Assignment 1
Amal Roy
14989

1. Using random projections matrix as given in the reference “Random projections and applications to dimensionality reduction”, k dimensional dataset is reduced to D=2,4,…,[k/2] dimensions. The function reduce_dim(X,D,location of output file) implemented in projections.py file takes care of this. The files are then stored in the given location.
Uncomment function call in main.py to generate files.

2. Gaussian Naive Bayes and nearest neighbor algorithms are implemented in bayes.py and nn.py

3. As the dimension of data increases, the accuracy and f1 scores increases only marginally or remains constant in most cases. In the case of very high dimensions , for example, the twitter dataset, random projections doesn’t do a very good job as the data matrix is very sparse and features mightnot be picked up during random projection. So, in this case, the original data gives very high accuracy compared to data in reduced dimension.
The plots are given in output_plots/task_3_*

The takeaway is that for most datasets, random projections gives acceptable performance at a reduced runtime and is very feasible for large dimensiond(d>10) datasets. So, Amar or Akbar maybe right but there is a payoff.

4. The plots are found in output_plots/task_4_*

5. The accuracy of my implementation and sklearn’s implementation are around the same (within 0.1 for all cases). However, the running time of sklearn’s implementation is much lower. This could be due to the various optimizations in the library code.

6. The random_hash function implemented in lsh.py reduces the dimension of data by making hash bins as given in section 4.1.2 “Locality sensitive hashing using stable distributions” (Andoni et al.)

7. Plots for PCA reduced data are found in output_plots/task_7_*