CSE-601: Data Mining and Bioinformatics

Project 1.1

Dimensionality Reduction

SUBMITTED BY:
DEBANJAN PAUL (50208716)
JAY BAKSHI (50206954)
SHIVAM GUPTA (50206323)

Objective:

The aim of this project is to perform dimensionality reduction of the given data set by using three methods viz., Principal Component Analysis, Singular Value Decomposition and t-Distributed Stochastic Neighbor Embedding. In some datasets, there are too many attributes which results in a lot of noise and ambiguities. This makes it difficult to project datasets and make predictions and analysis. Dimensionality reduction helps in better representation of data, by reducing the number of variables in consideration, which helps in performing better data analysis operations.

Principal Component Analysis (PCA) combines different attributes linearly and tries to capture the original variance of the given data. The newly generated attributes are called Principal components, such that the first principal component holds the largest variance, the second principal component holds the second largest variance and so on.

Flow of PCA Algorithm:

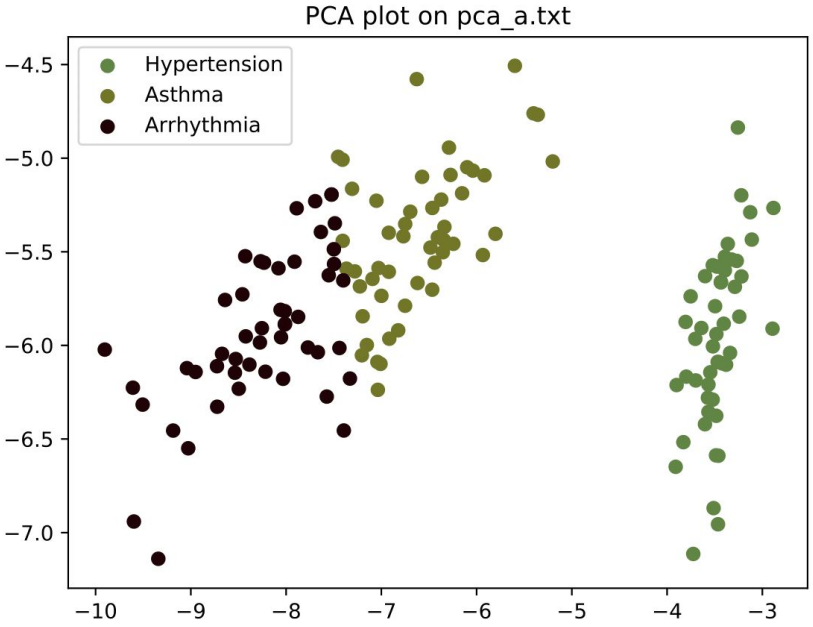
Packages used: Pandas (0.20.1), Numpy (v1.13.2), MatPlotLib
Here are the steps that we followed to implement PCA:

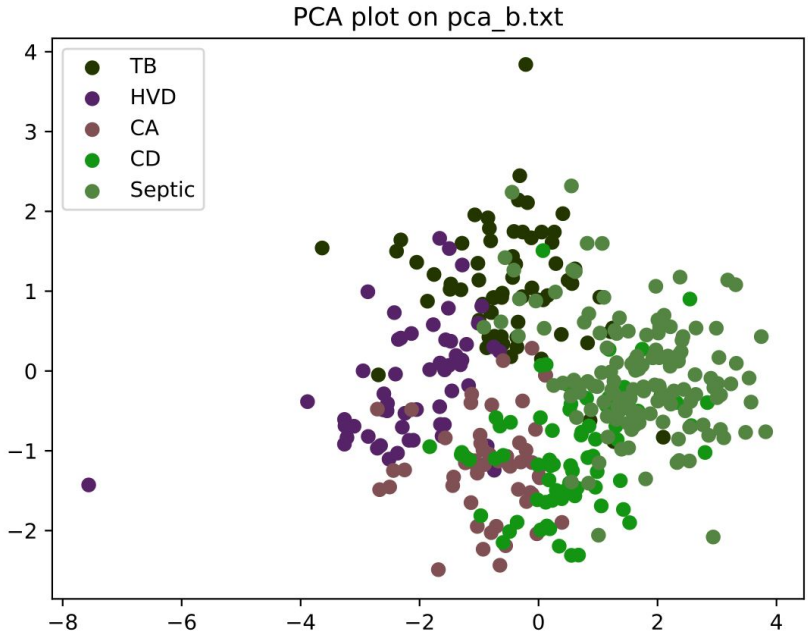
1. Read the .txt file name from Command Line Arguments.
2. Import the file, convert it into a numpy matrix and remove the last attribute/column.
3. Extract the labels from the last column into a list using pandas.
4. Compute the mean vector (by taking the mean of all rows) and normalize the numpy matrix using the following formulae:
   $X = x –$ (mean vector)

