The Appendix

```
library(acstats)
library(RJSONIO)
library(RCurl)
library(ngram)
library(tm)
library(wordcloud)
library(ggplot2)
library(glmnet)
library(text2vec)
library(data.table)
library(magrittr)
library(dplyr)
library(mosaic)
library(RTextTools)
library(e1071)
# library(caret)
library(tidyr)
library(stringr)
library(jsonlite)
library(tidytext)
library(rpart)
library(broom)
library(kableExtra)
#library(MASS) #lda
```

```
#Loading Data

setwd("~/Desktop/Statistics/Comps/Comps - Fayorsey/Comps-Fayorsey19E")
data <- read.csv("cleaned_hm.csv", stringsAsFactors = FALSE)

data$predicted_category <- as.factor(data$predicted_category)
```

```
#Creating Term Document Matrix

set.seed(7)
random_hm <- sample(1:nrow(data), 15000)
corpus    <- Corpus(VectorSource(data$cleaned_hm[random_hm]))
skipWords <- function(x) removeWords(x, words = c(stopwords(kind = "en"),'happy', 'day', 'got', 'went',
funcs <- list(skipWords, stripWhitespace, removeNumbers, removePunctuation, tolower)
a         <- tm_map(corpus, FUN = tm_reduce, tmFuns = funcs)
a_tdm     <- TermDocumentMatrix(a)
m         <- as.matrix(a_tdm)
v         <- sort(rowSums(m), decreasing = TRUE)
d         <- data.frame(word = names(v), freq = v)
d <- head(d, 10);d
```

```
#Displaying TDM for first three observations

a_dtm <- DocumentTermMatrix(a)
wa <- as.matrix(a_dtm)
```

```r
wa[1:3,1:4]
```

```r
#PieChart of Categories

ggplot(data, aes(x=predicted_category, fill=predicted_category))+
  geom_bar()+
  labs(title = "Distribution of Predicted Category")+
  guides(fill="none")+
  coord_polar(theta = "y", start=0)
```

```r
#Summary of wordcount

count <- sapply(data$cleaned_hm, wordcount) # Counts number of words
summary(count)
```

```r
#Distribution of Word Counts

category <- c("0-4","5-9","10-14","15-19","20-24","25-29","30-34","35-39",
              "40-44","45-49",">=50")
count_class <- cut(count, breaks = c(0,4,9,14,19,24,29,34,39,44,49,Inf),
                   labels = category, include.lowest = TRUE)
ggplot()+
  geom_bar(aes(x = count_class, fill = count_class))+
  ylim(0,30000)+
  labs(x = "Word Count", y = "Number of Happy Moments",
       title = "Word Count Distribution")+
  guides(fill = "none")
```

```r
#TD-IDF

words <- data %>%
  unnest_tokens(word, cleaned_hm) %>%
  count(predicted_category, word, sort = TRUE) %>%
  ungroup()

totalwords <- words %>%
  group_by(predicted_category) %>%
  summarize(total = sum(n))

words <- left_join(words, totalwords);words

tf <- words %>%
  bind_tf_idf(word, predicted_category, n);tf

tf %>%
  select(-total) %>%
  filter(n >=30) %>%
  arrange(desc(tf_idf));tf
```

```r
#TD-IDF Graph

tf %>%
  arrange(desc(tf_idf)) %>%
  mutate(word = factor(word, levels = rev(unique(word)))) %>%
  group_by(predicted_category) %>%
  top_n(10) %>%
  ungroup %>%
  ggplot() +
  geom_col(aes(word, tf_idf, fill = predicted_category),show.legend = FALSE) +
  labs(x = NULL, y = "tf-idf") +
  facet_wrap(~predicted_category, ncol = 3, scales = "free") +
  coord_flip()
```

```r
#Creating containers for models

set.seed(57)

bin_data <- data[sample(1:nrow(data), 15000), ]
my.matrix1 <- create_matrix(bin_data$cleaned_hm, language="english", removeNumbers=TRUE, stemWords=FALSE

container1 <- create_container(my.matrix1, bin_data$predicted_category, trainSize=1:8000, testSize =800
```

```r
#SVM model

set.seed(6)
start_svm <- Sys.time()
svm_model <- train_model(container1, "SVM")
end_svm <- Sys.time()

svm_results <- classify_model(container1, svm_model)
