rm(list = ls())

Data = read.csv("Yelp\_train.csv", stringsAsFactors = FALSE)

n = dim(Data)[1] # The number of rows of the data.frame

# Assign each observation a ID No.

Data$id = integer(n)

for (i in seq\_len(n)) {

Data$id[i] = i

}

# This function detects whether a character is using standard ASCII value.

# Returns true on ASCII, false on non-ASCII

isASCII = function(txt) {

return( all(charToRaw(txt) <= as.raw(127)) )

}

# This function counts the number of non-ASCII characters in a String of characters.

# Returns the count

ASCIICount = function(chars) {

count = 0

for (i in seq\_len(length(chars))) {

if (!isASCII(chars[i])) {

count = count + 1

}

}

return(count)

}

# Detect whether there are non-English comments in the file

# Record observation number of each non-English comment

num.lines = numeric(0)

for (i in 1:length(Data$text)) {

line = Data$text[i];

chars = strsplit(line, "")[[1]]

if (ASCIICount(chars) > 7) {

num.lines = append(num.lines, i)

}

}

# Display all non-English comments

num.lines = num.lines[-c(1, 2, 4)]

Data$text[num.lines]

Data$id[num.lines]

# Remove all non-English comments

Data = Data[-num.lines, ]

phrases = list()

words = list()

# Generate a list of phrases and words for all comments

for (i in seq\_len(n)) {

phrases[[i]] = unlist(strsplit(Data$text[i], ",|\\.|\\?|!|\""))

words[[i]] = unlist(strsplit(Data$text[i], " +"))

}

# This function is used to create a new predictor

# First it counts the number of occurances the word is in the text

# Then it updates on the specific cell in the data.frame

# Finally it prints out how many observations contain at least one occurance of this word

# option 1: look into words

# option 2: look into phrases

new.var.count = function(Data, word, option = 1) {

n = length(Data$text)

Data[[word]] = integer(n)

if (option == 1) {

# option 1: words

for (i in seq\_len(n)) {

Data[[word]][i] = length(grep(word, words[[i]], ignore.case = TRUE))

}

} else if (option == 2) {

# option 2: phrases

for (i in seq\_len(n)) {

Data[[word]][i] = length(grep(word, phrases[[i]], ignore.case = TRUE))

}

}

print(sum(Data[[word]] != 0))

return(Data)

}

# Adding new predictors...

Data = new.var.count(Data, "appealing") # 141

Data = new.var.count(Data, "\\<appeti") # 2275

Data = new.var.count(Data, "(\\<delicious)|(delectable)|(tasty)|(yum)|(\\<tasteful)") # 9695

Data = new.var.count(Data, "choice") # 1819

Data = new.var.count(Data, "excellent") # 2981

Data = new.var.count(Data, "(superior)|(superb)") # 341

Data = new.var.count(Data, "(top( |-)notch)|(first( |-)rate)", option = 2) # 353

Data = new.var.count(Data, "high( |-)quality", option = 2) # 290

Data = new.var.count(Data, "(mouth( |-)watering)|(scrumptious)|(luscious)", option = 2) # 161

Data = new.var.count(Data, "(\\<enjoy)|(\\<palatable)|(\\<delight)|(\\<pleasing)|(\\<satisf)|(\\<pleasant)") # 5726

Data = new.var.count(Data, "nausea") # 17

Data = new.var.count(Data, "\\<flav(o|ou)r[^l]") # 3602

Data = new.var.count(Data, "(distasteful)|(repulsive)|(sickening)|(unappetizing)|(unsavory)") # 67

