IR Assignment-5

- 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. K-Means Clustering using Bag-of-words and Word2Vec features:

Pre-Processing the dataset:

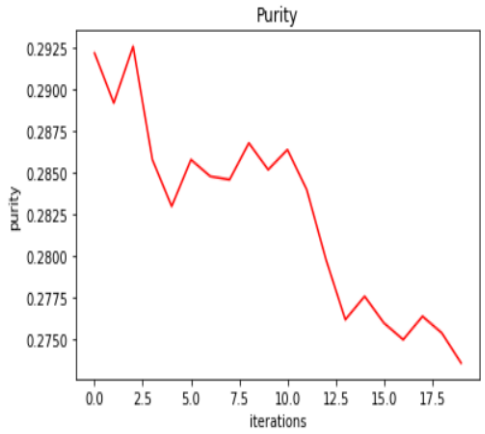
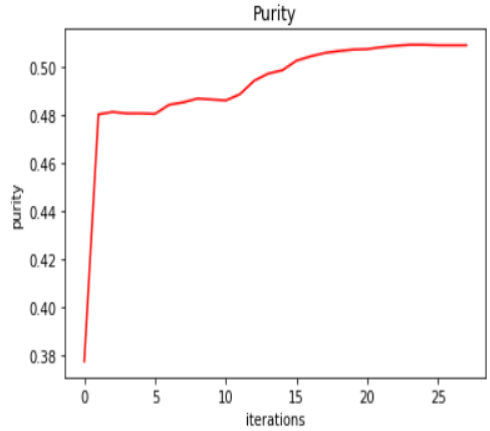
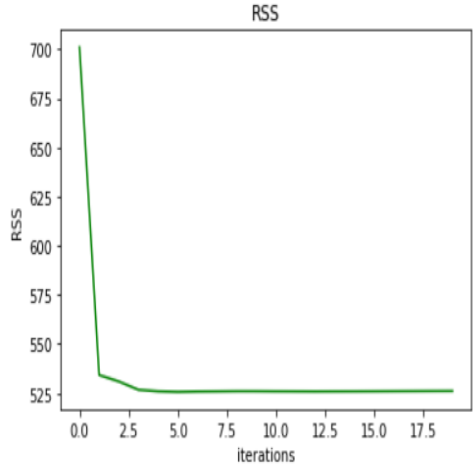
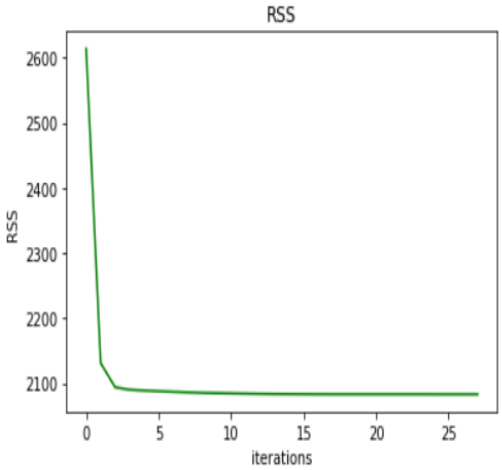
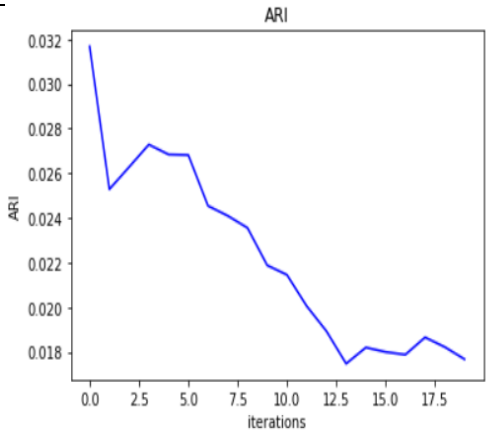
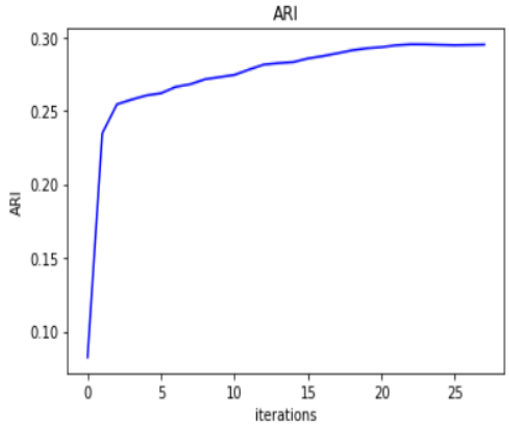
- Initially read all the class document names from each folder and stored to a list.
- Read those files and done the pre-processing for both train and test data.
- For pre-processing the text, used NLTK library and done following steps:
    - Conversion to lowercase
    - Contractions
    - Removed unnecessary characters and punctuations and tokenized using RegexpTokenizer.
    - Lemmatization
    - Stop words removal.
- **Feature Extraction:**
    - For **Bag-of-words** features: used term-frequency (here our vector dimension is 65309).
    - For **Word2Vector**: extracted the 300 dimension vector for each word.

