BDS516 Homework #9

Madeline Mauboussin

4/21/2021

```r
library(rtweet)
library(httpuv)
library(tidytext)
library(tidyverse)
library(dplyr)
library(lubridate)
library(ggplot2)
library(scales)
library(readr)
library(syuzhet)
library(mlbench)
library(caret)
library(vtreat)
library(InformationValue)
setwd("~/Desktop/Madeline_R_Stuff")
```

Packages

```r
## I set up authentication to get to twitter via create_tokwn with my consumer_key, consumer_secret, ac

## Getting tweets from Obama and Kanye
obama_raw <- get_timeline("@BarackObama", n = 3200)
kanye_raw <- get_timeline("@kanyewest", n = 3200)

## Cleaning the data sets and saving them as csv files

# Obama
obama_clean <- obama_raw %>% select("source", "status_id", "text","created_at",
                                    "retweet_count", "favorite_count", "is_retweet", "screen_name")

# Kanye
kanye_clean <- kanye_raw %>% select("source", "status_id", "text","created_at",
                                    "retweet_count", "favorite_count", "is_retweet", "screen_name")

## Saving files as csvs for ease of use
obama_clean <- as.data.frame(obama_clean)
write.csv(x=obama_clean, file="obama_tweets.csv")
```

```
kanye_clean <- as.data.frame(kanye_clean)
write.csv(x=kanye_clean, file="kanye_tweets.csv")
```

```
## Loading in csv files
obama <- read_csv("obama_tweets.csv")
kanye <- read_csv("kanye_tweets.csv")
```

Set up

Feature Extraction

```
obama %>% count(source) %>% arrange(-n)
kanye %>% count(source) %>% arrange(-n)
```

(1.) Source

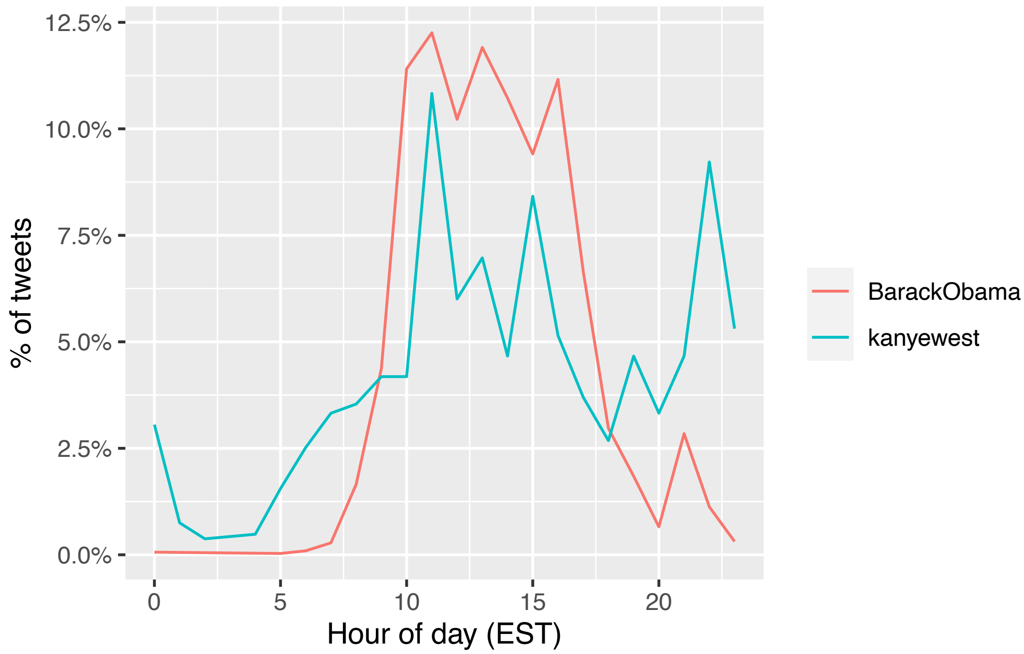
```
# A tibble: 5 x 2
  source                  n
  <chr>               <int>
1 Twitter Web Client   2444
2 Twitter for iPhone    473
3 Twitter Web App       200
4 Twitter Media Studio   77
5 Thunderclap             5
# A tibble: 2 x 2
  source                n
  <chr>             <int>
1 Twitter for iPhone  1827
2 Twitter Web App       38
```

