J.F.P. (Richard) Scholtens – s2956586 – Assignment 1 – Information Retrieval

Task 1:

Run the source code for the seven categories. It uses the naive Bayes algorithm, 90% of the data for training and the remaining 10% for testing. You should see an overall accuracy score, and precision and recall scores for each category. Report these values. Note: if there is any error check below the section troubleshooting.

```
Accuracy: 0.521415
 |------------------ |------------------- |------------------- |------------------- |
 |category           |precision           |recall              |F-measure           |
 |------------------ |------------------- |------------------- |------------------- |
 |ECONOMIE           |1.000000            |0.200000            |?                   |
 |SPORT              |1.000000            |0.240000            |?                   |
 |KUNST              |0.300000            |0.176471            |?                   |
 |BINNENLAND         |NA                  |NA                  |NA                  |
 |BUITENLAND         |0.909091            |0.192308            |?                   |
 |RECENSIE           |0.897436            |0.790960            |?                   |
 |INTERVIEW          |0.302395            |0.990196            |?                   |
 |------------------ |------------------- |------------------- |------------------- |
```

Task 2:

Implement the F-score (in the "empty" function provided to that end). Re-run the classification of task 1 and now report all the metrics (accuracy, precision, recall and f-measure for each category).

```
Accuracy: 0.489758
 |------------------ |------------------- |------------------- |------------------- |
 |category           |precision           |recall              |F-measure           |
 |------------------ |------------------- |------------------- |------------------- |
 |ECONOMIE           |1.000000            |0.205128            |0.340426            |
 |SPORT              |1.000000            |0.228070            |0.371429            |
 |KUNST              |0.000000            |0.000000            |0                   |
 |BINNENLAND         |0.000000            |0.000000            |0                   |
 |BUITENLAND         |1.000000            |0.314815            |0.478873            |
 |RECENSIE           |0.847222            |0.802632            |0.824324            |
 |INTERVIEW          |0.276163            |0.922330            |0.425056            |
 |------------------ |------------------- |------------------- |------------------- |
```

Task 3:

Report which are the 10 most informative words in your categories. You can use the function "classifier.show_most_informative_features".

```
 Most Informative Features
           Reuter = True          BUITEN : RECENS =    282.5 : 1.0
              dat = None           SPORT : INTERV =    259.8 : 1.0
              [?] = True          RECENS : ECONOM =    203.1 : 1.0
              een = None           SPORT : RECENS =    155.9 : 1.0
               en = None           SPORT : INTERV =    130.2 : 1.0
             ISBN = True          RECENS : BUITEN =    112.4 : 1.0
        wedstrijd = True           SPORT : ECONOM =     97.5 : 1.0
        president = True          BUITEN : SPORT  =     96.2 : 1.0
          miljard = True          ECONOM : SPORT  =     92.4 : 1.0
              met = None           SPORT : INTERV =     85.9 : 1.0
```

Task 4:
Implement 10-fold cross-validation. Your program should then report at the end 10 accuracy scores (1 per line), and as the last line the average of those 10 accuracy values. I.e. if I run your program I expect that the last 11 lines of its output contain float numbers and nothing else.

```
Accuracies
0.4944029850746269
0.46828358208955223
0.5167910447761194
0.539179104477612
0.48789571694599626
0.4925373134328358
0.4832089552238806
0.5335820895522388
0.4832089552238806
0.49906890130353815

Accuracies average
0.4998158648100281
```

Task 5:
Try to improve the average accuracy. Things you can try include (non exhaustive list): • Pre-process the data. E.g. tokenise, remove punctuation and stop words, lowercase the text, stemming, etc. • Use a subset of features. Instead of using all the words, you can use the top N most informative ones (you can try different values of N and plot a graph showing the average accuracy for each value of N). You can obtain the top informative words using the function “high_information_words” (in featx.py). • Use a different classifier. E.g. “MaxEntClassifier” or “DecisionTreeClassifier” are implemented in NLTK. You could also use a classifier from another library, e.g. sklearn. You should explain in your own words how the classifier

Added following pre-processing features:

- Tokenisation
- The exclusion of punctuation
- The exclusion of stopwords
- Lower casing of text
- Stemming

Used different classifier:

- Multinomial Naïve Bayes from the Sklearn package

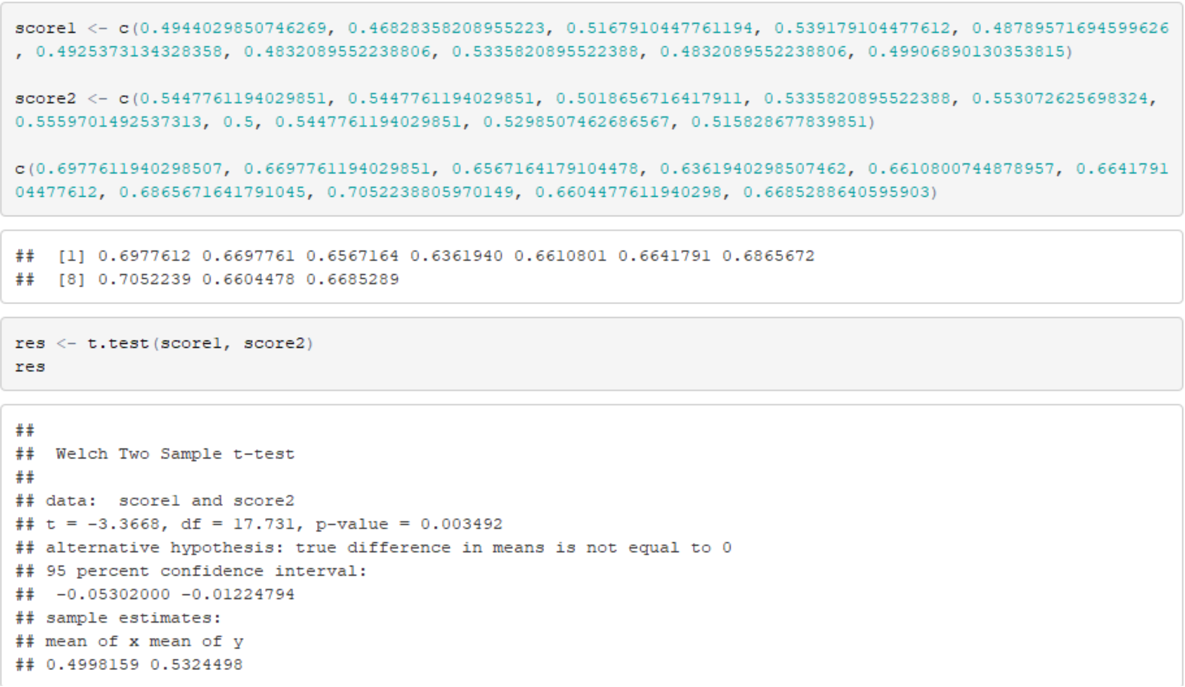
I tried implementing a feature which would have used high information words. This did work but it would sometimes fail to provide words for a specific category which lead to early termination of the program. This was because there were no words left within this category. Therefor I had chosen to leave this implementation out of my program.

I used the Multinomial Naïve Bayes from the Sklearn package. The Multinomial Naïve Bayes classifier does not differ very much from the original classifier. It only difference is that it considers the distribution of the data. Whereas the original classifier is a general term which refers to conditional independence of each of the features in the mode. Multinomial Naive Bayes classifier is a specific instance of a Naive Bayes classifier which uses a multinomial distribution for each of the features. Because the data exists out very high quantity of words, we can easily apply the Multinomial Naïve Bayes. This is because it is easy to calculate word frequencies.

Accuracies
0.6977611940298507
0.6697761194029851
0.6567164179104478
0.6361940298507462
0.6610800744878957
0.664179104477612
0.6865671641791045
0.7052238805970149
0.6604477611940298
0.6685288640595903

Accuracies average

0.6706474610189277
Task 6:



```
score1 <- c(0.4944029850746269, 0.46828358208955223, 0.5167910447761194, 0.539179104477612, 0.48789571694599626
, 0.4925373134328358, 0.4832089552238806, 0.5335820895522388, 0.4832089552238806, 0.49906890130353815)

score2 <- c(0.5447761194029851, 0.5447761194029851, 0.5018656716417911, 0.5335820895522388, 0.553072625698324,
0.5559701492537313, 0.5, 0.5447761194029851, 0.5298507462686567, 0.515828677839851)

c(0.6977611940298507, 0.6697761194029851, 0.6567164179104478, 0.6361940298507462, 0.6610800744878957, 0.6641791
04477612, 0.6865671641791045, 0.7052238805970149, 0.6604477611940298, 0.6685288640595903)
```

```
##  [1] 0.6977612 0.6697761 0.6567164 0.6361940 0.6610801 0.6641791 0.6865672
##  [8] 0.7052239 0.6604478 0.6685289
```

```
res <- t.test(score1, score2)
res
```

```
##
##  Welch Two Sample t-test
##
## data:  score1 and score2
## t = -3.3668, df = 17.731, p-value = 0.003492
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.05302000 -0.01224794
## sample estimates:
## mean of x mean of y
## 0.4998159 0.5324498
```

In this t-test the accuracy scores of a NLTK Naive Bayes classifier was used. The corpora included many Dutch newspapers. One vector includes the scores of without pre-processed data. The other includes the scores with pre-processed data. We use the t-test because the data is symmetric.

H0: The accuracy of the scores of non-pre-processed data is not significantly different than that of the accuracy scores of the pre-processed data.

Ha: The accuracy of the scores of non-pre-processed data is significantly different than that of the accuracy scores of the pre-processed data.

This hypothesis is tested with a T-test.
Significance level: 0.05:
t = -3.3668, df = 17.731, p-value = 0.003492

Conclusion:

The p-level is lower than the significance level. Therefor we reject H0 and accept Ha.
This means that the accuracy scores of the pre-processed data is significantly different than that of the accuracy scores of the pre-processed data.