Part Two - Data Cleaning

Samad Patel

9/29/2018

Summary

This document contains all the steps taken to clean the data before modeling.

Load and Merge Data

I collected and aggregated data from two different sources into two respective csvs. They can be joined based on the date.

```
# data regarding Trump's followers
follows <- read_csv('~/Documents/GitHub/Trump-Twitter-Predictions/Data/trump_followers_data.csv')

# data with tweets
tweets <- read_csv('~/Documents/GitHub/Trump-Twitter-Predictions/Data/trump_tweets.csv')
# Remove rows where we don't have favorite_counts
tweets <- tweets %>% filter(!is.na(favorite_count))
# Remove rows where favorite count is 0
tweets <- tweets %>% filter(favorite_count != 0)

# Make dates into datetime
tweets$created_at <- mdy_hm(tweets$created_at)
# Create column for left_join
tweets$Date <- date(tweets$created_at)

# Merge
df <- left_join(x = tweets, y = follows, by = "Date")
# Merge didn't work out quite as expected - there are more rows than I want.
# Some tweets are being repeated. Only include unique id_strings
df <- df %>% distinct(id_str, .keep_all = T)

kable(df[1,]) %>% column_spec(2, '2cm') %>%
   kable_styling(position = 'center', 'striped', row_label_position = 'c')
```

source	text	created_at	retweet_count	favorite_count	is_retweet	id_str
Twitter for iPhone	Judge Kavanaugh showed America exactly why I nominated him. His testimony was powerful honest and riveting. Democrats<d5> search and destroy strategy is disgraceful and this process has been a total sham and effort to delay obstruct and resist. The Senate must vote!	2018-09-27 22:46:00	81880	303263	FALSE	1.045445e+18

Missing Data

There are 40 observations that don't have any follower information. I will proceed to impute these values in various ways.

```
kable(head(df[!complete.cases(df), 8:11])) %>% kable_styling(position = 'center', 'striped',
                                                              row_label_position = 'c')
```

Date	Followers	Follower_Change	Num_Tweets
2017-03-20	NA	NA	NA
2017-03-20	NA	NA	NA
2017-03-20	NA	NA	NA
2017-03-20	NA	NA	NA
2017-03-20	NA	NA	NA
2017-03-20	NA	NA	NA

Num_Tweets

I will simply aggregate the number of tweets that Trump posted for each respective day that is missing.

```
dates_without_followers <- df[!complete.cases(df), ]$Date %>% unique()
# Number of Tweets will be easy - sum up unique ids for each given date
imputed_num_tweets <- df %>% group_by(Date) %>% summarize('Tweets' = n()) %>%
  filter(Date %in% dates_without_followers)
```

```r
# Place those values into their proper place in df
# Convoluted, but gets the job done
df <- left_join(df, imputed_num_tweets, 'Date')
df[!is.na(df$Tweets),'Num_Tweets'] <- df[!is.na(df$Tweets),'Tweets']
# Remove Tweets variable
df <- df %>% select(-Tweets)
```

Followers_Change

Every time there is a gap in days, the Follower_Change variable at the end of the gap accounts for how many followers were gained in that entire span. So I will simply divide that number by the number of days in the gap to determine the Followers_Change for any given day.

```r
# Just divide total change between a gap by number of days in gap.
delta1 <- df[df$Date == '2017-03-21', 'Follower_Change'] %>% unique() %>% as.numeric() / 5
delta2 <- df[df$Date == '2016-07-06', 'Follower_Change'] %>% unique() %>% as.numeric() / 4

# Since we must change the dates at end of gaps, throw those into relevant_dates
relevant_dates <- append(dates_without_followers, date('2017-03-21'), after = 0)
relevant_dates <- append(relevant_dates, date('2016-07-06'), after = 5)
# Create df
change <- tibble('Date' = relevant_dates, 'relevant_changes' = c(rep(delta1, 5), rep(delta2, 4)))
# Place those values into their proper place in df
df <- left_join(df, change, 'Date')
df[!is.na(df$relevant_changes),'Follower_Change'] <- df[!is.na(df$relevant_changes),'relevant_changes']
# Remove Tweets variable
df <- df %>% select(-relevant_changes)
```

