Principal Component Analysis Report

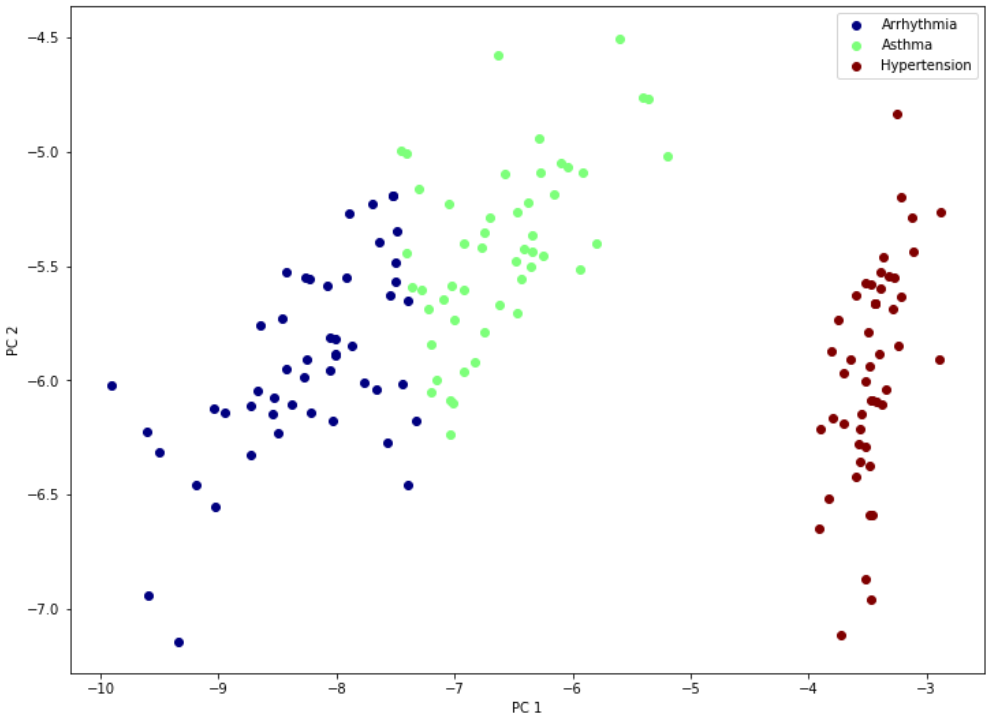
Team members

Mitali Vijay Bhiwande
Sumedh Sadanand Ambokar
Tejasvi Balaram Sankhe

SCATTER PLOTS:

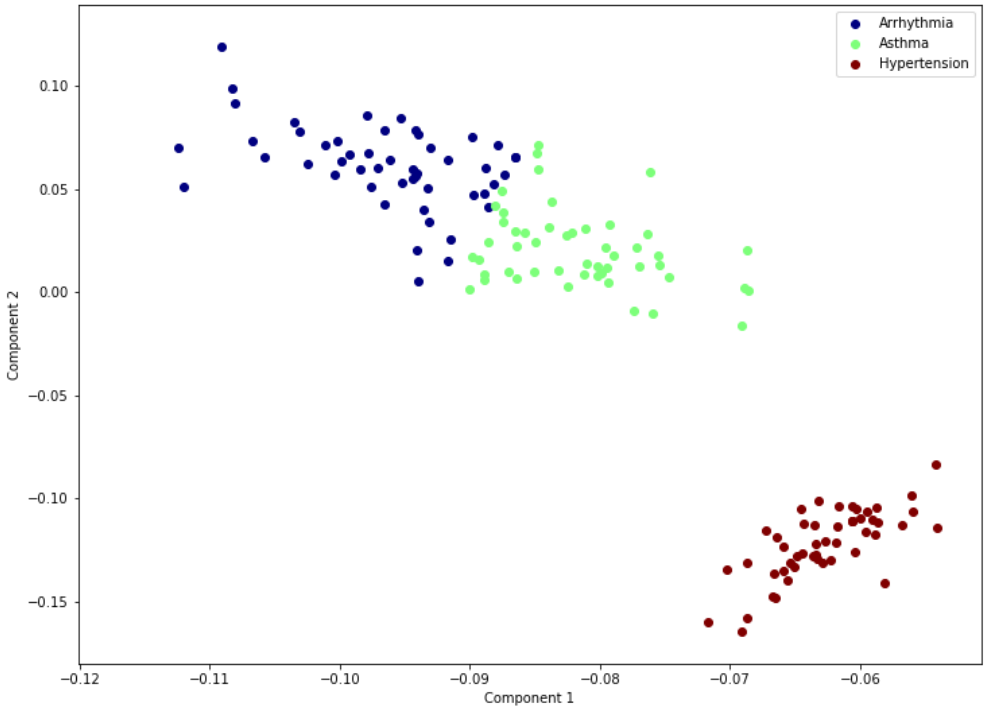
1. **File Name:** **pca_a.txt**
   Diseases: **Arrhythmia, Asthma, Hypertension**

