Predicting_Disaster_Tweets

Rawan Hammad

2/15/2022

Welcome to the Predicting Disaster Tweets R-Markdown file! The goal of this project is to practice Natural Language Processing (NLP). I am given a training and testing set that is split already. This is my very first take at NLP but I hope to get more comfortable with NLP in the future.

First, let's import all necessary libraries.

```
library(tidyr)
library(caret)
library(e1071)
library(irlba)
library(xgboost)
library(quanteda)
library(ggplot2)
```

Next, let's import the training set/data.

```
train_raw <- read.csv("C:/Users/rawan/OneDrive/Desktop/DePaul/Job
Assessments/nlp-getting-started/train.csv",stringsAsFactors = FALSE)

head(train_raw)
```

```
##   id keyword location
## 1  1
## 2  4
## 3  5
## 4  6
## 5  7
## 6  8
##
text
## 1                                                                    Our
Deeds are the Reason of this #earthquake May ALLAH Forgive us all
## 2
Forest fire near La Ronge Sask. Canada
## 3 All residents asked to 'shelter in place' are being notified by
officers. No other evacuation or shelter in place orders are expected
## 4
13,000 people receive #wildfires evacuation orders in California
## 5                                                Just got sent this photo
from Ruby #Alaska as smoke from #wildfires pours into a school
## 6                        #RockyFire Update => California Hwy. 20 closed in
```

```
both directions due to Lake County fire - #CAfire #wildfires
##   target
## 1      1
## 2      1
## 3      1
## 4      1
## 5      1
## 6      1
```

We can clean up the data and remove unnecessary columns and only keep the text and target.

```
train_raw <- train_raw[,4:5]

head(train_raw)

##
text
## 1                                                                    Our
Deeds are the Reason of this #earthquake May ALLAH Forgive us all
## 2
Forest fire near La Ronge Sask. Canada
## 3 All residents asked to 'shelter in place' are being notified by
officers. No other evacuation or shelter in place orders are expected
## 4
13,000 people receive #wildfires evacuation orders in California
## 5                                              Just got sent this photo
from Ruby #Alaska as smoke from #wildfires pours into a school
## 6                        #RockyFire Update => California Hwy. 20 closed in
both directions due to Lake County fire - #CAfire #wildfires
##   target
## 1      1
## 2      1
## 3      1
## 4      1
## 5      1
## 6      1
```

We can also rename the columns, for clarity, where the text is the tweet and the target is whether the tweet is about real disasters (1) or not (0).

```
names(train_raw) <- c("Text", "Target")

head(train_raw)

##
Text
## 1                                                                    Our
Deeds are the Reason of this #earthquake May ALLAH Forgive us all
## 2
Forest fire near La Ronge Sask. Canada
```

```
## 3 All residents asked to 'shelter in place' are being notified by
officers. No other evacuation or shelter in place orders are expected
## 4
13,000 people receive #wildfires evacuation orders in California
## 5                                                  Just got sent this photo
from Ruby #Alaska as smoke from #wildfires pours into a school
## 6                        #RockyFire Update => California Hwy. 20 closed in
both directions due to Lake County fire - #CAfire #wildfires
##   Target
## 1      1
## 2      1
## 3      1
## 4      1
## 5      1
## 6      1
```

We can quickly view a brief summary and structural summary of the training set.

```
summary(train_raw)

##      Text               Target
##  Length:7613        Min.   :0.0000
##  Class :character   1st Qu.:0.0000
##  Mode  :character   Median :0.0000
##                     Mean   :0.4297
##                     3rd Qu.:1.0000
##                     Max.   :1.0000

str(train_raw)

## 'data.frame':    7613 obs. of  2 variables:
##  $ Text  : chr  "Our Deeds are the Reason of this #earthquake May ALLAH
Forgive us all" "Forest fire near La Ronge Sask. Canada" "All residents asked
to 'shelter in place' are being notified by officers. No other evacuation or
shelter in pla"| __truncated__ "13,000 people receive #wildfires evacuation
orders in California " ...
##  $ Target: int  1 1 1 1 1 1 1 1 1 1 ...
```

Let's check for incomplete values.

```
length(which(!complete.cases(train_raw)))

## [1] 0
```

Great! What are the unique values in Target?

```
unique(train_raw$Target)

## [1] 1 0
```

Ones and zeroes only. Let's make factors.

```
train_raw$Target <- as.factor(train_raw$Target)
str(train_raw$Target)

##  Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

Since these are tweets, we know that the maximum length of a tweet is currently 280 characters. However, prior to 2020, the maximum length was only 140 characters. I don't have a timestamp on this dataset so I'm unsure if it's in the 140 or 280 range.

Let's add a column that describes the number of characters in the tweet.

