IR Assignment-4

- SUBHANI SHAIK (MT18117)

Dataset: 20newsgroups dataset

In this dataset, used **comp.graphics, sci.med, talk.politics.misc, rec.sport.hockey, sci.space [5 classes]** which have 1000 documents each i.e., total 5000 documents.

1. Naive Bayes Classification for different splits:

Pre-Processing the dataset:

- Initially read all the class document names from each folder and stored to a list.
- Read those files and done the pre-processing for both train and test data.
- For pre-processing the text, used NLTK library and done following steps:
    - Conversion to lowercase
    - Contractions
    - Removed unnecessary characters and punctuations and tokenized using RegexpTokenizer.
    - Lemmatization
    - Stop words removal.

Procedure:

- Initially split the dataset according to our need as training and test data.
- Prepare the 'Dictionary' for this training data as dictionary structure as-
  {class_num:[tokenized data of all docs in that class]}
- Prepare 'Vocabulary' for that split by merging all classes tokenized data and take set of them.
- Next, train the data by calculating 'class prior'(class probability) and 'word conditional probability'(these words are from vocabulary which is prepared from above) and stored them into a dictionary.

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

$$\begin{aligned} p(C_k \mid x_1, \ldots, x_n) &\propto p(C_k, x_1, \ldots, x_n) \\ &= p(C_k)\, p(x_1 \mid C_k)\, p(x_2 \mid C_k)\, p(x_3 \mid C_k)\, \cdots \\ &= p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)\,, \end{aligned}$$

Where,

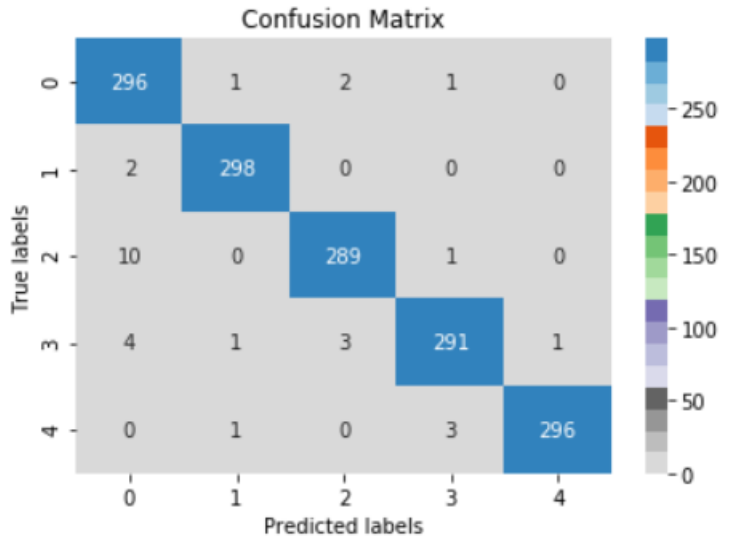
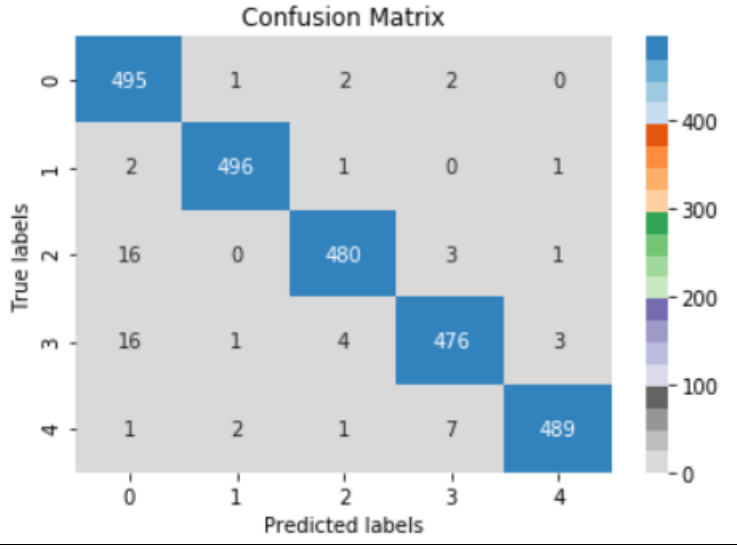
$p(C_k)$ = class prior probability,

