Assign_5_Solutions

Helen Catanese

November 5, 2018

Setup and library imports

```
# Key 85c8fdb6-c036-406d-b379-8bf5f42abec6

library(RJSONIO)
library(class)
library(e1071)
library(RTextTools)
library(tm)
library(tidytext)
library(caret)
```

Question 1

a

```
probX = (exp(-7 + 0.1 * 32 + 3 + -0.04 * 12))/(1 + exp(-7 + 0.1 *
    32 + 3 + -0.04 * 12))
probX
```

```
## [1] 0.2175502
```

b

You need to do some simple algebra here. If exp(-7+0.1*X+3.0+-.02*12))/(1+exp(-7+0.1*X+3.0+-.02*12)=.5, then exp(-7+0.1*X+3.0+-.02*12) = 1.The natural log of this gives you -7+0.1*X+3.0+-.02*12=0. And finally X=(7-3+.02)/0.1

```
hours = (7 - 3 + 0.04 * 12)/(0.1)
hours
```

```
## [1] 44.8
```

c

Following the same procedure, we get:

```
hours = (7 - 3 + 0.04 * 3)/(0.1)
hours
```

```
## [1] 41.2
```

Question 2

Parts a & b

```
data <- setNames(data.frame(matrix(ncol = 5, nrow = 0)), c("id",
    "title", "body", "wc", "section"))

i = 1
section = "culture"
sections = c("artanddesign", "business", "culture", "sport",
    "technology", "world")

for (i in 1:10) {
    for (section in sections) {
        newdata <- fromJSON(paste("http://content.guardianapis.com/search?section=",
            section, "&show-fields=wordcount%2Cbody&page=", i,
            "&page-size=200&api-key=85c8fdb6-c036-406d-b379-8bf5f42abec6",
            sep = ""))$response$results

        for (j in 1:200) {
            if (newdata[[j]]["type"] != "article") {
                # print (paste('Non-article
                # type',newdata[[j]]['type'],'removed.'))
                next
            }
            data[nrow(data) + 1, ] <- c(newdata[[j]]$id, newdata[[j]]$webTitle,
                newdata[[j]]$fields[[1]], newdata[[j]]$fields[[2]],
                newdata[[j]]$sectionId)
            Encoding(data[nrow(data), ][["body"]]) <- "UTF-8"
            Encoding(data[nrow(data), ][["title"]]) <- "UTF-8"
            # data[nrow(data),]$body<-
            data[nrow(data), ]$body <- gsub("<.*?>", "", data[nrow(data),
                ][["body"]])
            data[nrow(data), ]$body <- gsub("[^[:alnum:][:space:]]",
                "", data[nrow(data), ][["body"]])
            data[nrow(data), ]$body <- tolower(data[nrow(data),
                ][["body"]])
            # data[nrow(data),]$body<-gsub('[\\n]+',' ',
            # data[nrow(data),][['body']])
        }

    }
}

# write.csv(data, 'GuardianArticles.csv')
```

```
# print article 137
strwrap(data[137, ]$body, width = 80)
```

```
##  [1] "rip it up in this neat inversion of the bible story the temptress delilah is"
##  [2] "transformed into a freedom fighter the underdog who slays the giant brave new"
##  [3] "girl the girl with her louise brooks bob and bandit mask is a familiar dzama"
##  [4] "creation alongside bears and treemen masked and armed girl gangs have long"
##  [5] "populated his drawings where childhood makebelieve collides with adult"
```

```
##  [6] "brutality the time is now created in 2017 this drawings politics are however"
##  [7] "wellattuned to a world where the us presidents locker room talk had recently"
##  [8] "been aired seeing red the characters antique palette is steeped in the visual"
##  [9] "language of old movies the sharp background recalls both traditional sweets and"
## [10] "the agitprop graphic design of russian constructivism strangely sweet since the"
## [11] "early 2000s the winnipegraised artists mix of whimsy sex and sadism has made"
## [12] "him a star his artistic universe has expanded to include dolls dioramas and"
## [13] "film photograph dan bradicacourtesy of the artist st albans museum gallery"
## [14] "included in hand drawn action packed st albans museum art gallery to 11"
## [15] "november"
```

