JohnYe_HW5Part2

John Ye

2024-10-29

```
library(tokenizers)
library(SnowballC)
library(tm)

library(quanteda)

## Package version: 4.1.0
## Unicode version: 15.1
## ICU version: 74.1

## Parallel computing: 32 of 32 threads used.

## See https://quanteda.io for tutorials and examples.

##

## The following object is masked from 'package:tm':
##
##     stopwords

## The following objects are masked from 'package:NLP':
##
##     meta, meta<-

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────────
tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2

## ── Conflicts ────────────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ ggplot2::annotate() masks NLP::annotate()
## ✖ dplyr::filter()     masks stats::filter()
## ✖ dplyr::lag()        masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to
force all conflicts to become errors
```

1. Tokenization

```r
# Read the data from the csv file
news <- read.csv("bbc.csv")
#head(news)

# Tokenize the news
# create a vector containing only text
text <- news$text
# create a corpus
corpus <- VCorpus(VectorSource(text))

corpus <- corpus %>%
  tm_map(content_transformer(tolower)) %>%
  tm_map(removeNumbers) %>%
  tm_map(removePunctuation) %>%
  tm_map(stripWhitespace) %>%
  tm_map(content_transformer(wordStem), language = "en")

# Create Document-Term Matrix
dtm <- DocumentTermMatrix(corpus)
dtmMatrix <- as.matrix(dtm)

# Get the frequency of the works
frequencies <- colSums(dtmMatrix)

# Remove 15% of the words with the least frequency in the document
leastWord <- quantile(frequencies, probs = 0.15)
dtmMatrix <- dtmMatrix[,frequencies > leastWord]

# print the feature vector of the words that are appear 4 or more times
in the 2205th article in the dataset
article <- dtmMatrix[2205, ]
featureVector <- article[article >= 4]
print(featureVector)
```

```
##         and         are        base   eduvision     eslates
from
##          12           7           4           5           5
4
##      herren information     project   satellite        says
school
##           4           4           5           5           4
8
##    students         the        this        with
##           5          43           5           4
```

2. Classification

```
library(e1071)
library(nnet)
library(caret)

## The following object is masked from 'package:purrr':
##
##     lift

# Reduce the features set by removing highly correlated features
dtmDataFrame <- as.data.frame(dtmMatrix)
correlatedFeature <- cor(dtmDataFrame)
highCorrelation <- findCorrelation(correlatedFeature, cutoff = 0.99)
newDTM <- dtmDataFrame[, -highCorrelation]
newDTM$category <- news$category
```

Naive Bayes

```
# Split the data where 80% for training and 20% for testing
set.seed(1)
trainIndex <- createDataPartition(newDTM$category, p=0.8, list = FALSE)
train <- newDTM[trainIndex, ]
test <- newDTM[-trainIndex, ]
X_train <- train[, -ncol(train)]
y_train <- train$category
X_test <- test[, -ncol(test)]
y_test <- test$category

# Build a Multinomial Naive Bayes classifier
nb.fit <- naiveBayes(X_train, y_train, laplace = 1)
#nb.fit

# Predict using test data
nb.class <- predict(nb.fit, X_test)
# Print a confusion matrix
y_test <- factor(y_test)
nb.class <- factor(nb.class, levels = levels(y_test))
nb.matrix <- confusionMatrix(nb.class, y_test)
print(nb.matrix$byClass[, c("Precision", "Recall")])
```

```
##                      Precision    Recall
## Class: business      0.8333333 0.3921569
## Class: entertainment 0.5892857 0.4285714
## Class: politics      0.9444444 0.4096386
## Class: sport         0.3322034 0.9607843
## Class: tech          1.0000000 0.1125000
```

Logistic Regression (gives me a “protection stack overflow” error)

```
#{r} #lr.fit <- multinom(category ~ ., data = X_train, MaxNWts = 50000)
#lr.class <- predict(lr.fit, X_test) #y_test <- factor(y_test)
```

```
#lr.class <- factor(lr.class, levels = levels(y_test)) #lr.matrix <-
confusionMatrix(lr.class, y_test) #print(lr.matrix$byClass[,
c("Precision", "Recall")]) #
```

Problem with Logistic Regression

- The logistic regression gave me a protection stack overflow error. I tried to use “MaxNWts” to limits the number of weights, use “glmnet”, and PCA, but none of these methods worked.