DE SIMONI Pietro
SESSI Michela

Stockholm University

Report of Laboratory Exercise 3 :

Opinion Mining

Group 15

Introduction

Nowadays, products, services and locations are exposed to comments and opinions in the web. It is really easy to find some reviews if you want to evaluate, choose or buy something. On the other hand, it is useful to have a global vision of people's opinion: sometimes blogs and sites provide a structured form and some aggregated measures are reported, however the majority of the data are unstructured and have just a text format.

This assignment wants to evaluate the opinion that people has about a product as an example. The method exploits the WEKA machine learning tool in order to train and test a model to classify book reviews as positive or negative.
The aim is to understand how well the SVM model in WEKA could classify, how the result is affected by the size of the training and how stop word filtering and stemming influence the performance.

The paper body follows the instruction schema. In order to be clear, we provide the text of the questions in addition to our content.

Content & Discussion

We start this exercise just analyzing the content of the dataset provided.

Question 1: Does this look like texts containing opinions? Are they correctly classified as positive and negative? Why could it sometimes be difficult to determine if a review should be classified as positive or negative?

The file contains reviews, but the format is not really user friendly.
Investigating better, it is possible to understand the meaning of all the information contained in the file. For each review is provided:

- text
- user
- rating (from 1 to 5)
- location
- product

The classification of these reviews seems to be always correct. Obviously we couldn't check them all by hand: we just looked some.

Sometimes it could be hard for a computer to judge if a review is positive or negative because it's difficult for a machine to detect irony, and also for other reasons like ambiguous words. For example a review like *"I hate people who say that Haruf is a bad writer, I can't agree with them!"* Contains the words *hate* and *bad* that could be misleading.

The created file book_review.arff contains all the reviews and the class: positive or negative. In order to use it on WEKA tool, we transform this file from string to vector format.

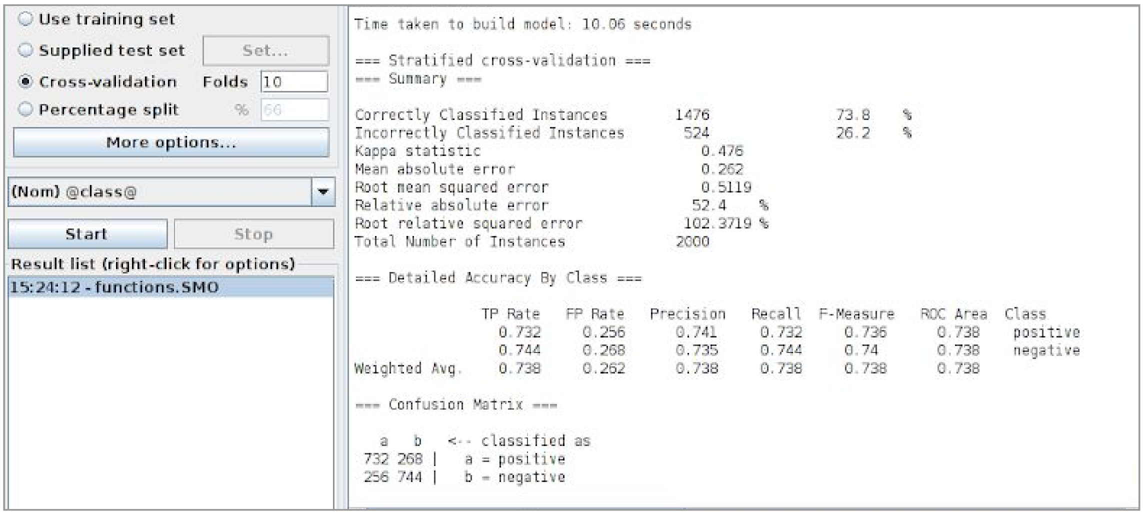
Question 2: Check the file book_review_vector.arff, what does it contain?