   Algorithm: PCA



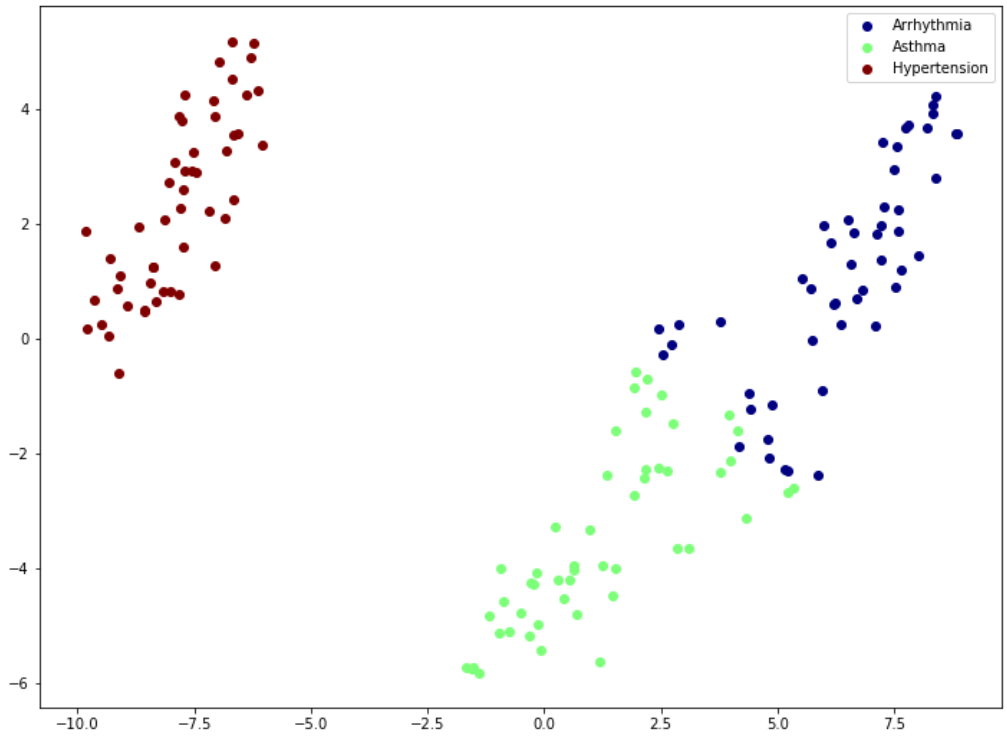
PCA: pca_a.txt

   Algorithm: SVD



SVD: pca_a.txt

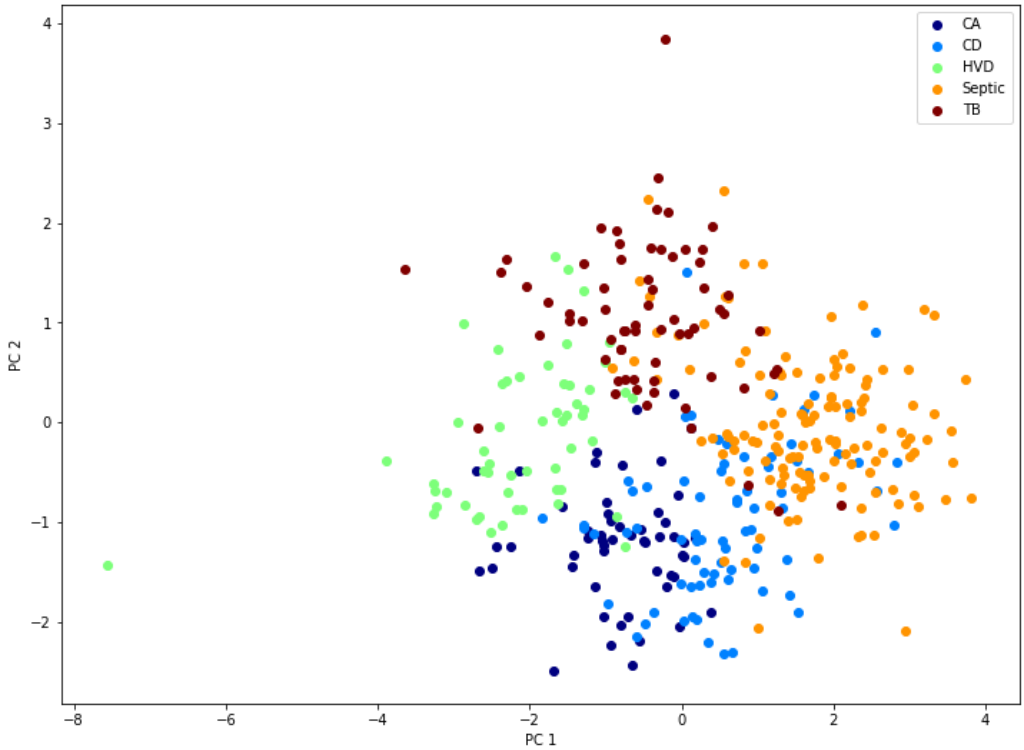
