**DSBA/MBAD 6211 Assignment 3**

Due Date: 11:59 pm @ 7/29/2020

**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.



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

|  |
| --- |
| > as.list(term\_sim,n=5)  $price  high reproductx accomid provid unorgan  0.2900886 0.2393096 0.2393096 0.2393096 0.2393096 |
|  |
| |  | | --- | | > I have used correlation measurement method used for this.  Other method such as cosine can be used. | |

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

> as.list(term\_sim2,n=5)

$servic

bad cramp lousi program custom

0.3142026 0.2965125 0.2965125 0.2908260 0.2777017

> I have used correlation measurement method used for this.

Other method such as cosine can be used.

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

Graph 1: Mostly the topic is related to parking and getting reward points.

Graph 2: This graph might be talking about CLEAN and USABLE facilities like – shower, restrooms etc.

Graph 3: This must for Non availability of certain services like food, drink etc.

Graph 4: Food and Card usage and getting free services

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

> confusionMatrix(model1.prediction,text2.valid$Target)

Confusion Matrix and Statistics

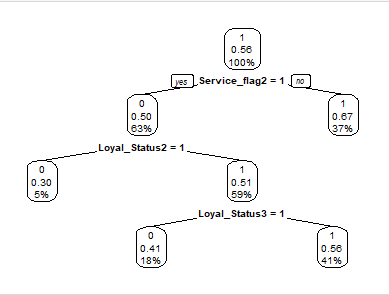
Reference

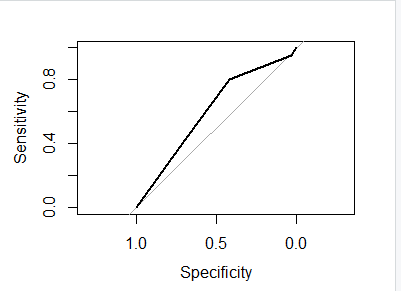
Prediction 0 1

0 13 8

1 18 32

Accuracy : 0.6338





> tree.ROC$auc

Area under the curve: 0.6024

* + 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

> confusionMatrix(model2.prediction,modelData.valid$Target)