The file is composed of two parts. The first one contains the features, which are the class label containing the class, and the 1251 words contained in the reviews.
In the second part there is a line for each review, reporting the id of the words contained in it and the number of times that they appear in it.

As first step we train a SVM model with 10 folds cross-validation.

Question 3: What percentage correctly classified instances did you obtain?

We have 73.8% correctly classified instances. This is a good start but maybe it is possible to improve it.



○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66
(Nom) @class@
Result list (right-click for options)
15:24:12 - functions.SMO

```
Time taken to build model: 10.06 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1476               73.8    %
Incorrectly Classified Instances       524               26.2    %
Kappa statistic                          0.476
Mean absolute error                      0.262
Root mean squared error                  0.5119
Relative absolute error                 52.4    %
Root relative squared error            102.3719 %
Total Number of Instances             2000

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.732     0.256      0.741     0.732     0.736      0.738    positive
                 0.744     0.268      0.735     0.744     0.74       0.738    negative
Weighted Avg.    0.738     0.262      0.738     0.738     0.738      0.738

=== Confusion Matrix ===

   a   b   <-- classified as
 732 268 |   a = positive
 256 744 |   b = negative
```

Figure 1 - Output SVM 10 folds cross-validation

Figure 1 shows the output: some more statistics are reported and the confusion matrix is provided. A feature that could be relevant is the time taken to build the model: 10.05 seconds in this case.

We train the model again this time dividing the dataset in training and test set. We want to divide the dataset in different proportions in order to see how it affects the accuracy on the test set.

Question 4: How do the performance results of the classifier change with the size of training set? Why do you think that is?

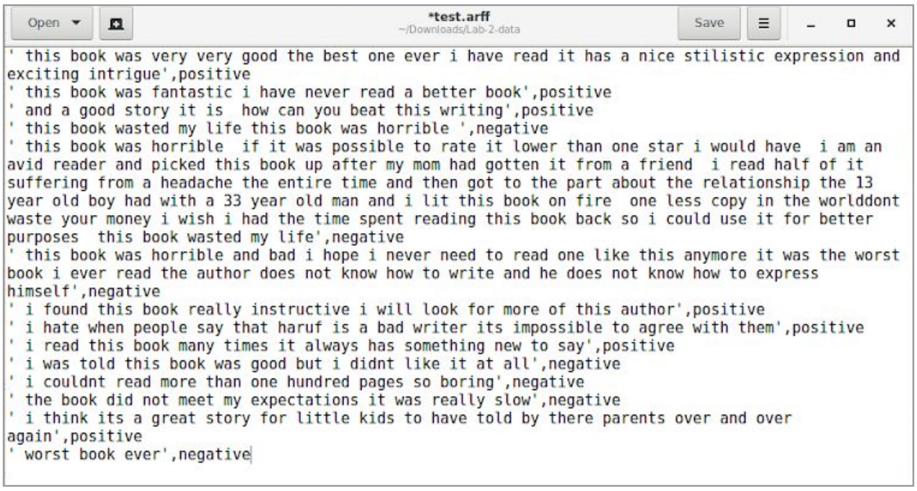
The results are given in the subsequent *Table 1*.

Training size	10%	20%	30%	40%	50%	60%	70%	80%	90%
Accuracy (%)	67.1667	69.375	69.2143	70.0833	71.2	72.25	71	71.25	74

Table 1 - Training percentage and related accuracy from SVM

We can observe that performance are improved increasing the percentage of data used for training the model. We could expect a result like this because the more data we use for training, the better we can train the model.

Following the exercise, we now use all the previous dataset to train the SVM model. As test set a file is provided: we add some more reviews in order to have more observations to test. *Figure 2* shows the file test.arff with the added reviews and corresponding feature value *positive* or *negative*.