ac_svm <- mean(svm_results$SVM_LABEL != bin_data[8001:10000,9])

table(svm_results$SVM_LABEL, bin_data[8001:10000,9])


end_svm - start_svm

pr_svm <- create_precisionRecallSummary(container1, svm_results)
```

```r
#Tree Model

set.seed(1)
start_tree <- Sys.time()
tree_model <- train_model(container1, "TREE")
end_tree <- Sys.time()

tree_results <- classify_model(container1, tree_model)
table(tree_results$TREE_LABEL, bin_data[8001:10000,9])

ac_tree <- (682+363+202+39)/2000
end_tree - start_tree

pr_tree <- create_precisionRecallSummary(container1, tree_results)
```

```r
#Multinomial Model

set.seed(2)
start_multi <- Sys.time()
multinomial_model <- train_model(container1, "GLMNET", family="multinomial")
end_multi <- Sys.time()
```

```
multi_results <- classify_model(container1, multinomial_model)
table(multi_results$GLMNET_LABEL, bin_data[8001:10000,9])
ac_multi <- mean(multi_results$GLMNET_LABEL != bin_data[8001:10000,9])
end_multi - start_multi

pr_multi <- create_precisionRecallSummary(container1, multi_results)
```

```
#Bagging Model

# set.seed(3)
# start_bagging <- Sys.time()
# lda_model <- train_model(container1, "BAGGING")
# end_bagging <- Sys.time()
#
# bagging_results <- classify_model(container1, lda_model)
# table(bagging_results$BAGGING_LABEL, bin_data[8001:10000,9])
# ac_bagging <- mean(bagging_results$BAGGING_LABEL != bin_data[8001:10000,9])
#
# end_bagging - start_bagging
# pr_bag <- create_precisionRecallSummary(container1, bagging_results)
```

```
#Random Forests Model

set.seed(4)
start_rf <- Sys.time()
rf_model <- train_model(container1, "RF", ntree=10)
end_rf <- Sys.time()

rf_results <- classify_model(container1, rf_model)
table( rf_results$FORESTS_LABEL, bin_data[8001:10000,9])
ac_rf <- mean(rf_results$FORESTS_LABEL != bin_data[8001:10000,9])
end_rf-start_rf

pr_rf <- create_precisionRecallSummary(container1, rf_results)
```

```
#LDA Model

set.seed(5)
start_lda <- Sys.time()
lda_model <- train_model(container1, "SLDA")
end_lda <- Sys.time()

lda_results <- classify_model(container1, lda_model)
table(lda_results$SLDA_LABEL, bin_data[8001:10000,9])
ac_lda <- mean(lda_results$SLDA_LABEL != bin_data[8001:10000,9])
end_lda - start_lda

pr_lda <- create_precisionRecallSummary(container1, lda_results)
```

```
#Runtime table

time_svm <- end_svm - start_svm

# time_bagging <- end_bagging - start_bagging
```

```
time_rf <- end_rf - start_rf

time_multi <- end_multi - start_multi

time_lda <- end_lda - start_lda

time_tree <- end_tree - start_tree

times <- c(time_svm, time_multi, time_tree, time_rf, time_lda)
accuracy <- c(ac_svm, ac_multi, ac_tree, ac_rf, ac_lda)

times <- as.data.frame(as.numeric(unlist(times)))

colnames(times) <- c("runtime")


model <- c("SVM", "Multinomial", "Decision Trees", "Random Forests", "LDA")

run_table <- cbind(model, accuracy, times)
```

```
#Plot of Misclassification rate

ggplot()+
  geom_bar(aes(x= reorder(run_table$model, run_table$accuracy), y=run_table$accuracy, fill=run_table$mo
  labs(x="Model", y="Percentage Misclassified", title="Missclasstion Rates")+
  guides(fill="none")
```

```
#Precision Recall Table for achievement category

pr_table <- rbind(pr_svm[1,], pr_multi[1,], pr_tree[1,], pr_rf[1,], pr_lda[1,])

rownames(pr_table) <- c("SVM", "Multinomial", "Decision Tree", "Random Forests", "LDA")

pr_table
```

```
#Decision Tree Confusion Matrix

table(tree_results$TREE_LABEL, bin_data[8001:10000,9])
```

```
#Multinomial Confusion Matrix

table(multi_results$GLMNET_LABEL, bin_data[8001:10000,9])
```