfmRC)

head(tstat\_freqRC, 20)

#visualize most frequent terms

library(ggplot2)

myDfmRC %>%

textstat\_frequency(n = 20) %>%

ggplot(aes(x = reorder(feature, frequency), y = frequency)) +

geom\_point() +

labs(x = NULL, y = "Frequency") +

theme\_minimal()

textplot\_wordcloud(myDfmRC,max\_words=100)

#preprocessing

# Remove stop words and perform stemming

library(stopwords)

myDfmRC <- dfm(myCorpusRC,

remove\_punc = T,

remove = c(stopwords("english")),

stem = T)

dim(myDfmRC)

topfeatures(myDfmRC,50)

#remove unwanted words or symbols in this case

stopwords1 <-c('t','')

myDfmRC <- dfm\_remove(myDfmRC,stopwords1)

topfeatures(myDfmRC,30)

stopwords2 <-c('get','use', '2','per','1','don','anymor', 'take', 'can', 'cant','use','get')

myDfmRC <- dfm\_remove(myDfmRC,stopwords2)

topfeatures(myDfmRC,50)

dim(myDfmRC)

#frequency analysis

tstat\_freqRC <- textstat\_frequency(myDfmRC)

head(tstat\_freqRC)

dim(myDfmRC)

#remove infrequent terms #significantly reduced dimentionality

myDfmRC<- dfm\_trim(myDfmRC,min\_termfreq=2, min\_docfreq=2)

dim(myDfmRC)

textplot\_wordcloud(myDfmRC,max\_words=100)

#how terms are related

#price

term\_priceRC <- textstat\_simil(myDfmRC,

selection="price",

margin="feature",

method="correlation")

as.list(term\_priceRC,n=5)

#service

term\_serviceRC <- textstat\_simil(myDfmRC,

selection="servic",

margin="feature",

method="correlation")

as.list(term\_serviceRC,n=5)

# TOPIC MODELING

library(topicmodels)

library(tidytext)

#remove shower and point

stopwords3 <-c('shower','point')

myDfmRC <- dfm\_remove(myDfmRC,stopwords3)

topfeatures(myDfmRC,30)

dim(myDfmRC)

#remove all 0 rows

myDfmRC <- as.matrix(myDfmRC)

myDfmRC <-myDfmRC[which(rowSums(myDfmRC)>0),]

myDfmRC <- as.dfm(myDfmRC)

dim(myDfmRC)

#explore LDA\_VEM topic model with 4 topics

myLdaRC <- LDA(myDfmRC,k=4,control=list(seed=50))

myLdaRC

# Term-topic probabilities

myLda\_tdRC <- tidy(myLdaRC)

myLda\_tdRC

# Visulize most common terms in each topic

library(ggplot2)

library(dplyr)

top\_terms <- myLda\_tdRC %>%

group\_by(topic) %>%

top\_n(8, beta) %>%

ungroup() %>%

arrange(topic, -beta)

top\_terms %>%

mutate(term = reorder(term, beta)) %>%

ggplot(aes(term, beta, fill = factor(topic))) +

geom\_bar(stat = "identity", show.legend = FALSE) +

facet\_wrap(~ topic, scales = "free") +

coord\_flip()

# View topic 8 terms in each topic

Lda\_termRC<-as.matrix(terms(myLdaRC,8))

View(Lda\_termRC)

# Document-topic probabilities

ap\_documentsRC <- tidy(myLdaRC, matrix = "gamma")

ap\_documentsRC

# View document-topic probabilities in a table

#how each topic is related

Lda\_documentRC<-as.data.frame(myLdaRC@gamma)

#RUN MODELS

# Prepare the corpus by adding the ID and Target columns

textRC[,3:15]<-lapply(textRC[,3:15],factor)

cols.dont.want <- c("Comments")

nctextRC <- textRC[, ! names(textRC) %in% cols.dont.want, drop = F]

nctextRC <- textRC[-c(1)]

str(textRC)

str(nctextRC)

#train/valid sets made

trainIndexRC <- createDataPartition(nctextRC$Target,

p=0.7,

list=FALSE,

times=1)

df1.trainRC <- nctextRC[-trainIndexRC,]

df1.validRC <- nctextRC[trainIndexRC,]

#DEC TREE MODEL (All only numeric/no comments)

Tree1.RC <- train(Target~.,

data = df1.trainRC,

method = 'rpart',

na.action = na.pass)

Tree1.RC

prp(Tree1.RC$finalModel,type=2,extra=106)

#simple dec model analysis

pred\_tree1.RC <- predict(Tree1.RC,

newdata = df1.validRC,

na.action = na.pass)

confusionMatrix(pred\_tree1.RC,df1.validRC$Target)

#DECISION TREE MODEL (with comments)

myDfmRC <- dfm(myCorpusRC,

remove\_punc=T,

remove=c(stopwords('english'),stopwords1,stopwords2),

stem=T)

myDfmRC <- dfm\_trim(myDfmRC,min\_termfreq=4, min\_docfreq=2)

dim(myDfmRC)

myDfm\_tfidfRC <- dfm\_tfidf(myDfmRC)

#wvd for dimention reduction

library(quanteda.textmodels)

#combine

modelSvdRC <- textmodel\_lsa(myDfm\_tfidfRC, nd=8)

modelSvdRC <- as.data.frame(modelSvdRC$docs)

df2RC <- cbind(nctextRC,modelSvdRC)

trainIndex2RC <- createDataPartition(df2RC$Target,

p=0.7,

list=FALSE,

time=1)

df2.trainRC <- df2RC[trainIndex2RC,]

df2.validRC <- df2RC[-trainIndex2RC,]

Tree2.RC <- train(Target~.,

data=df2.trainRC,

method='rpart',

na.action=na.pass)

Tree2.RC

prp(Tree2.RC$finalModel,type=2,extra=106)

pred\_tree2.RC <- predict(Tree2.RC,

newdata = df2.validRC,

na.action = na.pass)

confusionMatrix(pred\_tree2.RC, df2.validRC$Target)