Confusion Matrix and Statistics

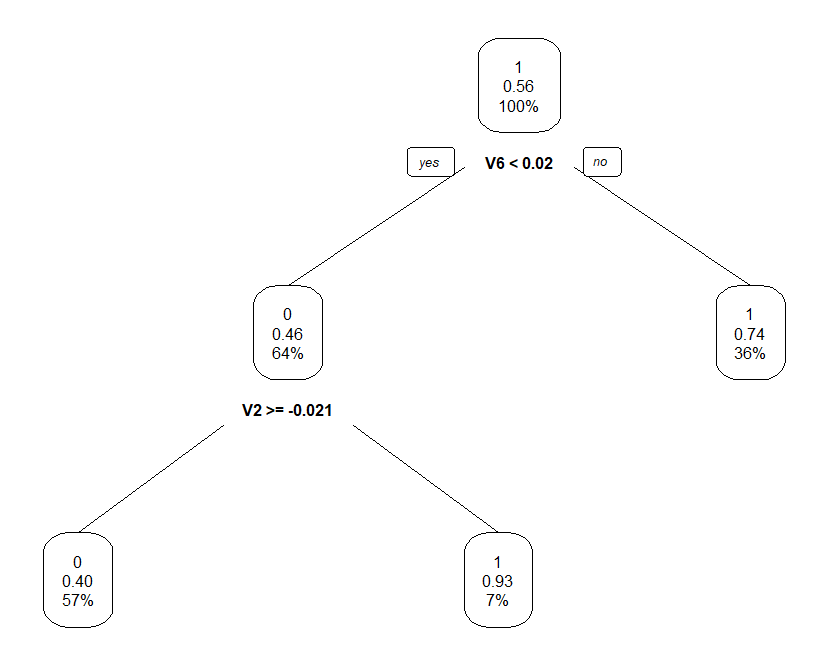
Reference

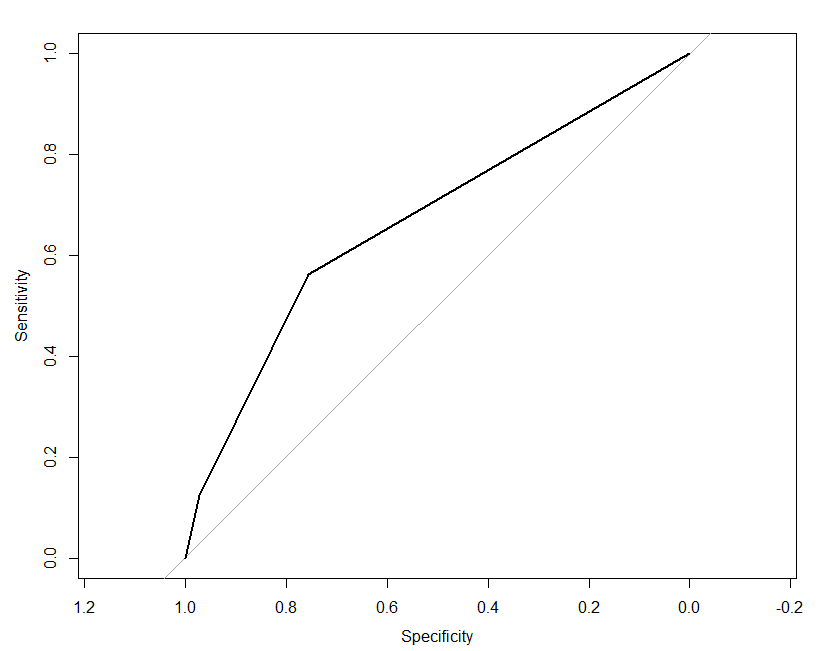
Prediction 0 1

0 28 21

1 9 27

Accuracy : 0.6471





> tree.ROC$auc

Area under the curve: 0.6672

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

Based on the confusion matrix, model 2 which has accuracy of 64% is performing better.

Model 1 has accuracy of 63%.

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

getwd()

setwd("C:\\Users\\P2190101\\Desktop\\NK Personal\\NK Study\\UNCC\\6211 - Advanced BI\\Assignment 3") # Set working directory

# Read data

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

# Convert column 3-15 to factor

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

str(text)

library(quanteda)

myCorpus <- corpus(text$Comment)

summary(myCorpus) # How many Sentences , Types -> Unique words, Token -> Total words

#Step 1: Create Corpus

myDfm <- dfm(myCorpus)

topfeatures(myDfm)

#Step 2: Get Document Term ready

library(stopwords)

myDfm <- dfm(myCorpus,

remove\_punc = T,

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

stem = T) # Remove English STOP WORDS such as 'a''an','the' etc.

dim(myDfm)

topfeatures(myDfm,30)

###########################

####### Question 1 ########

###########################

tstat\_freq <- textstat\_frequency(myDfm) # Simple frequency analysis - get unique list of Words

head(tstat\_freq, 20)

# Visulize the most frequent terms

library(ggplot2)

myDfm %>%

textstat\_frequency(n = 20) %>%

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

geom\_point() +

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

theme\_minimal()

# Wordcloud

textplot\_wordcloud(myDfm,max\_words=200)

###########################

####### Question 2 ########

###########################

# Top 5 terms similar to "price"

term\_sim <- textstat\_simil(myDfm,

selection="price",

margin="feature",

method="correlation")

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

###########################

####### Question 3 ########

###########################

# Top 5 terms similar to "servic"

term\_sim2 <- textstat\_simil(myDfm,

selection="servic",

margin="feature",

method="correlation")

as.list(term\_sim2,n=5)

###########################

####### Question 4 ########

###########################

library(topicmodels)

library(tidytext)

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

myDfm <- as.matrix(myDfm)

