CSE 601 : Data Mining and Bioinformatics

Project 1 PCA Report

Rohan Hemanshu Sheth

Niranjan Deshpande

Implementation details:

Principal Component Analysis (PCA):

· We read the input file, say pca_a.txt in python in a dataframe and save the last column containing class labels in a list which is required for plotting. We save the remaining columns in a two-dimensional numpy array.

· Next, we find the mean vector of all the columns in the array consisting of raw data and subtract the means from data in their corresponding columns.

· Then we calculate the covariance matrix as follows:

```
cov_mat = ((pcaa - mean_vec).T.dot(pcaa - mean_vec)) / (pcaa.shape[0]-1)
```

· Using the numpy function np.linalg.eig(cov_mat), we find eigenvectors and eigenvalues of the covariance matrix and sort eigenvalues in decreasing order. We only need to select the top two eigenvalues and their corresponding eigenvectors to reduce the dimensionality of our data to two features or principal components where eigenvalues represent the variance.

· Then we calculate the dot product of the array containing raw data (dimensions mxn) with a matrix containing the two eigenvectors (dimension nx2) to get the data with a reduced dimensionality of two features (dimension mx2).

· Then we use matplotlib.pyplot to find the scatterplot of the reduced dimensions (P1 and P2) to visualize the data with unique colors corresponding to respective unique class labels.

.

Singular Value Decomposition (SVD):

· For SVD, we use the scikitlearn library TruncatedSVD to perform dimensionality reduction as follows:

```
from sklearn.decomposition import TruncatedSVD
tsvd = TruncatedSVD(n_components=2)
tsvd.fit(pcaa)
result = tsvd.transform(pcaa)
```

t-Distributed Stochastic Neighbor Embedding (t-SNE):

· Similarly, we use the TSNE library from sklearn.manifold to perform dimensionality reduction:

```
from sklearn.manifold import TSNE
result =
TSNE(n_components=2,random_state=0).fit_transform(pcaa)
```

We have used random_state=0 as a seed for reproducibility.

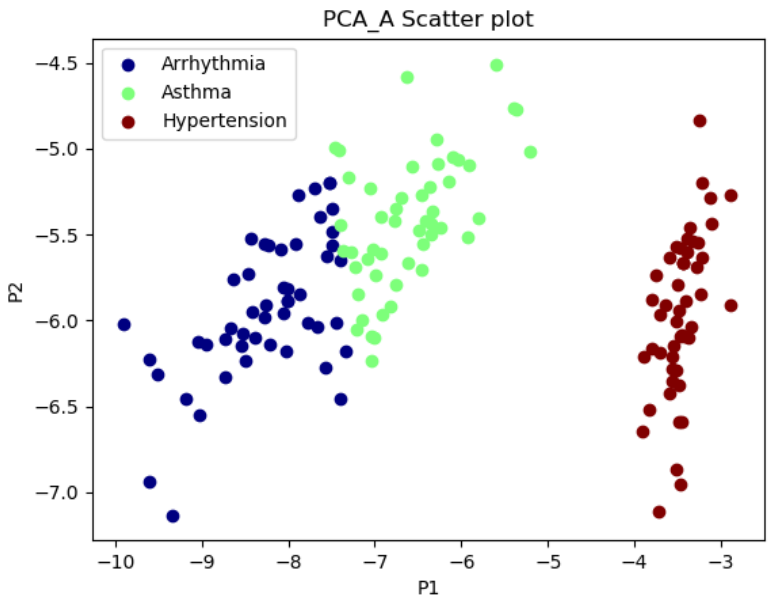
Discussion:

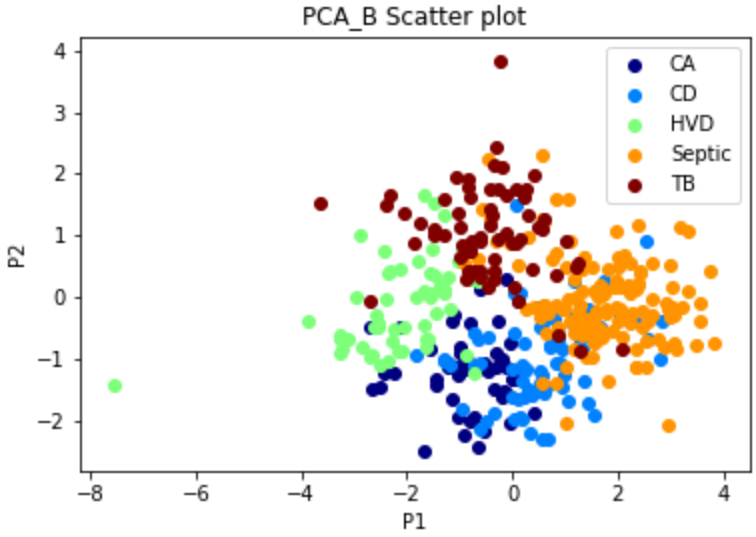
SVD and PCA are both linear transformation methods of dimensionality reduction, SVD is a matrix factorization method that can be used to calculate PCA. Calculating PCA involves eigen decomposition of covariance matrix while SVD involves singular value decomposition of the data matrix. Singular values of the data matrix are square roots of eigenvalues of the covariance matrix. Their results are related. In many cases, working on data matrix is more efficient and SVD can produce more accurate results. t-SNE is a non-linear technique for dimensionality reduction that is more suited for visualization of high dimensional datasets. While PCA and SVD are mathematical techniques, t-SNE is probabilistic. With same hyperparameters, it can produce different results on different runs. It is mainly a data exploration and visualization technique and it is

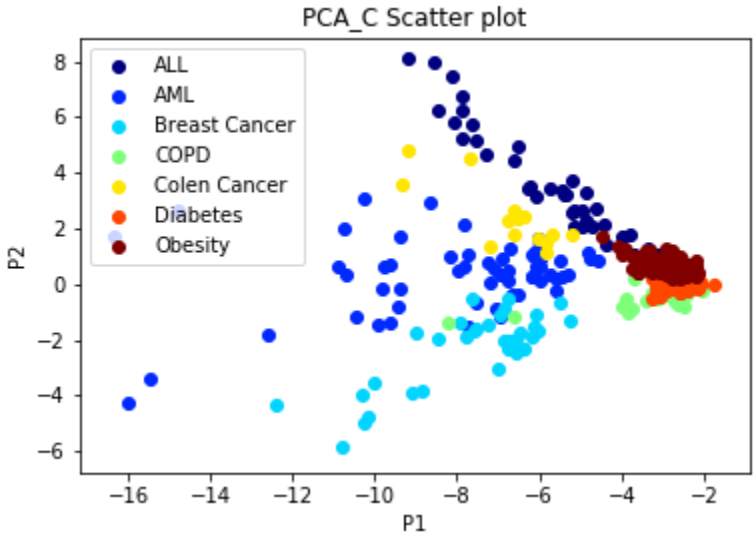
difficult to make an inference based on its output. However, it represents similar data points close together.

Plots:

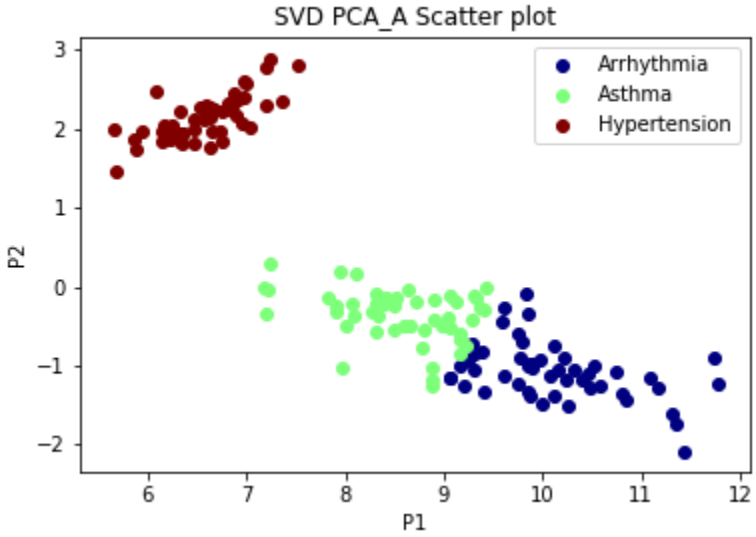
For PCA:

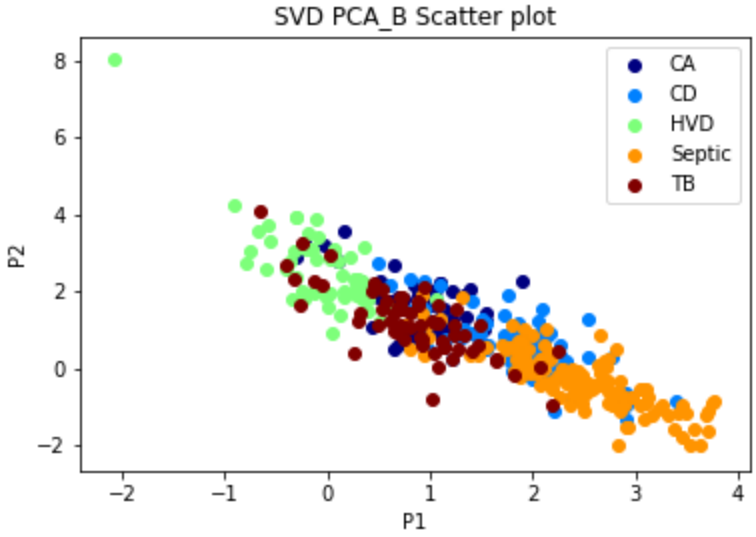


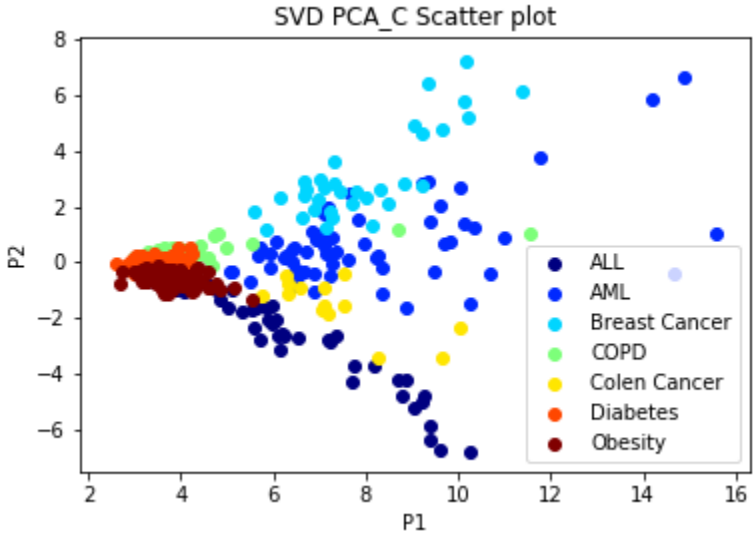

PCA_B Scatter plot
CA
CD
HVD
Septic
TB
P1
P2


PCA_C Scatter plot
ALL
AML
Breast Cancer
COPD
Colen Cancer
Diabetes
Obesity
P1
P2

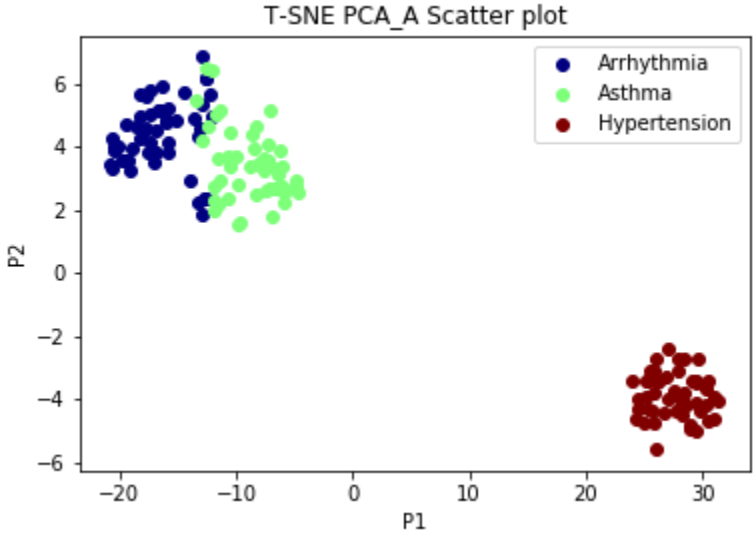
For SVD:

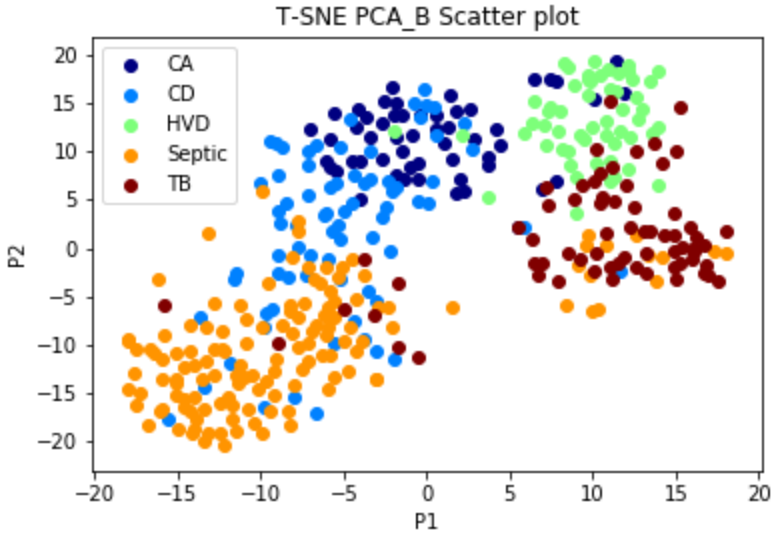

SVD PCA_A Scatter plot
Arrhythmia
Asthma
Hypertension
P1
P2

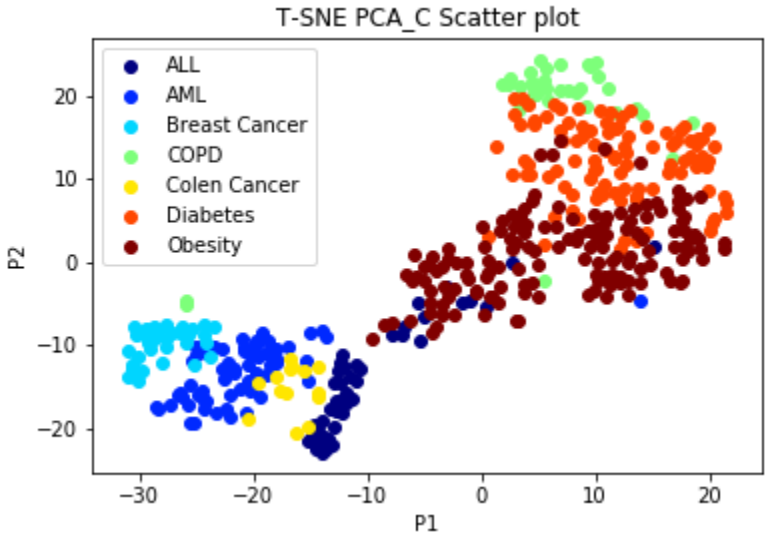

SVD PCA_B Scatter plot
CA
CD
HVD
Septic
TB
P1
P2


SVD PCA_C Scatter plot
P1
P2
ALL
AML
Breast Cancer
COPD
Colen Cancer
Diabetes
Obesity

For t-SNE:


T-SNE PCA_A Scatter plot
P1
P2
Arrhythmia
Asthma
Hypertension


T-SNE PCA_B Scatter plot
CA
CD
HVD
Septic
TB
P1
P2


T-SNE PCA_C Scatter plot
ALL
AML
Breast Cancer
COPD
Colen Cancer
Diabetes
Obesity
P1
P2