Data = new.var.count(Data, "(rancid)|(stale)|(rotten)") # 209

Data = new.var.count(Data, "bad") # 3364

Data = new.var.count(Data, "not bad", option = 2) # 361

Data$bad = Data$bad - Data$`not bad` #3057

Data = new.var.count(Data, "(holy)|(wow)|(god)") # 1127

Data = new.var.count(Data, "(nice)|(happy)|(easy)") # 8994

Data = new.var.count(Data, "(wonderful)|(beautiful)|(best)") # 8306

Data = new.var.count(Data, "lovely") # 540

Data = new.var.count(Data, "better") # 4305

Data = new.var.count(Data, "(like)|(love\\>)|(great)") # 22117

Data = new.var.count(Data, "(look.\* forward)|(worth)", option = 2) # 3224

Data = new.var.count(Data, "(many)|(plenty of)", option = 2) # 3661

Data = new.var.count(Data, "well") # 6009

Data = new.var.count(Data, "fine") # 1559

Data = new.var.count(Data, "cheap") # 1352

Data = new.var.count(Data, "(friendly)|(welcome)|(attentive)|(passion)") # 6547

Data = new.var.count(Data, "(unique)|(creative)") # 1303

Data = new.var.count(Data, "(fresh)|(authentic)|(healthy)") # 5786

Data = new.var.count(Data, "disappoint") # 3098

Data = new.var.count(Data, "return") # 1679

Data = new.var.count(Data, "not good", option = 2) # 323

Data = new.var.count(Data, "good") # 16592

Data$good = Data$good - Data$`not good` # 16450

Data = new.var.count(Data, "(noisy)|(dirty)|(nothing)") # 2954

Data = new.var.count(Data, "(quiet)|(comfortable)") # 1111

Data = new.var.count(Data, "(never)|(forever)") # 4557

Data = new.var.count(Data, "(but)|(however)") # 21441

Data = new.var.count(Data, "(terrible)|(trouble)|(weird)") # 1445

Data = new.var.count(Data, "reserv") # 996

Data = new.var.count(Data, "crowded") # 755

Data = new.var.count(Data, "(wait)|(even)|(slow)") # 12699

Data = new.var.count(Data, "have to", option = 2) # 2169

Data = new.var.count(Data, "(worse)|(awful)") # 783

Data = new.var.count(Data, "expensive") # 1059

Data = new.var.count(Data, "serious") # 936

Data = new.var.count(Data, "awesome") # 2421

Data = new.var.count(Data, "average") # 1135

Data = new.var.count(Data, "(clean)|(tidy)|(fast)") # 5177

Data = new.var.count(Data, "(center)|(convenient)") # 568

Data = new.var.count(Data, "close") # 1896

Data = new.var.count(Data, "pretty") # 4944

Data = new.var.count(Data, "(definitely)|(truly)|(especially)") # 6174

Data = new.var.count(Data, "(though)|(while)") # 8875

Data = new.var.count(Data, "wrong") # 1215

Data = new.var.count(Data, "small") # 3702

Data = new.var.count(Data, "(avoid)|(skip)") # 859

Data = new.var.count(Data, "insane") # 117

Data = new.var.count(Data, "back") # 8800

Data = new.var.count(Data, "come") # 5610

Data = new.var.count(Data, "!") # 15663

Data = new.var.count(Data, "\\?") # 4647

Data = new.var.count(Data, "(outstanding)|(extraordinary)") # 755

Data = new.var.count(Data, "interesting") # 891

Data = new.var.count(Data, "recommend") # 3816 needs interaction

Data = new.var.count(Data, "slip") # 79

Data = new.var.count(Data, "limit") # 691

Data = new.var.count(Data, "so( |-)so", option = 2) # 500

write.csv(Data, "Data.csv", row.names = FALSE)

Data = read.csv("Data.csv", stringsAsFactors = FALSE)

n = dim(Data)[1] # The number of rows of the data.frame

### ID is the 43th column in the dataset

# The code below generates a spread sheet with 101 predictors (X1 - X101)

# and the word corresponding to the names of predictors

iters = c(5:7, 13:42, 44:dim(Data)[2])

k = integer(1)

predictor.table = data.frame(predictor = character(101), idno = 1:101, stringsAsFactors = FALSE)

for (i in iters) {

k = k + 1

predictor.table$predictor[k] = names(Data)[i]

names(Data)[i] = paste("X", k, sep = "")

}

write.csv(predictor.table, "predictor\_table.csv", row.names = FALSE)