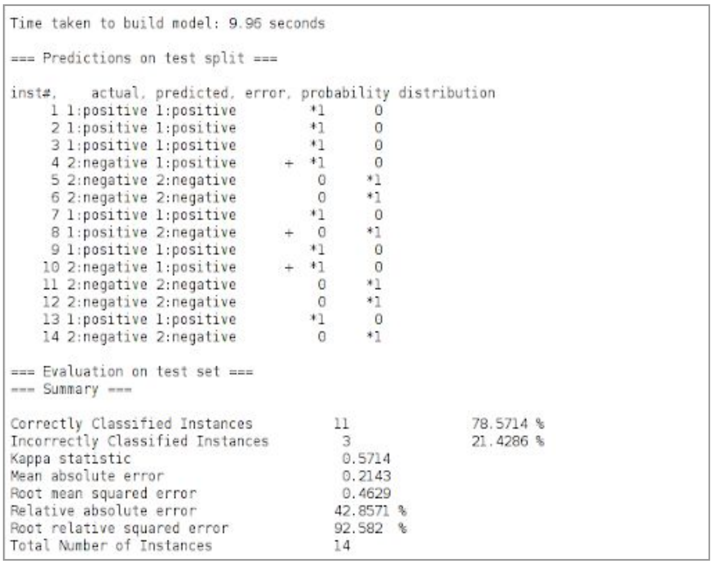


```
' this book was very very good the best one ever i have read it has a nice stilistic expression and
exciting intrigue',positive
' this book was fantastic i have never read a better book',positive
' and a good story it is  how can you beat this writing',positive
' this book wasted my life this book was horrible ',negative
' this book was horrible  if it was possible to rate it lower than one star i would have  i am an
avid reader and picked this book up after my mom had gotten it from a friend  i read half of it
suffering from a headache the entire time and then got to the part about the relationship the 13
year old boy had with a 33 year old man and i lit this book on fire  one less copy in the worlddont
waste your money i wish i had the time spent reading this book back so i could use it for better
purposes  this book wasted my life',negative
' this book was horrible and bad i hope i never need to read one like this anymore it was the worst
book i ever read the author does not know how to write and he does not know how to express
himself',negative
' i found this book really instructive i will look for more of this author',positive
' i hate when people say that haruf is a bad writer its impossible to agree with them',positive
' i read this book many times it always has something new to say',positive
' i was told this book was good but i didnt like it at all',negative
' i couldnt read more than one hundred pages so boring',negative
' the book did not meet my expectations it was really slow',negative
' i think its a great story for little kids to have told by there parents over and over
again',positive
' worst book ever',negative
```

Figure 2 - Reviews in the test set

Question 5: How many correctly classified sentences in percentage did you obtain?

As shown in *Figure 3*, now the correctly classified instances are about 78.6% of the test.



```
Time taken to build model: 9.96 seconds

=== Predictions on test split ===

inst#,    actual, predicted, error, probability distribution
     1 1:positive 1:positive        *1      0
     2 1:positive 1:positive        *1      0
     3 1:positive 1:positive        *1      0
     4 2:negative 1:positive     +  *1      0
     5 2:negative 2:negative         0     *1
     6 2:negative 2:negative         0     *1
     7 1:positive 1:positive        *1      0
     8 1:positive 2:negative     +   0     *1
     9 1:positive 1:positive        *1      0
    10 2:negative 1:positive     +  *1      0
    11 2:negative 2:negative         0     *1
    12 2:negative 2:negative         0     *1
    13 1:positive 1:positive        *1      0
    14 2:negative 2:negative         0     *1

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          11               78.5714 %
Incorrectly Classified Instances         3               21.4286 %
Kappa statistic                          0.5714
Mean absolute error                      0.2143
Root mean squared error                  0.4629
Relative absolute error                 42.8571 %
Root relative squared error             92.582  %
Total Number of Instances               14
```

Figure 3 - Output SVM to our test set

Question 6: How does this result compare to previous results? What are the drawbacks of this evaluation?

