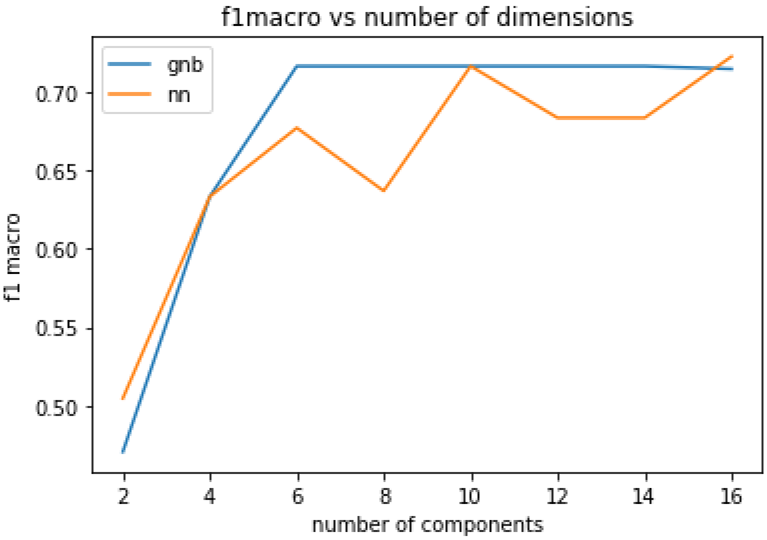
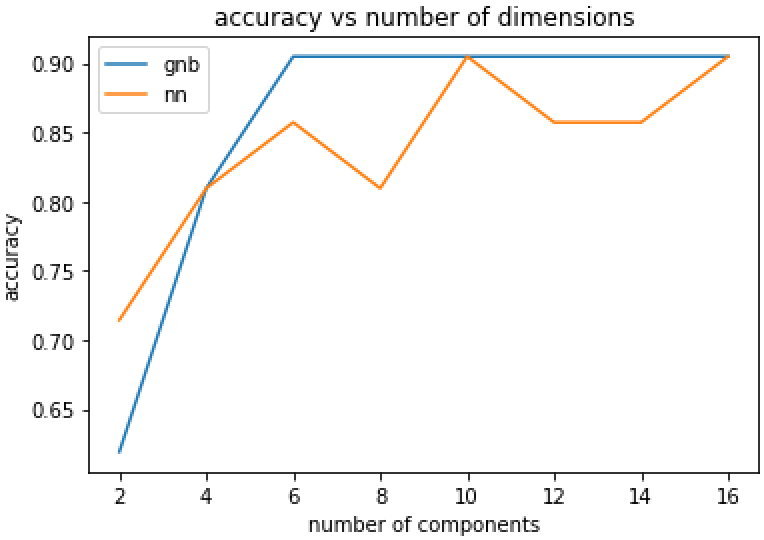
TIPR Assignment 1
Vishay Raina
14860

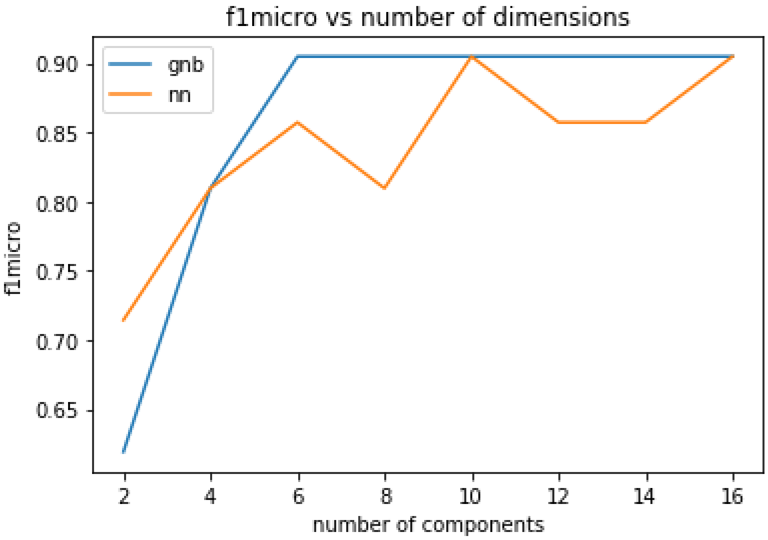
The Procedure:

