CSE 601
DATA MINING AND BIOINFORMATICS

PROJECT 1 - DIMENSIONALITY REDUCTION

Submitted by:

Hrishikesh Nitturkar (hnitturk@buffalo.edu)
Naga Vaishnavi Pakyala (nagavais@buffalo.edu)

Project Description:

In real-world, data collected contains many attributes related to various features which is very good since these features can be used for various analysis. But these features sometimes introduce a lot of noise in the data which makes it is really difficult to visualize the whole data. So, we consider dimensionality reduction where we reduce the number of features, without losing the overall importance of each feature, to make the data more manageable. In this project, we use Principal Component Analysis (PCA), Singular Value Decomposition (SVD) and t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithms to reduce the dimensionality of the data.

Principal Component Analysis converts a set of observations that may be correlated into set of linearly uncorrelated variables using orthogonal transformation. These new linearly uncorrelated variables called principal components, are sorted by the variance i.e. the first principal component has the largest variance, then second principal component has the next highest variance and so on. These principal components are also orthogonal to the preceding components.

PCA Algorithm:

1. Read data from the file. Ex: pca_a.txt file.
2. Convert the data into a dataframe with all features as columns except the last, and convert the last column into a dataframe as labels for each row in the features matrix.
3. It is important to center(or scale or normalize) the data in the input for PCA since PCA is really sensitive to differences in the values. For this, we calculate the mean of each column in data and subtract this mean from each value in the column.

$$X - \overline{x}$$

4. Then calculate the covariance matrix of the resulting data using the formula,

$$S = \frac{1}{n} XX^T$$

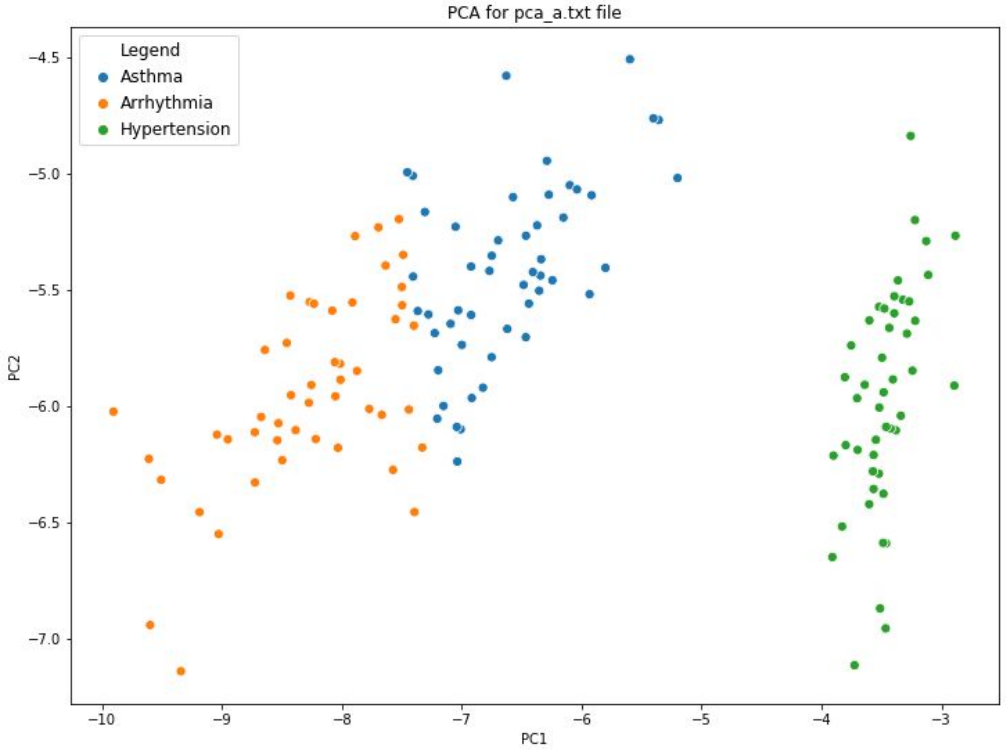
5. Then calculate the Eigen Values and Eigen Vectors of the covariance matrix using the linalg.eig function of numpy library.
6. Generally the eigen values are returned in a sorted order, with the highest value first. Accordingly the corresponding vectors are also returned. Since we are reducing the dimensionality of the original data to 2 columns, we consider only the first two eigen values and eigen vectors.
7. Then perform dot product of the eigen vectors with the data and get the two dimensional data points.
8. Plot the points using scatterplot function grouping by the values in disease column.

