rm(list = ls())

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)

############################################################

imputed\_missForest = read.csv("imputed\_missForest.csv", stringsAsFactors = FALSE)

Data$X5 = imputed\_missForest$X5

model = lm(stars ~ X1 + X2 + X3 + X4 + X5 + X15 + X18 + X23 + X24 + X25 + X26 + X27 + X29 + X30 + X33 + X36 + X38 +

X52 + X54 + X55 + X61 + X63 + X66 + X68 + X71 + X75 + X77 + X86 + X94 + X71:X75, data = Data)

############################################################

Test1 = read.csv("Yelp\_test.csv", stringsAsFactors = FALSE)

Test2 = read.csv("Yelp\_validate.csv", stringsAsFactors = FALSE)

n1 = dim(Test1)[1]

n2 = dim(Test2)[1]

phrases1 = list()

phrases2 = list()

words1 = list()

words2 = list()

# Generate a list of phrases and words for all comments

for (i in seq\_len(n1)) {

phrases1[[i]] = unlist(strsplit(Test1$text[i], ",|\\.|\\?|!|\""))

words1[[i]] = unlist(strsplit(Test1$text[i], " +"))

}

for (i in seq\_len(n2)) {

phrases2[[i]] = unlist(strsplit(Test2$text[i], ",|\\.|\\?|!|\""))

words2[[i]] = unlist(strsplit(Test2$text[i], " +"))

}

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)

}

words = words1

phrases = phrases1

Test1 = new.var.count(Test1, "appealing") # 141

Test1 = new.var.count(Test1, "\\<appeti") # 2275

Test1 = new.var.count(Test1, "(\\<delicious)|(delectable)|(tasty)|(yum)|(\\<tasteful)") # 9695

Test1 = new.var.count(Test1, "choice") # 1819

Test1 = new.var.count(Test1, "excellent") # 2981

Test1 = new.var.count(Test1, "(superior)|(superb)") # 341

Test1 = new.var.count(Test1, "(top( |-)notch)|(first( |-)rate)", option = 2) # 353

Test1 = new.var.count(Test1, "high( |-)quality", option = 2) # 290

Test1 = new.var.count(Test1, "(mouth( |-)watering)|(scrumptious)|(luscious)", option = 2) # 161

Test1 = new.var.count(Test1, "(\\<enjoy)|(\\<palatable)|(\\<delight)|(\\<pleasing)|(\\<satisf)|(\\<pleasant)") # 5726

Test1 = new.var.count(Test1, "nausea") # 17

Test1 = new.var.count(Test1, "\\<flav(o|ou)r[^l]") # 3602

Test1 = new.var.count(Test1, "(distasteful)|(repulsive)|(sickening)|(unappetizing)|(unsavory)") # 67

Test1 = new.var.count(Test1, "(rancid)|(stale)|(rotten)") # 209

Test1 = new.var.count(Test1, "bad") # 3364

Test1 = new.var.count(Test1, "not bad", option = 2) # 361

Test1$bad = Test1$bad - Test1$`not bad` #3057

Test1 = new.var.count(Test1, "(holy)|(wow)|(god)") # 1127

Test1 = new.var.count(Test1, "(nice)|(happy)|(easy)") # 8994

Test1 = new.var.count(Test1, "(wonderful)|(beautiful)|(best)") # 8306

Test1 = new.var.count(Test1, "lovely") # 540

Test1 = new.var.count(Test1, "better") # 4305

Test1 = new.var.count(Test1, "(like)|(love\\>)|(great)") # 22117

Test1 = new.var.count(Test1, "(look.\* forward)|(worth)", option = 2) # 3224

Test1 = new.var.count(Test1, "(many)|(plenty of)", option = 2) # 3661

Test1 = new.var.count(Test1, "well") # 6009

Test1 = new.var.count(Test1, "fine") # 1559

Test1 = new.var.count(Test1, "cheap") # 1352

Test1 = new.var.count(Test1, "(friendly)|(welcome)|(attentive)|(passion)") # 6547