Algorithm: TSNE



TSNE: pca_a.txt

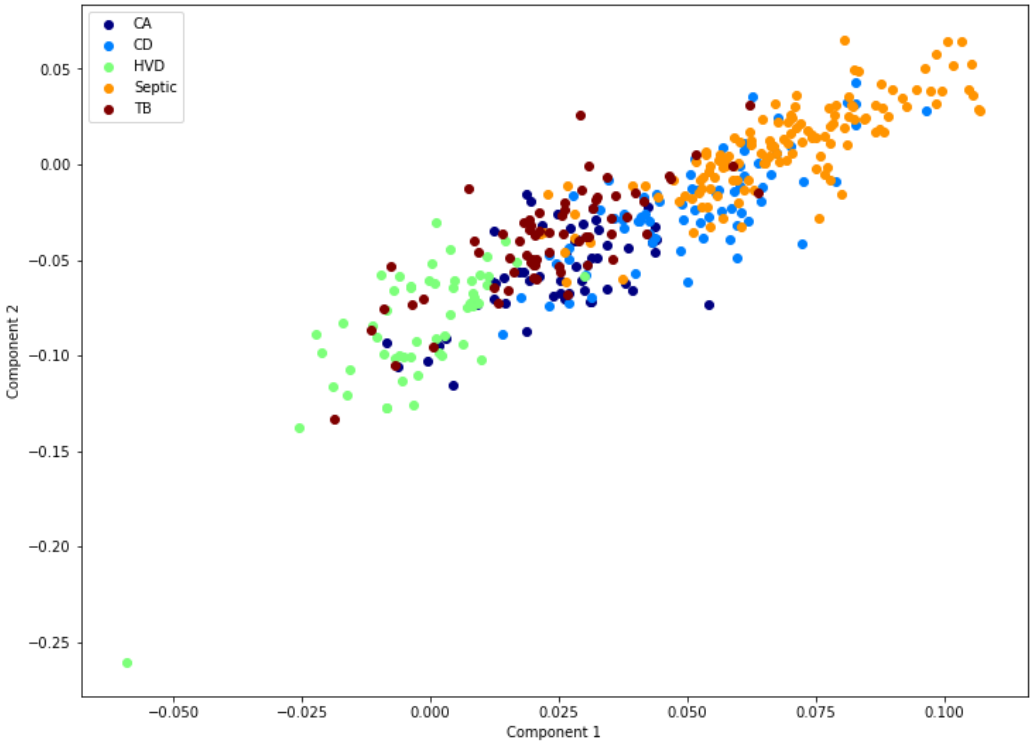
2. **File Name: pca_b.txt**
   Diseases: CA, CD, HVD, Septic, TB