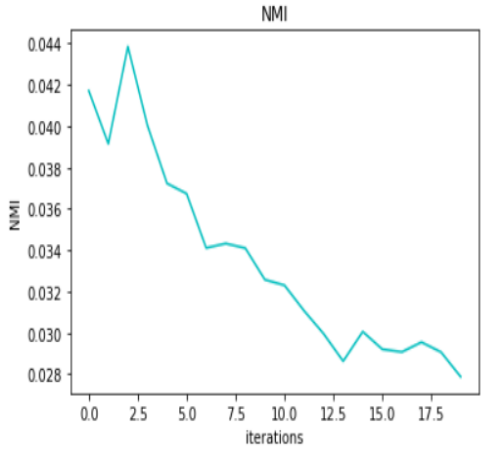
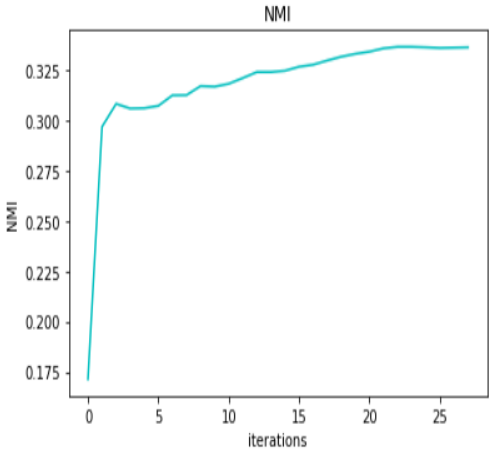
Procedure:

- Prepare the 'terms' list, 'document_ids', Vector space model(VSM) vector and saved them as '.sav'(serializable) files.
- Done the K-means clustering for k=5 classes by taking seed by random initialization.
- Repeated the K-means algorithm and restricted it by setting maximum iteration as 20.
- Computed the metrics like Purity, ARI(Adjusted Rand Index), NMI(Normalized Mutual Information), RSS(Residual Sum of Squares), analysed and plotted them.

Results and Analysis:

- Ran the algorithm for maximum iteration=20. In each iteration calculated the metrics.
- For each iteration 'Purity' was fluctuating, 'RSS' was decreasing, 'ARI' was fluctuating, 'NMI' was fluctuating.

Metric	Plot and Analysis	
	Bag-of-Words	**Word2Vec**
Purity		
RSS		
ARI		

NMI		

2. KNN Classification for different splits:

Pre-Processing the dataset:

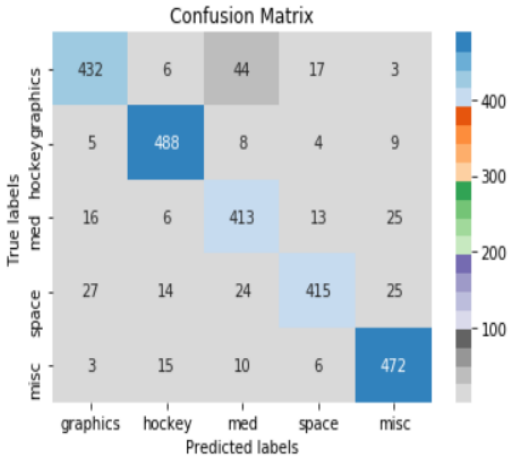
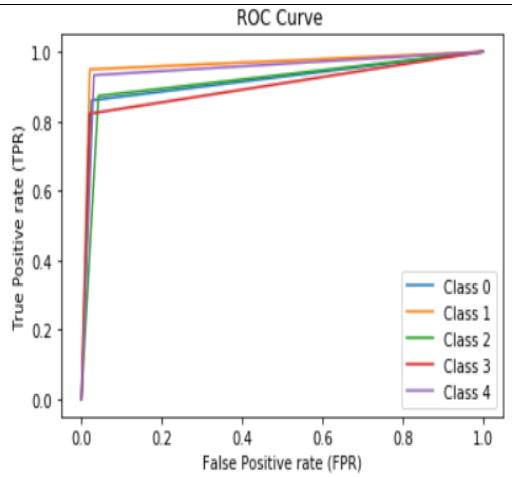
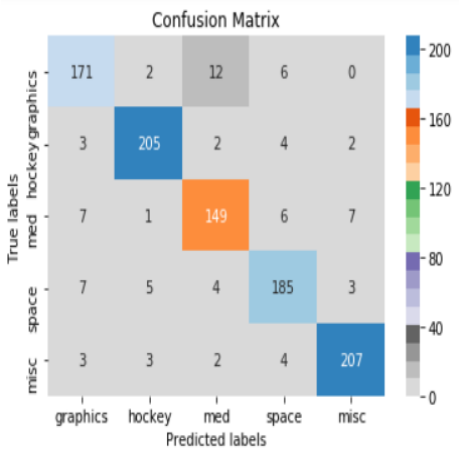
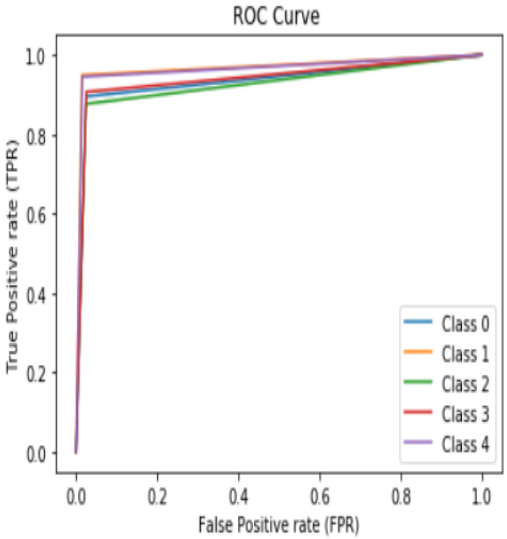
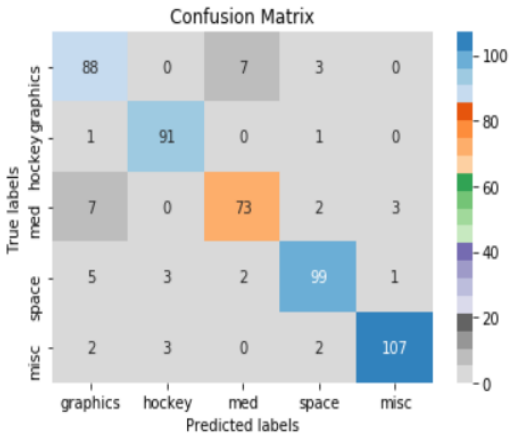
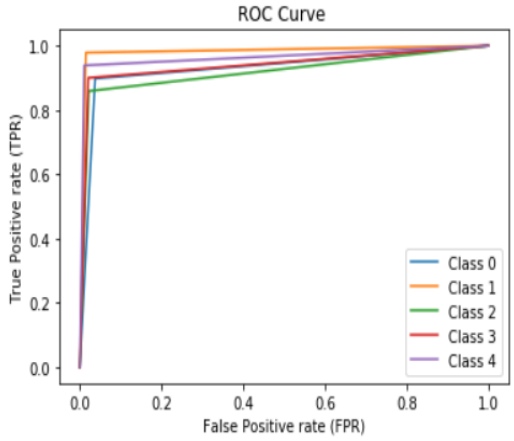
- Initially read all the class document names from each folder and stored to a list.
- Read those files and done the pre-processing for both train and test data.
- For pre-processing the text, used NLTK library and done following steps:
    - Conversion to lowercase
    - Contractions
    - Removed unnecessary characters and punctuations and tokenized using RegexpTokenizer.
    - Lemmatization
    - Stop words removal.
- **Feature Extraction:**
    - For **Word2Vector**: extracted the 300 dimension vector for each word.

