Intelligent Systems NLP Project

Rodrigo Carbajo Benito

Problem

In this project, the reviews of different drugs will be analysed to then be classified into the categories bad, neutral, and good based on the rating that the users of the drug gave them. Knowing which drugs are more accepted or useful for the consumers can be very useful for pharmaceutical companies, not only to define the drugs produced by the company with the best and worst acceptance in the market, but also to do so with the products of other competitor companies.

Experiments done

To address the problem explained above, the following steps have been followed.

1. Loading the data

The data set was extracted from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Drugs.com%29). It is split into train a test csv, but it has also been loaded all together. As it can be seen, the data set contains 9 attributes, which correspond to the identifier, the name of the drug, the condition the drug is used for, the review of the user, the rating that the user gives to the drug, the date, a count of people who found that review useful, the label, which was constructed by discretizing the ratings in three intervals: bad (1-4), neutral (5-7) and good (8-10), and finally whether the instance corresponds to the train or test set.

```
drug_train <- read.csv("UCIdrug_train.csv")
drug_test <- read.csv("UCIdrug_test.csv")
drug <- read.csv("UCIdrugs.csv")

head(drug)
```

```
##   uniqueID            drugName                 condition
## 1        2 Medroxyprogesterone                Amenorrhea
## 2        3 Medroxyprogesterone Abnormal Uterine Bleeding
## 3        4 Medroxyprogesterone             Birth Control
## 4        5 Medroxyprogesterone Abnormal Uterine Bleeding
## 5        7 Medroxyprogesterone Abnormal Uterine Bleeding
## 6        8 Medroxyprogesterone Abnormal Uterine Bleeding
##
review
## 1
"I&#039;m 21 years old and recently found out I might have PCOS. I
haven&#039;t gotten my period since 2010. On October 13th 2015 I started
taking medroxyprogesterone 10mg for 10 days. It actually worked. My
period came on 4 days after finishing the pill. It started with very
```

light cramping and very light bleeding. This is only my second day of being on my period. Looking forward to seeing the outcome, if the bleeding gets heavier or not. This pill does cause mood swings. One minute I feel a certain way, the next minute it changes. I'm pretty sure I'm driving my fianc"
2
"I have been on the shot 11 years and until a month ago, never 1 period or even spotting. A month ago, dark brown blood, awful odor, & now I have had a full blown period for 2 days. Not sure what to think. I have continued the shot all of these years only because of the absent monthly."
3 "Ive had four shots at this point. I was on birth control pills for years due to excessive bleeding and extreme pain. Never had a regular period and would have them for over a week with HEAVY bleeding-im talking pad and tampon at the same time and it would still overflow. The pills only made it worse. Since the shot, i have had no issues. I have gained about 25 pounds, but that didnt happen until 2.5 months in. And i quit antidepressants so that could have been why i gained weight. After 2.5 months, there is no bleeding and no cramps. I almost forget what its like to have a period. And to be thruthful, my boyfriend cums inside me and has since ive been on the shot. Ive had no pregnancy scare or issues at all. Its super convenient and easy!"
4 "I had a total of 3 shots. I got my first one before leaving the hospital after giving birth to my first and only child. I chose it on reccomendation from a friend who loves it. I stopped taking it September 2014, counting the 9 months I was pregnant I had not had a period in 2 years. I finally went to the Doctor since we were taking about having another baby. They did an ultrasound and said that the wall of my uterus was pretty thick, but nothing was done about it. About a week later it finally started. May 9 2015 I started spotting, then a regular to heavy flow. Its is not Oct. 25 2015 and I am still bleeding. My OB told me the only thing she could do for my is a D and C, which I seem to have to keep rescheduling. Don't take the shot!"
5
"I'm 18 and got this for heavy bleeding. I've always heard that bc makes you gain weight but was NOT expecting what I go. I gained 20 lbs. from just one shot. Sent from my usual 135 to 155. Made me nauseated after I ate anything. Had to buy a whole new wardrobe because I couldn't fit anything. And my sex drive! Omg I felt like a 90 yr old women because I was as dry as a desert and couldn't get wet no matter what. It was awful. I really really don't suggest this. I would much rather have cramps that make me throw up rather than take this shot ever again"
6
"Im 19 and have been having heavy and painful periods since forever! I got my depo shot last month and I spotted yesterday at school. I was with my boyfriend but glad i took a good shower that he didn't notice...i hope!!! But same thing happened today morning too!! All that's happening is my head is hurting alittle but it started of brownish to a proper red. I really hope its not for long!!!! But in terms of weight

```
gain, i have been eating alotttt more and my mood is a bit harder to
control. I wont take the shot anymore, ill go for the pill but I would
recommend dealing with the cramps and flow unless the doctor says
something"
##   rating      date usefulCount   label   set
## 1     10 27-Oct-15          11    Good train
## 2      8 27-Oct-15           7    Good train
## 3      9 26-Oct-15          12    Good train
## 4      1 25-Oct-15           4     Bad train
## 5      5 22-Oct-15           6 Neutral train
## 6      5 21-Oct-15           2 Neutral train
```