Algorithm: PCA



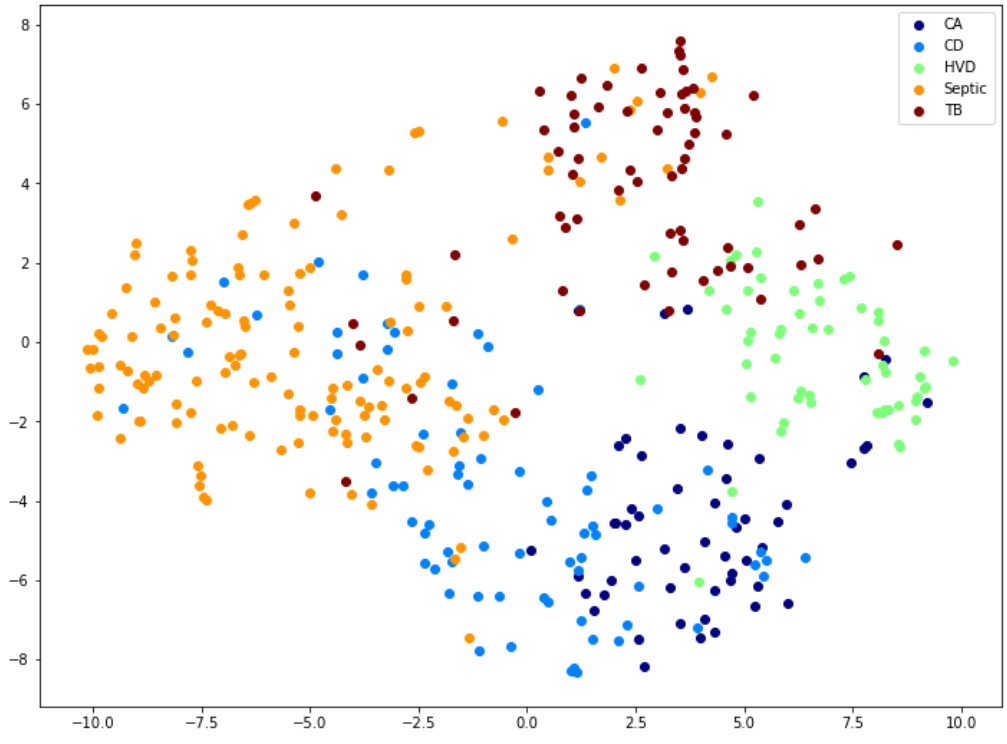
PCA: pca_b.txt

Algorithm: SVD



SVD: pca_b.txt

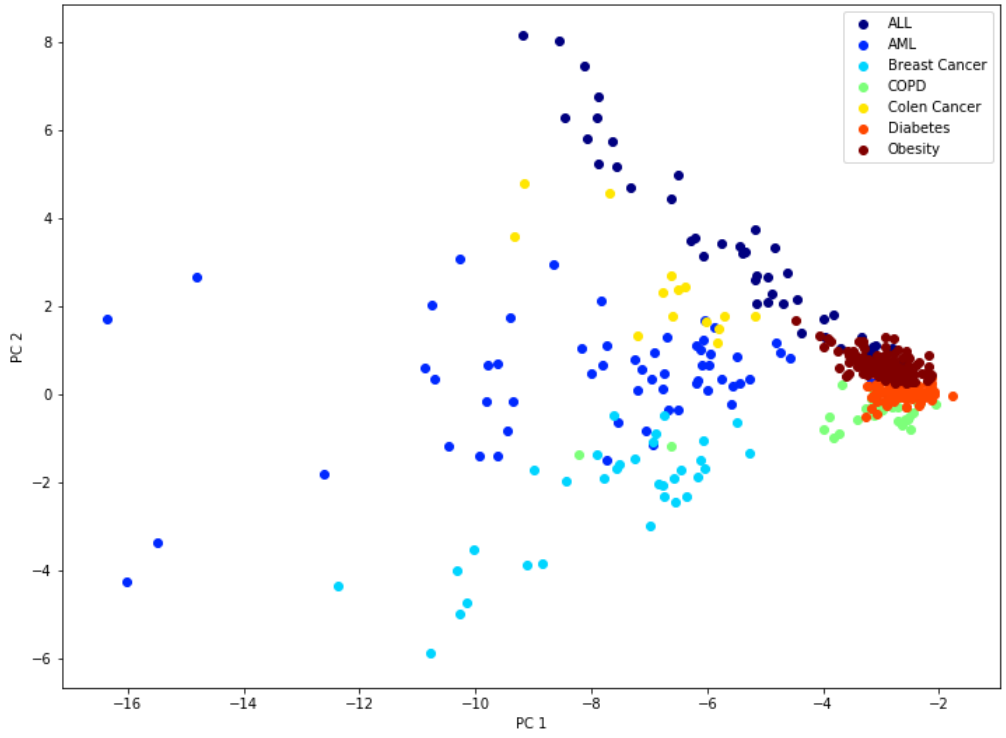
Algorithm: TSNE



