Amartejas Manjunath

1001742606
axm2606

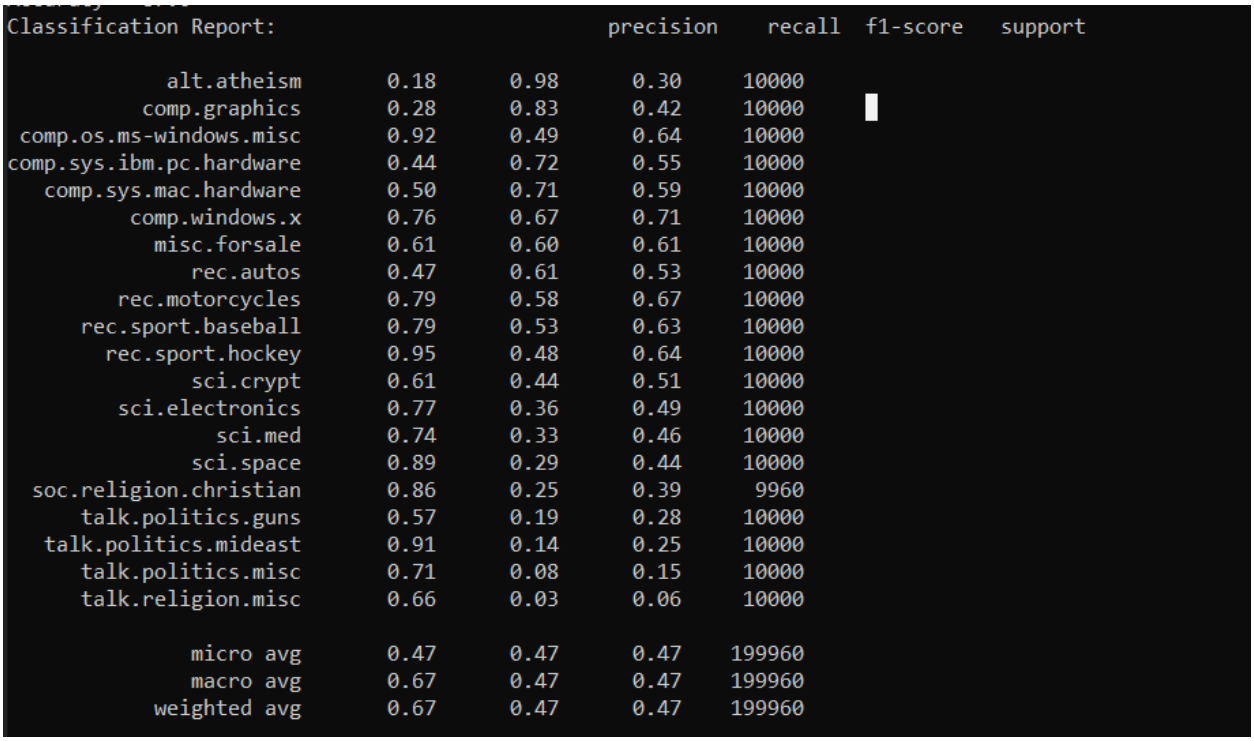
Machine Learning

Project 2

Functions Used:

1. **StopWords():** This function prepares the stopwords that will be used later for cleaning the data. This function opens the "stopwords.txt" from the system and reads it. The words are placed next to each and a list of stopwords are returned.
2. **Clean():** This function cleans the data that is passed to the function. It removes the symbols ['<', '\\','=',',',':'….], converts all the words to lowercase and removes stopwords. It splits the file into different words and creates a list of these words and returns it.
3. **Get_ran_file():** This function generates a random number from 0 to the length of no. of files and a random file is returned for testing. If the length of the folder list is less than 1 then the function returns a NULL value which is used to break the testing loop.
4. **Predict():** This function calls the get_ran_file to generate a random file cleans the file and calls the probability function and keeps a track of all the probability of all the in a list. The max value of this probability list the predicted value.
5. **Probability():** This function applies the Naive Bayes formula to the test file and the counter Vectorized dictionary return the probability for the test file.
6. **Main():** The main function is used to pre-process the dataset. It lists out all the directories in the dataset and also the files in those folders. It performs Counter Vectorization, this a process of counting how many times a particular word is repeated in a particular document.

Classification Report



Classification Report:

	precision	recall	f1-score	support
alt.atheism	0.18	0.98	0.30	10000
comp.graphics	0.28	0.83	0.42	10000
comp.os.ms-windows.misc	0.92	0.49	0.64	10000
comp.sys.ibm.pc.hardware	0.44	0.72	0.55	10000
comp.sys.mac.hardware	0.50	0.71	0.59	10000
comp.windows.x	0.76	0.67	0.71	10000
misc.forsale	0.61	0.60	0.61	10000
rec.autos	0.47	0.61	0.53	10000
rec.motorcycles	0.79	0.58	0.67	10000
rec.sport.baseball	0.79	0.53	0.63	10000
rec.sport.hockey	0.95	0.48	0.64	10000
sci.crypt	0.61	0.44	0.51	10000
sci.electronics	0.77	0.36	0.49	10000
sci.med	0.74	0.33	0.46	10000
sci.space	0.89	0.29	0.44	10000
soc.religion.christian	0.86	0.25	0.39	9960
talk.politics.guns	0.57	0.19	0.28	10000
talk.politics.mideast	0.91	0.14	0.25	10000
talk.politics.misc	0.71	0.08	0.15	10000
talk.religion.misc	0.66	0.03	0.06	10000
micro avg	0.47	0.47	0.47	199960
macro avg	0.67	0.47	0.47	199960
weighted avg	0.67	0.47	0.47	199960