2. Basic checks

After loading the data, some basic checks such as the encoding or the character normalization are performed.

```
library(utf8)
library(dplyr)

# Column selection
reviews_train <- pull(select(drug_train, "review"))
reviews_test <- pull(select(drug_test, "review"))

# Check the encoding of the reviews for training and test texts. If the
output is character(0), all the reviews are made of correct UTF-8
characters.
reviews_train[!utf8_valid(reviews_train)]

## character(0)

reviews_test[!utf8_valid(reviews_test)]

## character(0)

# Check character normalization (Normalized Composed Form) for training
and test texts
reviews_train_NFC <- utf8_normalize(reviews_train)
reviews_test_NFC <- utf8_normalize(reviews_test)

# If the outputs are 0, the texts are in NFC
sum(reviews_train_NFC != reviews_train)

## [1] 0

sum(reviews_test_NFC != reviews_test)

## [1] 0
```

3. Sentiment analysis

Prior to the classification of the reviews, sentiment analysis is used to get an idea of the words present in the texts that contribute more to a positive or negative feeling. This words will be probably related with the classification of a review as good, bad or neutral.

To do this, the data set is transformed into a corpus and then into a document-feature matrix, with tokenization and cleaning of the data performed at the same time.

```
library(quanteda)

# Use data as corpus
corpus <- corpus(drug, text_field = "review")

# Transform corpus into DFM
dfmat <- dfm(tokens(corpus) %>% tokens_tolower(),
             remove_punct = TRUE, remove_numbers = TRUE,  remove_symbols
= TRUE) %>%
  dfm_remove(stopwords('english'))
```

As it is done in the work “Converting to and from Document-Term Matrix and Corpus objects” (Julia Silge and David Robinson), some tasks related to sentiment analysis are performed.

```
# Prior to the classification, we can do some sentiment analysis
library(tidyr)
library(tidytext)

dfmat_sentiments <- tidy(dfmat)

dfmat_sentiments <- dfmat_sentiments %>%
  inner_join(get_sentiments("bing"), by = c(term = "word"))

dfmat_sentiments
```

```
## # A tibble: 1,345,422 x 4
##    document term   count sentiment
##    <chr>    <chr>  <dbl> <chr>
##  1 text1    worked     1 positive
##  2 text49   worked     1 positive
##  3 text74   worked     1 positive
##  4 text92   worked     1 positive
##  5 text101  worked     1 positive
##  6 text105  worked     1 positive
##  7 text107  worked     1 positive
##  8 text181  worked     1 positive
##  9 text192  worked     1 positive
## 10 text194  worked     1 positive
## # ... with 1,345,412 more rows
```

