ASSIGNMENT-5

SWAGATAM CHAKRABORTI(MT18146)

1. KMEANS CLUSTERING ALGORITHM:

PREPROCESSING:

1. Each line of the file is pre processed individually
2. Word tokenization is done using nltk library using the regextokenization which handles the formation of the tokens and also the removal of the punctuations
3. Stopwords have been removed from the tokens formed.
4. Lemmatization have been performed over the tokens using the nltk library
5. If the line contains any numbers, it is converted into words using inject library and is stored in the vocab along with the number itself

BAG OF WORD MODEL DATASET FORMATION:

Datasets are formed by the tokens of and its term frequency for each documents.

WORD 2 VEC MODEL DATASET FORMATION:

Datasets are formed by the stacking of the 300 length word to vectors of the tokens of the documents each weighed by its tf-idf score and then averaging out.

AASUMPTIONS:

The stopping criteria for the algorithm is the number of iterations.

RESULTS AND ANALYSIS:

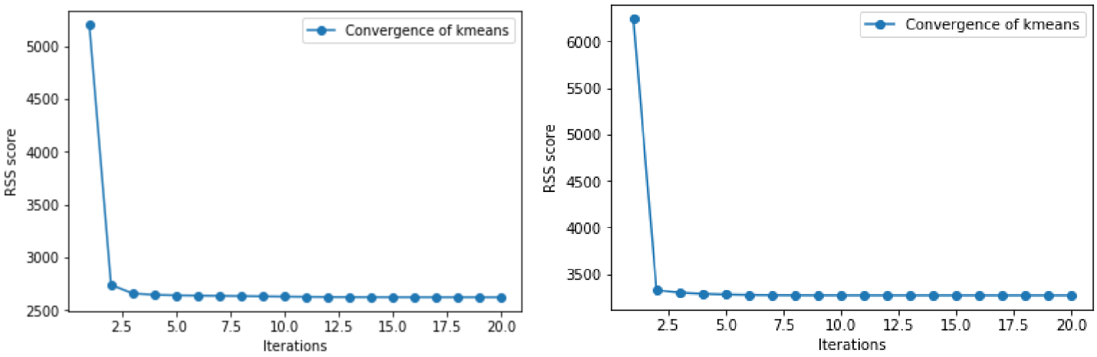

Figure 1: Comparison of the convergence across different models

	Purity	ARI	RSS
Bag of words	0.4202	0.04137615932008366	3273.0060947627376
Word2vec	0.5548	0.22792542521549608	2621.6488836541757

Table 2: Comparison of the evaluating matrices of the two models

	Cluster1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
Bag of words	884	1680	936	788	712
Word2vec	1042	1184	1222	963	589

Table 3: Cluster analysis across different models

INFERENCES: From the comparison analysis it is evident that word2vec model performs better than bag of model.

2. KNN CLUSTERING ALGORITHM:

PREPROCESSING:

6. Each line of the file is pre processed individually
7. Word tokenization is done using nltk library using the regextokenization which handles the formation of the tokens and also the removal of the punctuations
8. Stopwords have been removed from the tokens formed.
9. Lemmatization have been performed over the tokens using the nltk library
10. If the line contains any numbers, it is converted into words using inject library and is stored in the vocab along with the number itself

RESULTS AND ANALYSIS:

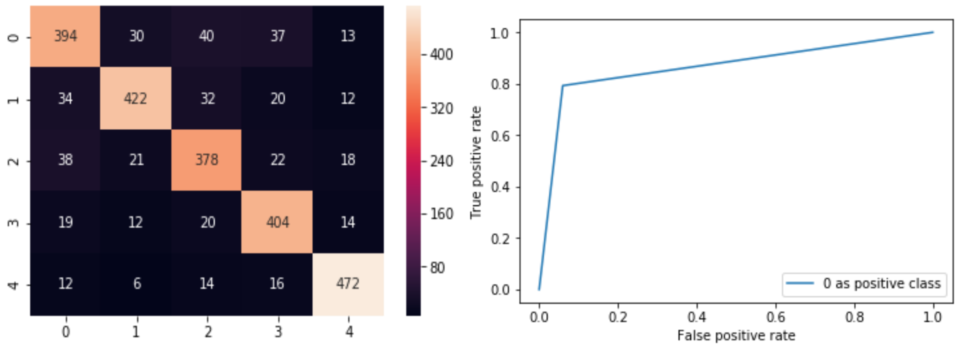

Figure 2 confusion matrix and ROC for k=1 and 50:50 split data

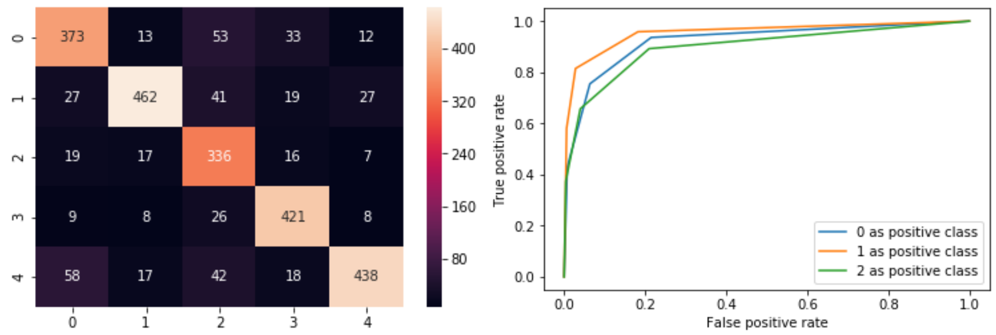

Figure 3: confusion and ROC for k=3 and 50:50 split

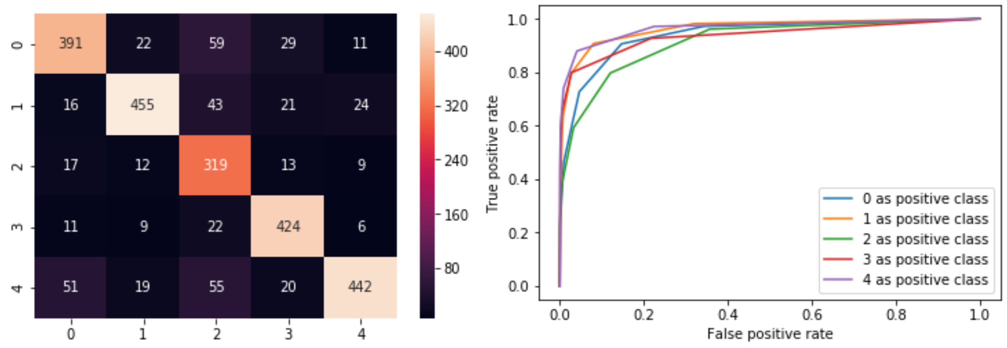

Figure 4: confusion and ROC for k=5 and 50:50 split

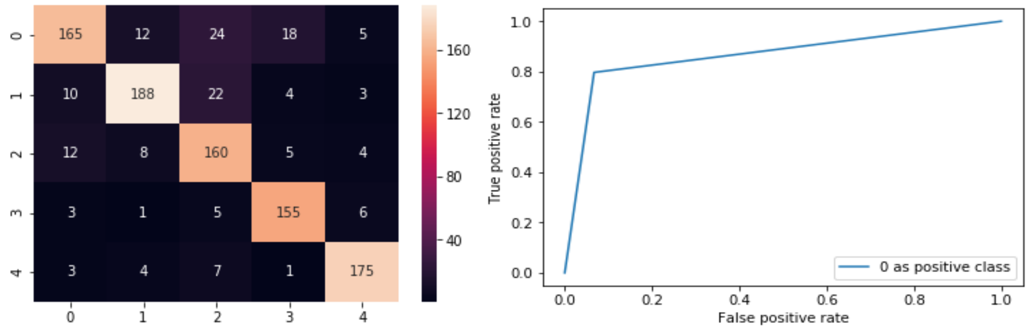

Figure 5: confusion and ROC for k=1 and 80:20 split

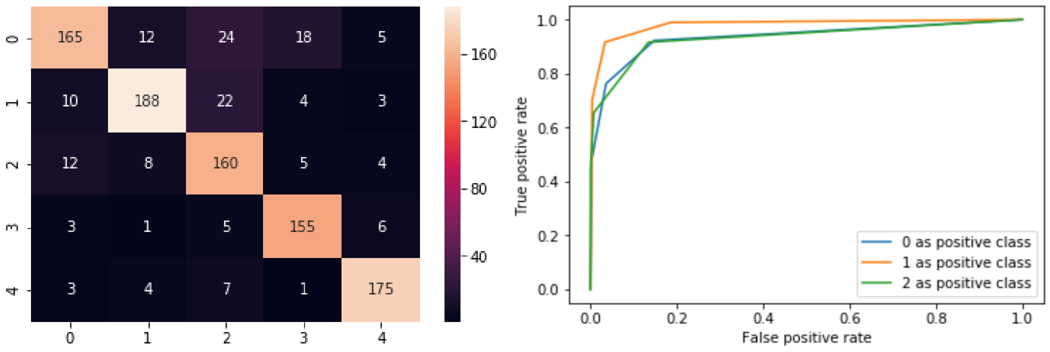