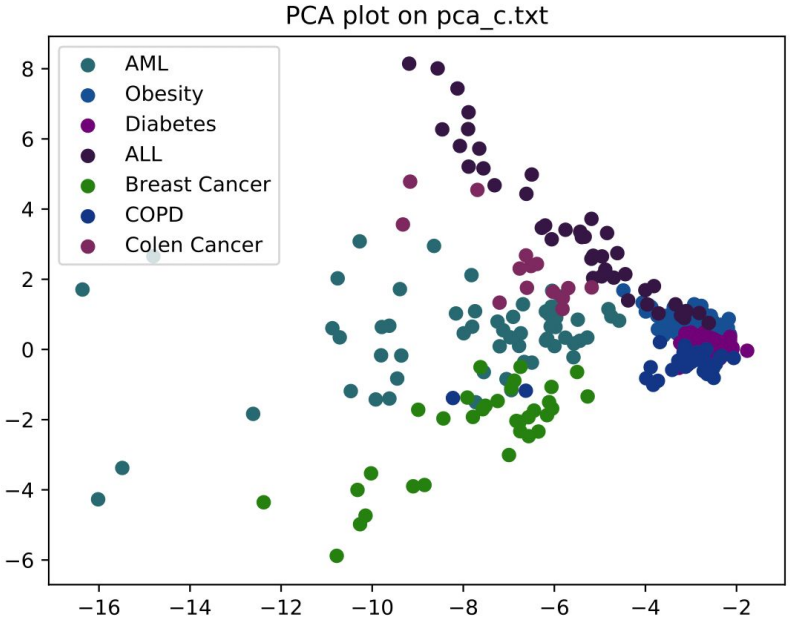
5. Calculate the Co-Variance of the resultant matrix which was obtained from the previous steps. We used following formulae to get the Co-Variance matrix:
   $S = [1 / (\text{Total number of rows})] * X * X^T$
6. Compute the Eigen Vectors from the Co-Variance Matrix S, using
   np.linalg.eig(S)
   function of numpy library.
7. The extracted Eigen vectors are stored in an increasing order of the Eigen Values. Therefore, we selected the first two columns of the Eigen vector matrix as the Principle Components.
8. For plotting purposes, we mapped the labels from Strings to Integers. Transform the resulting Principle Components and labels into a single Data frame.

Results:

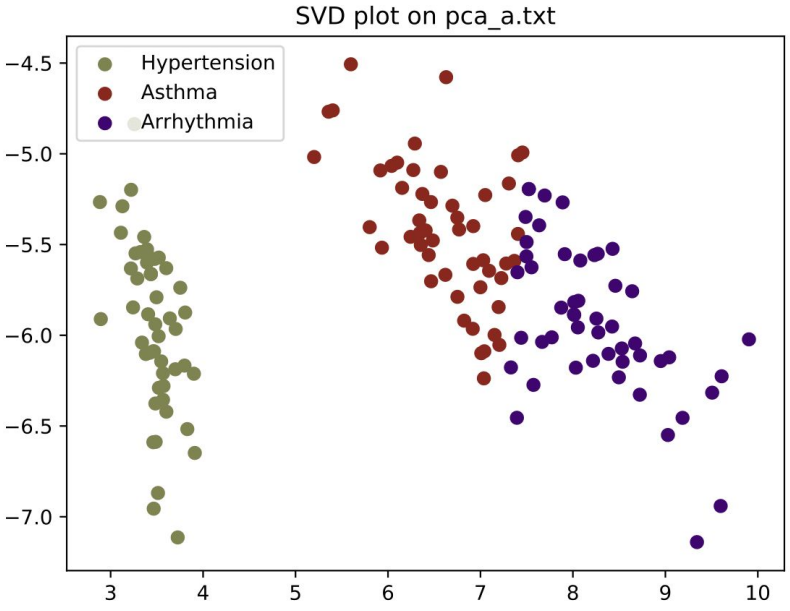
1. **PCA:**

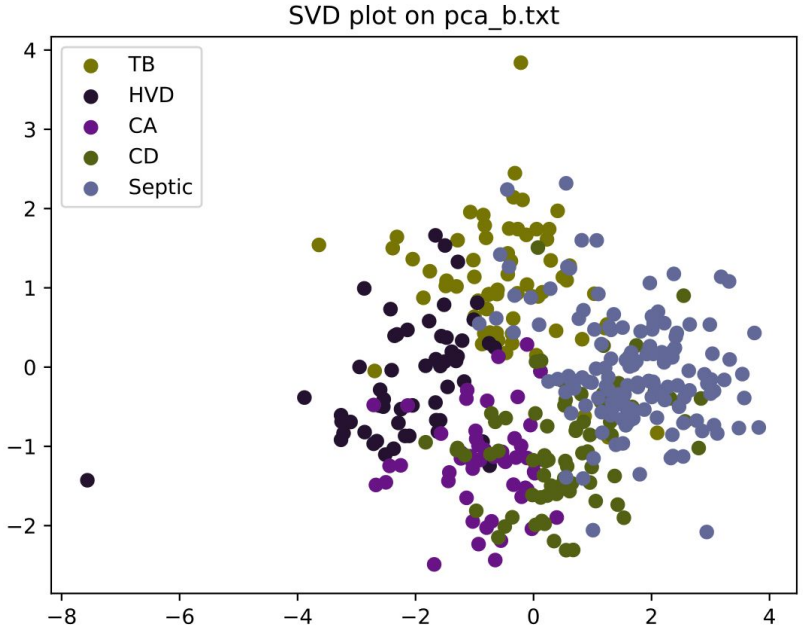


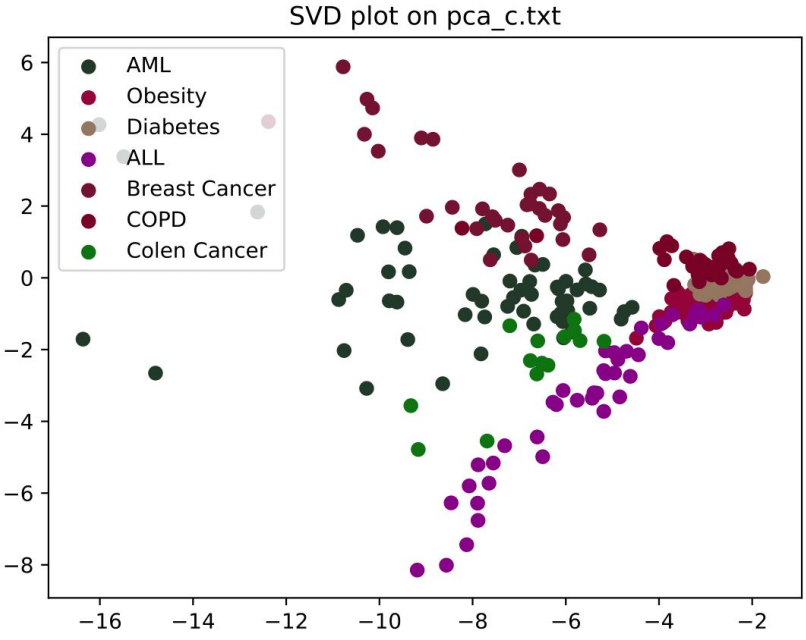



PCA plot on pca_c.txt
AML
Obesity
Diabetes
ALL
Breast Cancer
COPD
Colen Cancer
8
6
4
2
0
−2
−4
−6
−16
−14
−12
−10
−8
−6
−4
−2