#Part c

```
corpus <- Corpus(VectorSource(data$body))

# build a stemmed term document matrix
dtm = DocumentTermMatrix(corpus, control = list(removeNumbers = TRUE,
    stopwords = TRUE, stemming = TRUE))

# print a single row forarticle 137
as.matrix(dtm[137, which(as.matrix(dtm[137, ]) != 0)])
```

```
##      Terms
## Docs  action adult agitprop air alban alongsid antiqu arm art artist
##   137      1     1        1   1     2        1      1   1   1      3
##      Terms
## Docs  background bandit bear bibl bob bradicacourtesi brave brook brutal
##   137          1      1    1    1   1               1     1     1      1
##      Terms
## Docs  charact childhood collid constructiv creat creation dan delilah
##   137       1         1      1           1     1        1   1       1
##      Terms
## Docs  design diorama doll draw drawn dzama earli expand familiar fighter
##   137      1       1    1    2     1     1     1      1        1       1
##      Terms
## Docs  film freedom galleri gang giant girl graphic hand howev includ
##   137    1       1       2    1     1    3       1    1     1      2
##      Terms
## Docs  invers languag locker long louis made makebeliev mask mix movi
##   137      1       1       1    1     1    1          1    2   1    1
##      Terms
## Docs  museum neat new novemb now old pack palett photograph polit popul
##   137      2    1   1      1   1   1    1      1          1     1     1
##      Terms
## Docs  presid recal recent red rip room russian sadism see sex sharp sinc
##   137      1     1      1   1   1    1       1      1   1   1     1    1
##      Terms
## Docs  slay star steep stori strang sweet talk temptress time tradit
##   137    1    1     1     1      1     2    1         1    1      1
##      Terms
## Docs  transform treemen underdog univers visual wellattun whimsi
##   137         1       1        1       1      1         1      1
##      Terms
## Docs  winnipegrais world
##   137            1     1
```

#Part d

```
# first remove words that appear in too few documents
dtm <- removeSparseTerms(dtm, 0.99)

# also remove correlated terms
correlation_matrix = cor(as.matrix(dtm))
correlated_terms = findCorrelation(correlation_matrix, cutoff = 0.85)
correlated_terms = sort(correlated_terms)
dtm = dtm[, -c(correlated_terms)]

# split test and training data Note, these are only not
# randomized because they were not sorted by section
# originally
dtm.train = dtm[1:9000, ]
dtm.test = dtm[9001:11458, ]
corpus.train = corpus[1:9000]
corpus.test = corpus[9001:11458]
data.train = data[1:9000, ]
data.test = data[9001:11458, ]
data.train$section = as.factor(data.train$section)
data.test$section = as.factor(data.test$section)
```

```
# build your model
m <- naiveBayes(as.matrix(dtm.train), data.train$section)

# generate predictions
p = predict(m, as.matrix(dtm.test))

# create a confusion matrix, and compute prec/recall
confusionMatrix(p, data.test$section)
```

```
## Confusion Matrix and Statistics
##
##               Reference
## Prediction     artanddesign business culture sport technology world
##   artanddesign          317        3     119     5          8    20
##   business               18      333      16     5         66    84
##   culture                33        3     190    10          9    27
##   sport                  14        1      39   321         11    25
##   technology              8       21       8     0        292    31
##   world                  10        6      10     5         13   377
##
## Overall Statistics
##
##                 Accuracy : 0.7445
##                   95% CI : (0.7268, 0.7617)
##     No Information Rate : 0.2295
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                    Kappa : 0.6934
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
```

```
##                      Class: artanddesign Class: business Class: culture
## Sensitivity                       0.7925          0.9074         0.4974
## Specificity                       0.9247          0.9096         0.9605
## Pos Pred Value                    0.6716          0.6379         0.6985
## Neg Pred Value                    0.9582          0.9824         0.9122
## Prevalence                        0.1627          0.1493         0.1554
## Detection Rate                    0.1290          0.1355         0.0773
## Detection Prevalence              0.1920          0.2124         0.1107
## Balanced Accuracy                 0.8586          0.9085         0.7289
##                      Class: sport Class: technology Class: world
## Sensitivity                0.9277            0.7318       0.6684
## Specificity                0.9574            0.9670       0.9768
## Pos Pred Value             0.7810            0.8111       0.8955
## Neg Pred Value             0.9878            0.9490       0.9082
## Prevalence                 0.1408            0.1623       0.2295
## Detection Rate             0.1306            0.1188       0.1534
## Detection Prevalence       0.1672            0.1465       0.1713
## Balanced Accuracy          0.9426            0.8494       0.8226
```