Figure 6: confusion and ROC for k=3 and 80:20 split

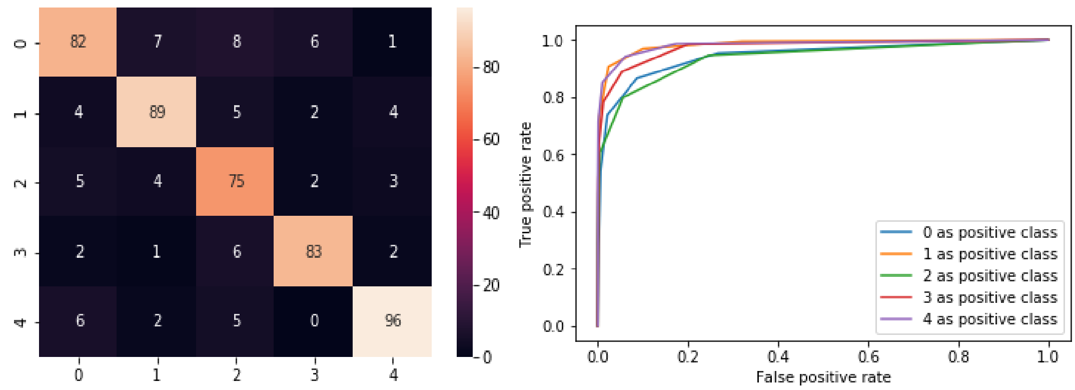

Figure 7: confusion and ROC for k=5 and 80:20 split

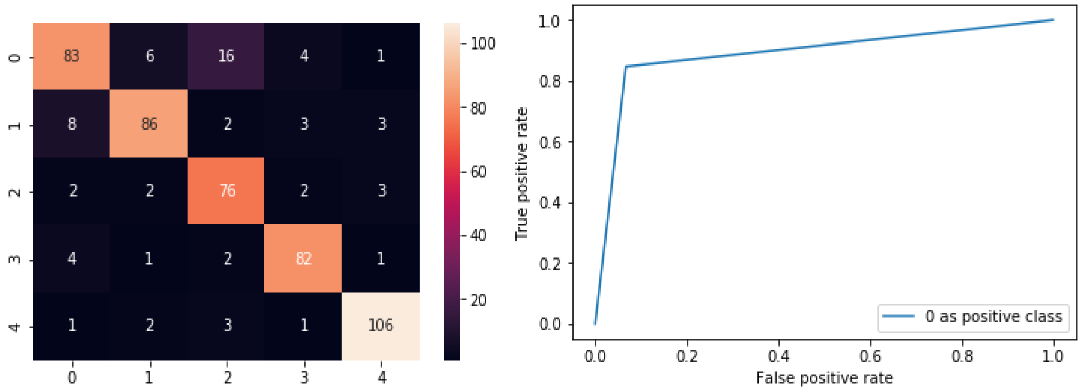

Figure 8: confusion and ROC for k=1 and 90:10 split

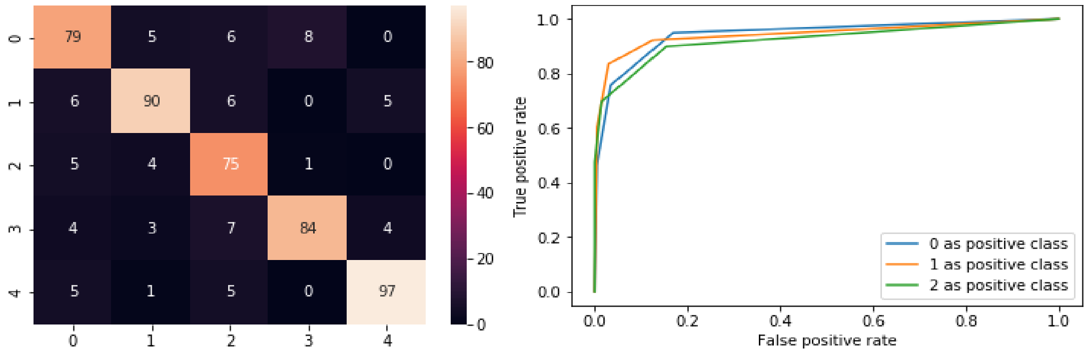

Figure 9: confusion and ROC for k=3 and 90:10 split

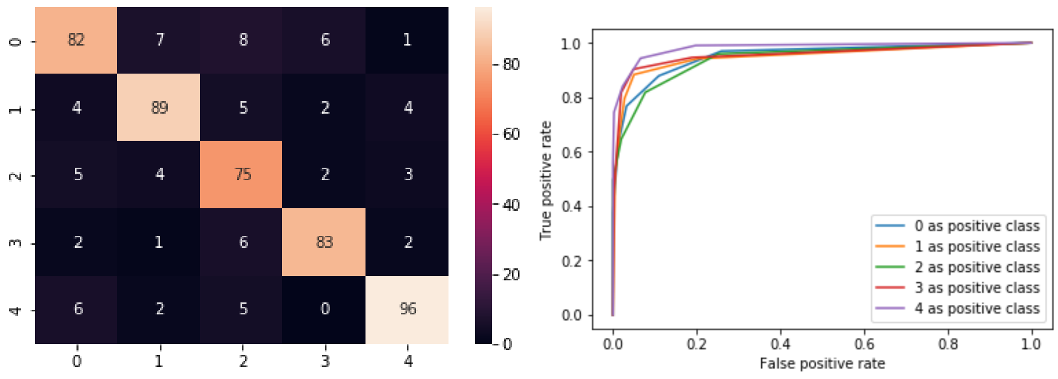

Figure 10: confusion and ROC for k=5 and 90:10 split

