CSCI 720 Big Data Analytics
Assignment 5 – Trisha P Malhotra

Multinomial Naïve Bayes classifier on 20 newsgroup dataset.

1]
For 4 categories : [alt.atheism', 'talk.religion.misc','comp.graphics', 'sci.space']:
Results are :
Accuracy : 90.5 %
Alpha: 0.01
Confusion matrix
newsgroups on atheism is often confused with sci.space;
and vice-versa
Best : 379 correct predictions for comp.graphics
Worst : 195 correct predictions for sci.space

```
/Library/Frameworks/Python.framework/Versions/3.6/bin/python3.6 /Users/tpm/PycharmProjects/bda-hw5/hw5.py
Data set loaded successfully
2034 documents - 3.980MB (training set)
1353 documents - 2.867MB (test set)
4 categories

Extracting features from the training data using a sparse vectorizer
Using HashingVectorizer:
n_samples: 2034, n_features: 65536
Extracting features from the test data using the same vectorizer
n_samples: 1353, n_features: 65536
********************************
Naive Bayes
********************************
Training:
MultinomialNB(alpha=0.01, class_prior=None, fit_prior=True)
Training complete
Train time: 0.010s
test time:  0.003s
accuracy:   0.905
confusion matrix:
[[279   2   6  32]
 [  5 371  11   2]
 [  3  12 379   0]
 [ 43   3  10 195]]


Process finished with exit code 0
```

2]
For all categories:

i) For alpha: 0.000001:
accuracy: 0.776

ii) For alpha: 1.0:
accuracy: 0.791

iii) **Alpha seems to give the best accuracy for smoothing alpha : 0.01**

Thus, finally for :
Alpha:
0.01
Accuracy:
83.1 %
Sparsity:
111 non-zero components by sample in a more than 65000-dimensional space (less than 0.17% non-zero features)

```
/Library/Frameworks/Python.framework/Versions/3.6/bin/python3.6 /Users/tpm/PycharmProjects/bda-hw5/hw5.py
Data set loaded successfully
11314 documents - 22.055MB (training set)
7532 documents - 13.801MB (test set)

Extracting features from the training data using a sparse vectorizer
Using HashingVectorizer:
n_samples: 11314, n_features: 65536
Sparsity for training data
114.64786989570443
Extracting features from the test data using the same vectorizer
n_samples: 7532, n_features: 65536
Sparsity for testing data
111.30045140732874
********************************
Naive Bayes
********************************
Training:
MultinomialNB(alpha=0.01, class_prior=None, fit_prior=True)
Training complete
Train time: 0.080s
test time:  0.028s
accuracy:   0.831
```

Categories for confusion matrix:
['alt.atheism', 'comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware', 'comp.windows.x', 'misc.forsale', 'rec.autos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey', 'sci.crypt', 'sci.electronics', 'sci.med', 'sci.space', 'soc.religion.christian', 'talk.politics.guns', 'talk.politics.mideast', 'talk.politics.misc', 'talk.religion.misc']

Deductions:

1.) 'talk.politics.misc' is often confused with 'talk.politics.guns'

2.) 'comp.windows.x' is also mostly confused to be 'comp.graphics'

3.) Value: 389 – best prediction for Category : "**talk.religion.misc**"

4.) Value 152 – worst prediction value for 'rec.sport.hockey'

```
confusion matrix:
[[250   1   0   3   0   1   0   1   2   1   1   1   0   4   2  24   5   4
    1  18]
 [  1 284  23  12   7  24   5   0   0   2   3   5  13   1   6   2   0   1
    0   0]
 [  1  28 268  48   3  16   3   0   1   4   0   6   4   0   3   2   1   0
    4   2]
 [  0  13  26 292  23   2   9   1   0   1   0   0  22   0   3   0   0   0
    0   0]
 [  0   7  14  25 307   1  11   4   1   3   0   1   8   1   2   0   0   0
    0   0]
 [  0  49  24  10   4 298   2   1   1   0   0   0   0   2   4   0   0   0
    0   0]
 [  0   4   7  23   5   0 319  11   4   1   3   0   7   4   1   0   1   0
    0   0]
 [  0   2   1   4   0   0  13 357   7   1   1   0   4   2   1   0   2   0
    1   0]
 [  0   0   0   1   0   0   5   6 381   1   0   0   3   0   0   0   1   0
    0   0]
 [  0   0   0   0   1   0   4   4   0 368  13   0   0   0   4   0   2   0
    1   0]
 [  0   0   0   0   0   2   0   0   0   3 389   1   0   1   1   1   0   0
    1   0]
 [  1   5   2   1   2   3   4   3   0   0   0 368   2   1   0   0   3   0
    1   0]
 [  1  10  11  28   6   2  10   5   7   0   0   9 292   7   3   0   0   1
    1   0]
 [  2  12   2   2   2   2   9   8   2   3   0   2   9 318   4   4   1   2
    9   3]
 [  1   5   2   2   1   4   0   0   0   3   0   0   2   4 363   1   3   0
    3   0]
 [  2   1   1   1   0   0   0   0   1   1   1   0   0   3   1 378   1   0
    1   6]
 [  0   0   0   1   2   0   1   1   0   1   0   4   0   1   1   0 333   1
   10   8]
 [  3   1   0   0   0   1   0   0   1   0   1   0   0   0   0   3   2 352
   11   1]
 [  4   4   0   0   1   0   0   2   2   0   1   4   0   2   8   1  76   3
  192  10]
 [ 32   1   3   0   0   0   2   1   0   1   0   1   0   3   4  28  14   2
    7 152]]
```