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

myDfm <- as.dfm(myDfm)

myLda <- LDA(myDfm,k=4,control=list(seed=101))

myLda

# Term-topic probabilities

myLda\_td <- tidy(myLda)

myLda\_td

library(ggplot2)

library(dplyr)

top\_terms <- myLda\_td %>%

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\_term<-as.matrix(terms(myLda,8))

View(Lda\_term)

# Document-topic probabilities

ap\_documents <- tidy(myLda, matrix = "gamma")

ap\_documents

# View document-topic probabilities in a table

Lda\_document<-as.data.frame(myLda@gamma)

View(Lda\_document)

###########################

####### Question 5 ########

###########################

library(readxl)

library(caTools)

library(ROCR)

library(rpart)

library(rpart.plot)

library(caret)

library(e1071)

library(caret)

library(pROC)

library(rpart)

library(rpart.plot)

text2 <- text[ -c(1,2) ]

# Data partition with the Caret package

# Set a random see so your "random" results are always same

set.seed(101)

trainIndex <- createDataPartition(text2$Target,

p=0.7,

list=FALSE,

times=1)

# Create Training Data

text2.train <- text2[trainIndex,]

# Create Validation Data

text2.valid <-text2[-trainIndex,]

####### TREE MODEL (Model 1) ########

# Build a decision tree model

tree.model <- train(Target~.,

data=text2.train,

method="rpart",

na.action=na.pass)

# Display decision tree results

tree.model

# Display decision tree plot

prp(tree.model$finalModel,type=2,extra=106)

# Validate the model result

model1.prediction <- predict(tree.model,newdata=text2.valid,na.action = na.pass)

confusionMatrix(model1.prediction,text2.valid$Target)

tree.probabilities <- predict(tree.model,newdata=text2.valid,type='prob',na.action=na.pass)

tree.ROC <- roc(predictor=tree.probabilities$`1`,

response=text2.valid$Target,

levels=levels(text2.valid$Target))

plot(tree.ROC)

tree.ROC$auc

#############################################################################################################################

####### non text and text mining (Model 2) ########

docvars(myCorpus, "Target") <- text$Target

# We will first generate SVD columns based on the entire corpus

# Pre-process the training corpus

modelDfm <- dfm(myCorpus,

remove\_punc = T,

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

stem = T)

#Remove highly infrequent words. Take only those which are at least 4 times and in two document

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

dim(modelDfm)

# Weight the predictiv DFM by tf-idf

modelDfm\_tfidf <- dfm\_tfidf(modelDfm)

dim(modelDfm\_tfidf)

# Use SVD to reduce dimensionality

#install.packages('quanteda.textmodels')

library(quanteda.textmodels)

modelSvd <- textmodel\_lsa(modelDfm\_tfidf, nd=8) # We reduce dim to 8 column

head(modelSvd$docs)

# Add the author information as the first column

modelData <-cbind(docvars(myCorpus,"Target"),as.data.frame(modelSvd$docs))

colnames(modelData)[1] <- "Target"

head(modelData)

# Split the data into training & test

trainIndex <- createDataPartition(modelData$Target,

p=0.7,

list=FALSE,

times=1)

# Create Training Data

modelData.train <- modelData[trainIndex,]

# Create Validation Data

modelData.valid <-modelData[-trainIndex,]

####### TREE MODEL (Model 2) ########

# Build a decision tree model

tree.model <- train(Target~.,

data=modelData.train,

method="rpart",

na.action=na.pass)

# Display decision tree results

tree.model

# Display decision tree plot

prp(tree.model$finalModel,type=2,extra=106)

# Validate the model result

model2.prediction <- predict(tree.model,newdata=modelData.valid,na.action = na.pass)

confusionMatrix(model2.prediction,modelData.valid$Target)

tree.probabilities <- predict(tree.model,newdata=modelData.valid,type='prob',na.action=na.pass)

tree.ROC <- roc(predictor=tree.probabilities$`1`,

response=modelData.valid$Target,

levels=levels(modelData.valid$Target))

plot(tree.ROC)

tree.ROC$auc

Hints:

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