```
train_raw$Tweet_Length <- nchar(train_raw$Text)

head(train_raw)

##
Text
## 1                                                                    Our
Deeds are the Reason of this #earthquake May ALLAH Forgive us all
## 2
Forest fire near La Ronge Sask. Canada
## 3 All residents asked to 'shelter in place' are being notified by
officers. No other evacuation or shelter in place orders are expected
## 4
13,000 people receive #wildfires evacuation orders in California
## 5                                              Just got sent this photo
from Ruby #Alaska as smoke from #wildfires pours into a school
## 6                        #RockyFire Update => California Hwy. 20 closed in
both directions due to Lake County fire - #CAfire #wildfires
##   Target Tweet_Length
## 1      1           69
## 2      1           38
## 3      1          133
## 4      1           65
## 5      1           88
## 6      1          110
```

Let's see what the maximum length of a tweet was here

```
max(train_raw$Tweet_Length)

## [1] 163
```

Alas! Some of these tweets must've been created after 2020.

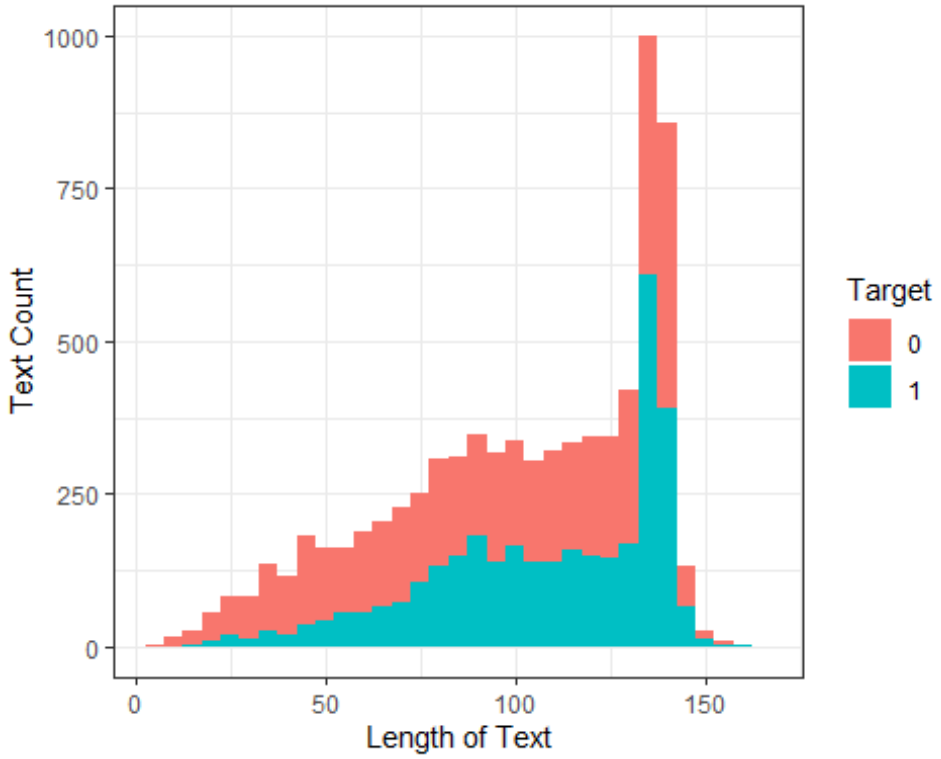
Now I'll just check how many 0 and 1 are available in the dataset.

