CmpE 493
Spring 2020
Assignment 2: Movie Review Sentiment Classification

Gurkan Demir
2015400177

May 2, 2020

Contents

1 Introduction **3**

2 Implementation **3**

3 Assumptions **4**

4 How to Run? **4**
4.1 Notes 4

5 Screenshots of Running System **5**

6 Results & Analysis **7**
6.1 Performance Values 7
6.2 Randomization Test 8

7 Improvements **8**

1 Introduction

In this assignment, we are expected to implement three Naive Bayes(NB) classifiers, Multinomial NB, Bernoulli NB, and Binary NB, for identifying the sentiment(positive/negative) of a given movie review. We are given a dataset which has following attributes:

1. The positive reviews are in the pos folder and the negative reviews are in the neg folder.
2. The training set contains 700 positive and 700 negative movie reviews.
3. The test set contains 300 positive and 300 negative movie reviews.

Before creating a dictionary, we are expected to tokenize the file and perform case-folding.

Also, we are expected to use Laplace smooting with alpha = 1 while classifying movie reviews.

Our program is expected to take a path of dataset as input, and reports macro-averaged, micro-averaged precision,recall,and F-measure values obtained by classifiers on the test set, as well as the performance values obtained for each class separately. Also, it reports significance of the differences between the micro-averaged F-scores of the algorithms by performing randomization tests.

2 Implementation

1. Read train set.
   (a) Construct model for Multinomial NB.
   (b) Construct model for Bernoulli NB.
   (c) Construct model for Binary NB.
2. Read test set.
   (a) Classify review using Multinomial NB.
   (b) Classify review using Bernoulli NB.
   (c) Classify review using Binary NB.
3. Report the macro-averaged and micro-averaged precision,recall,and F-measure values obtained by each classifiers on the test set, as well as the performance values obtained for each class separately.
4. Perform randomization tests to measure the significance of the differences between the micro-averaged F-scores of the algorithms.

After performing case-folding while reading movie reviews in dataset, tokenization is done. Tokenization is performed by removing punctuation, digits, and extra white spaces.

3 Assumptions

1. All reviews are classified correctly in dataset.
2. While implementing randomization test, R is chosen as **1000**.
3. While implementing randomization test, threshold for p-value is chosen as **0.005**.

4 How to Run?

In order to run movie Review Sentiment Classification, execute the following from the command line:

python3 classifier.py –dataset [DATASET]
where;

1. **DATASET :** Path of dataset (***required***).

4.1 Notes

1. Dataset must contain two folders named **train**, and **test**.
2. Both **train** and **test** folders must contain two folders named **pos**, and **neg**.
3. **pos** folder must contain many positive movie reviews in *txt* format.
4. **neg** folder must contain many negative movie reviews in *txt* format.

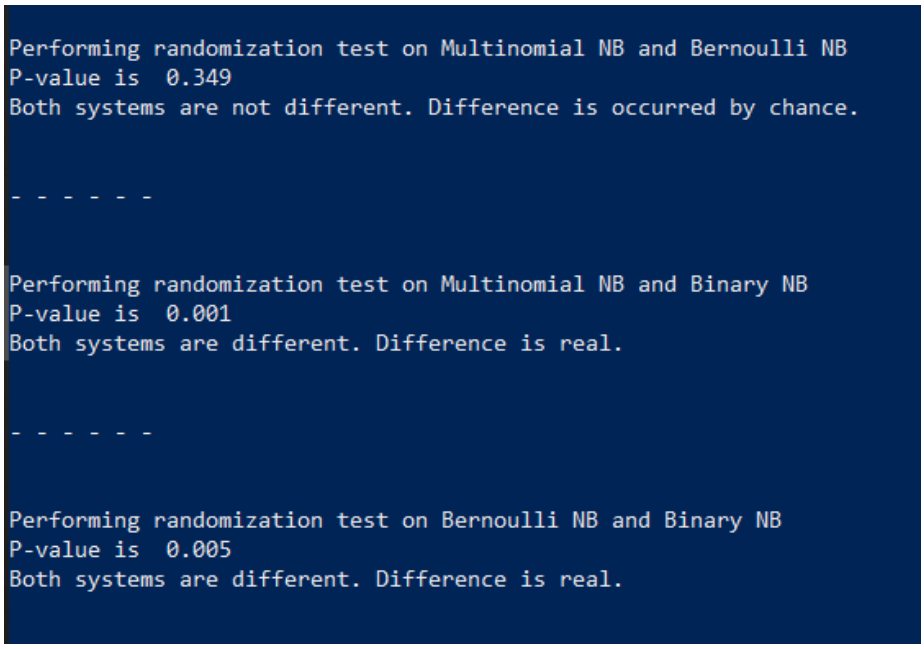
Details of starting execution is mentioned in ReadME file.

5 Screenshots of Running System

- Performance Values

```
Statistics for Multinomial Naive Bayes Classifier
                    precision         recall        f-score
       positive         0.837          0.787          0.811
       negative         0.799          0.847          0.822
 micro-averaged         0.817          0.817          0.817
 macro-averaged         0.818          0.817          0.817


- - - - - -


Statistics for Bernoulli Naive Bayes Classifier
                    precision         recall        f-score
       positive         0.895          0.683          0.775
       negative         0.744           0.92          0.823
 micro-averaged         0.802          0.802          0.802
 macro-averaged          0.82          0.802          0.799


- - - - - -


Statistics for Binary Naive Bayes Classifier
                    precision         recall        f-score
       positive         0.934          0.567          0.706
       negative         0.689           0.96          0.802
 micro-averaged         0.763          0.763          0.763
 macro-averaged         0.811          0.763          0.754
```

- Randomization Test



6 Results & Analysis

6.1 Performance Values

- Multinomial NB

	precision	**recall**	**f-score**
positive	0.837	0.787	0.811
negative	0.799	0.847	0.822
micro-averaged	0.817	0.817	0.817
macro-averaged	0.818	0.817	0.817

Table 1: Performance Values for Multinomial NB

- Bernoulli NB

	precision	**recall**	**f-score**
positive	0.895	0.683	0.775
negative	0.744	0.92	0.823
micro-averaged	0.802	0.802	0.802
macro-averaged	0.82	0.802	0.799

Table 2: Performance Values for Bernoulli NB

- Binary NB

	precision	**recall**	**f-score**
positive	0.934	0.567	0.706
negative	0.689	0.96	0.802
micro-averaged	0.763	0.763	0.763
macro-averaged	0.811	0.763	0.754

Table 3: Performance Values for Binary NB

6.2 Randomization Test

- Multinomial NB - Bernoulli NB

p-value	0.349

Table 4: P-Value between Multinomial NB and Bernoulli NB

Both systems are not different. Difference is occurred by chance.

- Multinomial NB - Binary NB

p-value	0.001

Table 5: P-Value between Multinomial NB and Binary NB

Both systems are different. Difference is real.

- Bernoulli NB - Binary NB

p-value	0.005

Table 6: P-Value between Bernoulli NB and Binary NB

Both systems are different. Difference is real.

7 Improvements

There are various ways to tokenize a file. My system implements tokenization by only removing punctuation, digits and extra white spaces. In this case, tokenization is not done perfectly. We need to care about the words stop words too.

Also, context plays a significant role in spelling error correction. If there exists a way to understand what text means, we can produce more related outputs.