1. First the dolphins dataset was loaded and arranged in a numpy array.
2. Then the gaussian random projections transform was applied to it for 2,4 ...16 components and this was saved in a .csv format for later use.
3. Then these datasets were loaded and split into train and test data.
4. A gaussian naive bayes algorithm and nearest neighbour algorithm was trained on the training data and then used to classify the test data.
5. The plots for that are given below.
6. The same procedure was applied to Pubmed data for number of components = 2,4, ... upto 64.
7. For twitter data, the text data was transformed into vectors using CountVectorizer method in SKlearn.
8. Since the dimension of the vector was too high ie: 2978, the random projections were only applied for a number of components = 2,4,... upto 500
9. The new instances the previous classification algorithms were trained and tested on PCA reduced data with number of components = 5,6,... upto15 for all 3 datasets.

Dolphin data with random projections:

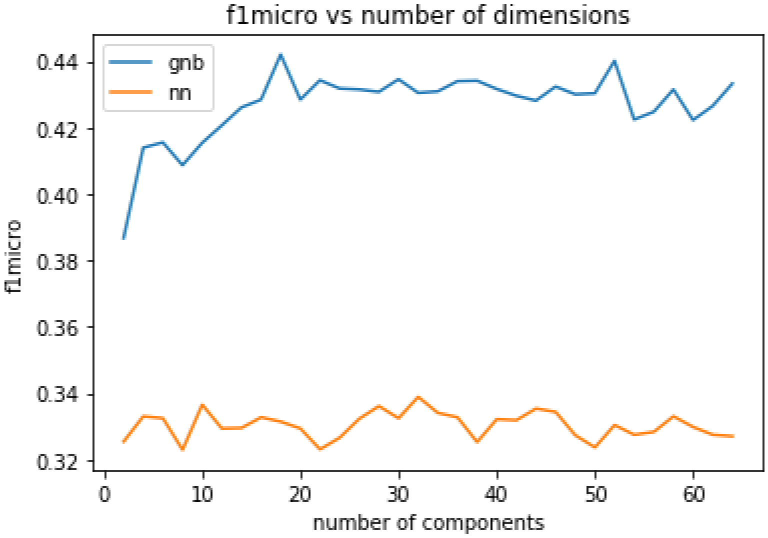


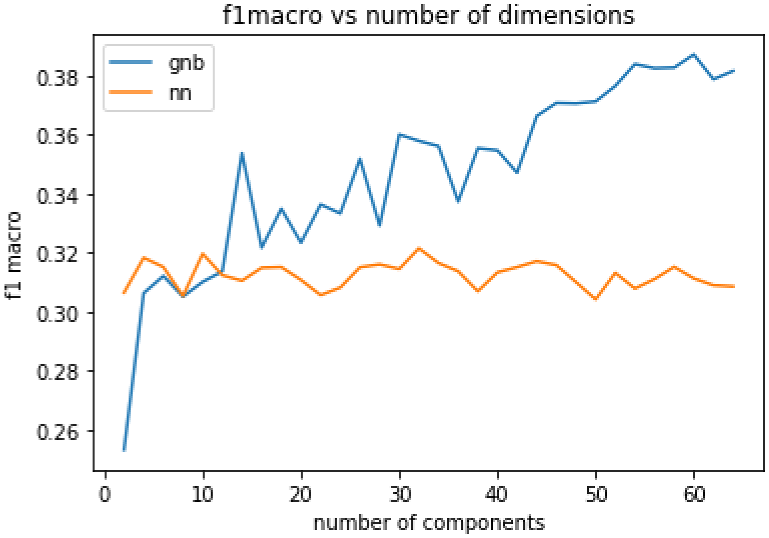


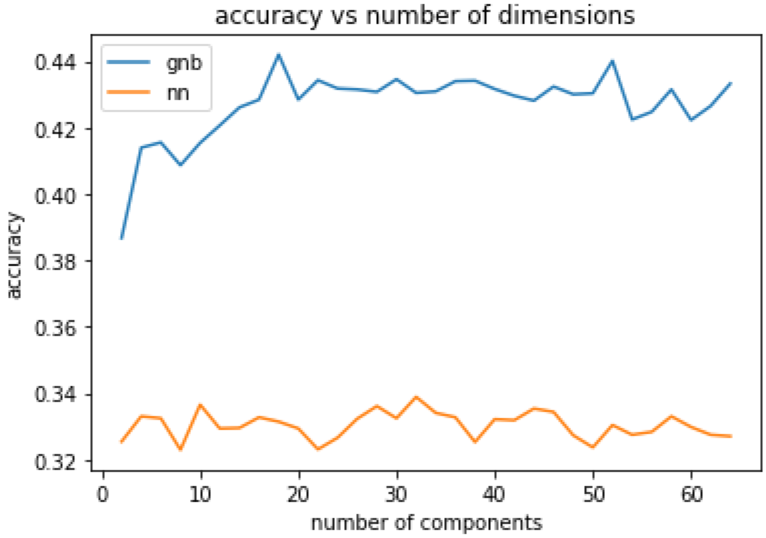


Conclusions: From these plots, we can see a clear elbow forming at number of components = 6. This suggests that 6 is the optimal number of components for getting better results for lesser computation time.

Pubmed data with random projections:

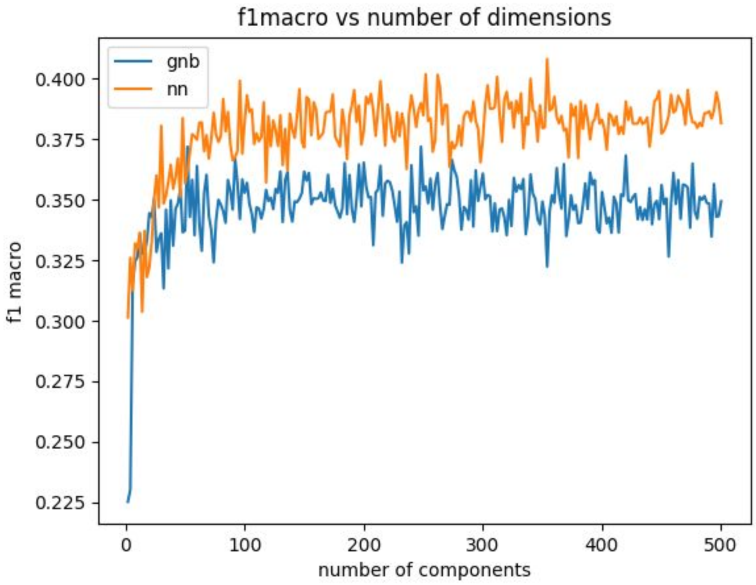

f1micro vs number of dimensions

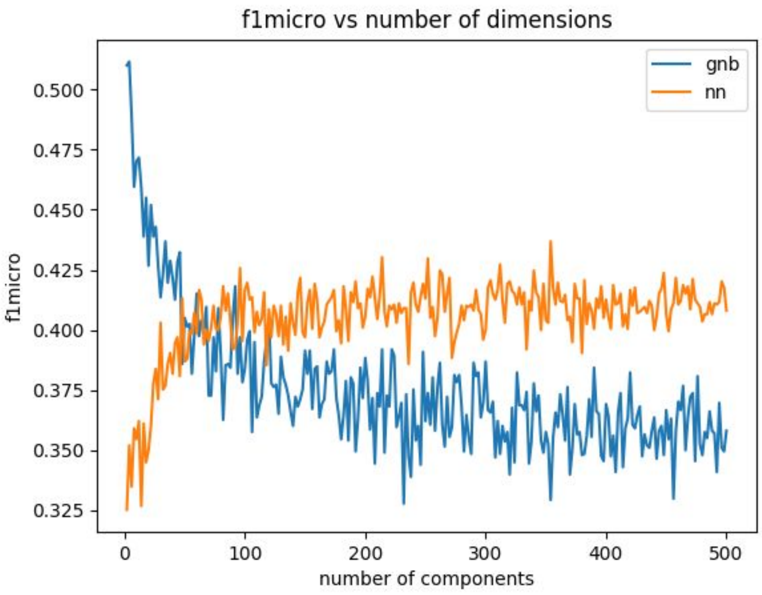

f1macro vs number of dimensions

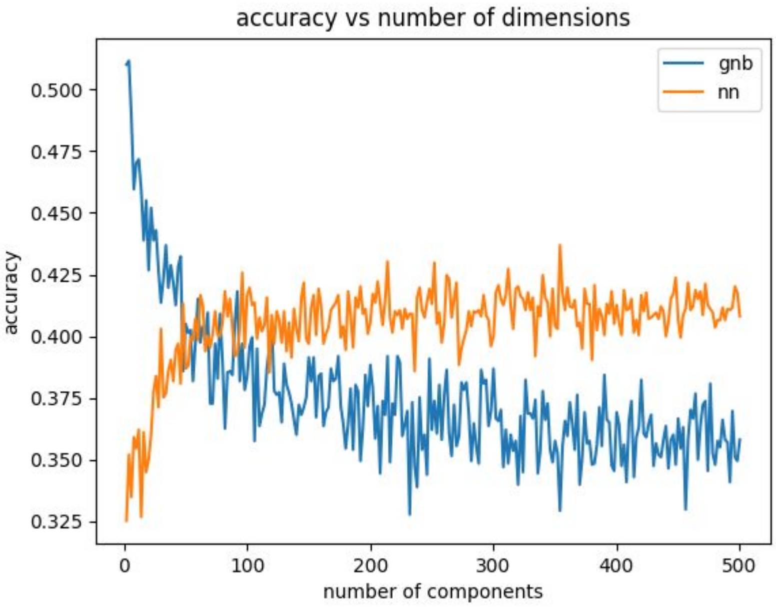

accuracy vs number of dimensions

Conclusions: There is not a clear elbow but number of components = 20 wins for both accuracy as well as f1 micro.