Obama most frequently tweets from a desktop/laptop while Kanye most frequently tweets from an iPhone.

```
merged_df <- rbind(obama, kanye)

merged_df %>% group_by(screen_name) %>%
  count(hour = hour(with_tz(created_at, "EST"))) %>%
  mutate(percent = n/sum(n)) %>%
  ggplot(aes(x = hour, y = percent, color = screen_name)) +
  labs(x = "Hour of day (EST)", y = "% of tweets", color = "") +
  scale_y_continuous(labels = percent_format()) +
  geom_line()
```

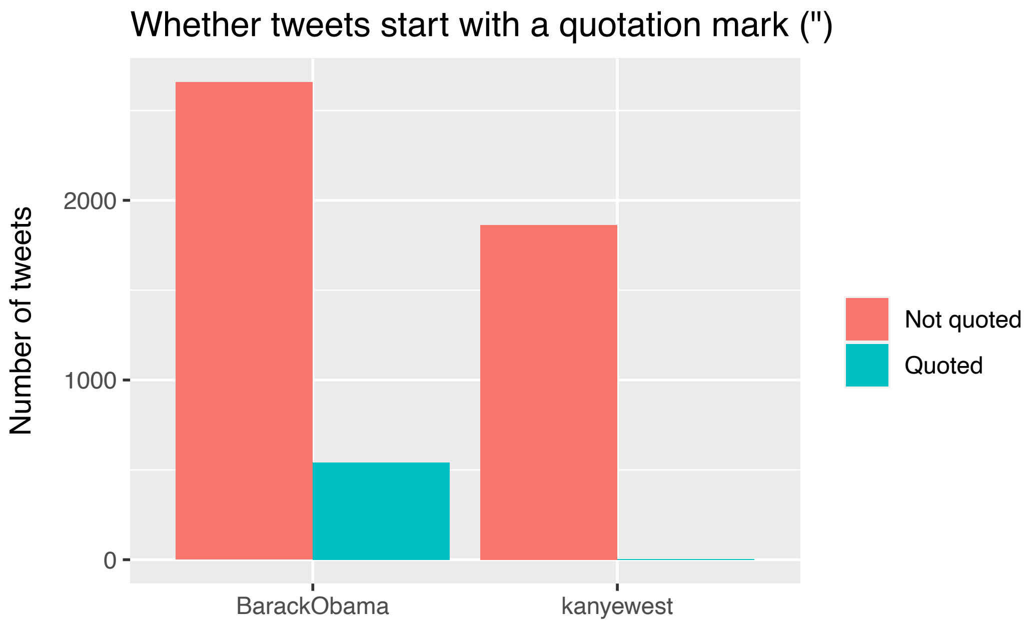
(2.) Time of Day



The vast majority of Obama's tweets are posted between 10am and 4pm, while, Kanye's tweets have much more variability.

```
## Plot of tweets with quotes vs. no quotes
merged_df %>% group_by(screen_name) %>%
  count(quoted = ifelse(str_detect(text, '^"'), "Quoted", "Not quoted")) %>%
  ggplot(aes(x = screen_name, y = n, fill = quoted)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "", y = "Number of tweets", fill = "") +
  theme(axis.title.y = element_text(margin = margin(t = 0, r = 10, b = 0, l = 0))) +
  ggtitle('Whether tweets start with a quotation mark (")')
```

(3.) Quotes




```
# Table of tweets with quotes vs. no quotes
merged_df %>% group_by(screen_name) %>%
  count(quoted = ifelse(str_detect(text, '^"'), "Quoted", "Not quoted")) %>%
  mutate(percent_quote = n/sum(n)*100)
```

```
# A tibble: 4 x 4
# Groups:   screen_name [2]
  screen_name quoted         n percent_quote
  <chr>       <chr>      <int>         <dbl>
1 BarackObama Not quoted  2657        83.1
2 BarackObama Quoted       542        16.9
3 kanyewest   Not quoted  1862        99.8
4 kanyewest   Quoted         3         0.161
```

Both Obama and Kanye do not quote very much in their tweets; however, ~17% of Obama's tweets use quotes, while Kanye quotes less than 1% of the time.

```
merged_df %>%
  group_by(screen_name) %>%
  filter(!str_detect(text, '^"')) %>%
  count(picture = ifelse(str_detect(text, "t.co"),
                       "Picture/link", "No picture/link")) %>%
  mutate(percent_picture = n/sum(n)*100)
```

