Text Classification

Stuti Chaturvedi
202161006

Haripriya Goswami
202161003

3rd Yashita Vajpayee
202162012

Abstract—**In this experiment we aim to perform text classification on a dataset of 20 news-groups through three different classifiers and compare the results from all the three on the basis of their F-score.**

I. Introduction

Classification means putting similar objects in the same class. Text classification works in the similar sense, we use features and combination of those features to represent the class, the text is then labeled to the class it will belong to. We can do this classification manually, through unsupervised learning and through a supervised machine learning approach. Text classification find it's application in many domains like email sorting, sentiment detection, topic-specific(vertical) search and many more. Classification has become one of the primary requirements information retrieval and natural language processing in today's times.

II. Theory: Types of Classifications

A. *Naive-Bayes Classification*

Naive Bayes classification is a supervised classification model which works on the Bayes theorem. We can find the probability of a document to belong to a class by:

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

here P(c) is the prior probability and evidence is P(tk/c) where tk(t1,t2,t3...) are the tokens from the document.

B. *K-Nearest Neighbour Classification*

The concept behind k-Nearest Neighbour or KNN is that objects belonging to same class or similar class are distributed around each other in the vector space. For a given document we take it's k nearest neighbours from the training set, and check which training class will it belong to. We calculate a score through cosine similarity and assign the given document to the class with highest score. The score is calculated as below:

$$\text{score}(c, d) = \sum_{d' \in S_k} I_c(d') \cos(\vec{v}(d'), \vec{v}(d))$$

C. *Rocchio Classification*

Rocchio classification use vector approach by creating boundaries between classes. These boundaries are known as decision boundaries. In this classification, centroid is calculated using data points and point having equal distance from this centroid forms boundary. Formula to calculate centroid is given by:

$$\vec{\mu}(c) = \frac{1}{|D_c|} \sum_{d \in D_c} \vec{v}(d),$$

III. Approach

A. *General*

We performed the experiment on 20-newsgroup dataset. Python is used to implement this experiment.Firstly we fetched the 20-newsgroup data which contained 20 unique classes, 11314 training samples and 7532 test samples.We then used CountVectorizer() for training and test sets to tokenize the set and get the count of each object. TfidfTransformer() is used to transform both the sets to tf-idf matrix. *Pipeline* from *sklearn.pipeline* can also be used to add vectorizer, transformer and classifier, all in one compound classifier. All the classifier are checked based on following metrics: Precision - fraction of retrieved documents that are relevant

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant}|\text{retrieved}).$$

Recall - fraction of relevant documents that are retrieved

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = P(\text{retrieved}|\text{relevant}).$$

F1 Score - Measured through of Precision and recall as:

$$F = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

B. *Naive-Bayes Classifier*

We used *MultinomialNB* from *sklearn.naive bayes* to fit the model to the training set and then predicted the test set against it. We used *Metrics* from *sklearn* to calculate Accuracy, Precision, Recall and F-score, and also constructed a confusion matrix.

C. *KNN Classifier*

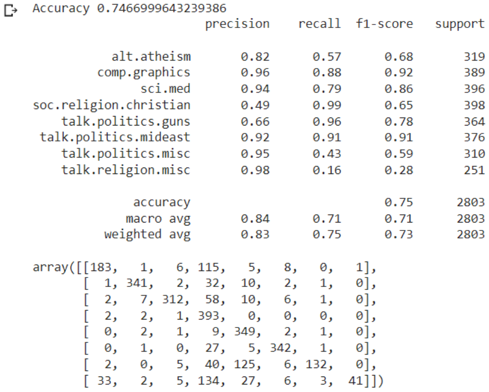
We used *KNeighborsClassifier* from *sklearn.neighbors* (k=7 used) to fit the model to the training set and then predicted the test set against it. We used *Metrics* from *sklearn* to calculate Accuracy, Precision, Recall and F-score, and also constructed a confusion matrix.

D. Roccchio Classifier

We used *NearestCentroid* from *sklearn.neighbors* for the training set and then predicted the test set against it. We used *Metrics* from *sklearn* to calculate Accuracy, Precision, Recall and F-score, and also constructed a confusion matrix.