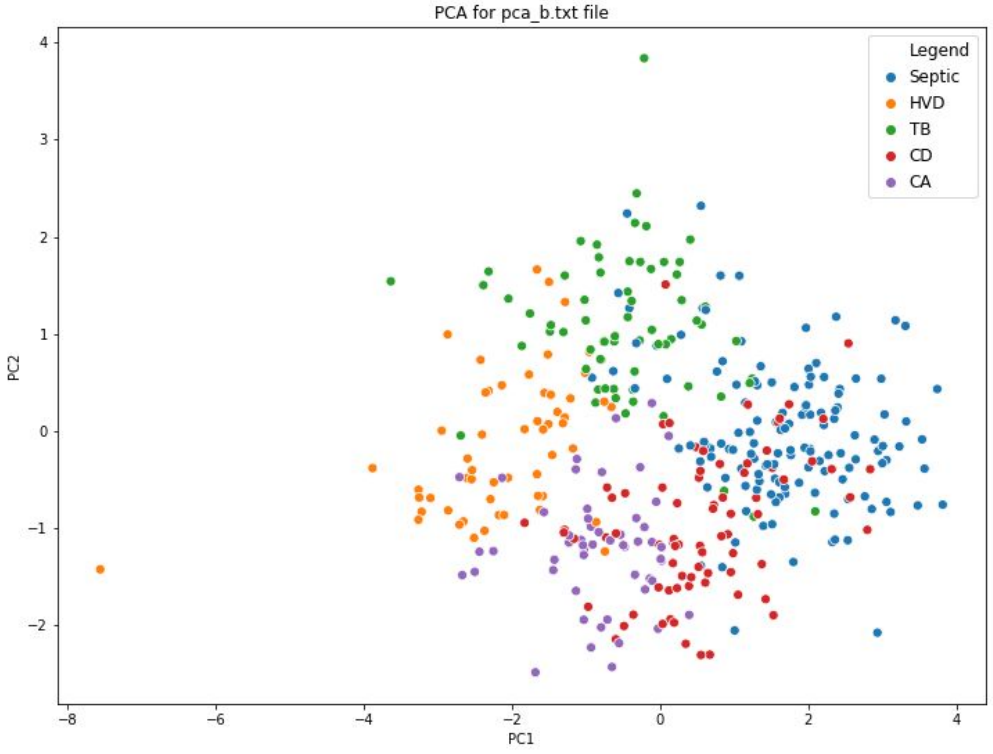
The graphs for PCA and SVD algorithms appear to be similar because these two algorithms use similar mathematical methods to calculate the eigen values and eigen vectors.

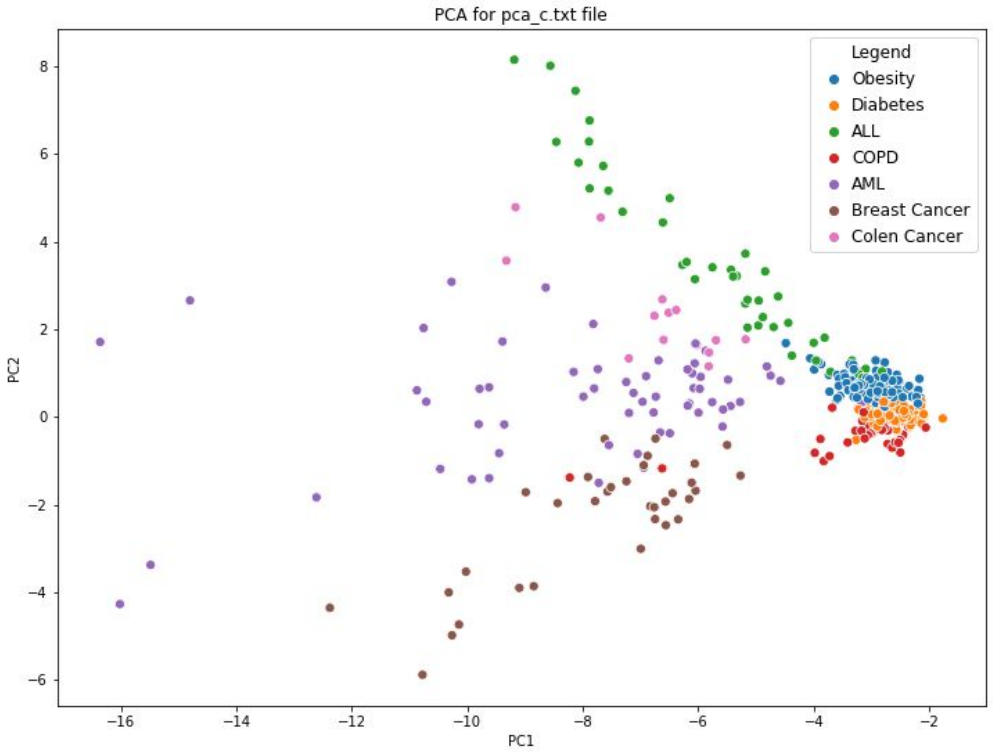
PCA uses: $XX^T = UDU^T$, where U is the data matrix.
SVD uses: $XX^T = U??^2 V$, where U is the data matrix.