Test1 = new.var.count(Test1, "(unique)|(creative)") # 1303

Test1 = new.var.count(Test1, "(fresh)|(authentic)|(healthy)") # 5786

Test1 = new.var.count(Test1, "disappoint") # 3098

Test1 = new.var.count(Test1, "return") # 1679

Test1 = new.var.count(Test1, "not good", option = 2) # 323

Test1 = new.var.count(Test1, "good") # 16592

Test1$good = Test1$good - Test1$`not good` # 16450

Test1 = new.var.count(Test1, "(noisy)|(dirty)|(nothing)") # 2954

Test1 = new.var.count(Test1, "(quiet)|(comfortable)") # 1111

Test1 = new.var.count(Test1, "(never)|(forever)") # 4557

Test1 = new.var.count(Test1, "(but)|(however)") # 21441

Test1 = new.var.count(Test1, "(terrible)|(trouble)|(weird)") # 1445

Test1 = new.var.count(Test1, "reserv") # 996

Test1 = new.var.count(Test1, "crowded") # 755

Test1 = new.var.count(Test1, "(wait)|(even)|(slow)") # 12699

Test1 = new.var.count(Test1, "have to", option = 2) # 2169

Test1 = new.var.count(Test1, "(worse)|(awful)") # 783

Test1 = new.var.count(Test1, "expensive") # 1059

Test1 = new.var.count(Test1, "serious") # 936

Test1 = new.var.count(Test1, "awesome") # 2421

Test1 = new.var.count(Test1, "average") # 1135

Test1 = new.var.count(Test1, "(clean)|(tidy)|(fast)") # 5177

Test1 = new.var.count(Test1, "(center)|(convenient)") # 568

Test1 = new.var.count(Test1, "close") # 1896

Test1 = new.var.count(Test1, "pretty") # 4944

Test1 = new.var.count(Test1, "(definitely)|(truly)|(especially)") # 6174

Test1 = new.var.count(Test1, "(though)|(while)") # 8875

Test1 = new.var.count(Test1, "wrong") # 1215

Test1 = new.var.count(Test1, "small") # 3702

Test1 = new.var.count(Test1, "(avoid)|(skip)") # 859

Test1 = new.var.count(Test1, "insane") # 117

Test1 = new.var.count(Test1, "back") # 8800

Test1 = new.var.count(Test1, "come") # 5610

Test1 = new.var.count(Test1, "!") # 15663

Test1 = new.var.count(Test1, "\\?") # 4647

Test1 = new.var.count(Test1, "(outstanding)|(extraordinary)") # 755

Test1 = new.var.count(Test1, "interesting") # 891

Test1 = new.var.count(Test1, "recommend") # 3816 needs interaction

Test1 = new.var.count(Test1, "slip") # 79

Test1 = new.var.count(Test1, "limit") # 691

Test1 = new.var.count(Test1, "so( |-)so", option = 2) # 500

words = words2

phrases = phrases2

Test2 = new.var.count(Test2, "appealing") # 141

Test2 = new.var.count(Test2, "\\<appeti") # 2275

Test2 = new.var.count(Test2, "(\\<delicious)|(delectable)|(tasty)|(yum)|(\\<tasteful)") # 9695

Test2 = new.var.count(Test2, "choice") # 1819

Test2 = new.var.count(Test2, "excellent") # 2981

Test2 = new.var.count(Test2, "(superior)|(superb)") # 341

Test2 = new.var.count(Test2, "(top( |-)notch)|(first( |-)rate)", option = 2) # 353

Test2 = new.var.count(Test2, "high( |-)quality", option = 2) # 290

Test2 = new.var.count(Test2, "(mouth( |-)watering)|(scrumptious)|(luscious)", option = 2) # 161

Test2 = new.var.count(Test2, "(\\<enjoy)|(\\<palatable)|(\\<delight)|(\\<pleasing)|(\\<satisf)|(\\<pleasant)") # 5726

Test2 = new.var.count(Test2, "nausea") # 17

Test2 = new.var.count(Test2, "\\<flav(o|ou)r[^l]") # 3602

Test2 = new.var.count(Test2, "(distasteful)|(repulsive)|(sickening)|(unappetizing)|(unsavory)") # 67

Test2 = new.var.count(Test2, "(rancid)|(stale)|(rotten)") # 209

Test2 = new.var.count(Test2, "bad") # 3364

Test2 = new.var.count(Test2, "not bad", option = 2) # 361

Test2$bad = Test2$bad - Test2$`not bad` #3057

Test2 = new.var.count(Test2, "(holy)|(wow)|(god)") # 1127

Test2 = new.var.count(Test2, "(nice)|(happy)|(easy)") # 8994

Test2 = new.var.count(Test2, "(wonderful)|(beautiful)|(best)") # 8306

Test2 = new.var.count(Test2, "lovely") # 540

Test2 = new.var.count(Test2, "better") # 4305

Test2 = new.var.count(Test2, "(like)|(love\\>)|(great)") # 22117

Test2 = new.var.count(Test2, "(look.\* forward)|(worth)", option = 2) # 3224

Test2 = new.var.count(Test2, "(many)|(plenty of)", option = 2) # 3661

Test2 = new.var.count(Test2, "well") # 6009

Test2 = new.var.count(Test2, "fine") # 1559

Test2 = new.var.count(Test2, "cheap") # 1352

Test2 = new.var.count(Test2, "(friendly)|(welcome)|(attentive)|(passion)") # 6547