```
summary(train_raw$Target)

##    0    1
## 4342 3271
```

Let's visualize the text details here.

```
ggplot(train_raw, aes(x = Tweet_Length, fill =
Target))+theme_bw()+geom_histogram(binwidth = 5)+labs(y = "Text Count", x =
"Length of Text")
```



Based on the graph above, we can see that most tweets in this dataset are not about disasters (are 0).

Now let's begin splitting the dataset into a training and testing sets.

```
set.seed(1246949)

index <- createDataPartition(train_raw$Target,
                             times = 1, p = 0.7, list = FALSE)

train <- train_raw[index,]
test <- train_raw[-index,]

#check number of rows in the training set vs testing
nrow(train)

## [1] 5330

nrow(test)

## [1] 2283
```

Now let's tokenize the text. Each word is considered a token.

```
library(quanteda)
train_tokens <- tokens(train$Text, what = "word",
                       remove_numbers = TRUE, remove_punct = TRUE,
                       remove_symbols = TRUE, remove_url = TRUE)

train_tokens[[80]]
```

```
##  [1] "So"         "i"          "guess"      "no"         "one"
##  [6] "actually"   "wants"      "any"        "free"       "Aftershock"
## [11] "TC"
```

Notice that some tokens have uppercase letters. Let's convert them all to lowercase to keep things simple!

```
train_tokens <- tokens_tolower(train_tokens)

train_tokens[[80]]
```

```
##  [1] "so"         "i"          "guess"      "no"         "one"
##  [6] "actually"   "wants"      "any"        "free"       "aftershock"
## [11] "tc"
```

We can now use the built-in function stopwords to remove words like "a", "the", "is", or "are" and many more. Stopwords is a popular term/concept/function used in NLP modeling, so it's not really unique to this library only (this is mostly a note to myself since I'm new to this stuff).

```
train_tokens <- tokens_select(train_tokens, stopwords(), selection =
"remove")

train_tokens[[80]]
```

```
## [1] "guess"      "one"        "actually"   "wants"      "free"
## [6] "aftershock" "tc"
```

Let's do some stemming now. "Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma."

```
train_tokens <- tokens_wordstem(train_tokens, language = "english")

train_tokens[[80]]
```

```
## [1] "guess"      "one"        "actual"     "want"       "free"
## [6] "aftershock" "tc"
```

I don't see a difference… Let's try a different tweet.

```
train_tokens[[90]]
```

```
## [1] "thing"   "stand"   "dream"   "tri"     "belief"  "actual"  "possibl"
## [8] "joel"    "brown"
```

I see possible has a missing e. It's not perfect, but it worked!

We can now go into NLP and create bag of words. Dfm is a type of "Matrix-Class" object. Dim below is used to get its dimensions.

```
train_tokens_dfm <- dfm(train_tokens, tolower = FALSE)

dim(train_tokens_dfm)

## [1]  5330 12038

#colnames(train_tokens_dfm)    prints the column names in the dfm;
#                    commented out since its too long to show in the document
```

Transforming the dfm to a matrix below.

```
train_tokens_matrix <- as.matrix(train_tokens_dfm)

dim(train_tokens_matrix)

## [1]  5330 12038
```

We are now ready to set up the features data frame with all the labels.

```
train_tokens_df <- cbind(Target = train$Target, convert(train_tokens_dfm, to
= "data.frame"))

#clean up the column names
names(train_tokens_df) <- make.names(names(train_tokens_df))

dim(train_tokens_df)

## [1]  5330 12040
```

Find any NA's

```
sum(is.na(train_tokens_df))

## [1] 0
```

We'll use the 10 fold cross validation method to create stratified folds to fix imbalances.

```
library(doSNOW)

## Loading required package: foreach

## Loading required package: iterators

## Loading required package: snow
```

```
cv_folds <- createMultiFolds(train$Target, k = 10 , times = 3 )
cv_control <- trainControl(method = "repeatedcv", number = 10 ,
                           repeats = 3, index = cv_folds)

start.time <- Sys.time()
```

Creating the cluster

```
library(xgboost)
library(lme4)

cl <- makeCluster(4)

clusterExport(cl, "train_tokens_df")

#clusterEvalQ(cl, train_tokens_df)
```