Results:

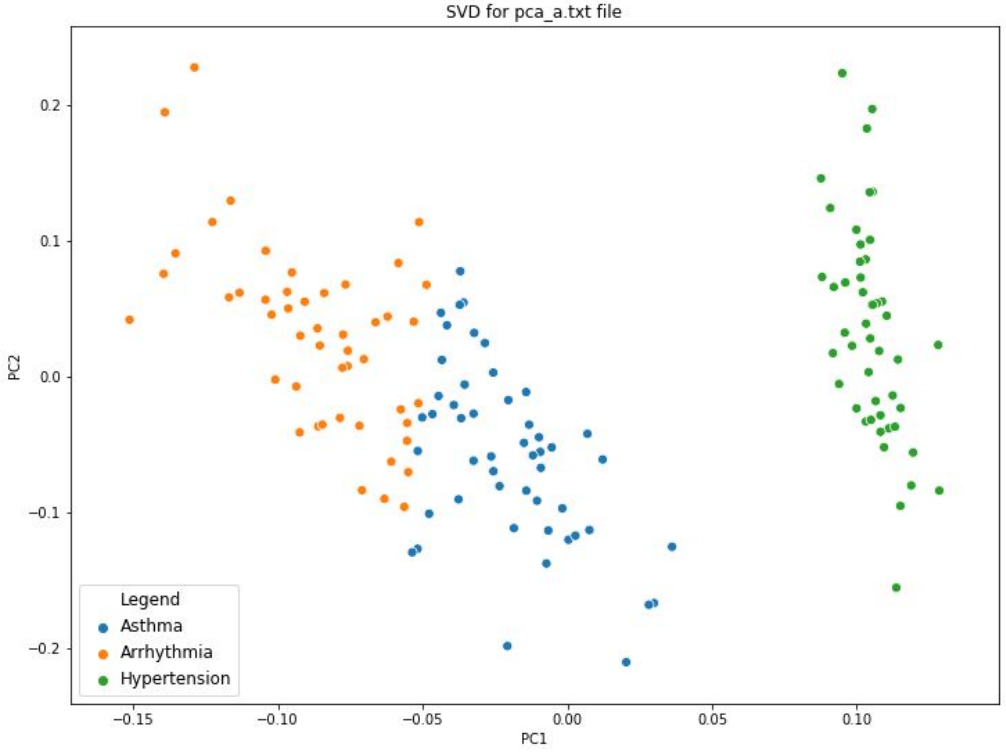
Principal Component Analysis (PCA) Algorithm:

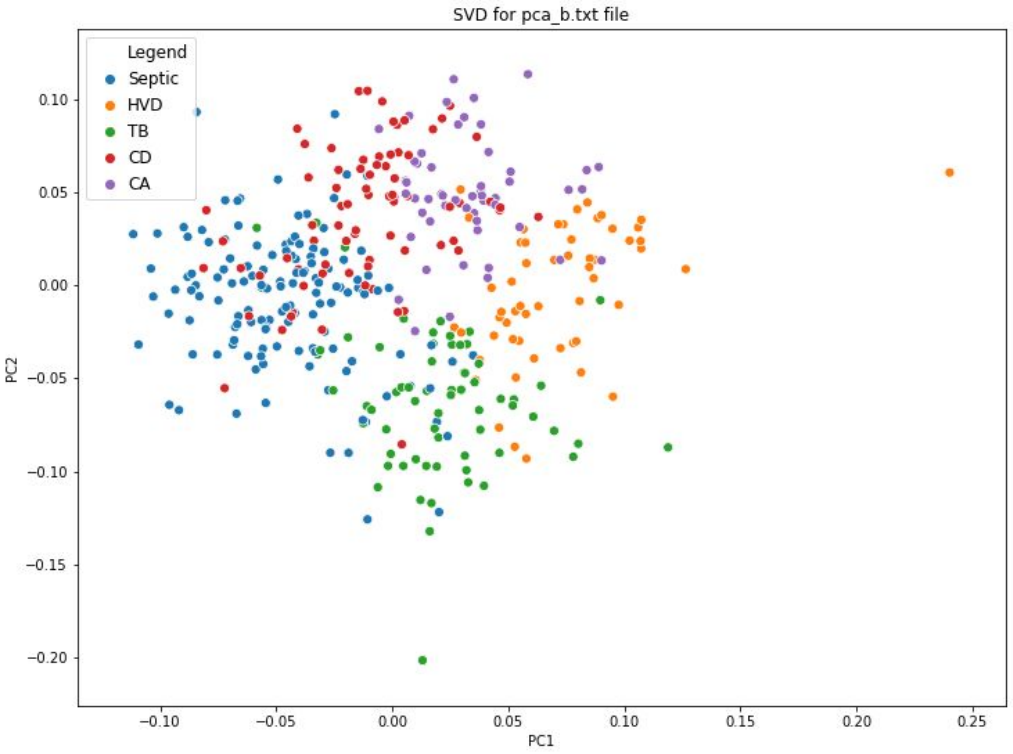


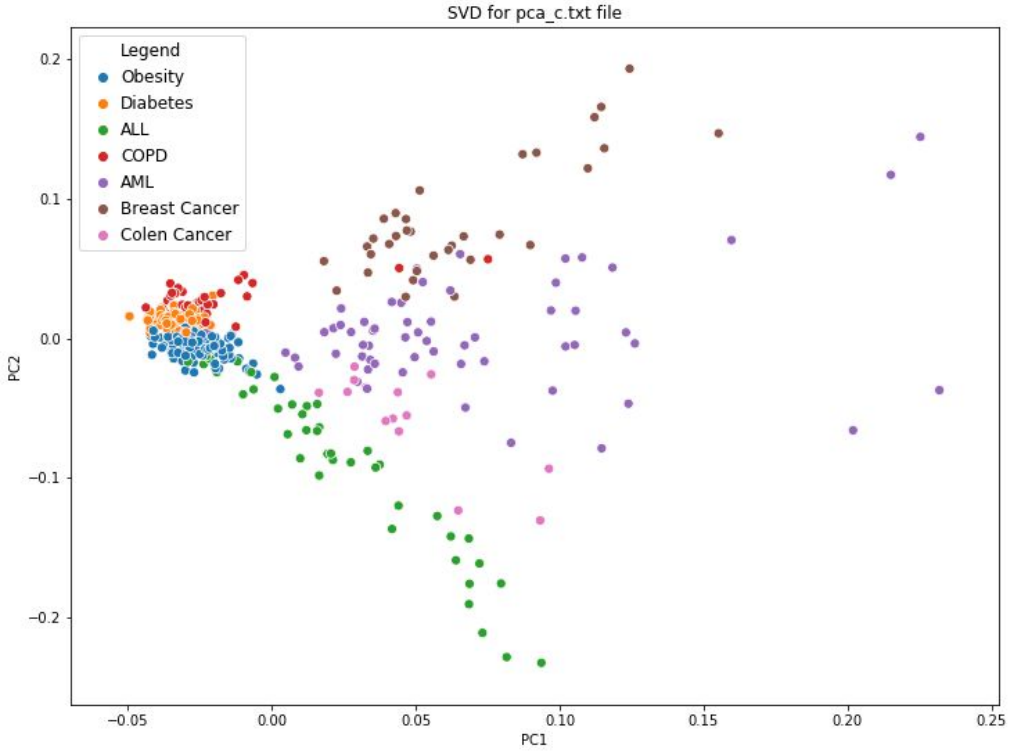