We think this is a good result, even if there's a just little raise in the accuracy. We need in fact to consider that the test contains only 14 reviews, and two of the ones we added were really tricky:

"I was told that this book was good but i didn't like it at all" (negative)

"I hate when people say that Haruf is a bad writer. It's impossible to agree with them" (positive)

The first one contains the word *good* and if we consider the words one by one we don't consider "*didn't like*" but we divide *didn't* and *like.* So probably *"good"* and *"like"* will contribute to make this review misclassified as positive. Similarly the second example contains words that can mislead the classifier.
A part from these two examples only another one was misclassified.

Question 7: There are many ways to try to improve classification results. Discuss and motivate the methods you would do to try to improve the results you got during the lab so far.

The result is already good but it's possible to improve it with some techniques. Since we are dealing with text data it is good to remove stop-words because they don't have a value themselves. Including them in our dataset just adds noise to our trained model.
Another good technique to use when dealing with text data is stemming (reducing the inflected words to their basic form). This makes possible to consider two or more inflected forms of the same word as the same.

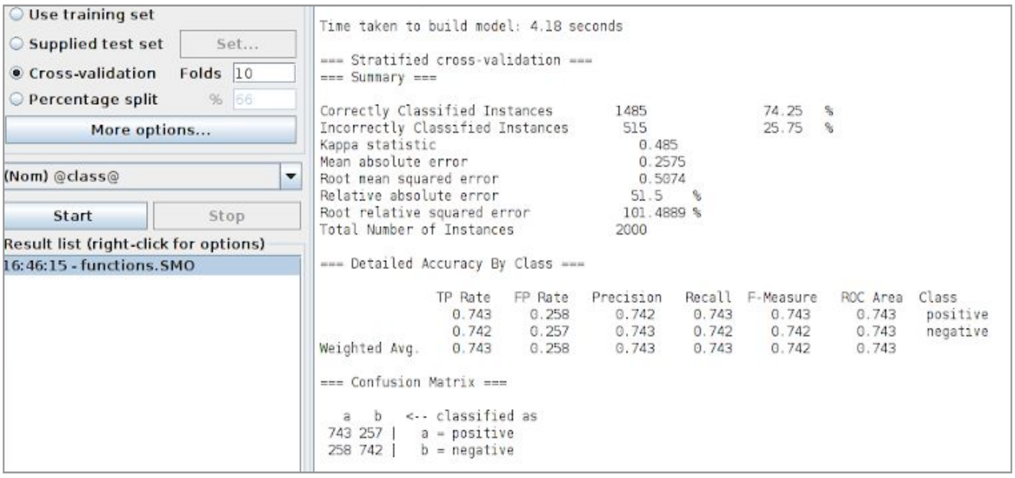
In order to improve our model it could be useful to employ the misclassified reviews inside the training set, in this way we could learn something new from this data and maybe obtain a better result.
For sure it is important to expand our training set, we want to remember that ideally the more data we train the more accurate the result will be.

The exercise goes on with the implementation of a new model. Now we use a stopwords list in order to filter the reviews and to train a better model.

Question 8: Did the stop word filtering improve or impair the results after 10-fold cross validation? How much? What size does the stop word filtered and the not stop word filtered file have respectively? Would you recommend using stop word filtering?

As the output in *Figure 4* shown, accuracy is improved a little bit (74.25 %) The best result is that we reduced the dataset. It was about 2MB and now it's around 400kB. We are now dealing with only 2000 reviews but in a real life case, you can have millions of reviews, so reducing the size of the dataset could be great.

Another improvement is that now the model trains faster. If before it took around 10 seconds to train it, now it takes only 4 seconds. One could say that it's just six seconds, but again if you imagine to be dealing with millions of reviews training the model in half the time would be great.



