**CSE 601 – Data Mining and Bioinformatics**

**Project 1: Dimensionality Reduction & Association Analysis**

**Team members:**

Kiran Ketan Nevrekar, 50336915, kiranket

Revathy Narayanan, 50336857, revathyn

Saranya Saravanan, 50314931, saravan2

**Part 1: Dimensionality Reduction**

**Objective:** The aim of this part of the project is to implement dimensionality reduction techniques like Principal Component Analysis, Singular Value Decomposition and T-Stochastic Neighbor Embedding to reduce larger dataset to smaller valuable dataset so that it can be used for meaningful data interpretation.

**Principal Component Analysis:** Principal Component Analysis is a dimensionality reduction technique that is used to reduce dimensionality of large datasets by transforming a large set of attributes to a smaller set of attributes that contains valuable information.

**PCA implementation**

**The flow of PCA code implementation is as follows:**

1. A list named 'files' is created to contain paths for all the three data files.
2. Using pandas the data is converted into a dataframe which is then used throughout the code.
3. The last column in the dataset is the label. Therefore we store it in a separate array variable ‘disease’ and the rest to a variable ‘x’. Now dimensionality reduction is to be performed to this data in ‘x’.
4. The data points in ‘x’ are normalized using the min max normalization which is then used to generate mean adjusted dataset.
5. This mean adjusted dataset is used to find covariance matrix. Co-variance provides the strength between two or more variables.
6. Using the resultant covariance matrix, eigendecomposition of the matrix is calculated which produces a list of eigenvalues and eigenvectors.
7. Computing Eigenvectors from the resultant covariance matrix by passing the covariance matrix in the ‘np.linalg.eig()’ function. Before plotting the values for visualization, complex numbers are removed leaving only with real numbers.
8. Using plotly,the transformed two dimensional data are visualized in the form of scatter plots for all the given 3 text files. (Refer figure: 1,2 and 3 for the result)

**Packages used for PCA:** Pandas, numpy, plotly

**Singular Value Decomposition**

Singular Value Decomposition is a matrix decomposition method used to reduce matrices thereby making the calculations simpler.

**SVD Implementation:**

1. A list named 'files' is created to contain paths for all the three data files.
2. Using pandas the data is converted into a dataframe which is then used throughout the code.
3. The last column in the dataset is the label. Therefore we store it in a separate array variable ‘disease’ and the rest to a variable ‘x’. Now dimensionality reduction is to be performed to this data in ‘x’.
4. The data points in ‘x’ are normalized using the min max normalization which is then used to generate mean adjusted dataset.
5. The resultant data is then used to fit and transform. Transformation was performed using the inbuilt function **linalg.svd().**
6. Only the first two components are stored from the above step as we require only 2 dimensional data points.
7. Performed visualization using plotly.

**Packages used for SVD:** pandas, numpy, plotly, linalg

**T-Distributed Stochastic Neighbor Embedding:**

T-Distributed Stochastic Neighbor Embedding is a dimensionality reduction method which is used to analyze higher dimensional data. It gives a gist of how the data is arranged in higher dimensional space.

**T-SNE implementation:**

1. A list named 'files' is created to contain paths for the three data files
2. Using pandas the data is converted into a dataframe which is then used throughout the code.
3. Assigning last column to the variable ‘disease’ and the rest to a variable ‘x’
4. Using numpy , the data points in x are normalized.
5. The resultant data is then used to fit and transform using **tsne()** function from sklearn package by specifying the number of reduced components.
6. Performed visualization using plotly and the resultant data points.

