```r
# Week 5 - Twitter


# VIDEO 5

# Read in the data

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

str(tweets)


# Create dependent variable

tweets$Negative = as.factor(tweets$Avg <= -1)

table(tweets$Negative)


# Install new packages

install.packages("tm")
library(tm)
install.packages("SnowballC")
library(SnowballC)


# Create corpus

corpus = Corpus(VectorSource(tweets$Tweet))

# Look at corpus
corpus

corpus[[1]]


# Convert to lower-case

corpus = tm_map(corpus, tolower)

corpus[[1]]

# Remove punctuation

corpus = tm_map(corpus, removePunctuation)

corpus[[1]]
```

```r
# Look at stop words
stopwords("english")[1:10]

# Remove stopwords and apple

corpus = tm_map(corpus, removeWords, c("apple",
 stopwords("english")))

corpus[[1]]

# Stem document

corpus = tm_map(corpus, stemDocument)

corpus[[1]]



# Video 6

# Create matrix

frequencies = DocumentTermMatrix(corpus)

frequencies

# Look at matrix

inspect(frequencies[1000:1005,505:515])

# Check for sparsity

findFreqTerms(frequencies, lowfreq=20)

# Remove sparse terms

sparse = removeSparseTerms(frequencies, 0.995)
sparse

# Convert to a data frame

tweetsSparse = as.data.frame(as.matrix(sparse))

# Make all variable names R-friendly

colnames(tweetsSparse) = make.names(colnames(tweetsSparse))
```

```r
# Add dependent variable

tweetsSparse$Negative = tweets$Negative

# Split the data

library(caTools)

set.seed(123)

split = sample.split(tweetsSparse$Negative, SplitRatio = 0.7)

trainSparse = subset(tweetsSparse, split==TRUE)
testSparse = subset(tweetsSparse, split==FALSE)



# Video 7

# Build a CART model

library(rpart)
library(rpart.plot)

tweetCART = rpart(Negative ~ ., data=trainSparse, method="class")

prp(tweetCART)

# Evaluate the performance of the model
predictCART = predict(tweetCART, newdata=testSparse, type="class")

table(testSparse$Negative, predictCART)

# Compute accuracy

(294+18)/(294+6+37+18)

# Baseline accuracy

table(testSparse$Negative)

300/(300+55)


# Random forest model

library(randomForest)
set.seed(123)
```

```
tweetRF = randomForest(Negative ~ ., data=trainSparse)

# Make predictions:
predictRF = predict(tweetRF, newdata=testSparse)

table(testSparse$Negative, predictRF)

# Accuracy:
(293+21)/(293+7+34+21)
```