Procedure:

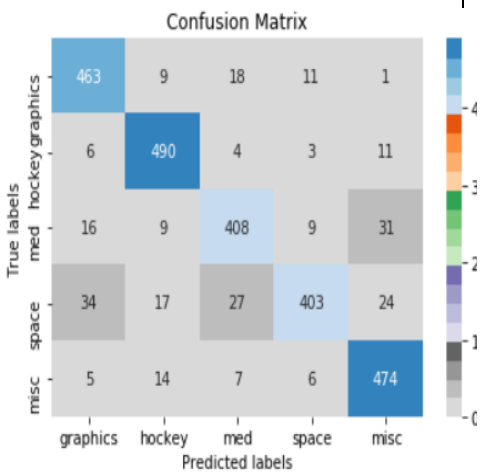
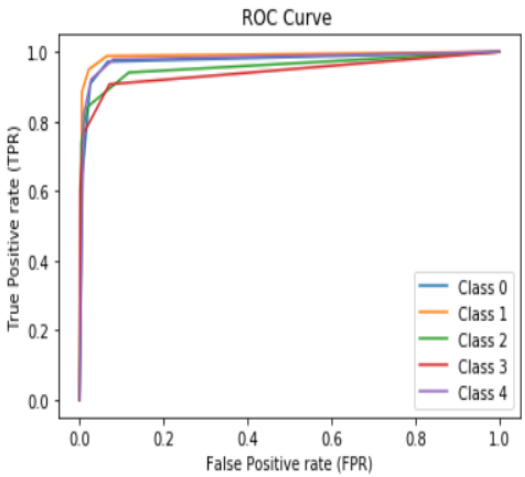
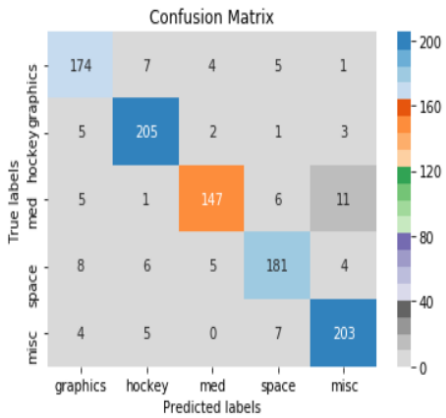
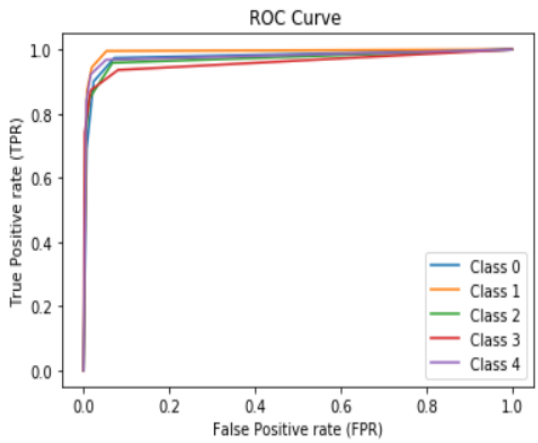
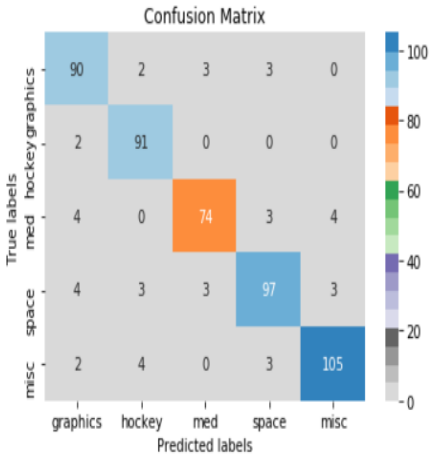
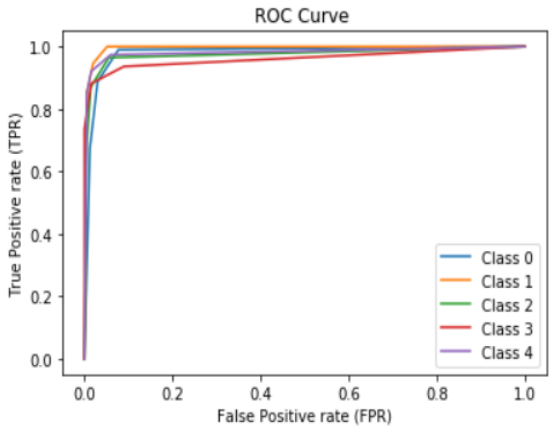
- Initially split the dataset according to our need as training and test data as 50:50, 80:20, 90:10 ratios.
- Done the KNN for k=1,3,5 values for each splitted data and computed the metrics like Accuracy, Confusion matrix and ROC curves.
- Save those variables into files by using ‘Joblib’ and load them for testing.
- Later after prediciting for all the docs, calculate the ‘Confusion matrix’ and ‘Accuracy’ for that split.
- Repeated the same process for other splits.