**Packages used for T-SNE:** pandas, numpy, plotly, linalg, TSNE

**Results Screenshots:**

**Principal Component Analysis:**

1. **pca\_a.txt**

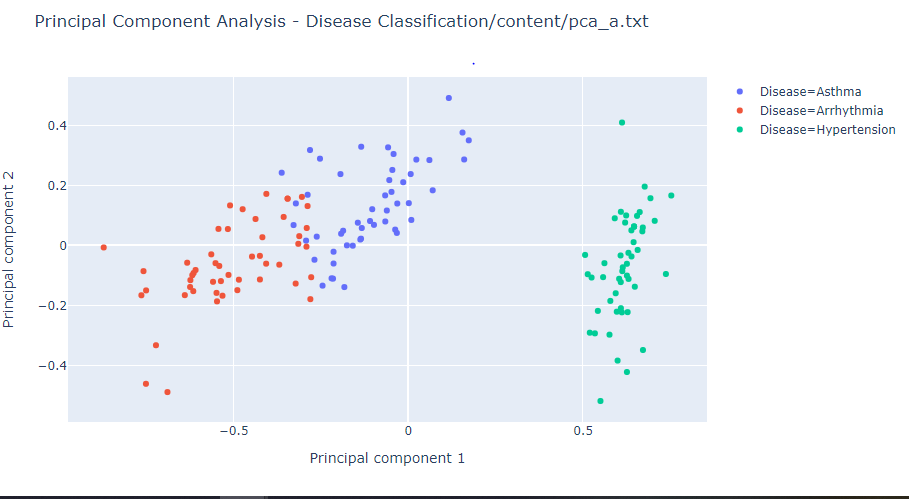
****

Figure:1

1. **pca\_b.txt**

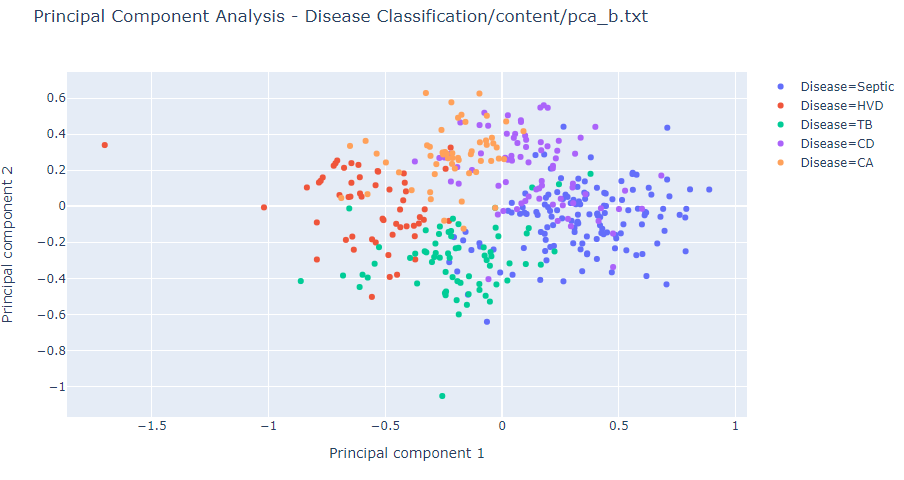
****

Figure:2

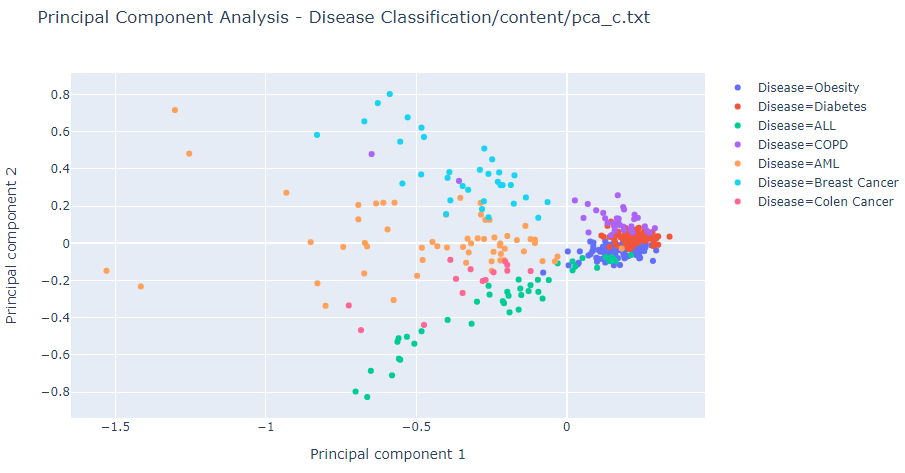
1. **pca\_c.txt **

Figure:3

**Singular Value Decomposition:**

1. **SVD on pca\_a.txt:**

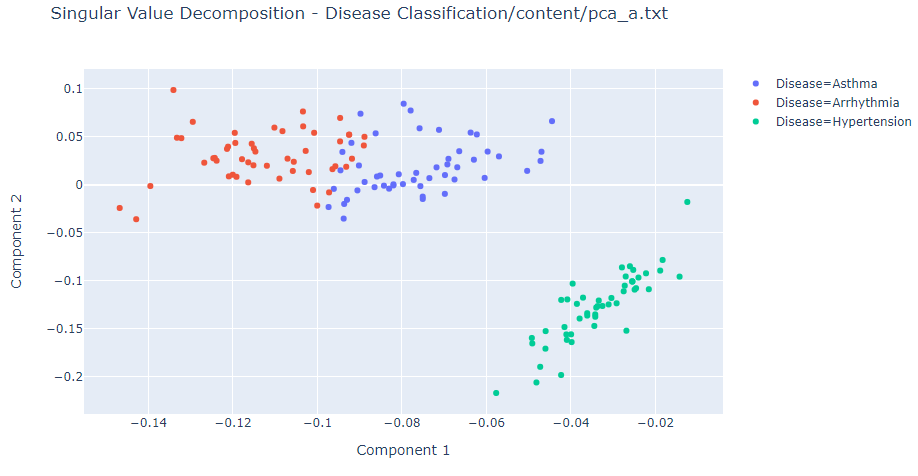
****

Figure:4

1. **SVD on pca\_b.txt:**

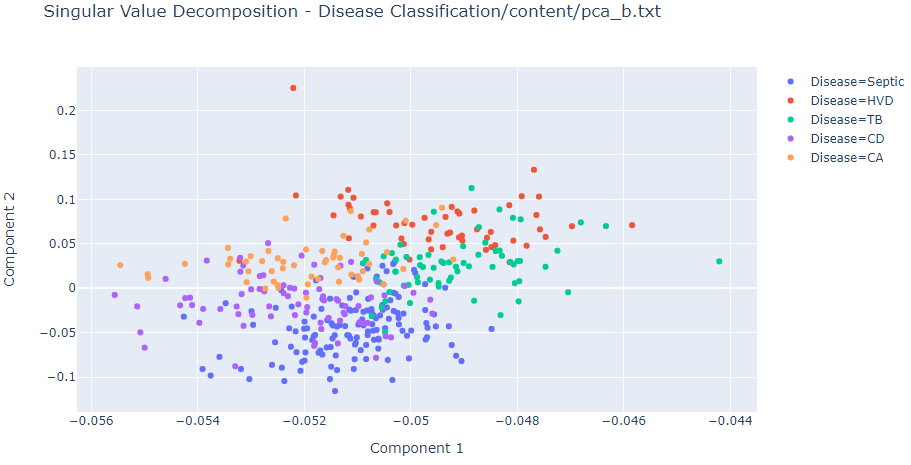
****

Figure:5

1. **SVD on pca\_c.txt:**

