Description

Sergey

January 9, 2017

Описание работы. После просмотра информации, было решено, что столбцы, влияющие на результат, будут следующие: - text - in_reply_to_user_id - user.followers_count.

Текст твита может оказывать большое влияние на то, будут ли его ретвитать или нет. Иногда независимо от того, кто является автором. Является ли автором твита тот пользователь, у которого он "на стене", также может является одним из факторов того, будут ли твит ретвитать. Чем больше подписчиков, тем больше пользователей увидят твит. (Сюда можно ещё добавить описание пользователя, так как он может содержать статус пользователя в обществе или другую информацию. Но у меня не хватает мощности компьютера для этого)

Считываем информацию из train.csv, выбираем нужные столбцы и переназываем их для удобства. Также было замечено, что в данных есть пропуски, поэтому удаляем их.

```
train <- read.csv("train.csv", header = T)

train.part <- train[, c("id", "text", "in_reply_to_user_id", "user.description",
                        "user.followers_count", "retweet_count")]

names(train.part) <- c("id", "text", "is_retweet", "description", "followers",
"retweet_count")
train.part <- na.omit(train.part)
```

Для предсказания будем строить модель random forest с 20 деревьями. Для этого понадобятся следующие библиотеки

```
install.packages("tm")
install.packages("SnowballC")
install.packages("e1071")
install.packages("randomForest")
install.packages("caret")

library(tm)
library(SnowballC)
library(e1071)
library(caret)
library(randomForest)
```

Перед тем, как строить модель преобразуем данные к нужной форме. Для того, чтобы учитывать текст твита построим "мешок слов" или словарь. Из текста твита нужно выбрать слова, построить словарь и выбрать из них наиболее встречающиеся, чтобы уменьшить затраты на память и время вычисления. Удаляем все ссылки и слова, не несущие информации(такие как: я, ты, он и др.)

```
onlyText <- function(x) {
  x <- gsub("[^a-zA-Z]", " ", x)
  return(gsub("http([[:alnum:]|[:blank:]]*)$", "", x))
}

tokenize <- function(x) {
  x <- tolower(x)
  x <- unlist(strsplit(x, split=" "))
  x
}

stopWords <- stopwords("en")
stopWords <- c(stopWords, "re", "ve", "ll", "dr")
```

Функция onlyText оставляет нужные слова. Функция tokenize разделяет слова(из предложения получаем массив слов). stopWords содержит английские слова, не несущие информацию. (Зная, что твиты написаны на разных языках, всё равно делал только для английского, так как на все нужно много времени и таких твитов гораздо больше, чем остальных, написанных на других языках).

Запускаем цикл, который для каждой строки(твита) в тексте удаляет "ненужные" слова.

```
removeStopWords <- sapply(1:nrow(train.part), function(x) {
  t <- train.part[x, 2]
  t <- onlyText(t)
  t <- tokenize(t)
  t <- t[nchar(t) > 1]
  t <- t[!t %in% stopWords]

  paste(t, collapse=" ")
})
```

Строим "мешок слов" и удаляем редкие термины.

```
train.part.vector <- VectorSource(removeStopWords)
train.part.corpus <- Corpus(train.part.vector,
                                        readerControl = list(language = "en"))
train.part.bag <- DocumentTermMatrix(train.part.corpus,
                                     control = list(stemming = TRUE))
train.part.bag <- removeSparseTerms(train.part.bag, 0.9988)
```

Столбец преобразуем в столбец, который будет показывать: является ли пользователь автором твита(1 - является, 0 - не является)

```
train.part[, 3] <- ifelse(train.part$is_retweet == "0.0" | train.part$is_retweet
== "", 1, 0)
```

Составим таблицу, столбцы которой будут слова из выше указанного "мешка слов", столбец, где показано: набрал ли твит 20 и более ретвитов за 48 часов, столбец, который будет показывать: является ли пользователь автором твита, столбец, показывающий количество подписчиков пользователя.

```
train.df <- data.frame(inspect(train.part.bag[1:nrow(train.part.bag), ]))
is.retweeted <- ifelse(train.part$retweet_count[1:nrow(train.part.bag)] >= 20,
1, 0)
train.df <- cbind(is.retweeted, train.df)
train.df <- cbind(train.part[1:nrow(train.part.bag), 3],
                  train.part[1:nrow(train.part.bag), 5],
                  train.df)
names(train.df)[1:2] <- c("is.retweet", "followers")
```

В vocab храним список слов из словаря.

```
vocab <- names(train.df)[-c(1,2,3)]
```

Строим модель random forest. Модель будет предсказывать: будет ли твит более 20 раз ретвитаться за 48 часов. Ответом будет 1 - будет, 0 - не будет. В качестве переменной отклика возьмём is.retweeted, предикторами являются все остальные столбцы таблицы train.df.

```
forest <- train(as.factor(is.retweeted) ~ ., data = train.df,
                method = "rf",
                trControl = trainControl(method = "cv", number = 2),
                prox = TRUE,
                ntree = 20,
                do.trace = 10,
                allowParallel = TRUE)
```

После построения модели сделаем прогноз для твитов из test.csv. Делаем все те же шаги, что и выше.

```
test <- read.csv("test.csv", header = T)

test.part <- test[, c("id", "text", "in_reply_to_user_id", "user.description",
                      "user.followers_count")]

names(test.part) <- c("id", "text", "is_retweet", "description", "followers")
test.part <- na.omit(test.part)
test.part[, 3] <- ifelse(test.part$is_retweet == "0.0" | test.part$is_retweet ==
"", 1, 0)



remove.stop.words.test <- sapply(1:nrow(test.part), function(x) {
  t <- test.part[x, 2]
  t <- onlyText(t)
  t <- tokenize(t)
  t <- t[nchar(t) > 1]
  t <- t[!t %in% stopWords]

  paste(t, collapse=" ")
})

test.part.vector <- VectorSource(remove.stop.words.test)
test.part.corpus <- Corpus(test.part.vector,
                           readerControl = list(language = "en"))
```

```
test.part.bag <- DocumentTermMatrix(test.part.corpus,
                                    control = list(stemming = TRUE, dictionary
= vocab))

test.df <- data.frame(inspect(test.part.bag[1:nrow(test.part.bag), ]))
test.df <- cbind(rep(2, nrow(test.df)), test.df)
test.df <- cbind(test.part[1:nrow(test.part.bag), 3],
                         test.part[1:nrow(test.part.bag), 5],
                         test.df)
names(test.df)[c(1,2)] <- c("is.retweet", "followers")
```

Построили таблицу с теми же столбцами, что и в train.df. Теперь предскажем результат, используя модель forest.

```
test.df[, 3] <- predict(forest, newdata = test.df, prob = T)
```

Запишем ответ в файл prediction.csv

```
result.df <- data.frame(test.part$id, result = test.df[, 3])
write.csv(result.df, "prediction.csv", quote = FALSE, row.names = FALSE)
```

В файле два стобца id - идентификатор твита, result - показывает, наберёт твит 20 и более ретвитов за 48 часов или нет (1 - наберёт, 0 - не наберёт).

(Свою модель forest я строил только на 10000 твитах, так как на большее мой компьютер не смог.)