TSNE: pca_b.txt

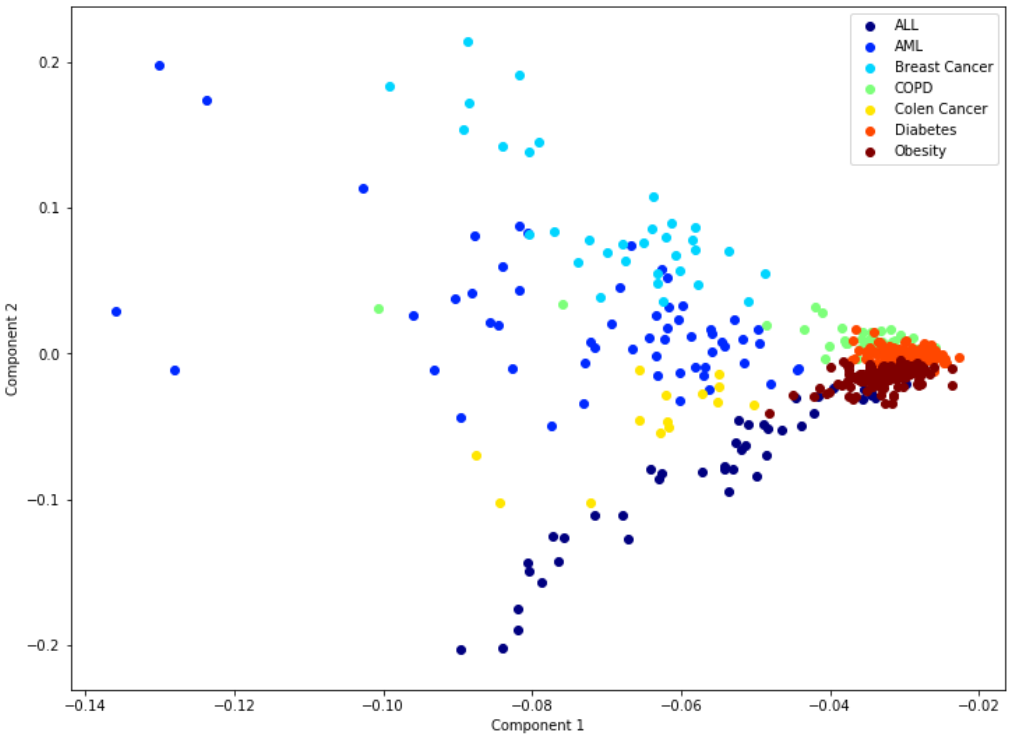
3. **File Name: pca_c.txt**
   Diseases: ALL, AML, Breast Cancer, COPD, Colen Cancer, Diabetes, Obesity