Results:

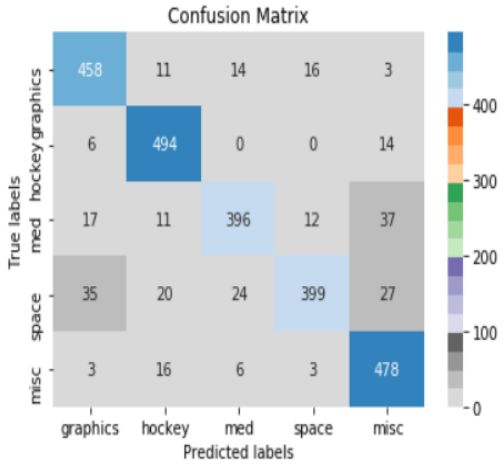
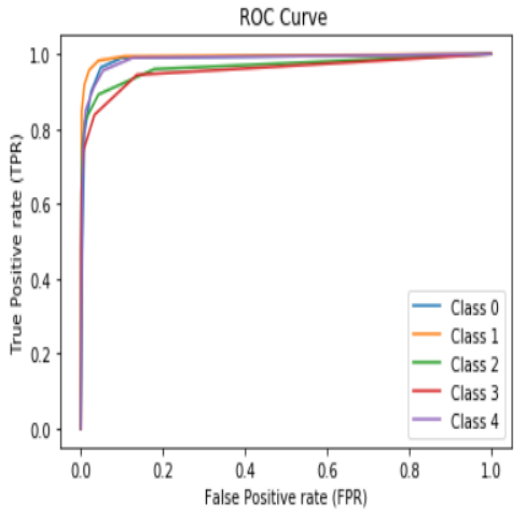
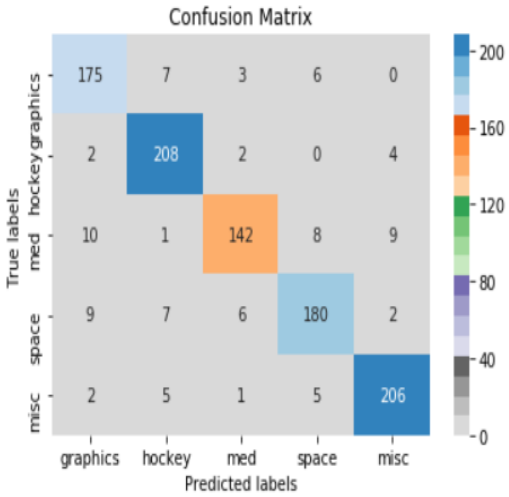
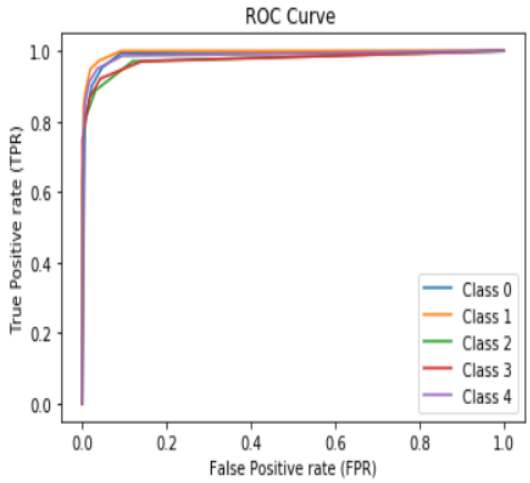
For k=1:

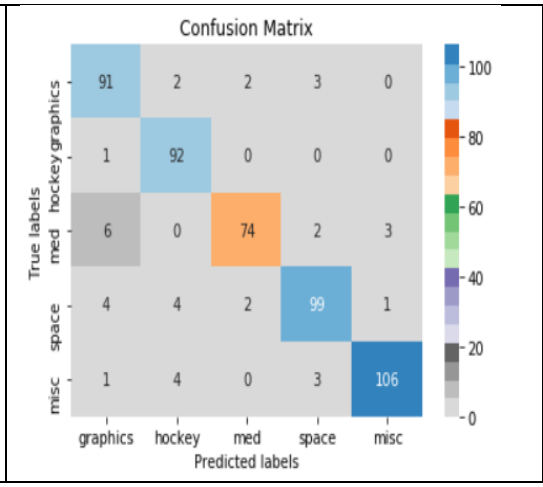
Split	Accuracy and Confusion matrix	ROC
50:50	Accuracy=0.88 	
80:20	Accuracy=0.917 	
90:10	Accuracy=0.916 	

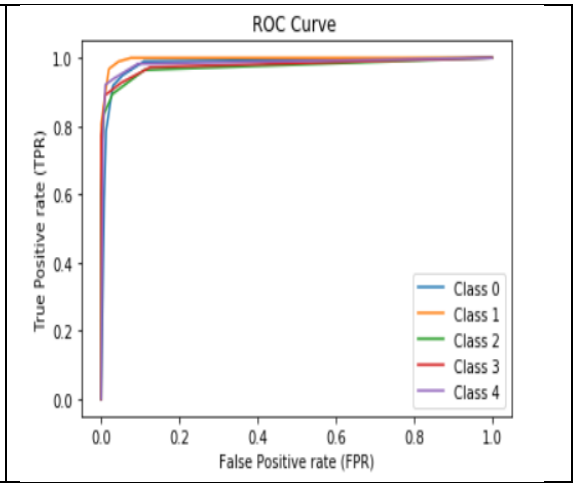
For k=3:

Split	Accuracy and Confusion matrix	ROC
50:50	Accuracy=0.8952 	
80:20	Accuracy=0.91 	
90:10	Accuracy=0.914 	

For k=5:

Split	Accuracy and Confusion matrix	ROC
50:50	Accuracy=0.89 	
80:20	Accuracy=0.911 	
90:10	Accuracy=0.924	





- In KNN, it is giving accuracy of max of 91% in a split.
- But in Naïve Bayes, it is giving accuracy of >90% in almost all splits.
- So Naïve bayes is better than KNN, because it was calculated depending on joint probability, whereas KNN is just predicts according nearest neighbour which is not robust.