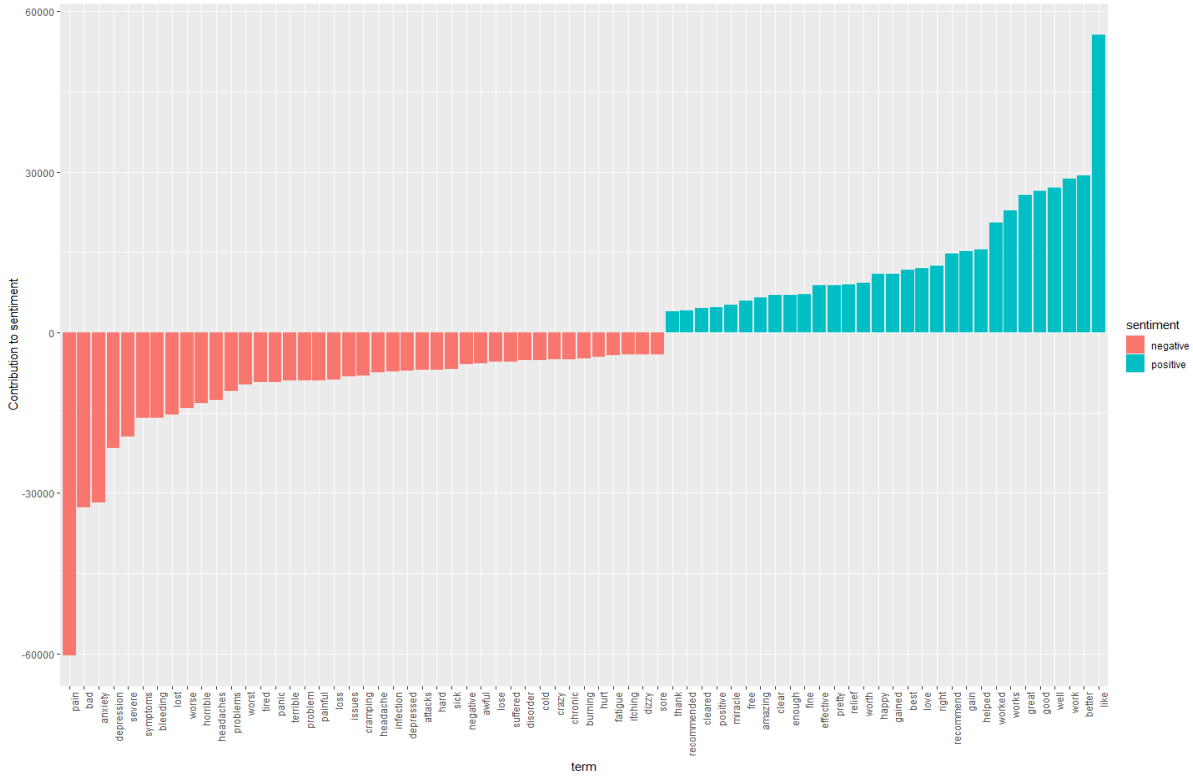
The count of different positive and negative words can be seen for each of the texts.

```
dfmat_sentiments %>%
  count(document, sentiment, wt = count) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative) %>%
  arrange(sentiment)
```

```
## # A tibble: 210,995 x 4
##    document   negative positive sentiment
##    <chr>         <dbl>    <dbl>     <dbl>
##  1 text126053       67       20       -47
##  2 text171678       67       20       -47
##  3 text130488       41        9       -32
##  4 text53986        41        9       -32
##  5 text192246       30        1       -29
##  6 text11812        44       21       -23
##  7 text124583       33       10       -23
##  8 text46694        44       21       -23
##  9 text79628        33       10       -23
## 10 text101177       21        0       -21
## # ... with 210,985 more rows
```

The documents with the most negative feelings can be detected.

```
library (ggplot2)
dfmat_sentiments %>%
  count(sentiment, term, wt = count) %>%
  filter(n >= 4000) %>%
  mutate(n = ifelse(sentiment == "negative", -n, n)) %>%
  mutate(term = reorder(term, n)) %>%
  ggplot(aes(term, n, fill = sentiment)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ylab("Contribution to sentiment")
```



Or the contribution of the words to a positive or negative feeling (the strongest ones). With a high probability, the reviews that contain these words will be easier to classify as positive feelings can be associated to a good review and negative feelings to a bad one.

4. Text classification: reviews into good, neutral or bad

After this exploration of the data, supervised classification models are going to be trained to predict whether a review is good or bad based on the words it contains.

```
# Split into train and test to train the machine learning models
dfm_train <- dfm_subset(dfmat, set == "train")
dfm_test <- dfm_subset(dfmat, set == "test")
```