Output:

```
reading files from 20_newsgroups/alt.atheism
reading files from 20_newsgroups/comp.graphics
reading files from 20_newsgroups/comp.os.ms-windows.misc
reading files from 20_newsgroups/comp.sys.ibm.pc.hardware
reading files from 20_newsgroups/comp.sys.mac.hardware
reading files from 20_newsgroups/comp.windows.x
reading files from 20_newsgroups/misc.forsale
reading files from 20_newsgroups/rec.autos
reading files from 20_newsgroups/rec.motorcycles
reading files from 20_newsgroups/rec.sport.baseball
reading files from 20_newsgroups/rec.sport.hockey
reading files from 20_newsgroups/sci.crypt
reading files from 20_newsgroups/sci.electronics
reading files from 20_newsgroups/sci.med
reading files from 20_newsgroups/sci.space
reading files from 20_newsgroups/soc.religion.christian
reading files from 20_newsgroups/talk.politics.guns
reading files from 20_newsgroups/talk.politics.mideast
reading files from 20_newsgroups/talk.politics.misc
reading files from 20_newsgroups/talk.religion.misc
Total words is 250737
10.0% of testing done
20.0% of testing done
30.0% of testing done
40.0% of testing done
50.0% of testing done
60.0% of testing done
70.0% of testing done
80.0% of testing done
90.0% of testing done
Testing done

Accuracy = 87.0
```

About Confusion Matrix:

- A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.
- The predicted classes are represented in the columns of the matrix, whereas the actual classes are in the rows of the matrix. We then have four cases for each newsgroup in the data:

1. True positives (TP): the cases for which the classifier predicted 'correct class' and the actual data was from same class.

2. True negatives (TN): the cases for which the classifier did not predicted class as 'correct' and the actual data was also not from same class.

3. False positives (FP): the cases for which the classifier predicted class as 'correct class' but the actual data had a different correct class.

4. False negatives (FN): the cases for which the classifier did not predicted class as 'correct class' and the actual data was rather a correct class.

```
Accuracy = 87.0
[[9819    5    0   19   10   14    0    0   12    0    0    9    0    0
     6   13    4    0    1   88]
 [ 502 8287   20  353  105  435  118   26   17    0   10    9   68   28
    12    0    4    4    2    0]
 [ 503 1508 4917 1570  622  637  140    0   12    0    0   14   64    7
     0    0    4    0    1    1]
 [ 527 1148  140 7174  676   75  144   13    0    0    0    0   96    7
     0    0    0    0    0    0]
 [ 543 1302   44  839 7071   61  133    0    0    0    0    0    0    7
     0    0    0    0    0    0]
 [ 500 2152   45  326  146 6693   56   26    0   11    0    9   32    0
     0    0    4    0    0    0]
 [ 619 1204   18  967  743   41 5991  116   65   11   12    0  153   31
     6    0   16    3    4    0]
 [1225  957   14  532  538   21  404 6116   91    0   10    0   64   14
     6    0    8    0    0    0]
 [1401  932    5  473  555   51  299  512 5770    0    0    0    0    0
     0    0    0    0    2    0]
 [1732  859    9  504  455   51  446  431  177 5296   10    0    8   14
     0    0    8    0    0    0]
 [1494 1211   35  523  309  108  531  359  144  428 4813    9   32    0
     0    0    2    0    2    0]
 [2070 2026   31  313  527  212   60  323   36   12    0 4362   24    0
     0    0    2    0    2    0]
 [ 690 1895   21 1403  964   73  543  659   60   17   12   39 3595    7
    18    0    4    0    0    0]
 [2857 1511   10  215  460   52  307  528  221  196   17  116  150 3325
    18    0    4    3   10    0]
 [1698 2708   27  299  438  176  276  615   72   97   13  207  264  200
  2899    0    4    0    6    1]
 [5615  450    0   52  115   27  107  394  153  289   16   36   15  192
     9 2486    0    3    0    1]
 [4427  227   20  249   55    3   65 1361  326   79   12 1044    7   94
    64    1 1889    1   67    9]
 [6304  356    0   78  161   16   36  293   56   93   49  251   61  190
    71  167  337 1432   48    1]
 [5232  289    0  195  125    5   64  788  106  134   70  812   32  311
   150   21  683  112  812   59]
 [7994   82    1   65   11    7   34  352   29   44   25  177   16   60
    16  219  356   10  192  310]]
```