*(4.) **Pictures***

```
# A tibble: 4 x 4
# Groups:   screen_name [2]
  screen_name picture              n percent_picture
  <chr>       <chr>            <int>           <dbl>
1 BarackObama No picture/link    209            7.87
2 BarackObama Picture/link      2448           92.1
3 kanyewest   No picture/link    841           45.2
4 kanyewest   Picture/link      1021           54.8
```

The vast majority of Obama's tweets include a picture or link (92%), while only 55% of Kanye's tweets contain a picture or link.

```
merged_df %>% group_by(screen_name) %>%
  count(is_retweet) %>%
  mutate(perc_retweet = n/sum(n)*100)
```

(5.) Re-tweets

```
# A tibble: 4 x 4
# Groups:   screen_name [2]
  screen_name is_retweet     n perc_retweet
  <chr>       <lgl>      <int>        <dbl>
1 BarackObama FALSE       2847         89.0
2 BarackObama TRUE         352         11.0
3 kanyewest   FALSE       1670         89.5
4 kanyewest   TRUE         195         10.5
```

Obama and Kanye have nearly the same percentage of tweets that are re-tweets.

```
merged_df %>% group_by(screen_name) %>%
  summarize(avg_retweet = mean(retweet_count),
            avg_fav = mean(favorite_count))
```

(6.) Re-tweet Counts & Favorite Counts

```
# A tibble: 2 x 3
  screen_name avg_retweet avg_fav
* <chr>             <dbl>   <dbl>
1 BarackObama      11364.  58132.
2 kanyewest         9319.  49022.
```

On average, a tweet posted by Obama is re-tweeted ~11,4000 times and is favorited by 60,000 people. For Kanye, the average tweet is re-tweeted ~9,000 times and favorited by ~50,000 people. However, these differences do not seem to be a meaningful metric of comparison given the fact that Obama has 130 million twitter followers, while Kanye has only 30 million.

```
merged_sentiment <- merged_df %>%
  mutate(text2 = str_replace_all(text, "[^[:alpha:]]", " "), # removes all non-alphabetic characters
         get_nrc_sentiment(text2)) # getting nrc scores for tweet texts
```

```
merged_sentiment %>%
  group_by(screen_name) %>%
  summarize(anger = mean(anger),
            anticipation = mean(anticipation),
            fear = mean(fear),
            disgust = mean(disgust),
            joy = mean(joy),
            sadness = mean(sadness),
            surprise = mean(surprise),
            trust = mean(trust),
```

```
            negative = mean(negative),
            positive = mean(positive))
```

(7.) Sentiment

```
  screen_name anger anticipation  fear disgust   joy sadness surprise trust
1 BarackObama 0.316        0.738 0.453   0.100 0.566   0.233    0.284 1.247
2   kanyewest 0.149        0.305 0.213   0.071 0.376   0.153    0.120 0.414
  negative positive
1    0.517    1.779
2    0.269    0.743
```

Obama's tweets (on average) seemingly score higher across all sentiment scores. This is particularly true for "anticipation", "trust", and "positive" sentiments

Part A

Develop an algorithm that allows to predict who of the politicians tweeted using just the information in the text of the tweet and the time of the tweets. You are not allowed to use the information about the user. You can use sentiments, individual words, punctuation and anything else as a source of features.

```
## Setting up data frame for logistic regression
obama_kanye <- merged_sentiment

# Changing names of sources (before filtering)
obama_kanye$source[obama_kanye$source=="Twitter Web Client"] <- "web"
obama_kanye$source[obama_kanye$source=="Twitter for iPhone"] <- "iphone"

obama_kanye2 <- obama_kanye %>%
  select(screen_name, source, created_at, text, status_id,
         anger, anticipation, fear, disgust, joy, sadness, surprise, trust, negative, positive) %>%
  # filtering twitter sources for only web/iPhone
  filter(source %in% c("web", "iphone")) %>%
  # creating variables for time of day, whether the tweet uses a quote, and whether
  # there is a picture or link in the tweet
  mutate(hour = hour(with_tz(created_at, "EST")),
         quoted = ifelse(str_detect(text, '^"'), "quote", "NO_quote"),
         picture = ifelse(str_detect(text, "t.co"), "picture_link", "NO_picture_link"),
         is_obama = case_when(screen_name == "BarackObama" ~ 1,
                              screen_name == "kanyewest" ~ 0))

# Selecting variables for regression
obama_kanye3 <- obama_kanye2 %>%
  select(is_obama, screen_name, source, hour, quoted, picture,
         anger, anticipation, fear, disgust, joy, sadness, surprise, trust, negative, positive)
```