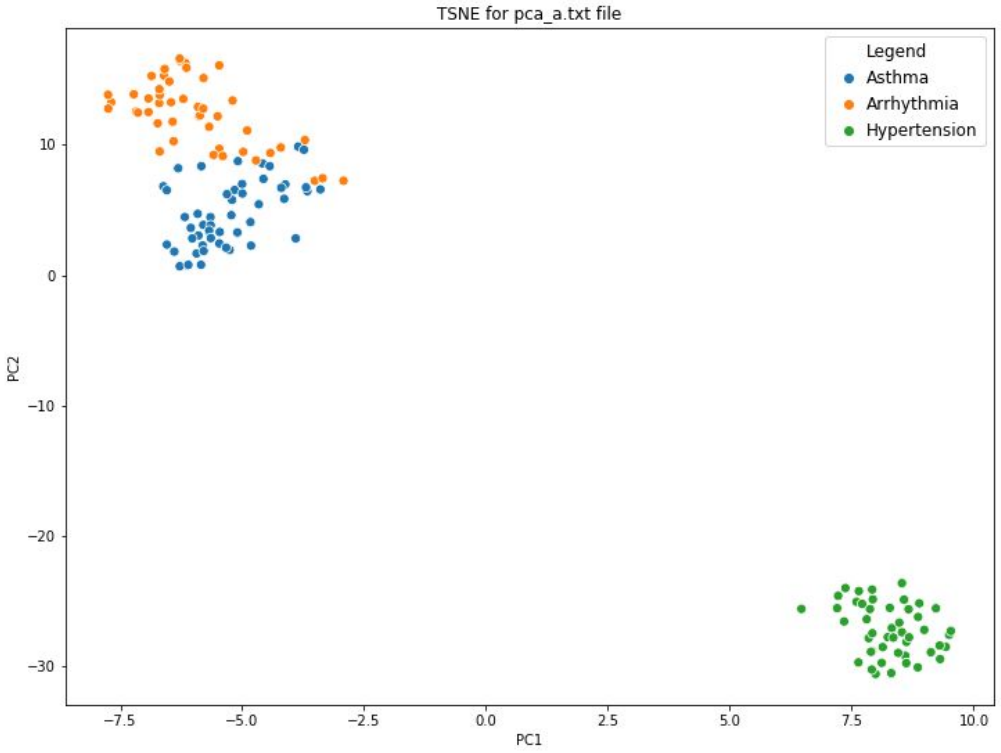
Singular Value Decomposition (SVD) Algorithm