2. SVD:


SVD plot on pca_a.txt
Hypertension
Asthma
Arrhythmia
−4.5
−5.0
−5.5
−6.0
−6.5
−7.0
3
4
5
6
7
8
9
10


SVD plot on pca_b.txt
TB
HVD
CA
CD
Septic
4
3
2
1
0
−1
−2
−8
−6
−4
−2
0
2
4


SVD plot on pca_c.txt
AML
Obesity
Diabetes
ALL
Breast Cancer
COPD
Colen Cancer
6
4
2
0
−2
−4
−6
−8
−16
−14
−12
−10
−8
−6
−4
−2

Explanation for the similarity between the graphs of SVD and PCA:
Singular value decomposition (SVD) and principal component analysis (PCA) dimensionality reduction methods which are based on computing the eigenvalues and eigenvectors, while retaining the important information. The formulas which are used to calculate the covariance matrix for the two methods are also similar:

PCA:

$$XX^T=WDW^T$$

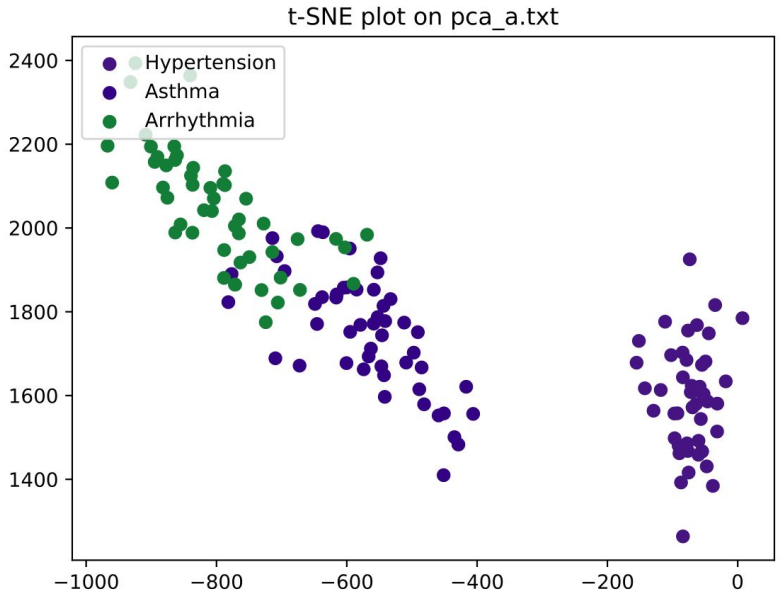
SVD:

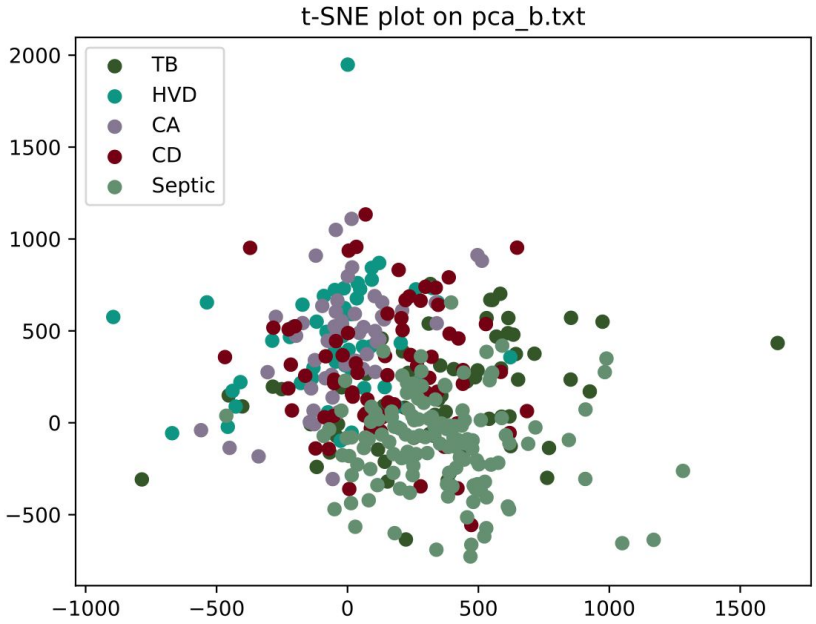
$$XX^T=U\,\Sigma^2 U^T$$

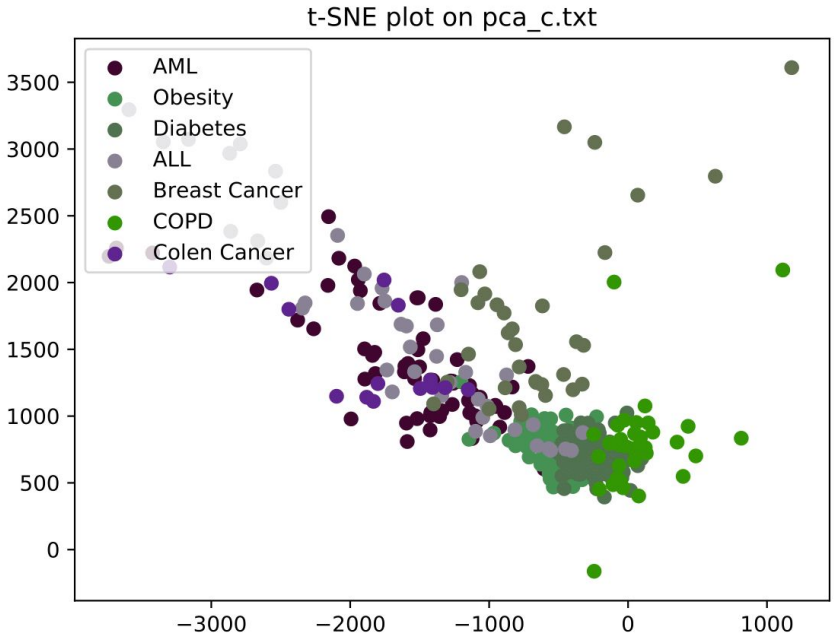
D and Σ are some constant values. W and U represents a data matrix or a dataframe.

Therefore, the end results for the two methods are also similar.

3. t-SNE:




t-SNE plot on pca_b.txt
TB
HVD
CA
CD
Septic
2000
1500
1000
500
0
−500
−1000
−500
0
500
1000
1500


t-SNE plot on pca_c.txt
AML
Obesity
Diabetes
ALL
Breast Cancer
COPD
Colen Cancer
3500
3000
2500
2000
1500
1000
500
0
−3000
−2000
−1000
0
1000

Comparison between t-SNE and PCA:

PCA uses mathematical formulas to compute linear combinations of attributes and to reduce the number of dimensions. It is using the correlation between some dimensions and tries to provide a minimum number of variables that keeps the maximum amount of variation or information about how the original data is distributed.
t-Distributed Stochastic Neighbor Embedding (t-SNE) is an another technique for dimensionality reduction and is particularly well suited for the visualization of high-dimensional datasets. Contrary to PCA it is not a mathematical technique but a probabilistic one.
The working of t-SNE can be explained as:
t-Distributed stochastic neighbor embedding (t-SNE) minimizes the divergence between two distributions: a distribution that measures pairwise similarities of the input objects and a distribution that measures pairwise similarities of the corresponding low-dimensional points in the embedding.

In simple words it looks at the original data and search to best represent this data using less dimensions by matching both distributions. This method can be used to show more clear variations in data as compared to PCA but it is computationally heavy.
The average time complexity is $O(n^2)$. Therefore, in case of very high dimensional data, we may need to apply another dimensionality reduction technique before using t-SNE.