Based off the feature extraction above, we believe that the features which most contribute to the prediction of whether a tweet was authored by Obama vs. Kanye are: source, quotes, pictures, and sentiment scores. We now will develop a classification algorithm using logistic regression model to predict the probability of a tweet being authored by Obama. As such,

the *outcome variable* will be a tweet by Obama (yes or no) and the *predictor variables* will be some combination of the features mentioned above. To that end, we will run several logistic regression models, but only include the model with the greatest predictive power.

```r
model <- glm(is_obama ~  factor(quoted) + factor(picture) +
                anticipation + fear + joy + trust + positive,
             family = "binomial",
             data = obama_kanye3)

summary(model)
```

Logistic Regression Model

```
Call:
glm(formula = is_obama ~ factor(quoted) + factor(picture) + anticipation +
    fear + joy + trust + positive, family = "binomial", data = obama_kanye3)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.2690  -0.5607   0.1226   0.6564   2.7439

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                 -3.74117    0.14644 -25.547  < 2e-16 ***
factor(quoted)quote          6.68990    0.72081   9.281  < 2e-16 ***
factor(picture)picture_link  3.40999    0.13460  25.334  < 2e-16 ***
anticipation                 0.55093    0.08229   6.695 2.15e-11 ***
fear                         0.58056    0.07469   7.773 7.68e-15 ***
joy                         -1.14234    0.09478 -12.052  < 2e-16 ***
trust                        0.73085    0.07298  10.014  < 2e-16 ***
positive                     0.81770    0.06215  13.157  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6323.9  on 4743  degrees of freedom
Residual deviance: 3745.6  on 4736  degrees of freedom
AIC: 3761.6

Number of Fisher Scoring iterations: 8
```

```r
exp(model$coefficients)
```

```
                (Intercept)         factor(quoted)quote
                 0.02372623                804.24385232
factor(picture)picture_link                anticipation
                30.26492569                  1.73487349
                       fear                         joy
                 1.78703885                  0.31907270
                      trust                    positive
                 2.07683779                  2.26527822
```

Interpretation of Model

- "**quoted**"
  - All else equal, tweets with quotes have a ~80,000% greater odds of being Obama's tweets.
  - *Calculation:* odds = (797.47143283 - 1)*100 = 79647.14
- "**picture_link**"
  - All else equal, tweets with pictures or links have a ~3,000% greater odds of being Obama's tweets.
  - *Calculation:* odds = (30.04893338 - 1)*100 = 2904.893
- "**anticipation**"
  - All else equal, a one-unit increase in the sentiment score for anticipation increases the odds of the tweet being authored by Obama by 74%.
  - *Calculation:* odds = (1.74400434 - 1)*100 = 74.40043
- "**fear**"
  - All else equal, a one-unit increase in the sentiment score for fear increases the odds of the tweet being authored by Obama by 78%.
  - *Calculation:* odds = 1.78329704 - 1)*100 = 78.3297
- "**joy**"
  - All else equal, a one-unit increase in the sentiment score for joy decreases the odds of the tweet being authored by Obama by 68%.
  - *Calculation:* odds = (0.31620393 - 1)*100 = -68.37961
- "**trust**"
  - All else equal, a one-unit increase in the sentiment score for trust increases the odds of the tweet being authored by Obama by 108%.
  - *Calculation:* odds = (2.07902310 - 1)*100 = 107.9023
- "**positive**"
  - All else equal, a one-unit increase in the sentiment score for positive increases the odds of the tweet being authored by Obama by 127%.
  - *Calculation:* odds = (2.27180811 - 1)*100 = 127.1808

Part B

Apply the algorithm to new tweets from both users to estimate how well the predictions work.

Given that our logistic regression model was developed using *all* of the tweets ever posted by Kanye West (n = 1,868), rather than applying the algorithm to a new tweets, we will evaluate the algorithm using a train-test split.