Naive Bayes Classifier

As in the hands on, a Naive Bayes model was used, which is simple but very powerful as it has a similar performance in general to some other more complex models.

```
library(quanteda.textmodels)
library(caret)

# Train the model (using multinomial distribution as it presented the
same results in this case, in contrast to the results of the hands on )
model_nb <- textmodel_nb(dfm_train,
                         dfm_train$label,
                         distribution = "multinomial")
```

```r
# Prediction of the labels
pred_nb <- predict(model_nb,
                   newdata = dfm_test)

# Compute the confusion matrix
confM_nb <- confusionMatrix(table(pred_nb, docvars(dfm_test)$label))

# Compute the accuracy
acc_coincidences <- sum(as.character(pred_nb) ==
as.character(docvars(dfm_test)$label))
acc_total <- length(as.character(pred_nb))
acc_nb <- acc_coincidences/acc_total
acc_nb
```

```
## [1] 0.6829595
```

```r
# Show the metrics by class
confM_nb[["byClass"]]
```

```
##                Sensitivity Specificity Pos Pred Value Neg Pred Value
Precision
## Class: Bad       0.6878566   0.8736994      0.6460682      0.8930602
0.6460682
## Class: Good      0.7407030   0.7815287      0.8366271      0.6661625
0.8366271
## Class: Neutral   0.4387626   0.8411857      0.3230755      0.8966519
0.3230755
##                    Recall        F1 Prevalence Detection Rate
## Class: Bad      0.6878566 0.6663078  0.2510323     0.17267418
## Class: Good     0.7407030 0.7857482  0.6016628     0.44565339
## Class: Neutral 0.4387626 0.3721354  0.1473050     0.06463192
##                Detection Prevalence Balanced Accuracy
## Class: Bad                 0.2672693         0.7807780
## Class: Good                0.5326786         0.7611158
## Class: Neutral             0.2000521         0.6399742
```

Support Vector Machine Classifier

In the hands on it was learned that to train a SVM model, a sample of the original data needed to be taken as otherwise an error will appear due to the dimensionality of the data. SVM models are widely used because they offer very good results in classification problems. In this experiment, different sample sizes have been used to compare how metrics changed when the sample was increased.

```r
set.seed(23)
svmPredictions <- function(x,
                           weight){ #weight can be "uniform", "docfreq"
or "termfreq".

  # Sample of documents
  dfmat_train <- dfm_sample(dfm_subset(dfmat, set == "train"), x)
```

```
  dfmat_test <- dfm_subset(dfmat, set == "test")

  # Train the SVM model with the sample
  model_svm <- textmodel_svm(dfmat_train,
                         dfmat_train$label,
                         weight = weight)

  # Prediction of the labels
  pred_svm <- predict(model_svm,
                  newdata = dfmat_test)

  # Compute the confusion matrix
  confM_svm <- confusionMatrix(table(pred_svm,
docvars(dfmat_test)$label))

  # Compute the accuracy
  acc_coincidences <- sum(as.character(pred_svm) ==
as.character(docvars(dfmat_test)$label))
  acc_total <- length(as.character(pred_svm))
  acc_svm <- acc_coincidences/acc_total
  acc_svm

  # Show the metrics by class
  confM_svm[["byClass"]]

}

# Results for a sample size of 10000 and uniform weight
svmPredictions(10000, "uniform")
```

```
##                Sensitivity Specificity Pos Pred Value Neg Pred Value
Precision
## Class: Bad       0.5705712   0.8603889      0.5780230      0.8566872
0.5780230
## Class: Good      0.7922347   0.6357566      0.7666397      0.6695186
0.7666397
## Class: Neutral   0.2372475   0.8879946      0.2678928      0.8707863
0.2678928
##                    Recall        F1 Prevalence Detection Rate
## Class: Bad      0.5705712 0.5742729  0.2510323     0.14323178
## Class: Good     0.7922347 0.7792271  0.6016628     0.47665811
## Class: Neutral 0.2372475 0.2516406  0.1473050     0.03494774
##                Detection Prevalence Balanced Accuracy
## Class: Bad                0.2477960         0.7154801
## Class: Good               0.6217498         0.7139957
## Class: Neutral            0.1304542         0.5626210
```