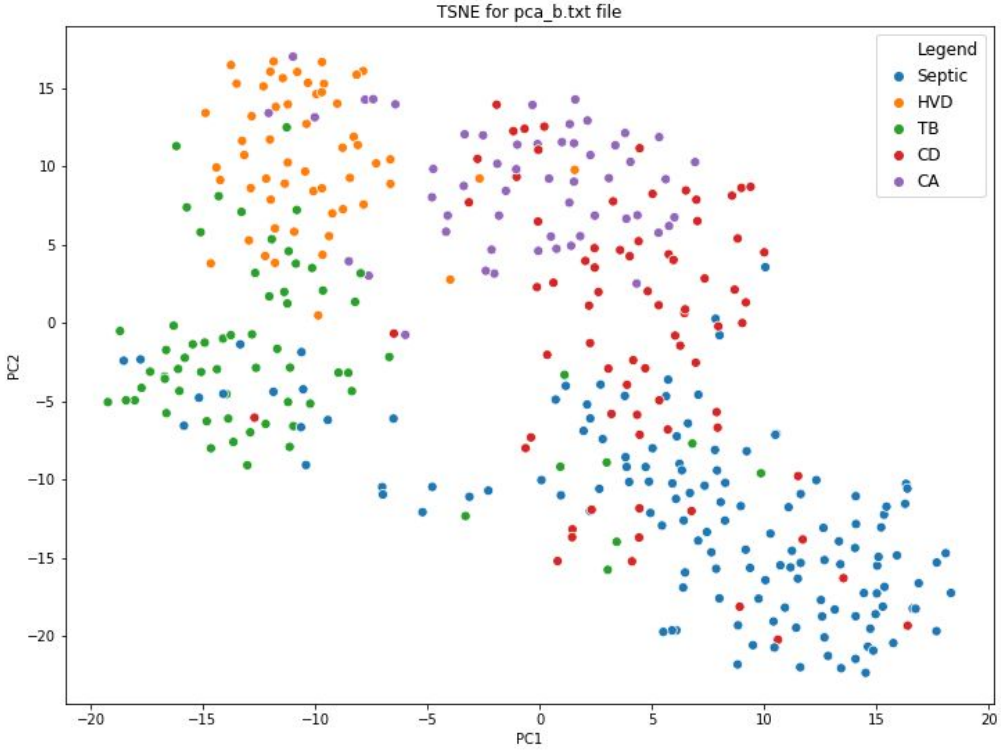


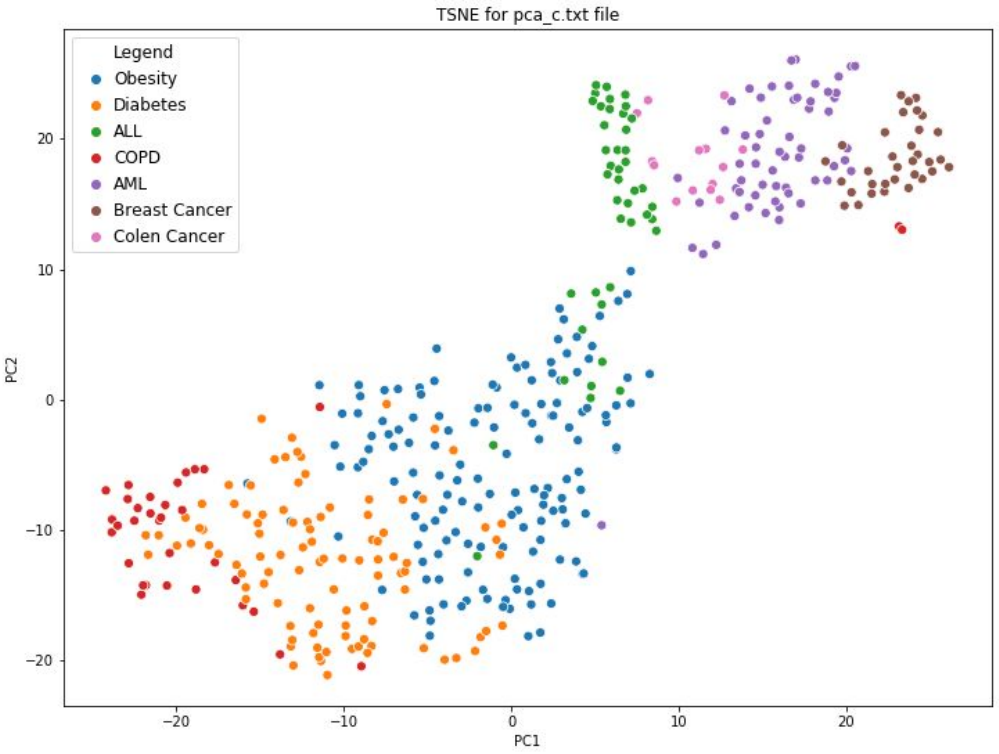

SVD for pca_b.txt file
Legend
Septic
HVD
TB
CD
CA
PC2
PC1
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.10
−0.05
0.00
0.05
0.10
0.15
0.20
0.25


SVD for pca_c.txt file
Legend
Obesity
Diabetes
ALL
COPD
AML
Breast Cancer
Colen Cancer
PC2
PC1
0.2
0.1
0.0
−0.1
−0.2
−0.05
0.00
0.05
0.10
0.15
0.20
0.25

t-Distributed Stochastic Neighbor Embedding() Algorithm:






TSNE for pca_c.txt file
Legend
Obesity
Diabetes
ALL
COPD
AML
Breast Cancer
Colen Cancer
PC2
PC1
20
10
0
−10
−20
−20
−10
0
10
20