Train-Test Split Evaluation

```
# Checking for class bias
table(obama_kanye3$is_obama)
```

```
##
##    0    1
## 1827 2917
```

```
## Creating train and test data

# Ensuring Train Data draws equal proportions of Obama (1) and Kanye (0))
set.seed(04917)
input_ones <- obama_kanye3[which(obama_kanye3$is_obama == 1), ]  # all 1's
input_zeros <- obama_kanye3[which(obama_kanye3$is_obama == 0), ]  # all 0's
```

```r
# 1's for training
input_ones_training_rows <- sample(1:nrow(input_ones), 0.7*nrow(input_ones))
training_ones <- input_ones[input_ones_training_rows, ]

# 0's for training. Pick as many 0's as 1's
input_zeros_training_rows <- sample(1:nrow(input_zeros), 0.7*nrow(input_zeros))
training_zeros <- input_zeros[input_zeros_training_rows, ]

#Row bind the 1's and 0's
train.data <- rbind(training_ones, training_zeros)

# Creating Test Data
test_ones <- input_ones[-input_ones_training_rows, ]
test_zeros <- input_zeros[-input_zeros_training_rows, ]

# Row bind the 1's and 0's
test.data <- rbind(test_ones, test_zeros)

## Building Logistical Model and Predicting on Test Data
model_train <- glm(is_obama ~  factor(quoted) + factor(picture) +
                anticipation + fear + joy + trust + positive,
                data=train.data,
                family=binomial(link="logit"))

predicted <- predict(model_train, test.data, type="response")
```

Model Diagnostics

```r
# Optimal prediction probability cutoff
optCutOff <- optimalCutoff(test.data$is_obama, predicted)
optCutOff #  = 0.52
```

```r
misClassError(test.data$is_obama, predicted, threshold = optCutOff)
```

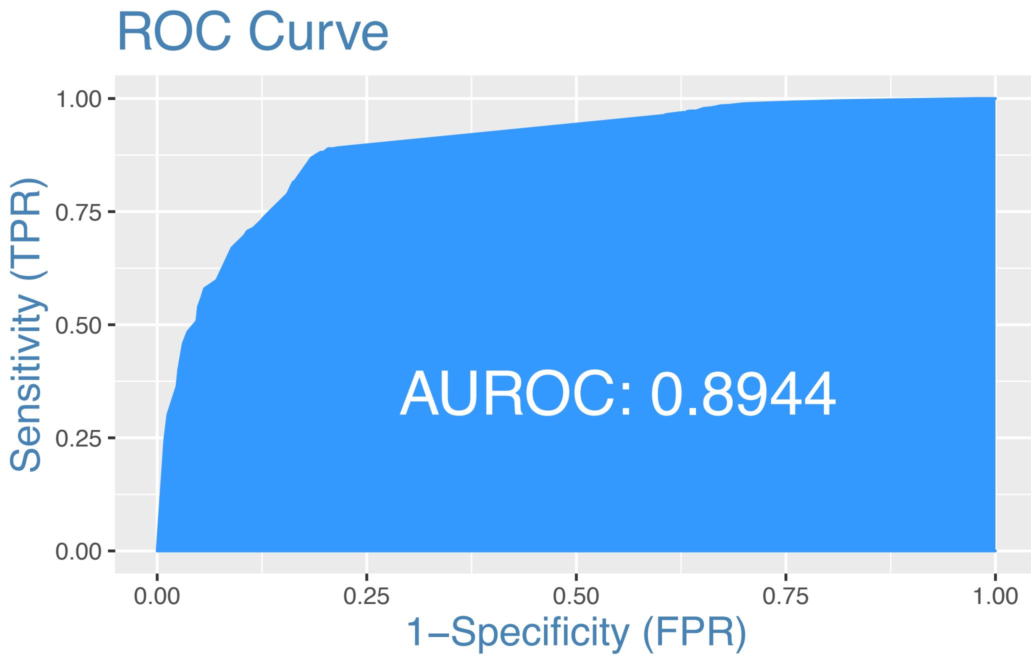
Misclassification Error

```
[1] 0.1467
```

The model's misclassification error (i.e. the percentage of incorrectly classified instances) is 14%.

```r
plotROC(test.data$is_obama, predicted)
```

AUC-ROC Curve



Our model has an AUC of .9, meaning there is a ~90% chance that the model will be able distinguish between positive class (i.e. Obama's tweets) and negative class (i.e. Kanye's tweets)

```
sensitivity(test.data$is_obama, predicted, threshold = optCutOff)
specificity(test.data$is_obama, predicted, threshold = optCutOff)
```

Sensitivity and Specificity

```
[1] 0.8892694
[1] 0.7959927
```

The model's true positive rate (i.e. sensitivity) and true negative rate (i.e specificity) are both about 85%.

All in all, our model has fairly strong predictive ability

Part C

Try the prediction algorithm with a different set of tweets from unrelated users. Discuss how the algorithm works / breaks in this case.