Algorithm: PCA



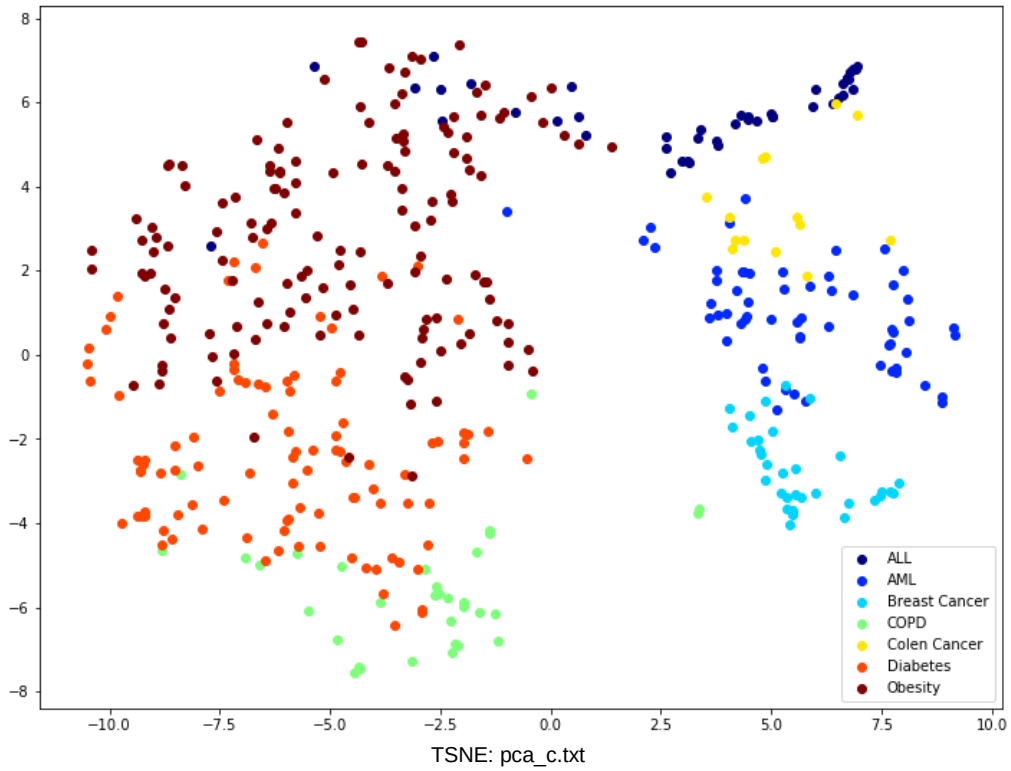
PCA: pca_c.txt

Algorithm: SVD



SVD: pca_c.txt

Algorithm: TSNE



TSNE: pca_c.txt

For PCA,

- Implemented PCA algorithm is used to obtain new 2-Dimensional co-ordinates of the original attributes.
- Each point is colored based on the disease it represents in the provided data.
- The scatter plot signifies the two principal components with the maximum variation.

For SVD,

- The original attributes are considered along with their diseases to plot the SVD result and color the representing points.
- Here, the components 1 and 2 signify the largest variations.
- Numpy's linear algebra package is used for computing the singular value decomposition of original attributes.

For TSNE,

- TSNE from sklearn.manifold package is used for implementing TSNE algorithm.
- Number of iterations are set to 1000 to optimize the resulting clusters.
- Initial embedding is set to 'pca' for more stable global initialization.
- Perplexity with values like 30, 40, 50 and 60 were tried but well defined clusters were obtained at the default value of 30.
- Similarly, learning_rate is set to 100 for obtaining well defined clusters.
- For fitting the results in 2-Dimensional space, n_components is set to 2.
- The resulting TSNE plot signifies the clusters corresponding to the diseases in the data set.

PCA Implementation:

1. Initially the data is split into two lists one representing the attributes and the other representing the diseases. This is done using the function, fetch_attributes() which takes the original dataset as input and returns the disease array and list of respective attributes.
2. **Steps for PCA Computation:**
   i. Mean is computed for all the attributes by taking the mean of all rows.
   ii. Original attributes are adjusted by subtracting the mean from them.
   iii. Covariance is computed as a dot product between the transpose of original attributes and the original attributes over the total number of attributes.
   iv. Eigen vectors and eigen values are computed using the above covariance matrix.
   v. Top 2 eigen vectors corresponding to largest variances each representing the principal components, are selected and new attributes are computed as the dot product of original attributes and top eigen vectors.
3. Obtained new co-ordinates are displayed across 2-Dimensions with the help of scatter plot.

```
def plot_pca(original_attrs, disease_array):
    attrs_mean = original_attrs.mean(axis=0)
    adjusted_attrs = original_attrs - attrs_mean
    covariance = np.dot(np.transpose(adjusted_attrs),adjusted_attrs)/len(adjusted_attrs)
    w, v = LA.eig(covariance)
    top_eigen_vectors = v[:,0:2]
    new_coordinates = np.dot(original_attrs,top_eigen_vectors)
    draw_scatter_plot(new_coordinates[:,0:1],new_coordinates[:,1:2],disease_array)
```

Fig 1. Function to compute and display PCA

4. Similarly, using the existing packages, SVD and TSNE algorithms are implemented to convert the high dimensional data into 2-Dimensional data and their respective scatter plots are displayed.

Discussion:

- PCA compactly represents the ways original data deviates from the mean.
- Thus, PCA corresponds to centering the dataset and then rotating it to obtain points with maximum variance as the top principal components.
- SVD corresponds to compactly summarizing the data and the way it deviates from zero.
- If mean centered data is used for SVD computation, the results will be similar to that of PCA and the plots will be same.
- TSNE provides well separated clusters corresponding to each disease by reducing the dimensions using probability distribution.
- In contrast to PCA and SVD, TSNE focuses more on nearest neighbor accuracy.

Existing Packages used:

i. numpy:
   - To convert dataset into array.

ii. linalg from numpy :
   - To compute eigen vectors and eigen values.
   - Also, used in SVD computation.

iii. TSNE from sklearn.manifold :
   - To implement TSNE algorithm.