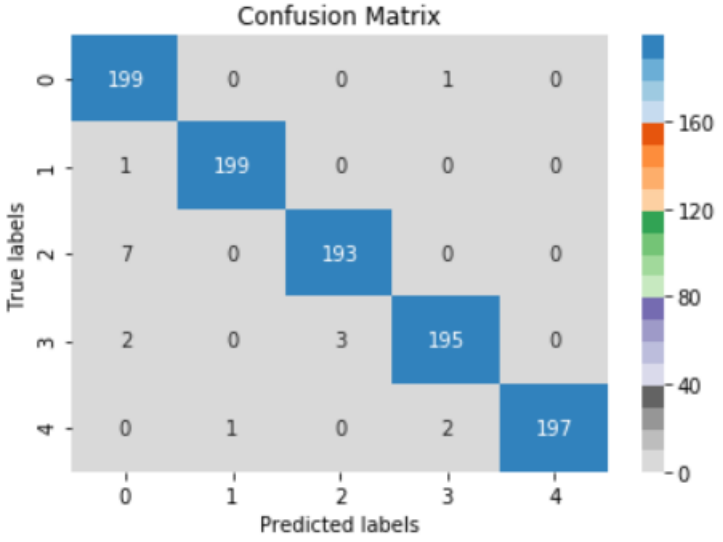
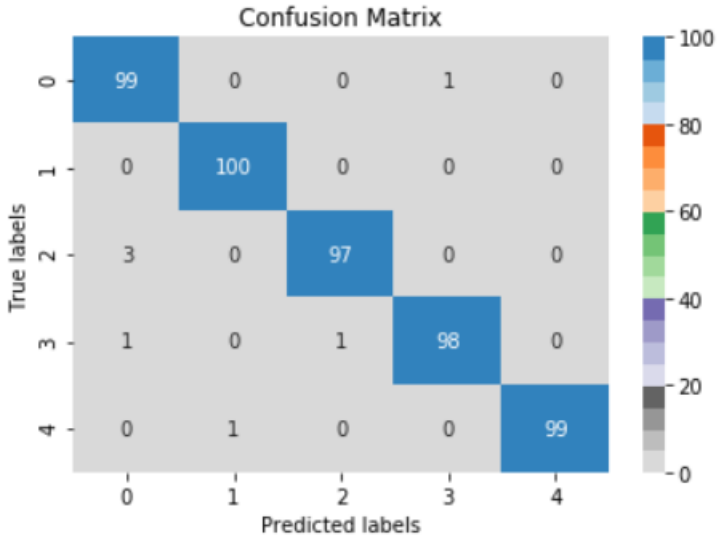
$p(x_i \mid C_k)$ = conditional probability for each word.

- These probability values will be used at testing time.
- Save those variables into files by using 'Joblib' and load them for testing.

- For each class and for each word from vocabulary, take the conditional probability and class prior and calculate the logarithm of them and add them.
- Take max of all class probability for that word and predict class for it.
- Later after prediciting for all the docs, calculate the ‘Confusion matrix’ and ‘Accuracy’ for that split.
- Repeated the same process for other splits.

Results:

Split ratio	Accuracy and Confusion matrix
70:30	0.98 
50:50	0.9744 

80:20	0.983 
90:10	0.986 

2. Naive Bayes Classification for different splits:

Pre-Processing the dataset:

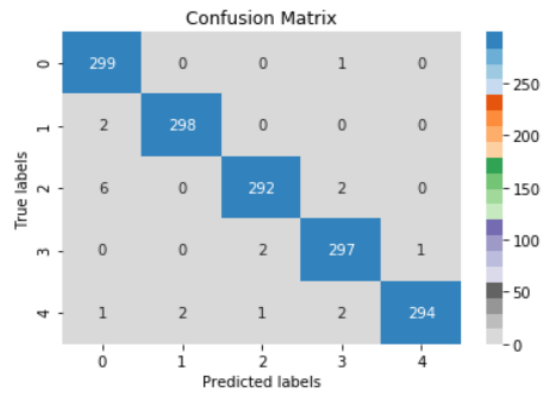
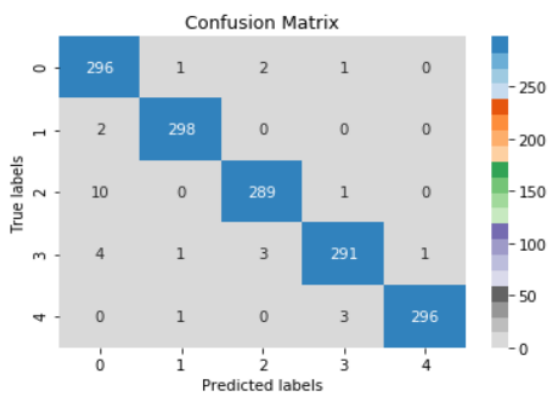
- Initially read all the class document names from each folder and stored to a list.
- Read those files and done the pre-processing for both train and test data.
- For pre-processing the text, used NLTK library and done following steps:
    - Conversion to lowercase
    - Contractions
    - Removed unnecessary characters and punctuations and tokenized using RegexpTokenizer.
    - Lemmatization
    - Stop words removal.

Procedure:

- Same as Q1 procedure but here taken features by calculating Tf-Idf values for each word in each document of class.
- After calculating Tf-Idf values and take top ‘k’ features from each class and merge them all which we consider as vocabulary for this question and do same as Q1.
- Construct the confusion matrix and accuracy for this.
- Compare with Q1 70:30 split.

Result and Analysis:

Accuracy and Confusion matrix for this question and Q1 70:30 split

Tf-Idf based Naïve Bayes	Naïve Bayes
0.9866666666666667	0.98
	

Here we have taken features based on Tf-Idf values and done the Naïve Bayes classification. It means we are selecting best features from each class. So that our accuracy is better than without feature selection model (Q1).