Test2 = new.var.count(Test2, "(unique)|(creative)") # 1303

Test2 = new.var.count(Test2, "(fresh)|(authentic)|(healthy)") # 5786

Test2 = new.var.count(Test2, "disappoint") # 3098

Test2 = new.var.count(Test2, "return") # 1679

Test2 = new.var.count(Test2, "not good", option = 2) # 323

Test2 = new.var.count(Test2, "good") # 16592

Test2$good = Test2$good - Test2$`not good` # 16450

Test2 = new.var.count(Test2, "(noisy)|(dirty)|(nothing)") # 2954

Test2 = new.var.count(Test2, "(quiet)|(comfortable)") # 1111

Test2 = new.var.count(Test2, "(never)|(forever)") # 4557

Test2 = new.var.count(Test2, "(but)|(however)") # 21441

Test2 = new.var.count(Test2, "(terrible)|(trouble)|(weird)") # 1445

Test2 = new.var.count(Test2, "reserv") # 996

Test2 = new.var.count(Test2, "crowded") # 755

Test2 = new.var.count(Test2, "(wait)|(even)|(slow)") # 12699

Test2 = new.var.count(Test2, "have to", option = 2) # 2169

Test2 = new.var.count(Test2, "(worse)|(awful)") # 783

Test2 = new.var.count(Test2, "expensive") # 1059

Test2 = new.var.count(Test2, "serious") # 936

Test2 = new.var.count(Test2, "awesome") # 2421

Test2 = new.var.count(Test2, "average") # 1135

Test2 = new.var.count(Test2, "(clean)|(tidy)|(fast)") # 5177

Test2 = new.var.count(Test2, "(center)|(convenient)") # 568

Test2 = new.var.count(Test2, "close") # 1896

Test2 = new.var.count(Test2, "pretty") # 4944

Test2 = new.var.count(Test2, "(definitely)|(truly)|(especially)") # 6174

Test2 = new.var.count(Test2, "(though)|(while)") # 8875

Test2 = new.var.count(Test2, "wrong") # 1215

Test2 = new.var.count(Test2, "small") # 3702

Test2 = new.var.count(Test2, "(avoid)|(skip)") # 859

Test2 = new.var.count(Test2, "insane") # 117

Test2 = new.var.count(Test2, "back") # 8800

Test2 = new.var.count(Test2, "come") # 5610

Test2 = new.var.count(Test2, "!") # 15663

Test2 = new.var.count(Test2, "\\?") # 4647

Test2 = new.var.count(Test2, "(outstanding)|(extraordinary)") # 755

Test2 = new.var.count(Test2, "interesting") # 891

Test2 = new.var.count(Test2, "recommend") # 3816 needs interaction

Test2 = new.var.count(Test2, "slip") # 79

Test2 = new.var.count(Test2, "limit") # 691

Test2 = new.var.count(Test2, "so( |-)so", option = 2) # 500

iters = c(5:7, 13:42, 44:dim(Test1)[2])

k = integer(1)

for (i in iters) {

k = k + 1

names(Test1)[i] = paste("X", k, sep = "")

}

predictors1 = data.frame(X1 = Test1$X1, X2 = Test1$X2, X3 = Test1$X3, X4 = Test1$X4, X5 = Test1$X5, X15 = Test1$X15,

X18 = Test1$X18, X23 = Test1$X23, X24 = Test1$X24, X25 = Test1$X25, X26 = Test1$X26,

X27 = Test1$X27, X29 = Test1$X29, X30 = Test1$X30, X33 = Test1$X33, X36 = Test1$X36,

X38 = Test1$X38, X52 = Test1$X52, X54 = Test1$X54, X55 = Test1$X55, X61 = Test1$X61,

X63 = Test1$X63, X66 = Test1$X66, X68 = Test1$X68, X71 = Test1$X71, X75 = Test1$X75,

X77 = Test1$X77, X86 = Test1$X86, X94 = Test1$X94)

result1 = predict(model, predictors1)

result.frame1 = data.frame(Id = 1:dim(Test1)[1], Prediction = result1)

write.csv(result.frame1, "group1\_1.csv", row.names = FALSE)

iters = c(5:7, 13:42, 44:dim(Test2)[2])

k = integer(1)

for (i in iters) {

k = k + 1

names(Test2)[i] = paste("X", k, sep = "")

}

predictors2 = data.frame(X1 = Test2$X1, X2 = Test2$X2, X3 = Test2$X3, X4 = Test2$X4, X5 = Test2$X5, X15 = Test2$X15,

X18 = Test2$X18, X23 = Test2$X23, X24 = Test2$X24, X25 = Test2$X25, X26 = Test2$X26,

X27 = Test2$X27, X29 = Test2$X29, X30 = Test2$X30, X33 = Test2$X33, X36 = Test2$X36,

X38 = Test2$X38, X52 = Test2$X52, X54 = Test2$X54, X55 = Test2$X55, X61 = Test2$X61,

X63 = Test2$X63, X66 = Test2$X66, X68 = Test2$X68, X71 = Test2$X71, X75 = Test2$X75,

X77 = Test2$X77, X86 = Test2$X86, X94 = Test2$X94)

result2 = predict(model, predictors2)

result.frame2 = data.frame(Id = 1:dim(Test2)[1], Prediction = result2)

write.csv(result.frame2, "group1\_2.csv", row.names = FALSE)

id = 1:(n1 + n2)

k = integer(1)

m = integer(1)

result.frame = data.frame(Id = id, Prediction = numeric(length(id)))

for (i in id) {

k = k + 1

if (k <= n1) {

result.frame$Prediction[i] = result.frame1$Prediction[k]

} else {

m = m + 1

result.frame$Prediction[i] = result.frame2$Prediction[m]

}

}

write.csv(result.frame, "group1.csv", row.names = FALSE)