CSE 6363 001 Machine Learning

Fall 2022

Project 2 Report

Instructor's Name: Dr. Dajiang Zhu

Student Name: Siva Srinivasa Sameer Miriyala

Student ID: 1002024871

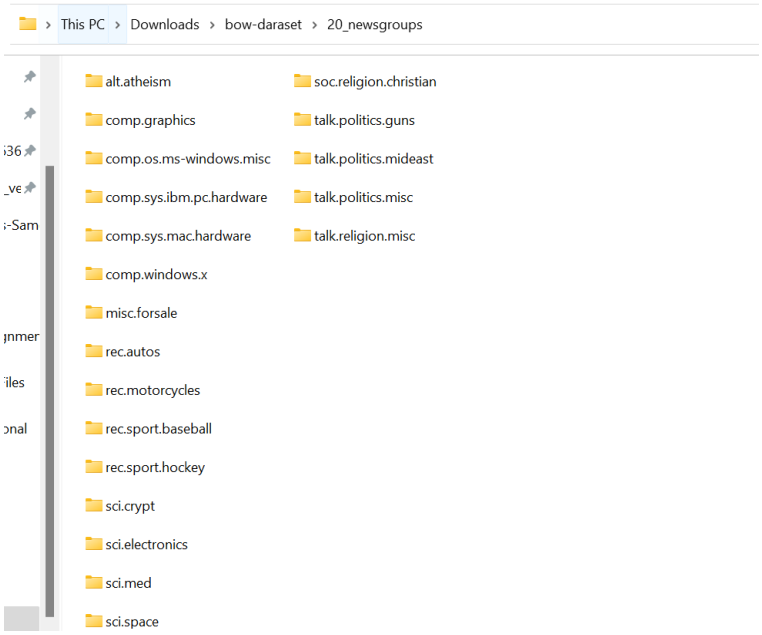
Given Problem Statement:

There is a dataset available with 20,000 newsgroup messages from the 20 newsgroups. The collection contains 1000 documents for each of the 20 newsgroups. We must test the accuracy of a naive bayes classifier on this data. Additionally, split the data into test and training sets so that cross validation may be used to assess the model's performance on a given set of data.

Dataset:

This data has been extracted from (http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html)

There are 20 folders containing 1000 documents each in this data set. The names of the newsgroups are implied by the folder name.



Implementation:

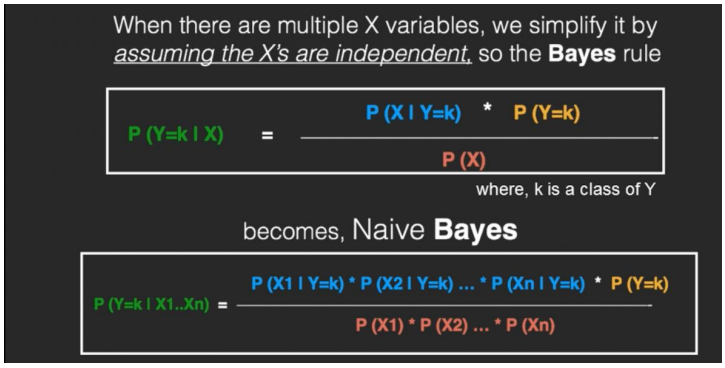
The classification of the text may be determined by looking at the frequency of the terms in it. The document is pre-processed to determine the number of words in each class, and the probability of each word in the class is then determined. For every class, we additionally compute the prior value. We next determine the anticipated class values using the Bayes theorem.

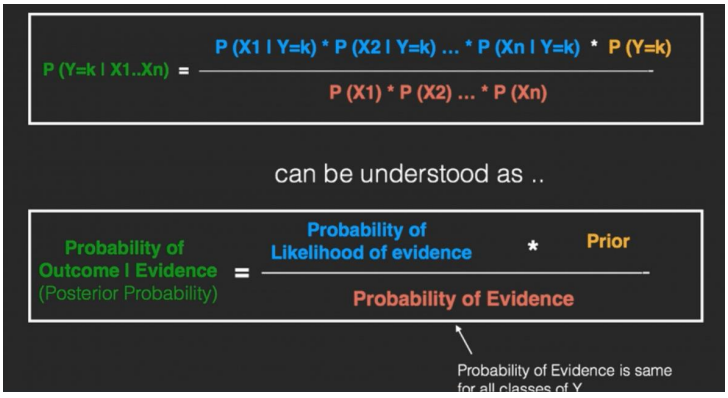
In order to obtain the desirable solution we use k-folds by creating the certain number of bin in cross validation. The beta mean usually generated by each of the beta value to gradually minimize overfitting.

The Naïve Bayes Classifier:

The following probability theorem from which Naïve Bayes is deduced.

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$





The final Naïve Bayes Formula is as follows:

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

Approach:

DATA PRE-PROCESSING:

1. **Stop words assortment**: Stop words are obtained and imported from the Python-compatible package nltk.corpus. A list of additional stop words that were compiled from web sources and are unimportant in this context is annexed to the list of stop words above. Additional appendices to the stop words list include a list of numerals and punctuation.
2. **Stop Word removal:** The Stop Words are further eliminated from the text in the files. So, at this point, the data pre-processing is finished.
3. **The feature List:** The collection of key words used to forecast the news group classes, together with their rates of occurrence, are maintained in the dictionary data structure in Python, which is employed in the construction of the feature list. After the stop words from the provided data are eliminated, 1270015 words remain. Out of which 114500 are unique terms in total. As a result, the feature list in this project uses a maximum of 2500 terms with the greatest frequency of occurrences.

