# Report :: TIPR Assignment - I

A. Lakshmi

11 Feb, 2019

## 1 Bayes Classifer

Implemented Bayes and naive Bayes classifier. The following comparisons are done with Bayes classifier. Bayes classifier was tested on the original dataset and dataset whose dimension was reduced using Gaussian random projection. The random projection matrix was formed from elements drawn from independent and identically distributed Gaussian distribution. Dimensions were varied from 2 to half of the original dimension in steps of two and saved in the respective folders. The dataset was split into 10 folds and leave one out cross validation was adopted. The datasets were analysed based on the accuracy and F1 micro and macro score

### 1.1 Bayes Classifer on Dolphin dataset

Fig. 1 shows the results of Bayes classifier on Dolphin dataset. The implemented version (Bayes classiifer) and python version (Naive Bayes) were performing better on dolphin dataset. However, the implemented version was performing better as Bayes does not assume feature independence whereas Python version is Naive Bayes, which assumes feature independence. There was decrease in the performance with decrease in dimension

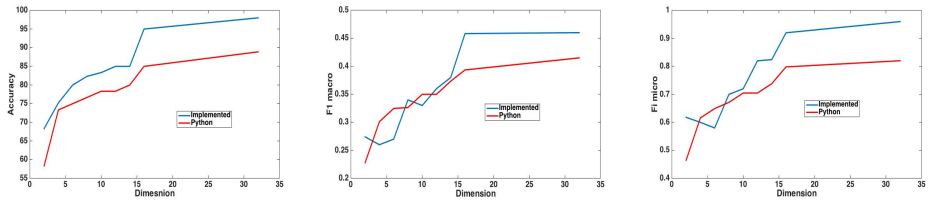


Figure 1: Comparison between implemented and python Bayes classifier on Dolphin dataset. Left: Accuracy, Middle: F1 macro, Right: F1 micro

## 1.2 Bayes Classifer on pubmed dataset

Fig. 2 shows the results of Bayes classifier on pubmed dataset. Performance of Bayes classifier on dolphin dataset was giving better performance than that of pubmed dataset This may be attributed to the singular nature of the high dimensional covariance matrix inversion in Bayes classifier. The performance was not degrading much with decrease in the dimension as dimension is large and certain features may be dependant and hence redundant.

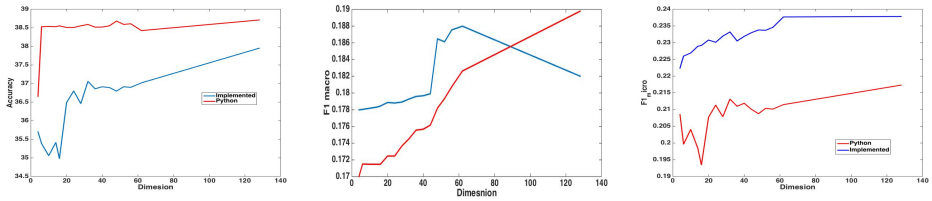


Figure 2: Comparison between implemented and python Bayes classifier on Pubmed dataset. Left: Accuracy, Middle: F1 macro, Right: F1 micro

## 1.3 Bayes Classifer on twitter dataset

The twitter dataset is converted into self implemented Bag of Words. As vocabulary size was too large (around 5000) (with most of the words occuring in one or two documents), they were attributing to noise and the run time was very high. Hence, the vocabulary size was reduced to decrease the run time complexity. As certain words were very rare, they were mostly attributing to noise and hence their removal from the vocabulary didn't decrease the performance of the algorithm. In this dataset, lower dimension seems to perform better than the higher dimensions. (Fig. 3)

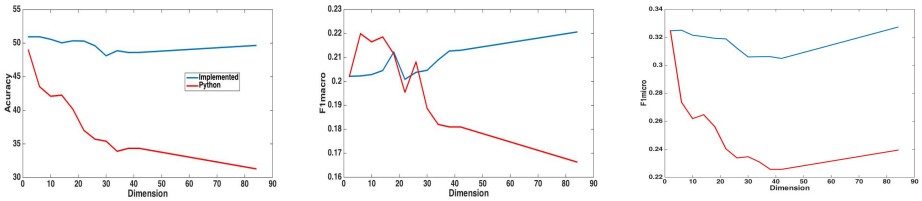


Figure 3: Comparison between implemented and python Bayes classifier on Twitter dataset. Left: Accuracy, Middle: F1 macro, Right: F1 micro

# 2 Nearest Neighbor Classifer

Nearest neighbor classifier was tested on the three datasets.

## 2.1 Nearest Neighbor on Dolphin dataset

The performance of the implemented NN classifier and python NN classifier were similar (Fig. 4). There was decrease in the performance of the algorithm with decrease in dimension.

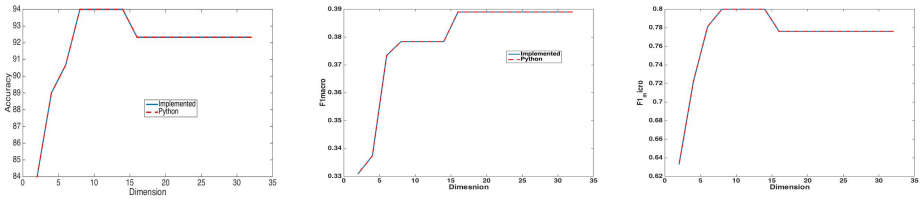


Figure 4: Comparison between implemented and python nearest neighbor classifier on Dolphin dataset. Left: Accuracy, Middle: F1 macro, Right: F1 micro

## 2.2 Nearest Neighbor on Pubmed dataset

The performance of the implemented NN classifier and python NN classifier were similar (Fig. 5). As the dimension is very huge, run time was very high. Hence, the following parameters were estimated on 100 test data of every split of cross fold validation and averaged over all the splits. There was no noticeable decrease in the perfromance with reduction in the dimension.

