CENG 463
Assignment 1

Cem Gündoğdu

cem.gundogdu@metu.edu.tr

December 16, 2021

Contents

1 Naive Bayes Classifier

1.1 Introduction

I used the `NaiveBayesClassifier` class from the NLTK library. I have tried various preprocessing techniques, and logged the accuracy along with the confusion matrix and metrics such as precision.

1.2 Metrics for different preprocessing techniques

In this section, I list the accuracies for different preprocessing techniques. I haven't tried all combinations of them, since the number grows exponentially. However, I added each technique **incrementally** (added the preprocessing step without removing previous ones), with the hope that the change caused by the added technique will be representative of its usefulness.

These trials were made on the *dev* set, since I did them during the development phase. The evaluation in subsection 1.3 will be done on the *test* set.

Complete logs from each try is available in the data directory. These include the confusion matrix, accuracy, and for each category the precision, recall and F_1 measure. I include only the last log here, to keep this report short.

Details of the calculation of precision, recall and F_1 measure are given in subsection 1.3.

A table of the accuracies for each technique is in Table 1. As seen in the table, each techniques improved the accuracy in SVC. But for Naive Bayes, removing the punctuation and stemming caused a decrease in accuracy.

Table 1: Accuracies for different preprocessing techniques.

Preprocessing type	Accuracy (%)	
	Naive Bayes	Support Vector
Simplest version	64.20	64.09
Lowercase conversion	65.92	66.77
Punctuation removal	65.38	68.70
Stopword removal	67.63	70.74
Stemming	66.67	72.34
Removing short words	66.99	72.78

1.2.1 Simplest version

The words in title are counted twice — in this version and in all the following ones. Other than this, there is no processing in this version. The text is given to `nltk.word_tokenize()` and the resulting tokens are used to train the classifier.

1.2.2 Lowercase conversion

Converted all words to lowercase.

This improved accuracy in both SVC and NBC. I think this is because our corpus size is rather small. So, this change allows us to make better use of the limited data.

I think, on a larger corpus, keeping the case could be more beneficial, since it would allow us to distinguish the words in the title from the words in the body.

1.2.3 Punctuation removal

Removed the following characters from the text:

```
!"#$%&'()*+,./:;<=>?@[\]^_`{|}~'
```

I did not replace them with spaces, but simply removed them. This made, for example, the text *Sophies's* to become the single word *sophies*.

Removing the punctuation caused a decrease in the accuracy of NB classifier. I was expecting a decrease, since punctuation can actually be helpful in understanding the book's genre. For example, one could expect to see more question marks in a mystery book's description. However, you wouldn't expect lots of exclamation marks in a science book, unless the author got very excited about whatever scientific topic they were writing about.

SVC's accuracy increased.

1.2.4 Stopword removal

I used the corpus `nltk.corpus.stopwords.words('english')` from the NLTK library. I removed every word that occured in this list.

This improved the accuracy of both classifiers.

1.2.5 Stemming

I passed each word to the `nltk.stem.PorterStemmer()` from the NLTK library.

This decreased the accuracy of NB, and resulted in a significant improvement in SVC.

1.2.6 Removing short words

I thought stopword removal should have been enough, but I also tried removing words that were shorter than 3 characters. Surprizingly, this resulted in an increase in the accuracy of both classifiers.

1.3 Evaluation

1.3.1 Calculation of metrics

Recall is the ratio of true positives for a class to the number of input documents of that type. To find recall, we divide each diagonal entry by the sum of corresponding row.

Precision is the ratio of true positives for a class to the number of documents that are identified to be in that class. To calculate it, we divide diagonal entries by the sum in that column.

Figure 1: Metrics for the best SVC version on test data set

```
Loaded classifier from cache.
Accuracy: 0.7194206008583691
                |                           s     |
                |                           c     |
                |                           i     |
                |                           e     |
                |                           n     |
                |           p               c     |
                |           h               e     |
                |           i   r           -     |
                |       m   l   e   r   s   f     |
                |   h   y   o   l   o   c   i   s |
                |   o   s   s   i   m   i   c   p |
                |   r   t   o   g   a   e   t   o |
                |   r   e   p   i   n   n   i   r |
                |   o   r   h   o   c   c   o   t |
                |   r   y   y   n   e   e   n   s |
----------------+---------------------------------+
         horror |<172> 22   .   1  14   2  22   1 |
        mystery |  27<182>   .   2  18   .  11   . |
     philosophy |   5   .<153> 26   2  36   5   1 |
       religion |  13   2  36<150>   6   9  14   . |
        romance |  18   5   2   2<162>   .  20  19 |
        science |   4   5  21  10   .<175> 15   . |
science-fiction |  36  10   3   3  13   8<166>   1 |
         sports |   8   1   .   1  34   4   5<181>|
----------------+---------------------------------+
(row = reference; col = test)



                 |  Precision  |     Recall  | F1-Measure
-------------------------------------------------------------
         horror  |     0.6078  |     0.7350  |     0.6654
        mystery  |     0.8018  |     0.7583  |     0.7794
     philosophy  |     0.7116  |     0.6711  |     0.6907
       religion  |     0.7692  |     0.6522  |     0.7059
        romance  |     0.6506  |     0.7105  |     0.6792
        science  |     0.7479  |     0.7609  |     0.7543
science-fiction  |     0.6434  |     0.6917  |     0.6667
         sports  |     0.8916  |     0.7735  |     0.8284
```

Figure 2: Metrics for the best NB version on test data set

```
Loaded classifier from cache.
Accuracy: 0.6738197424892703
                |                           s     |
                |                           c     |
                |                           i     |
                |                           e     |
                |                           n     |
                |          p                c     |
                |          h                e     |
                |          i   r            -     |
                |      m   l   e   r   s    f     |
                |  h   y   o   l   o   c    i   s |
                |  o   s   s   i   m   i    c   p |
                |  r   t   o   g   a   e    t   o |
                |  r   e   p   i   n   n    i   r |
                |  o   r   h   o   c   c    o   t |
                |  r   y   y   n   e   e    n   s |
----------------+---------------------------------+
         horror |<164> 19   .   3  18   2  27   1 |
        mystery |  34<171>  .   4  14   .  15   2 |
     philosophy |   4   2<147> 32   3  30   9   1 |
       religion |  13   5  46<132>  2  13  19   . |
        romance |  19   9   1   .<148>  .  26  25 |
        science |   4   6  26  12   .<166> 16   . |
science-fiction |  35  11   6   3  11   8<165>  1 |
         sports |  14   1   1   2  41   6   6<163>|
----------------+---------------------------------+
(row = reference; col = test)



                 |  Precision  |     Recall  | F1-Measure
-------------------------------------------------------------
         horror  |     0.5714  |     0.7009  |     0.6296
        mystery  |     0.7634  |     0.7125  |     0.7371
     philosophy  |     0.6476  |     0.6447  |     0.6462
       religion  |     0.7021  |     0.5739  |     0.6316
        romance  |     0.6245  |     0.6491  |     0.6366
        science  |     0.7378  |     0.7217  |     0.7297
science-fiction  |     0.5830  |     0.6875  |     0.6310
         sports  |     0.8446  |     0.6966  |     0.7635
```