DOC VECTOR: Only the necessary words remain after the document has been pre-processed. The key is the term, and the value is the frequency of occurrence, and we use this function to count each word's occurrences before adding them to a dictionary. The resulting vector will be used to create the probability.

PRIOR: The likelihood of each class over the whole dataset is predicted by this function. It creates a list for each class, which is then used to determine the final probabilities according to the Bayes theorem.

PROBABILITY: Finding the probabilities using Baye's theorem to predict to which class the each document belongs to.

Implementation:

1. To construct a vector containing the frequency of each word in each document, we iterate through all of them. We compile dictionaries for each class simultaneously, keeping track of how frequently each term is used in that class.
2. After everything is done, we determine the previous values and the likelihood that each word will appear in that class. The testing step follows, after which we begin making the vectors for the remaining documents.
3. After obtaining the frequencies of all the documents, we must next get the probabilities for each word from the class vector list and determine the likelihood that the document belongs to that class by adding up all the words that are included in the test document vector.
4. Once the likelihood for a given document has been determined, it is compounded by the preceding value to provide the final probability that the document belongs to the class.

5. Each class is subjected to the same procedure, with the maximum probability serving as the projected class. To get accuracy scores, then, for each document in the testing portion, repeat the process again.

Output:

Illustrating all the classification report of the following.

```
In [11]: class_predicted=model.predict_batch()

100%|██████████████████████████████████████████████████████████████| 7999/7999 [17:37<00:00,  7.56it/s]

In [12]: print(confusion_matrix(class_predicted,model.y_test))
         print(classification_report(class_predicted,model.y_test))

[[150   7   3   1   5   8   1   3   6   8   5   1   5  10   4   3   0  12
    6  49]
 [  0 167  34   7   0  54   0   1   0   0   0   3   6   4   8   0   0   0
    1   0]
 [  0   0  40   0   0   1   1   0   0   0   0   0   0   0   0   0   0   0
    0   0]
 [  0   5  33 239  32   5  18   2   0   0   1   2   3   0   1   0   0   1
    1   0]
 [  2  15  55  15 327  20   7   2   0   0   0   5   2   0   2   0   0   0
    3   3]
 [  0   4   2   3   0  78   0   0   0   0   0   1   1   0   0   0   0   0
    0   0]
 [ 24  77  93  61  17  82 336  24  24  45  72  21  20  38  24  39  35  36
   45  34]
 [  8   5   4   6   1   1   8 294   5   2   5   2   5   5   8   0  10   8
    9   7]
 [ 57  12  14   5   2   9   3  12 330   8  58  32   3   9  15   5  39  52
   48  35]
 [  1   0   1   2   0   0   2   0   0 297 154   0   0   0   2   0   0   0
    2   1]
 [  0   4   1   0   1   1   6   0   0  13  43   1   4   1   2   0   0   0
    0   0]
 [  3   1   7   1   0   8   1   1   1   1   0 227   2   0   1   1   2   2
    2   0]
 [ 34  54  81  43  10 100  12  15   4  13  31  42 312  19  27  20  38  29
   34  34]
 [ 24  14  17   4   2  23   5   3   4   4  11  20   4 271  20   5  19  48
   61  35]
 [  5   3   7   1   1   6   0   3   1   0   1   4   1   2 248   0   2   4
    8   5]
 [ 24   1   0   0   0   0   0   0   0   0   0   0   0   3   2 335   0   1
    1  27]
 [  6   3   7   4   2   0   1   9   3   2   5  13   9   4   9   1 238  10
   62  30]
 [ 16   4   4   1   1  16   1   2   4   6   5   1   7   5   4   2   5 171
   12  14]
 [ 19   1   2   2   1   1   0   6   4   3   9  15   2  15  11   0  19  13
  105  26]
 [ 53   1   0   2   0   1   1   0   2   1   4   0   1   5   3   4   8   8
   15 103]]
```

```
 15 103]]
                         precision    recall  f1-score   support

              alt.atheism       0.35      0.52      0.42       287
            comp.graphics       0.44      0.59      0.50       285
  comp.os.ms-windows.misc       0.10      0.95      0.18        42
 comp.sys.ibm.pc.hardware       0.60      0.70      0.65       343
    comp.sys.mac.hardware       0.81      0.71      0.76       458
           comp.windows.x       0.19      0.88      0.31        89
             misc.forsale       0.83      0.29      0.43      1147
                rec.autos       0.78      0.75      0.76       393
          rec.motorcycles       0.85      0.44      0.58       748
       rec.sport.baseball       0.74      0.64      0.69       462
         rec.sport.hockey       0.11      0.56      0.18        77
                sci.crypt       0.58      0.87      0.70       261
          sci.electronics       0.81      0.33      0.47       952
                  sci.med       0.69      0.46      0.55       594
                sci.space       0.63      0.82      0.72       302
   soc.religion.christian       0.81      0.85      0.83       394
       talk.politics.guns       0.57      0.57      0.57       418
    talk.politics.mideast       0.43      0.61      0.51       281
       talk.politics.misc       0.25      0.41      0.31       254
       talk.religion.misc       0.26      0.49      0.33       212

                accuracy                           0.54      7999
               macro avg       0.54      0.62      0.52      7999
            weighted avg       0.67      0.54      0.56      7999
```

Drawbacks:

1. It is taking significant longer time to load, calculate and pre-process the data.
2. Some of them can become not so good predictors.
3. We added additional stop codes for better predictability.

References:

Bayes theorem formula image reference:

https://www.machinelearningplus.com/wp-content/uploads/2018/11/03_bayes_rule_naive_bayes_new.png