Maral Dicle Maral
03.01.2022

CMPE 493 INFORMATION RETRIEVAL
ASSIGNMENT 2: A Simple Search System for Phrase and Free Text Queries

In this assignment Multinomial Naive Bayes (NB) and k Nearest Neighbor (kNN) algorithms for text classification is implemented. Reuters data was utilized for training and test sets.

Classification with kNN and NB:

```
python3 classify.py ./reuters21578/ ./stopwords.txt
```

Assignment consists of 3 main steps:

1- Data Preprocessing
2- Test and Training Set Creation
3- Naïve Bayes Classification
4- kNN Classification
5- Evaluation
6- Statistical significance

1- Data Preprocessing

Reuters data was read via an HTML file processing library, beautifulSoup. As the documents are read, they are trimmed from punctuations according to the string.punctuation list and cleaned from stop words specified by the user. Latin-1 encoding was used for the documents. Multiprocessing which creates a process for each document was used to speed up the data reading at data preprocessing step. Final vocabulary consists of 27045 words.

2- Top 10 Classes, Test and Training Set

Number of documents in test set is 2326 while it is 5999 for training set. The top ten classes in the documents are: 'earn', 'acq', 'money', 'fx', 'crude', 'trade', 'interest', 'supply', 'ship', 'fxinterest'. For **test set**, there are acq: 696, earn: 108, crude: 121, fx: 112, money: 182, trade: 75, interest: 81, ship: 36, grainwheat: 32, supply: 30 documents. For **training set,** there are acq:1596,

Maral Dicle Maral

03.01.2022

earn:2840, interest:190, supply:130, money:608, trade:251, fx:260, crude:253, grainwheat:102, ship:108 documents. Number of double labeled documents in the test set is 143, while it is 444 for training set.

3- Naïve Bayes Classification

For naïve bayes, I used word counts as features for the classification. I used P_cj and P_wk_cj values for classification as described in the class. I applied add-one smoothing.

4- K Nearest Neighbors Classification

For kNN classification, I first applied feature selection based on term frequencies of the whole words in the training set, I utilized 1 percent of the features due to the efficiency issues on my computer. Normally I should have left out only the words with counts less than 3.

After feature selection, I created the tf-idf vectors of both training and test set according to the selected features.

Later on, I created a development set from the training set in order to find the best performing k value (in terms of F measure). I run the kNN and accuracy_measure algorithm for k values ranging from 7 to 30 and the best performing k for my development set is 10.

Finally, I merged the development set with the rest of the training set and run the kNN algorithm for the all the training set and tried to identify the test set classes.

kNN algorithm finds the k closest neighbors to the given test point and evaluates its class accordingly. I applied cosine similarity on the tf-idf vectors to find the most k similar neighbors to the test documents. Because we already implemented cosine similarity algorithm in the second assignment, and because of the efficiency issues, I utilized numpy library for cosine similarity calculation.

5- Evaluation

Below precision and recall values for naïve bayes and kNN classification suggests the naïve bayes has overall better performance. The macro F1 score for naïve bayes is 0.755 while it is 0.6854 for kNN.

Maral Dicle Maral

03.01.2022

Accuracy Scores for Naive Bayes are:

Macro Precision: 0.8969 Macro Recall: 0.6519 Macro F1 Score: 0.755

Micro Precision: 0.8889 Micro Recall: 0.9327

Accuracy Scores for kNN are:

Macro Precision: 0.8382 Macro Recall: 0.5797 Macro F1 Score: 0.6854

Micro Precision: 0.8583 Micro Recall: 0.9006

P Value Evaluation:

In order to find the significance of this accuracy difference between two algorithms, I utilized the randomization test. The null hypothesis was used classification algorithms (naive bayes and kNN) are not different. Therefore, I swapped the predicted results (y_pred) of the two algorithms with 0.5 possibility and find the macro F1 accuracy measure for both swapped y_pred lists for 10 times and found the p value as 0.09090, which is scientifically significant. Therefore, I concluded that Naive Bayes algorithm performs better compared to kNN. **(Figure 1)**

```
Last login: Sun Jan  2 23:52:02 on ttys000
maral@192 Assignment3_Classification_NB_KNN % python3 classify.py ./reuters21578/ ./stopwords.txt
#### ACCURACY SCORES FOR NAIVE BAYES ####
Macro Precision:  0.8969    Macro Recall:  0.6519   Macro F1 Score:  0.755
Micro Precision:  0.8889    Micro Recall:  0.9327

#### ACCURACY SCORES FOR KNN ####
Macro Precision:  0.8382    Macro Recall:  0.5797   Macro F1 Score:  0.6854
Micro Precision:  0.8583    Micro Recall:  0.9006

P value 0.09090909090909091
- Classification process ends in 43.644900 seconds -
maral@192 Assignment3_Classification_NB_KNN %
```

Figure 1 (Accuracy Measures for Naïve Bayes and kNN, p Value for the randomization test)