Assignment 5

Submitted by

Swati Verma

MT19073

Question 1

Naive Bayes

Assumptions:

1. No metadata has been removed from any of the documents
2. User will input proper values.

Preprocessing steps:

1. Words in all the documents are converted to lowercase.
2. Punctuations have been removed from all the documents.
3. Numbers are converted into words.
4. Expansion such as don't to do not has not been handled because the library was not working on my system.
5. Stopwords have been removed from all the documents using nltk library.
6. To convert surface words to root word lemmatization is used instead of stemming because lemmatization gives meaningful words.
7. All the above steps are done both for training and testing documents..

Methodology:

1. Initially documents from all the classes are stored in a list called docs.
2. User is asked to enter the percentage of documents he/she wants to be in training and testing set.
3. Each file is read and all the pre processings steps are performed on both training and testing documents.
4. After that, for training data following values are stored which will help to calculate mutual information and tf-idf scores.

a. A dictionary is created where the class name acts as key and all the tokens corresponding to that particular document as values.
b. A dictionary is created where key is document name and value is a dictionary which contains term and term frequency pair for each term corresponding to each document. For calculating tf **logBase2(1+term frequency)** has been used.
c. A training vocab is created which contains all the tokens of training dataset
d. A unique training vocab is created which contains all the unique tokens of the training dataset.
e. A dictionary is created which contains class name as key and a dictionary of term and in how many documents that term occurs in a particular class as value.
f. A dictionary which contains each term of training set as key and the number of classes in which that term occurs as value. This is df dictionary which will help in calculating tf-idf values for feature selection.

5. After storing all the data mutual information and tf-idf scores are calculated for each term for each class. Mutual information is calculated using formula:

$$I(U;C) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} P(U=e_t, C=e_c) \log_2 \frac{P(U=e_t, C=e_c)}{P(U=e_t)P(C=e_c)}$$

$$I(U;C) = \frac{N_{11}}{N} \log_2 \frac{NN_{11}}{N_{1.}N_{.1}} + \frac{N_{01}}{N} \log_2 \frac{NN_{01}}{N_{0.}N_{.1}} + \frac{N_{10}}{N} \log_2 \frac{NN_{10}}{N_{1.}N_{.0}} + \frac{N_{00}}{N} \log_2 \frac{NN_{00}}{N_{0.}N_{.0}}$$

N11= #docs where term is present and document belongs to class c.

N01= #docs where term is not present but class is c

N10= #docs where term is present but class is not c.

N00= #docs where neither term is present nor class is c

6. Again users are asked to enter the percentage of features they want for top k features in each class.

7. Now top k% features are selected and stored for each class using mutual information and tf-idf feature selection techniques.
8. After that during **training phase of naive bayes,** for each class, for each term conditional probability is stored using the formula

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V}(T_{ct'} + 1)} = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B}$$

Where Tct is the frequency of term t present in that class, summation of Tct is total number of terms present in that particular class c and B is the total unique tokens in training dataset.

9. Prior probability for each class is calculated using the formula;

$$\hat{P}(c) = \frac{N_c}{N}$$ where Nc is the number of documents in class c and N is the total number of documents in the training set.

10. Now in the testing **phase,** one document is taken and each term is seen. If that term occurs in top k features(once using MI and next time using tf-idf score) then its conditional probability is taken for that term else probability is calculated using smoothing i.e. Tct =0.
11. For each document in the testing set, probability is calculated for each class using the formula:

$$c_{\text{map}} = \arg\max_{c \in \mathbb{C}} [\log \hat{P}(c) + \sum_{1 \leq k \leq n_d} \log \hat{P}(t_k|c)]$$

12. The class with highest probability is assigned to that particular document.
13. After that accuracy is taken for different split ratio of training and testing documents and a graph is plotted for different split ratios and accuracy using MI feature selection and tf-idf feature selection.
14. One more graph is plotted for different values of top k features and accuracy score using MI and tf-idf as feature selection.

Question 2

Assumptions:

1. No metadata has been removed from any of the documents.
2. User will input proper values.

Preprocessing steps:

1. Words in all the documents are converted to lowercase.
2. Punctuations have been removed from all the documents.
3. Numbers are converted into words.
4. Expansion such as don't to do not has not been handled because the library was not working on my system.
5. Stopwords have been removed from all the documents using nltk library.
6. To convert surface words to root word lemmatization is used instead of stemming because lemmatization gives meaningful words.
7. All the above steps are done both for training and testing documents..

Methodology:

1. All the steps till storing top k terms using mutual information and tf-idf technique as feature selection is the same as the question1.
2. Also for the testing and training set two dictionaries are created. Each dictionary has a document name as key and a list of lemmatized tokens as values.
3. In KNN, cosine similarity is calculated between each test document and all the training documents.
4. While calculating cosine similarity it is seen for testing documents, if terms in that testing document occurs in top k features then only its tf-idf value is calculated which is used to calculate cosine similarity.
5. After that the user is asked to enter the value of K, where K is K nearest neighbour.
6. Top K documents are then retrieved based on the cosine score.

7. In top K documents, scores of the documents which belong to the same class are summed up and then the class with the highest score is assigned to the test document.
8. Graph is plotted for different values of K, using both mutual information and tf_idf as feature selection technique.