IV. Results

A. Naive-Bayes Classifier



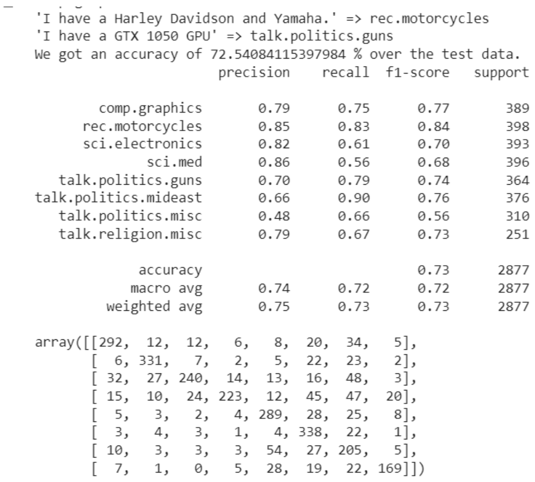
```
Accuracy 0.7466999643239386
                          precision    recall  f1-score   support

            alt.atheism       0.82      0.57      0.68       319
          comp.graphics       0.96      0.88      0.92       389
                sci.med       0.94      0.79      0.86       396
soc.religion.christian       0.49      0.99      0.65       398
     talk.politics.guns       0.66      0.96      0.78       364
  talk.politics.mideast       0.92      0.91      0.91       376
     talk.politics.misc       0.95      0.43      0.59       310
     talk.religion.misc       0.98      0.16      0.28       251

               accuracy                           0.75      2803
              macro avg       0.84      0.71      0.71      2803
           weighted avg       0.83      0.75      0.73      2803

array([[183,   1,   6, 115,   5,   8,   0,   1],
       [  1, 341,   2,  32,  10,   2,   1,   0],
       [  2,   7, 312,  58,  10,   6,   1,   0],
       [  2,   2,   1, 393,   0,   0,   0,   0],
       [  0,   2,   1,   9, 349,   2,   1,   0],
       [  0,   1,   0,  27,   5, 342,   1,   0],
       [  2,   0,   5,  40, 125,   6, 132,   0],
       [ 33,   2,   5, 134,  27,   6,   3,  41]])
```

B. KNN Classifier



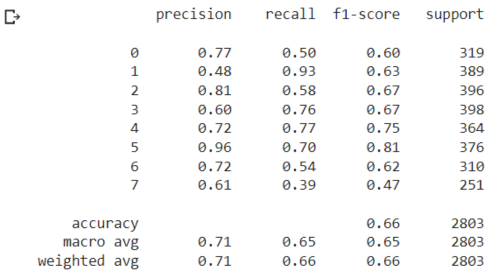
```
'I have a Harley Davidson and Yamaha.' => rec.motorcycles
'I have a GTX 1050 GPU' => talk.politics.guns
We got an accuracy of 72.54084115397984 % over the test data.
                       precision    recall  f1-score   support

        comp.graphics       0.79      0.75      0.77       389
      rec.motorcycles       0.85      0.83      0.84       398
      sci.electronics       0.82      0.61      0.70       393
              sci.med       0.86      0.56      0.68       396
   talk.politics.guns       0.70      0.79      0.74       364
talk.politics.mideast       0.66      0.90      0.76       376
   talk.politics.misc       0.48      0.66      0.56       310
   talk.religion.misc       0.79      0.67      0.73       251

             accuracy                           0.73      2877
            macro avg       0.74      0.72      0.72      2877
         weighted avg       0.75      0.73      0.73      2877

array([[292,  12,  12,   6,   8,  20,  34,   5],
       [  6, 331,   7,   2,   5,  22,  23,   2],
       [ 32,  27, 240,  14,  13,  16,  48,   3],
       [ 15,  10,  24, 223,  12,  45,  47,  20],
       [  5,   3,   2,   4, 289,  28,  25,   8],
       [  3,   4,   3,   1,   4, 338,  22,   1],
       [ 10,   3,   3,   3,  54,  27, 205,   5],
       [  7,   1,   0,   5,  28,  19,  22, 169]])
```

C. Roccchio Classifier



```
              precision    recall  f1-score   support

           0       0.77      0.50      0.60       319
           1       0.48      0.93      0.63       389
           2       0.81      0.58      0.67       396
           3       0.60      0.76      0.67       398
           4       0.72      0.77      0.75       364
           5       0.96      0.70      0.81       376
           6       0.72      0.54      0.62       310
           7       0.61      0.39      0.47       251

    accuracy                           0.66      2803
   macro avg       0.71      0.65      0.65      2803
weighted avg       0.71      0.66      0.66      2803
```

V. Conclusion

Through this experiment, we compared the three classifiers based on the different parameters. According to accuracy and average values F-score we conclude that Naive-Bayes classifier has the highest values of the among the three classifiers.

References

[1] Introduction to information retrieval by Christopher D Manning Prabhakar Raghavan Hinrich Schutze

[2] https://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html

[3] https://nlp.stanford.edu/IR-book/html/htmledition/k-nearest-neighbor-1.html

[4] https://nlp.stanford.edu/IR-book/html/htmledition/rocchio-classification-1.html