Text Mining_Project

Vivek

10/14/2023

R Markdown

```
#Import dataset
library(readr)
TISS_reviews <- read_csv("C:/Users/user/Desktop/reviews.csv")
```

```
## Rows: 304 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Reviewer, Review
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#view dataset
View(TISS_reviews)
```

```
#dimension of dataset
dim(TISS_reviews)
```

```
## [1] 304   2
```

```
#change the limit of the printing by using max.print option.
options(max.print = 100000)


#nchar will return the number of characters in a string.
nchar(TISS_reviews$Review)
```

```
##   [1] 1255  825  197  645  182  938  701  388  199  129   90   92  229  159  209
##  [16]  186  242  416  228  256  239  318  182  198   62  140  145  168   91  286
##  [31]  249  100  130  588  259  132  796   81  456  151   86  225   94  680   57
##  [46]   93  466  608   98   88  182  324   60  197  256   75   80 2394   81  375
##  [61]  158  188 1203  532   70   94  846  363  116   59  555   60  139   53  124
##  [76]   76   56   68   77  714   45   57  156  171   41  120  121  197  332   56
##  [91]  761  220  125   64   86  118  110  159   42   44  123   90   85   47   81
## [106]   62   65   34   47   63  144   30   45   58   88  197   47   88   24  133
## [121]   77   51  111  102   35   35   20   85  105   81   79   36   83  112   73
## [136]   52   60   42   45   44   60   82   69   48   55   53   57   19  111   51
## [151]  206   61   25   24   89   15  109   52   55   49   48   31   49   86   37
## [166]   33   59   45    9   55   56   63   35   15   21  132   28   64   30   37
## [181]   14   23   29   24   39   39   15   14   19    5   36   22    9   70   56
## [196]   20   67   12   29   37   33   54   44   14   14   37    2   25   14   19
## [211]   47    8   16   15   14   39   64   31   44    4   30   40   25   24   26
## [226]   30   45   43   31   37   22   32   20   34   37   25   16   36   21   37
## [241]   28   21   20   15   39   30    6   14    9   15    5   20   21   15   12
## [256]   36    4   32   14   15   16   28   14    4   27   22   28   11   21    6
## [271]   12    2   31    9   22    4    4   13   13   28   14   13   36   23    4
## [286]   29   21   21   13   10   11   20   16   11   13   16   14   12   10   10
## [301]   11   33   11   23
```

```
#Average number of characters
sum(nchar(TISS_reviews$Review))/304
```

```
## [1] 116.2599
```

```
#text-preprocessing functions from tm package.
library(tm)
```

```
## Loading required package: NLP
```

```
#The five most common tasks to be performed are
# 1.    lowering text,
# 2.    removing punctuation,
# 3.    stripping extra whitespace,
# 4.    removing numbers and
# 5.    removing "stopwords" - Stopwords are common words that often do not provide any additional insight.

getTransformations()
```

```
## [1] "removeNumbers"     "removePunctuation" "removeWords"
## [4] "stemDocument"      "stripWhitespace"
```

```
stopwords('english')
```

```
##   [1] "i"         "me"        "my"        "myself"    "we"
##   [6] "our"       "ours"      "ourselves" "you"       "your"
##  [11] "yours"     "yourself"  "yourselves" "he"        "him"
##  [16] "his"       "himself"   "she"       "her"       "hers"
##  [21] "herself"   "it"        "its"       "itself"    "they"
##  [26] "them"      "their"     "theirs"    "themselves" "what"
##  [31] "which"     "who"       "whom"      "this"      "that"
##  [36] "these"     "those"     "am"        "is"        "are"
##  [41] "was"       "were"      "be"        "been"      "being"
##  [46] "have"      "has"       "had"       "having"    "do"
##  [51] "does"      "did"       "doing"     "would"     "should"
##  [56] "could"     "ought"     "i'm"       "you're"    "he's"
##  [61] "she's"     "it's"      "we're"     "they're"   "i've"
##  [66] "you've"    "we've"     "they've"   "i'd"       "you'd"
##  [71] "he'd"      "she'd"     "we'd"      "they'd"    "i'll"
##  [76] "you'll"    "he'll"     "she'll"    "we'll"     "they'll"
##  [81] "isn't"     "aren't"    "wasn't"    "weren't"   "hasn't"
##  [86] "haven't"   "hadn't"    "doesn't"   "don't"     "didn't"
##  [91] "won't"     "wouldn't"  "shan't"    "shouldn't" "can't"
##  [96] "cannot"    "couldn't"  "mustn't"   "let's"     "that's"
## [101] "who's"     "what's"    "here's"    "there's"   "when's"
## [106] "where's"   "why's"     "how's"     "a"         "an"
## [111] "the"       "and"       "but"       "if"        "or"
## [116] "because"   "as"        "until"     "while"     "of"
## [121] "at"        "by"        "for"       "with"      "about"
## [126] "against"   "between"   "into"      "through"   "during"
## [131] "before"    "after"     "above"     "below"     "to"
## [136] "from"      "up"        "down"      "in"        "out"
## [141] "on"        "off"       "over"      "under"     "again"
## [146] "further"   "then"      "once"      "here"      "there"
## [151] "when"      "where"     "why"       "how"       "all"
## [156] "any"       "both"      "each"      "few"       "more"
## [161] "most"      "other"     "some"      "such"      "no"
## [166] "nor"       "not"       "only"      "own"       "same"
## [171] "so"        "than"      "too"       "very"
```