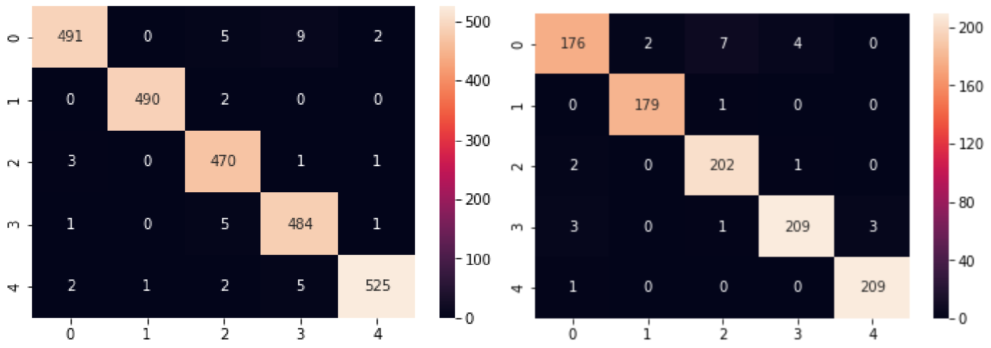

Figure 11: Confusion matrix with Naive Bayes at 80:20 and 90:10 splits

	K=1	K=3	K=5	Naïve Bayes
50:50	82.8	81.28	81.24	98.4
80:20	83.5	84.3	85.9	97.5
90:10	86.6	85.0	85.0	98.4

Table 2: Accuracy across different splits and different

INFERENCE: From the above table it is evident that the naïve bayes accuracy is always higher as compared to knn across different values of k.