Twitter data with random projections

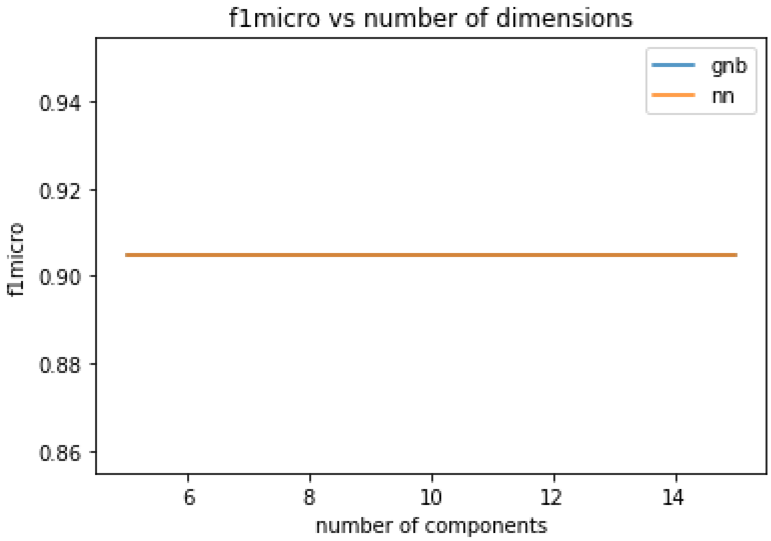


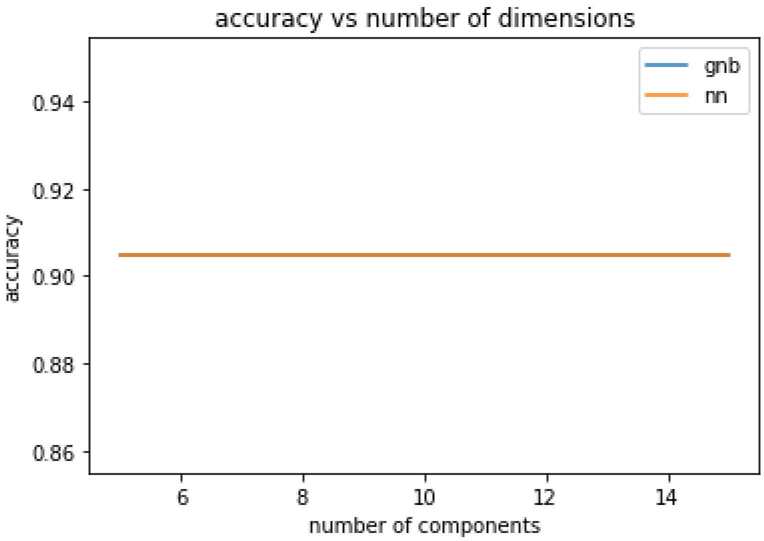


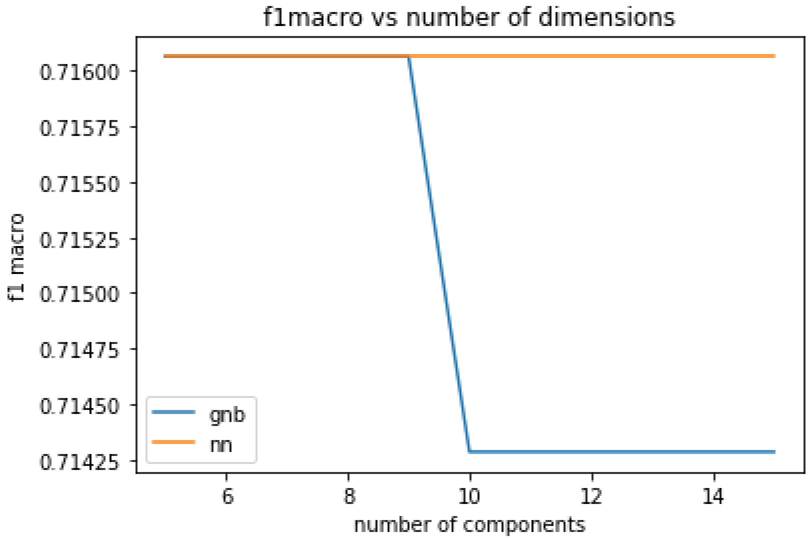


Conclusions: From these plots, we can see a elbow forming at number of components = 80. This suggests that 80 is the optimal number of components for getting better results for lesser computation time.

Dolphin data with PCA reduction:

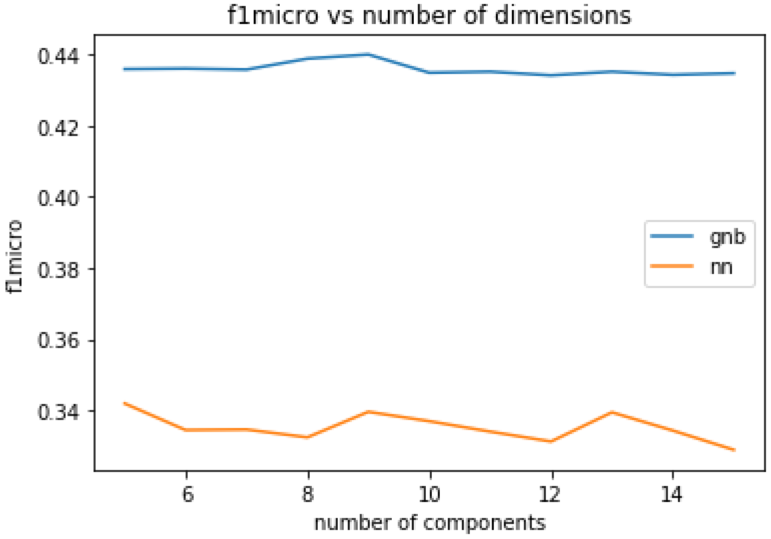


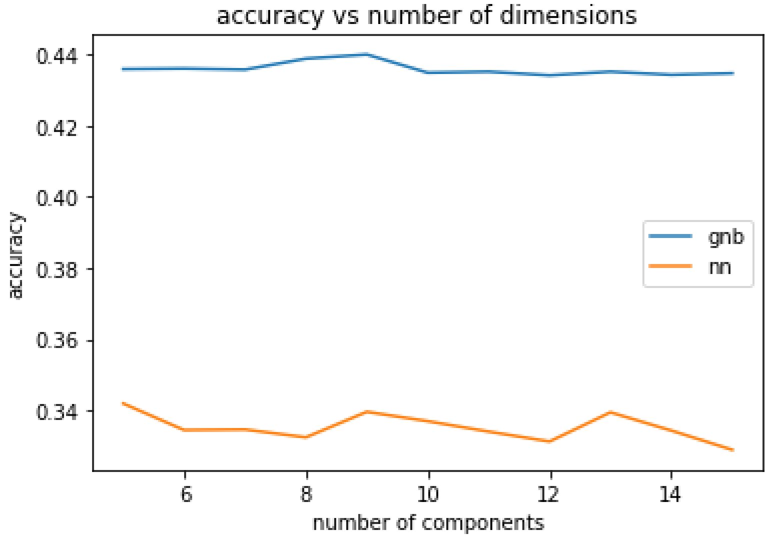


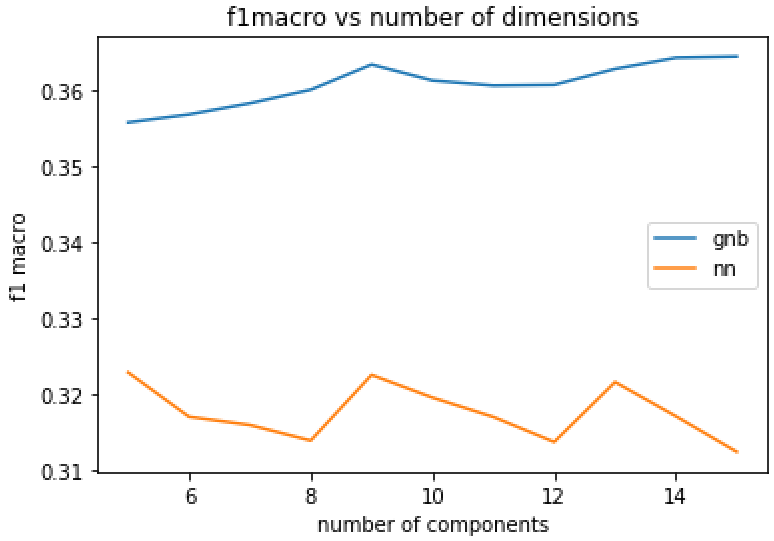


Conclusions: Here the gnb and nn classifiers mostly overlap and also perform better for # components <= 10 (see f1 macro plot). And the accuracy is equal to that from random projections.

Pubmed data with PCA reduction:

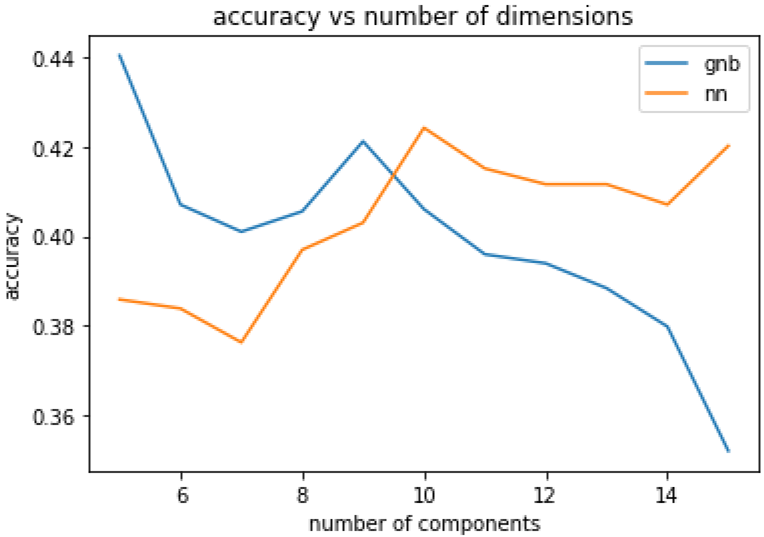

f1micro vs number of dimensions

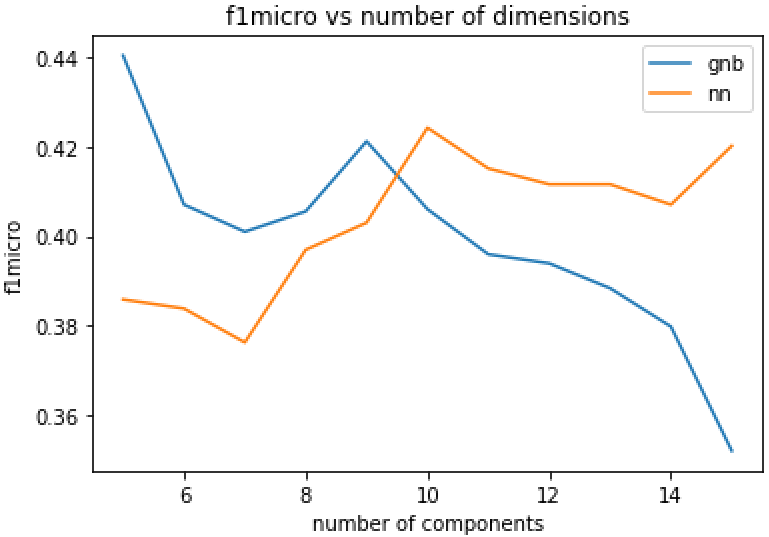

accuracy vs number of dimensions

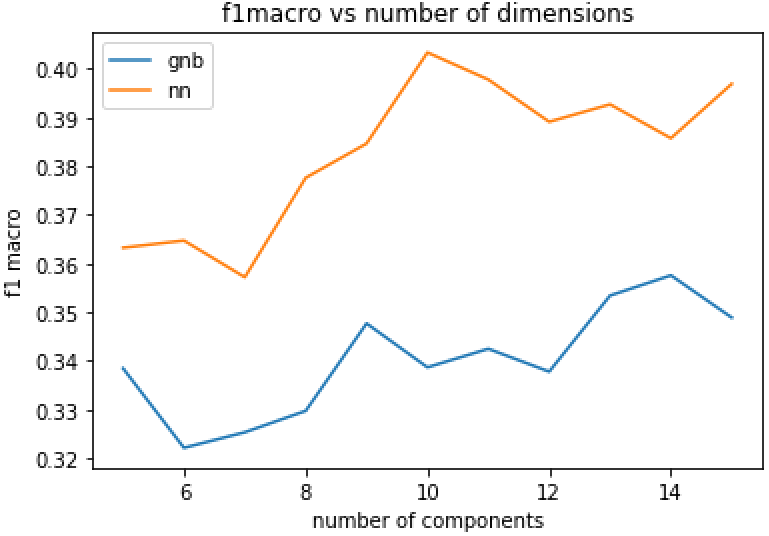

f1macro vs number of dimensions

Conclusions: Here the gnb outperforms nn classifiers and also it performs better for # components = 9 . And the accuracy is equal to that from random projections for much less number of components.

Twitter data with Truncated SVD reduction:







Conclusions: Here the gnb outperforms nn classifiers and also it performs better for # components = 10 . And the accuracy is equal to that from random projections for much less number of components.