```
#Customised stopwords
custom.stopwords <- c(stopwords('english'), 'tata', 'tiss', 'institute')

#Apply various preprocessing functions.

#For tolower function, you have to add an additional argument,
#"content_transformer" because you are using a customized version of "tolower" which modifies the content of an R object.

#tm_map() takes two arguments, a corpus and a cleaning function.

#In the clean_corpus function, the order of operations is important.

#tm_map a transformation function taking a text document as input and returning a text document.
#The function content_transformer can be used to create a wrapper to get and set the content of text documents.

clean.corpus<-function(corpus){
  corpus <- tm_map(corpus, content_transformer(tolower))
  corpus <- tm_map(corpus, removeWords, custom.stopwords)
  corpus <- tm_map(corpus, removePunctuation)
  corpus <- tm_map(corpus, stripWhitespace)
  corpus <- tm_map(corpus, removeNumbers)
  return(corpus)}

#Define the reviews object as your corpus or collection of natural language documents.
#Make a vector source: Tiss .
#A vector source interprets each element or each row as a document.
TISS <- VectorSource(TISS_reviews[,2])

#Make a volatile corpus: TISScorpus.
#Corpus is the main structure that tm uses for storing and manipulating text documents.
#There are two types VCorpus (Volatile Corpus) and PCorpus (Permanent Corpus).
#A volatile corpus is fully kept in memory and thus all changes only affect the corresponding R object.
TISScorpus <- VCorpus(TISS)
TISScorpus
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 1
```

```
#Apply text cleaning in Volatile corpus.
reviews_clean<-clean.corpus(TISScorpus)
reviews_clean
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 1
```

```
#TermDocumentMatrix: bag of words text mining methodology
reviews_tdm<-TermDocumentMatrix(reviews_clean)
#View(reviews_tdm)


#Reclassify TDM as a matrix
reviews <- as.matrix(reviews_tdm)
#View(reviews)
dim(reviews)
```

```
## [1] 1108    1
```

```
#First check the dimensions of the data frame.
#You will notice that it is 1 column and 1108 rows.
#This means that in total there are 1108 distinct words after cleaning steps.
```

```
#Word frequency
#Sum across each row because each row is a unique word in the corpus.
#You need to call the base R function rowSums and pass the matrix to it.
word.freq<-rowSums(reviews)

#Create a new data frame object by putting the original words and the term frequencies next to each other. The new freq.df o
bject has two columns, one for terms and another that is just the summation for each row.
freq.df<-data.frame(word = names(word.freq), frequency = word.freq)
freq.df<-freq.df[order(freq.df[,2], decreasing=T),]

#Now you can sort the data frame and then look at the first ten most frequent terms.
freq.df[1:10,]
```

```
##                word frequency
## social       social       114
## best           best        92
## campus       campus        83
## place         place        66
## great         great        49
## india         india        48
## sciences   sciences        47
## good           good        42
## one             one        42
## education education        39
```

```
#Word Clouds. A word cloud is a visualization based on frequency. In a word cloud, words are represented with varying font s
ize.
#The code snippet will create a word cloud with a maximum number of terms equal to 100. You can also specify the minimum fre
quency threshold as well. The code also specifies blue and darkred colours (color words from least to most frequent). You ca
n change the colors manually.
m <- as.matrix(reviews)
View(m)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq = v)
View(d)
```

```
#Display the TISS word Cloud
library(wordcloud2)
wordcloud2(d)
```


social
campus
best
place
great
good
sciences
studies
education
india
work
students
research
one
science
development
school
nice
dorabji
beautiful
amazing
excellent
institutes
institutions
institution
experience
college
courses
human
people
life
student
universities
public
green
academic
professional
knowledge
multicampus
management
faculty
programs
staff
hostel
policy
awesome
quality
like
terms
indias
trees
learn
oldest
health
graduate
library
many
proud
know
teaching
two
sir
new
can
old
also
way
course
trust
culture
deemed
since
service
reputed
world
highly
greenery
wonderful
admission
provides
international
friendly
different
premier
everything
established
humanities
presidency
including
field
make
years
high
area
focus
love
get
food
full
asias
seems
offers
part
british
tissian
every
studying
founded
support
unique
learning
economics
excellence
atmosphere
doctoral
around
global
facilities
teachers
actually
things
future
administration
bombay
within
based
indian
want
action
labour
history
group
collaboration
collaborations