● Cross-validation Folds 10
○ Percentage split % 66
(Nom) @class@

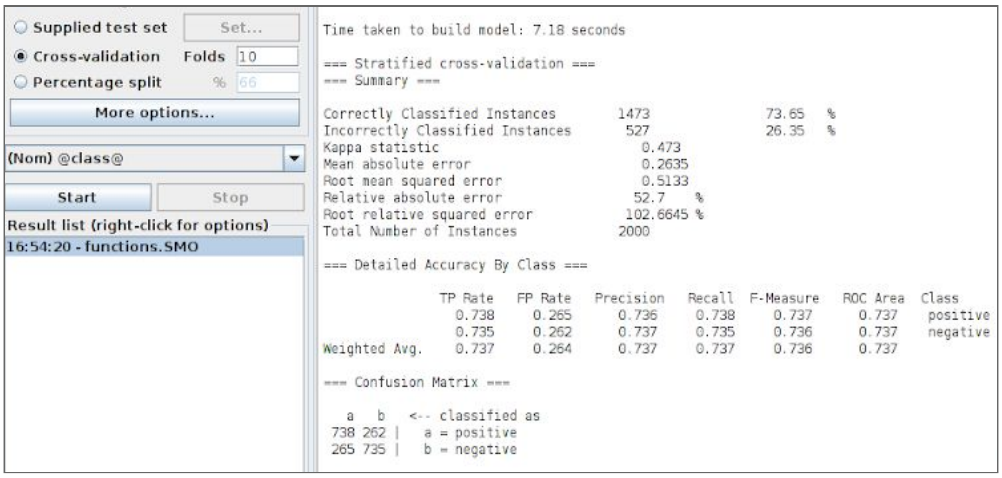
```
Time taken to build model: 4.18 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1485               74.25   %
Incorrectly Classified Instances       515               25.75   %
Kappa statistic                          0.485
Mean absolute error                      0.2575
Root mean squared error                  0.5074
Relative absolute error                 51.5    %
Root relative squared error            101.4889 %
Total Number of Instances             2000

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.743     0.258     0.742     0.743     0.743      0.743    positive
                 0.742     0.257     0.743     0.742     0.742      0.743    negative
Weighted Avg.    0.743     0.258     0.743     0.743     0.742      0.743

=== Confusion Matrix ===

   a   b   <-- classified as
 743 257 |   a = positive
 258 742 |   b = negative
```

Figure 4 - Output using stopwords filtering

In addition to the stopwords filtering, we apply stemming. In particular we use the function IteratedLovinsStemmer provided by WEKA.

Question 9: Did stemming improve or impair the results after 10-fold cross-validation? How much? Does the stemmer work properly (HINT check the attribute list)? Which of the stemmers did you try?

Looking at the word list generated by WEKA, when using stemming we notice that words are now reported in their non-inflected form. In some cases this is good but we don't understand how some words were generated and if they have been correctly stemmed.
For example there is a word: *"ww"* and we don't have any clue about what the original word could be so we doubt it has been stemmed properly.
Accuracy is not better than the one with the previous approaches. We can see the output reported in *Figure 5*



○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) @class@

Result list (right-click for options)
16:54:20 - functions.SMO

```
Time taken to build model: 7.18 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1473               73.65   %
Incorrectly Classified Instances       527               26.35   %
Kappa statistic                          0.473
Mean absolute error                      0.2635
Root mean squared error                  0.5133
Relative absolute error                 52.7     %
Root relative squared error            102.6645 %
Total Number of Instances             2000

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.738     0.265      0.736     0.738     0.737      0.737    positive
                 0.735     0.262      0.737     0.735     0.736      0.737    negative
Weighted Avg.    0.737     0.264      0.737     0.737     0.736      0.737

=== Confusion Matrix ===

   a   b   <-- classified as
 738 262 |   a = positive
 265 735 |   b = negative
```

Figure 5 - Output using stopwords filtering

Conclusion

The assignment shows how simple is to use a tool to analyze text data and extract useful information. The same classification could have been done by hand but that would take a lot of time and of course cost a lot of money. It is much more efficient to classify by home enough reviews (the more the better) and then train a classifier.