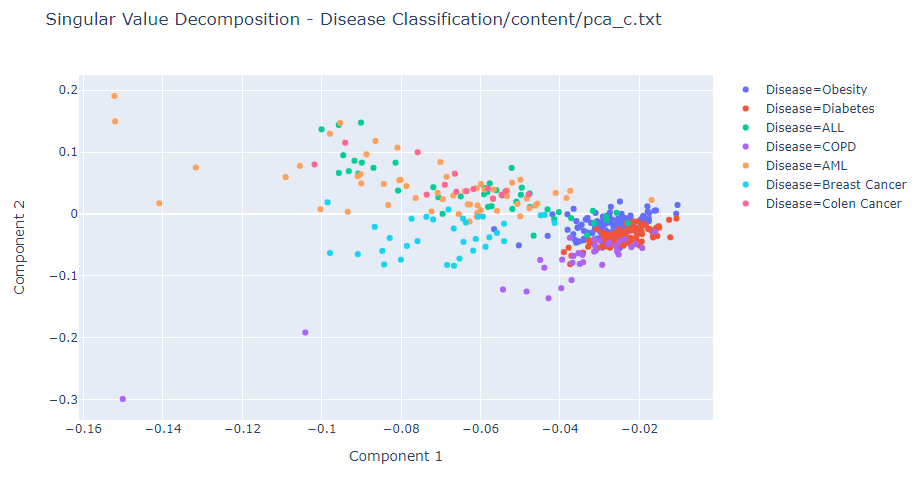
****

Figure:6

**T-Distributed Stochastic Neighbor Embedding:**

1. **T-SNE on pca\_a.txt:**

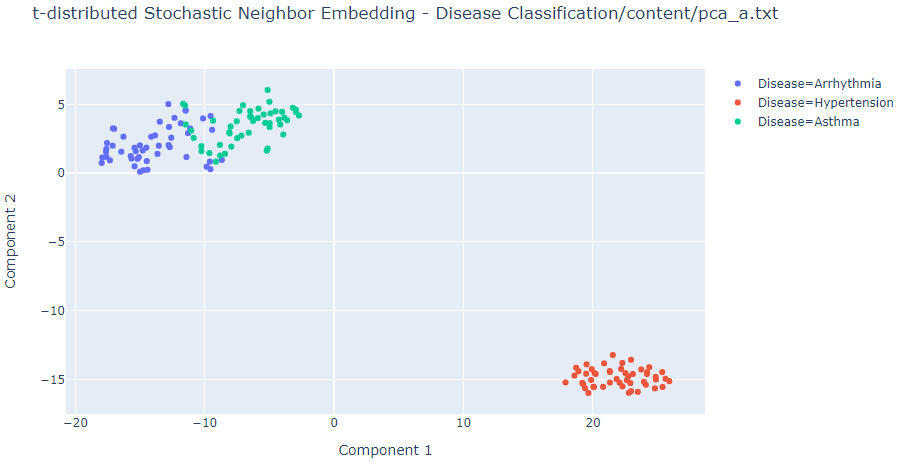
****

Figure:7

**T-SNE on pca\_b.txt:**

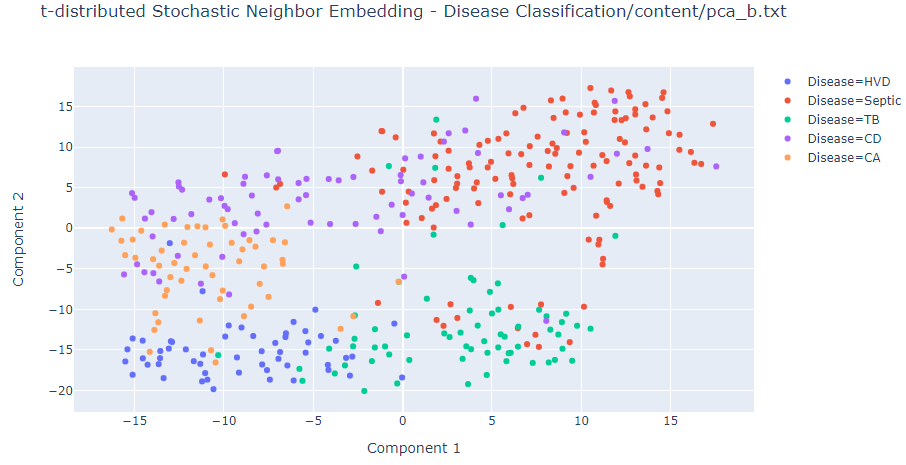
****

Figure:8

**T-SNE on pca\_c.txt:**

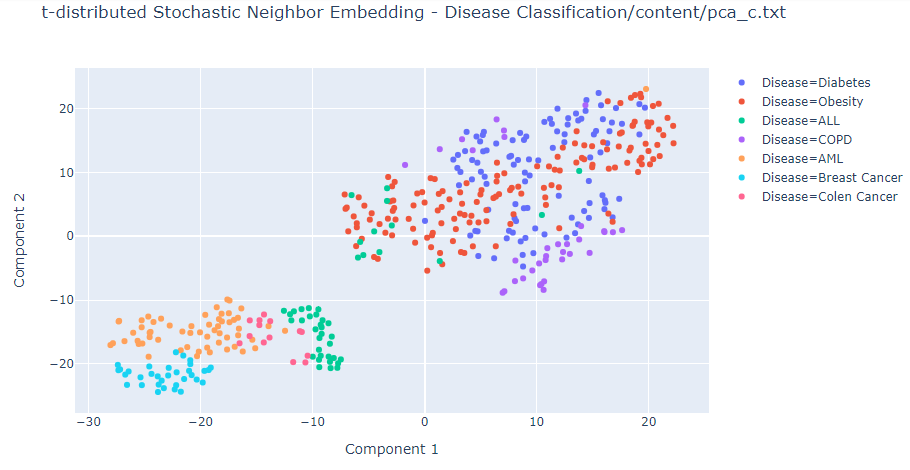
****

Figure:9

**Result:**

Even though Principal Component Analysis, Singular Value Decomposition and T- Stochastic Neighbor Embedding are used for dimensionality reduction, they differ in the way they deliver their result.

Their ability to project data in a higher dimensional space differs from each other as in some cases one gives better results than the other.

Here, T-SNE did a better job than PCA and SVD on separating the clusters and representing them on higher dimensional space. T-SNE differs from PCA by preserving only small pairwise distances or local similarities whereas PCA is concerned with preserving large pairwise distances to maximize variance.

When compared, the distance between the same cluster points seems smaller in T-SNE than the others. However, T-SNE takes longer time than the rest while computing the output.