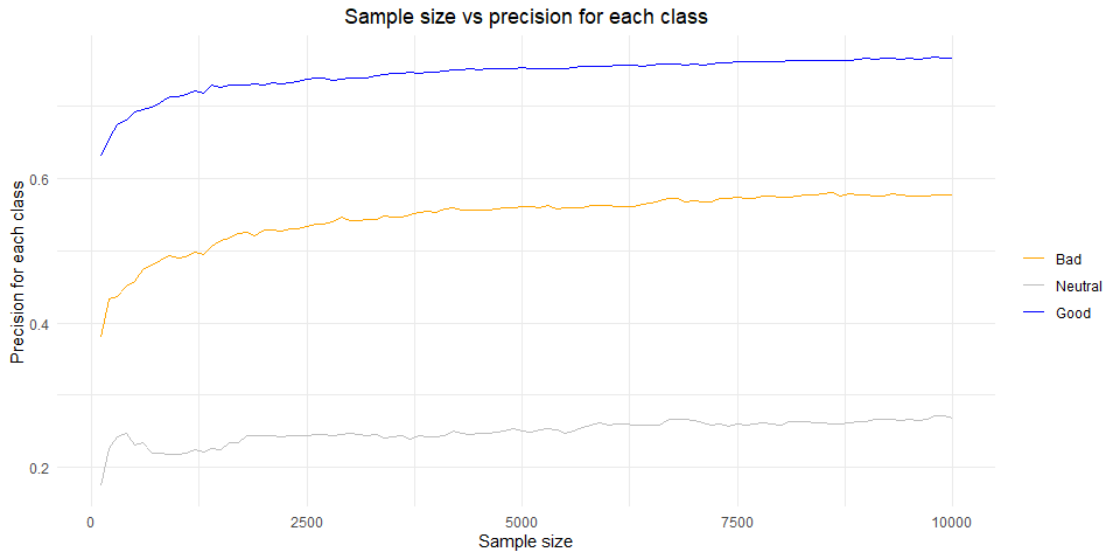
These are the results for the SVM when it is trained using 10,000 samples. Now, the results of the model trained with several sample sizes are going to be computed and stored to plot them in some graphs.

```
# Lists to save the precision and recall values for different sample
sizes
results_recall_bad <- list()
results_recall_neutral <- list()
results_recall_good <- list()

results_precision_bad <- list()
results_precision_neutral <- list()
results_precision_good <- list()


for (i in seq(100, 10000, by = 100)){
  set.seed(23)

  # Sample of documents
  dfmat_train <- dfm_sample(dfm_subset(dfmat, set == "train"), i)
  dfmat_test <- dfm_subset(dfmat, set == "test")

  # Train the SVM model with the sample
  model_svm <- textmodel_svm(dfmat_train,
                         dfmat_train$label,
                         weight = "uniform")

  # Prediction of the labels
  pred_svm <- predict(model_svm,
                  newdata = dfmat_test)

  # Compute the confusion matrix
  confM_svm <- confusionMatrix(table(pred_svm,
docvars(dfmat_test)$label))

  # Store the metrics on the lists
  results_recall_bad <- append(results_recall_bad, confM_svm$byClass[1,
1])
  results_recall_neutral <- append(results_recall_neutral,
confM_svm$byClass[3, 1])
  results_recall_good <- append(results_recall_good, confM_svm$byClass[2,
1])

  results_precision_bad <- append(results_precision_bad,
confM_svm$byClass[1, 3])
  results_precision_neutral <- append(results_precision_neutral,
confM_svm$byClass[3, 3])
  results_precision_good <- append(results_precision_good,
confM_svm$byClass[2, 3])
```

```
}


# Store data in a data frame
df <-data.frame(sample_size = seq(100, 10000, by = 100))
df$precision_bad = unlist(results_precision_bad)
df$precision_neutral = unlist(results_precision_neutral)
df$precision_good = unlist(results_precision_good)
df$recall_bad = unlist(results_recall_bad)
df$recall_neutral = unlist(results_recall_neutral)
df$recall_good = unlist(results_recall_good)
```

We now plot the sample size vs the precision for each class.

```
# Plot the precision evolution
ggplot(df, aes(x = sample_size)) +
  geom_line(data = df, aes(y = precision_bad, colour = "Bad")) +
  geom_line(data = df, aes(y = precision_neutral, colour = "Neutral")) +
  geom_line(data = df, aes(y = precision_good, colour = "Good")) +
  xlab("Sample size") +
  ylab("Precision for each class") +
  theme_minimal()+
  ggtitle("Sample size vs precision for each class") +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_colour_manual("",
                      breaks = c("Bad", "Neutral",  "Good"),
                      values = c("orange", "gray", "blue"))
```



And the sample class vs the recall of each class.

```
# Plot the recall evolution
ggplot(df, aes(x = sample_size)) +
  geom_line(data = df, aes(y = recall_bad, colour = "Bad")) +
  geom_line(data = df, aes(y = recall_neutral, colour = "Neutral")) +
  geom_line(data = df, aes(y = recall_good, colour = "Good")) +
  xlab("Sample size") +
  ylab("Recall for each class") +
  theme_minimal()+
  ggtitle("Sample size vs recall for each class") +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_colour_manual("",
                      breaks = c("Bad", "Neutral",  "Good"),
                      values = c("orange", "gray", "blue"))
```