TEXT MINNING

Source code:

#1. loading required library

library(stringr)

library(tm)

library(ggplot2)

library(qdap)

library(SnowballC)

library(wordcloud)

tweets=read.csv("C:/Users/I340968/Desktop/ed-code/R programming-elective/Week\_6/Tweets.csv",

stringsAsFactors = FALSE)

View(tweets)

#2.understanding the data structure.

str(tweets)

#3.setting up before text processing.

r1 = as.character(tweets$Tweet)

set.seed(100)

sample=sample(r1,(length(r1)))

sample

View(sample)

#4.data process

#4.1 create croups

tweets\_vs <- VectorSource(r1)

tweets\_co1=VCorpus(tweets\_vs)

#4.2 remove punctuations

v1<-tm\_map(tweets\_co1,removePunctuation)

v1[[1]][1]

#4.3 lower case

v2<-tm\_map(tweets\_co1,tolower)

#4.3 removing numbers

v3<-tm\_map(tweets\_co1,removeNumbers)

#4.4 removing whitespace

v4<-tm\_map(tweets\_co1,stripWhitespace)

#4.5 remvoing stop words

v5<-tm\_map(tweets\_co1,removeWords,stopwords('english'))

#4.6 stemming

v6 <- c("additional","additive","addition")

stem\_doc<-stemDocument(v6)

stem\_doc

v7<-'addition'

v7

#to make the completion of the word

stem\_text=stemCompletion(stem\_doc,v7)

stem\_text

#4.7 Document term matrix via croups

v10<-DocumentTermMatrix(tweets\_co1)

v10

#4.8data fram from the output

v11<-as.matrix(v10)

dim(v11)

View(v11)

#5. creating worldcloud

wordcloud(tweets$Tweet,min.freq = 5,random.order = TRUE, colors=brewer.pal(8, "Dark2"))

wordcloud(tweets$Tweet,min.freq = 5,random.order = FALSE, colors=brewer.pal(8, "Dark2"))

#Negative review

#1. Creating datafram.

str(tweets)

tweets$Avg = as.integer(tweets$Avg)

negativeTweets <- tweets[tweets$Avg < 0,]

View(negativeTweets)

#2. Running following code:

n1 = as.character(negativeTweets$Tweet)

set.seed(100)

sample2 = sample(n1, (length(n1)))

#1.1 create croups

tweets\_ng <- VectorSource(negativeTweets)

tweets\_co2=VCorpus(tweets\_ng)

#1.2 remove punctuations

v1<-tm\_map(tweets\_co2,removePunctuation)

v1[[1]][1]

#1.3 lower case

v2<-tm\_map(tweets\_co2,tolower)

#1.4 removing numbers

v3<-tm\_map(tweets\_co2,removeNumbers)

#1.5 removing whitespace

v4<-tm\_map(tweets\_co2,stripWhitespace)

#1.6 remvoing stop words

v5<-tm\_map(tweets\_co2,removeWords,stopwords('english'))

#1.7 stemming

v11 <- c("additional","additive","addition")

stem\_doc<-stemDocument(v11)

stem\_doc

v12<-'addition'

v12

#1.8 to make the completion of the word

stem\_text=stemCompletion(stem\_doc,v12)

stem\_text

#1.9 Document term matrix via croups

v13<-DocumentTermMatrix(tweets\_co2)

v13

#4.8data fram from the output

v14<-as.matrix(v13)

dim(v13)

View(v13)

#creating worldcloud

wordcloud(negativeTweets$Tweet,min.freq = 5,random.order = TRUE, colors=brewer.pal(8, "Dark2"))

wordcloud(negativeTweets$Tweet,min.freq = 5,random.order = FALSE, colors=brewer.pal(8, "Dark2"))

#possitive feedback

#5. Creating datafram.

str(tweets)

tweets$Avg = as.integer(tweets$Avg)

possitiveTweets <- tweets[tweets$Avg >= 0,]

View(possitiveTweets)

#6. Running following code:

p1 = as.character(possitiveTweets$Tweet)

set.seed(100)

sample2 = sample(p1, (length(p1)))

#7.1 create croups

tweets\_pg <- VectorSource(possitiveTweets)

tweets\_co3=VCorpus(tweets\_pg)

#7.2 remove punctuations

v1<-tm\_map(tweets\_co3,removePunctuation)

v1[[1]][1]

#7.3 lower case

v2<-tm\_map(tweets\_co3,tolower)

#7.4 removing numbers

v3<-tm\_map(tweets\_co3,removeNumbers)

#7.5 removing whitespace

v4<-tm\_map(tweets\_co3,stripWhitespace)

#7.6 remvoing stop words

v5<-tm\_map(tweets\_co3,removeWords,stopwords('english'))

#1.7 stemming

v15 <- c("additional","additive","addition")

stem\_doc<-stemDocument(v15)

stem\_doc

v16<-'addition'

v16

#1.8 to make the completion of the word

stem\_text=stemCompletion(stem\_doc,v16)

stem\_text

#1.9 Document term matrix via croups

v17<-DocumentTermMatrix(tweets\_co3)

v17

#4.8data fram from the output

v18<-as.matrix(v17)

dim(v18)

View(v18)

#creating worldcloud

wordcloud(possitiveTweets$Tweet,min.freq = 5,random.order = TRUE, colors=brewer.pal(8, "Dark2"))

wordcloud(possitiveTweets$Tweet,min.freq = 5,random.order = FALSE, colors=brewer.pal(8, "Dark2"))