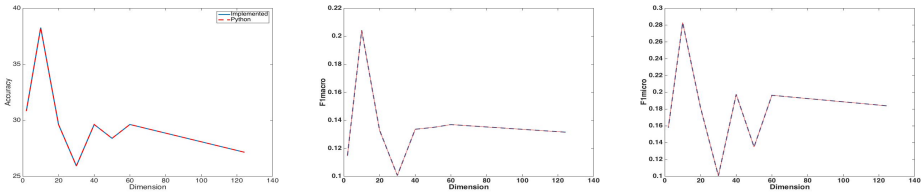


Figure 5: Comparison between implemented and python nearest neighbor classifier on Pubmed dataset. Left: Accuracy, Middle: F1 macro, Right: F1 micro

## 2.3 Nearest Neighbor on Twitter dataset

The performance of the implemented NN classifier and python NN classifier were similar (Fig. 6). It was performing better with low dimension.

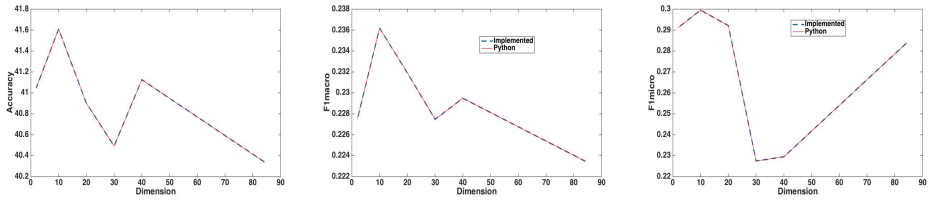


Figure 6: Comparison between implemented and python nearest neighbor classifier on Twitter dataset. Left: Accuracy, Middle: F1 macro, Right: F1 micro

# 3 Locality Sensitive Hashing

LSH was implemented with random projection method.

## 3.1 Locality Sensitive Hashing on Dolphin dataset

LSH was compared with PCA dimensionality reduction (Fig. 7). PCA performs better. However, the performance of LSH and PCA were not degrading much with decrease in dimension

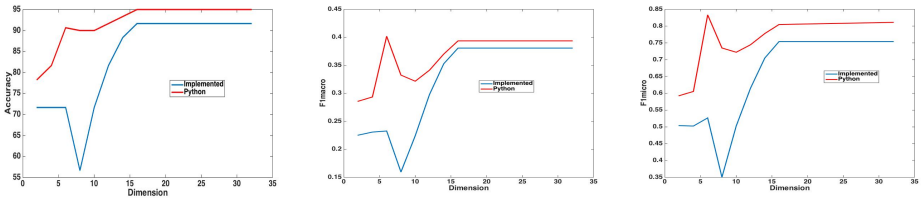


Figure 7: Comparison between LSH and PCA on Dolphin dataset. Left: Accuracy, Middle: F1 macro, Right: F1 micro

## 3.2 Locality Sensitive Hashing on Pubmed dataset

LSH was compared with PCA dimensionality reduction (Fig. 8). LSH performs better. However, the performance of LSH and PCA were not degrading much with decrease in dimension

## 3.3 Locality Sensitive Hashing on Twitter dataset

LSH was compared with PCA dimensionality reduction (Fig. 9). LSH performs better. However, the performance of LSH and PCA improved with decrease in dimension

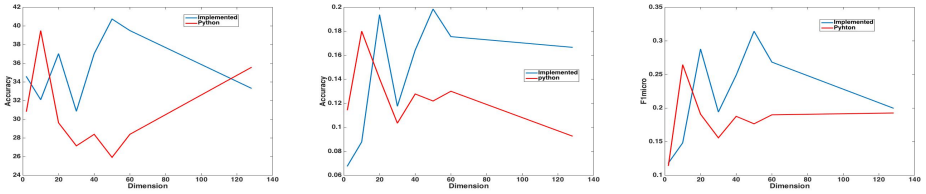


Figure 8: Comparison between LSH and PCA on Pubmed dataset. Left: Accuracy, Middle: F1 macro, Right: F1 micro

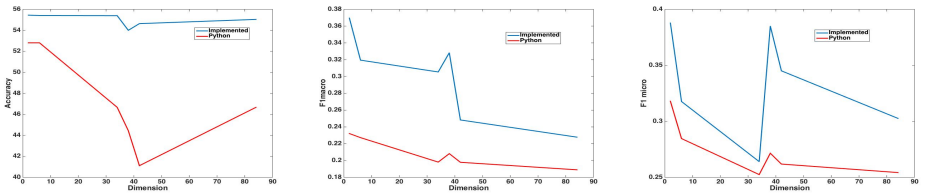


Figure 9: Comparison between LSH and PCA on Twitter dataset. Left: Accuracy, Middle: F1 macro, Right: F1 micro