In the following section, we will apply our prediction algorithm (i.e. the trained logistic model created above) to a set of tweets posted by the rapper, Drake.

```
## Creating a data set for Drake Tweets

# Extracting data from Twitter
drake_raw <- get_timeline("@Drake", n = 3200)
```

```r
# Cleaning the data sets
drake_clean <- drake_raw %>% select("source", "status_id", "text","created_at",
                                "retweet_count", "favorite_count", "is_retweet",
                                "screen_name")

drake <- as.data.frame(drake_clean)
```

```r
## Getting sentiment scores
drake_sentiment <- drake %>%
  mutate(text2 = str_replace_all(text, "[^[:alpha:]]", " "), # removes all non-alphabetic characters
         get_nrc_sentiment(text2)) # getting nrc scores for tweet texts
```

```r
## Preparing data set for testing
drake_test <- drake_sentiment

drake_test2 <- drake_test %>%
  # creating variables for whether the tweet uses a quote & whether there is a picture/link
  mutate(quoted = ifelse(str_detect(text, '^"'), "quote", "NO_quote"),
         picture = ifelse(str_detect(text, "t.co"), "picture_link", "NO_picture_link")) %>%
  select(screen_name, text2, quoted, picture,
         anticipation, fear, joy, trust, positive)

## Sanity check
head(drake_test2)
```

Preparing a data set of Drake Tweets

```
  screen_name
1       Drake
2       Drake
3       Drake
4       Drake
5       Drake
6       Drake

1 It s the biggest Ultimate Madness Tournament ever  I  m putting up     k to the winner so someone go g
2
3
4
5
6                                    Going live tonight at      pm EST with the return of OVO Sound Radi
    quoted      picture anticipation fear joy trust positive
1 NO_quote picture_link            2    1   1     0        2
2 NO_quote picture_link            0    0   0     0        0
3 NO_quote picture_link            0    0   0     0        0
4 NO_quote picture_link            0    0   0     1        0
5 NO_quote picture_link            0    0   0     0        0
6 NO_quote picture_link            1    0   0     0        2
```

```r
## Predicting the train logistical regression model on the drake data
predicted_drake <- predict(model_train, drake_test2, type="response")
predicted.classes <- ifelse(predicted_drake > 0.5, "Obama", "Kanye")
```

```r
table(predicted.classes)
```

Applying Prediction Algorithm on Drake Tweets

```
predicted.classes
Kanye Obama
 1584   164
```

When the original prediction algorithm was used on a data set of tweets posted by Drake, the algorithm classified 91% of the tweets as being Kanye West's tweets and 9% being Obama's. Given this result, it would be interesting to look at examples of Drake's tweets that were classified as Kanye's vs. Obama's.

```r
drake_test3 <- drake_test2
drake_test3$predictions <- predicted.classes
drake_test3 <- drake_test3 %>%
  mutate(n = row_number())
```

```r
drake_test3 %>% filter(n == 36) %>%
  pull(text2)
```

Example: **Predicted Classification of Tweet = Obama**

```
[1] "   Drake When to Say When  amp  Chicago Freestyle  Video  https   t co ZIAX R UCY"
```

```r
drake_test3 %>% filter(n == 121) %>%
  pull(text2)
```

Example: **Predicted Classification of Tweet = Kanye**

```
[1] "Seventh Annual OVOFEST https   t co Y KeKSHt R"
```

The first example shows a Drake tweet that was classified as an *Obama* tweet. This tweet did not begin with a quote/link but included a picture., and its sentiment scores were as follows: anticipation = 4; fear = 1; joy = 2; trust = 3; positive = 5. Given that our algorithm found that pictures and sentiments of anticipation, fear, trust, and positive *all* increase the odds of a tweet belonging to Obama (versus Kanye), it is unsurprising that example #1 was coded as an Obama tweet. The second example shows a Drake tweet that was classified as a *Kanye* tweet. This tweet also did not begin with a quote, included a picture, and its sentiment scores were as follows: anticipation = 0; fear = 0; joy = 1; trust = 0; positive = 0. While our algorithm found that a one unit increase in the sentiment score for joy *decreases* the odds of the tweet being authored by Obama by 68%, it also found that tweets with pictures/links have a ~3,000% greater odds of being Obama's tweets. As such, this classification seems to be rather odd. All in all, it is clear that our algorithm is only as good as the data it is provided.