Followers

The idea here is simple - start at the end of a gap, and subtract Followers_Change from Followers to get the number of Followers that were in the day before.

```r
# For each date
for (i in 1:(length(relevant_dates) - 1)){
  # If the followers for the previous date is missing
  if (is.na(df[df$Date == relevant_dates[i+1], 'Followers'] %>% unique() %>% as.numeric())){
    # Fill those NA vals with the Followers from date i - Followers_Change for date i
    # Followers from date i
    old_follows <- df[df$Date == relevant_dates[i], 'Followers'] %>% unique() %>% as.numeric()
    # Followers change for date i
    old_change <- df[df$Date == relevant_dates[i], 'Follower_Change'] %>% unique() %>% as.numeric()
    # Change df
    df[df$Date == relevant_dates[i+1], 'Followers'] <- old_follows - old_change
  }
}
```

Sparse Classes

97%% of Trump's tweets come from Twitter for iPhone, Twitter for Android, or Twitter Web Client. All the other classes should be recoded to 'Other'.

```r
# Recode
df$source <- car::recode(df$source, "c('Media Studio', 'Twitter Ads', 'Twitter for iPad', 'Instagram',
                         'Twitter for BlackBerry', 'Twitter QandA', 'Periscope', 'Facebook',
                         'TweetDeck', 'Mobile Web (M5)', 'Twitter Mirror for iPad') = 'Other'")
```

Date and Time

This is technically feature engineering, but since I want to delete irrelevant columns at the end of this document I might as well extract relevant date and time info now.

```r
# Year variable
df$Year <- year(df$created_at)
df$Month <- month(df$created_at)
df$Week <- week(df$created_at)
df$Day <- day(df$created_at)
df$Hour <- hour(df$created_at)
```

Adding Holidays

Whether or not a tweet is made on a holiday may be predictive of a tweet doing well. I will merge the current dataset with a dataset that marks major holidays between 2012 and 2020. I will also add in days that aren't necessarily holidays, but are highly important days regarding Trump, his campaign, and the American people. These include 9/11, New Years Eve, major election primary days, the Republican National Convention, Election Day + the day after, and Inaugration Day + the day after.

```r
# Load Data
usholidays <- read_csv("~/Documents/GitHub/Trump-Twitter-Predictions/Data/usholidays.txt",
                       col_names = c('Index', 'Date', 'Holiday'))
# Remove index
usholidays <- usholidays %>% select(-Index)

# Df with added holidays
added_holidays <- tibble('Date' = ymd(c('2015-09-11', '2016-09-11', '2017-09-11', '2018-09-11', # 9/11
                         '2016-11-08', '2016-11-09', # Gen Election
                         '2017-01-20', '2017-01-21', # Inauguration
                         '2015-12-31', '2016-12-31', '2017-12-31', '2018-12-31', # NYE
                         '2016-03-01', '2016-03-05', '2016-03-15', # March Primaries
                         '2016-04-26', '2016-06-07', # Other Primaries
                         '2016-07-18', '2016-07-19', '2016-07-20', '2016-07-21' # RNC
                         )),
                         'Holiday' = rep(1, 21))

# Add these rows to usholidays
usholidays <- rbind(usholidays, added_holidays)

# Join usholidays to df
df <- left_join(df, usholidays, 'Date')

# Any NA values in Holiday should be changed to 0
df$Holiday[is.na(df$Holiday)] <- 0
```

Delete Irrelevant Rows

There's no clear way to tell which tweets in this dataset were later deleted by Trump. This is a major issue as he often misspells things and then deletes them. I will at the very least delete all tweets from Trump that were liked fewer than 100 times.

```
df <- df %>% filter(favorite_count >=100)
```

Delete Irrelevant Columns

We don't need created_at, is_retweet, id_str, and Date. We may need retweet_count later, so I'll leave it for now.

```
df <- df %>% select(-which(colnames(df) %in% c('created_at', 'is_retweet', 'id_str', 'Date')))
```

I'm now prepared for extensive feature engineering. I'll conduct Part Three in Python. I'll write out the file for use there.

```
write.csv(x = df, file = '~/Documents/GitHub/Trump-Twitter-Predictions/Data/project_data.csv', row.names
```