# CSE 601: DATA MINING AND BIOINFORMATICS

# FALL 2018

# PROJECT 1: PCA AND APRIORI ALGORITHM

# PART 1: DIMENSIONALITY REDUCTION

# REPORT


**SIDDHARTH SELVARAJ #50247317**

**ALLEN DANIEL YESA #50246827**

**ANUSH RAVINDRA SHETTY #50247204**

# PART 1: DIMENSIONALITY REDUCTION

**INPUTTING THE DATA:**

The data files are inputted using the input () function. From understanding the input data, the last column contains the labels(diseases) and the rest other columns are the attributes.

**PCA IMPLEMENTATION:**

The following steps were followed in implementing the PCA algorithm.

1. From the original attributes, the mean is calculated.
2. Adjusted attributes are calculated by determining the difference between the mean and the original attributes.
3. The covariance matrix is calculated using numpy.cov() on the adjusted attributes.
4. The eigen values and eigen vectors are determined from the covariance matrix using the numpy.linalg.eig() function.
5. The eigen values and vectors are sorted in the descending order in order to get the top eigen vector and value pairs.
6. The new co-ordinates are calculated by taking dot product of the original attributes and the top eigen vectors.

**TSNE AND SVD IMPLEMENTATION:**

The TSNE is implemented using the existing package from sklearn.manifold. The learning rate is set to 100, the n_components is set to 2 and initialization of embedding is set as pca and the scatter plot is plotted.

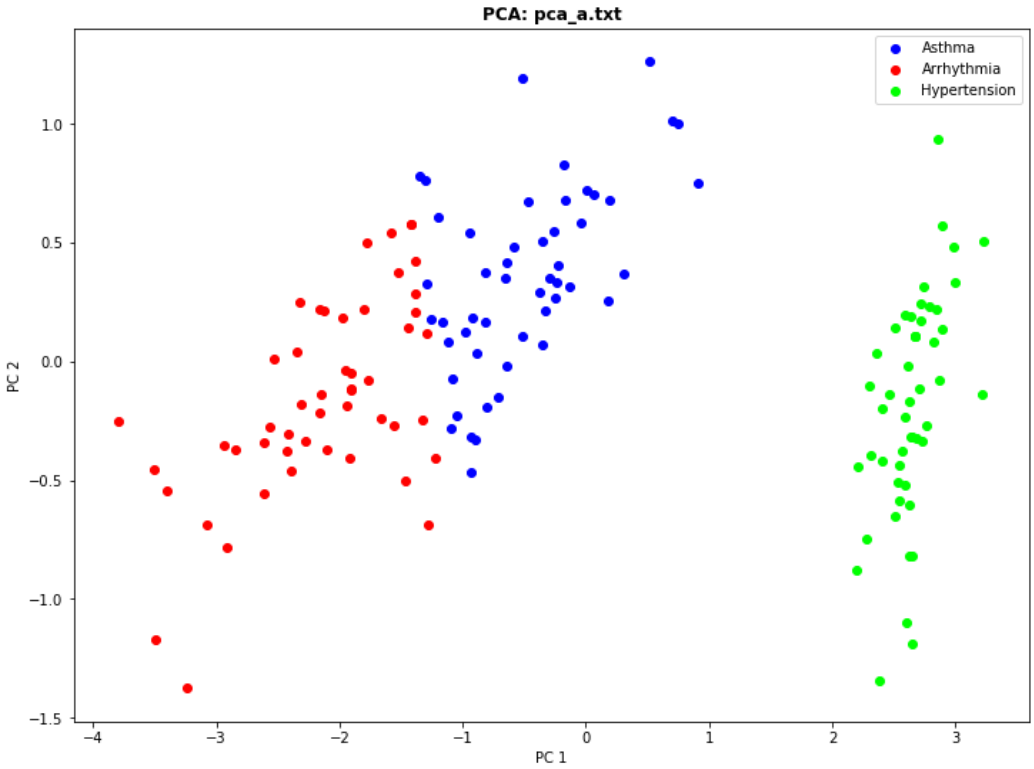
The SVD is implemented using numpy.linalg.svd() and the scatter plot is plotted for visualization.

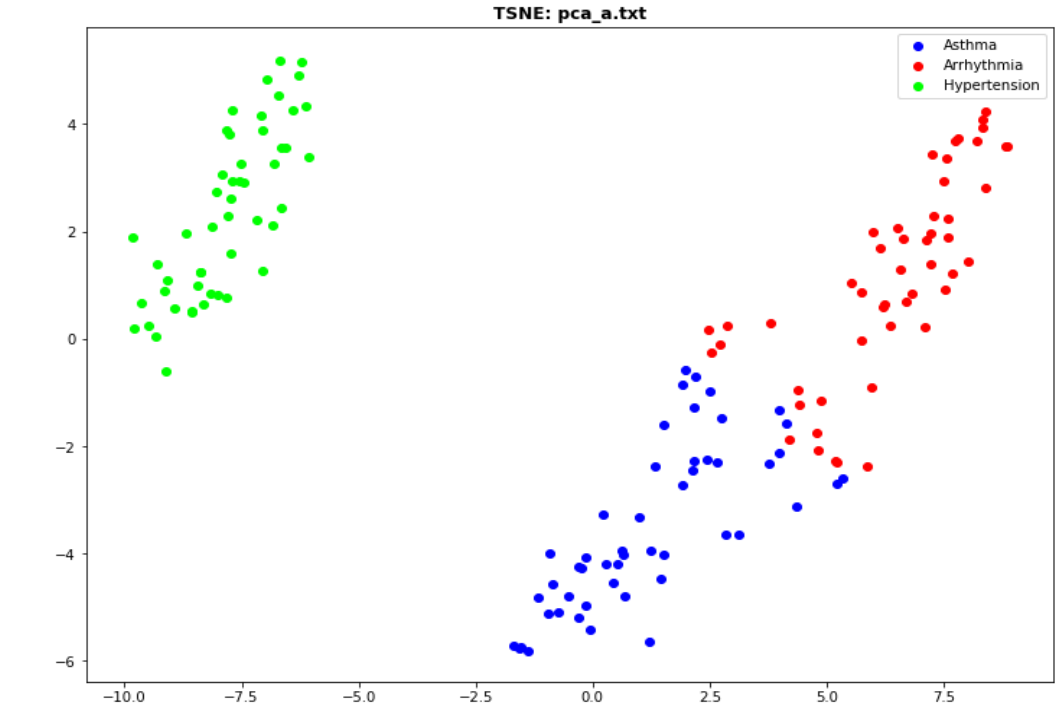
**VISUALIZATION:**

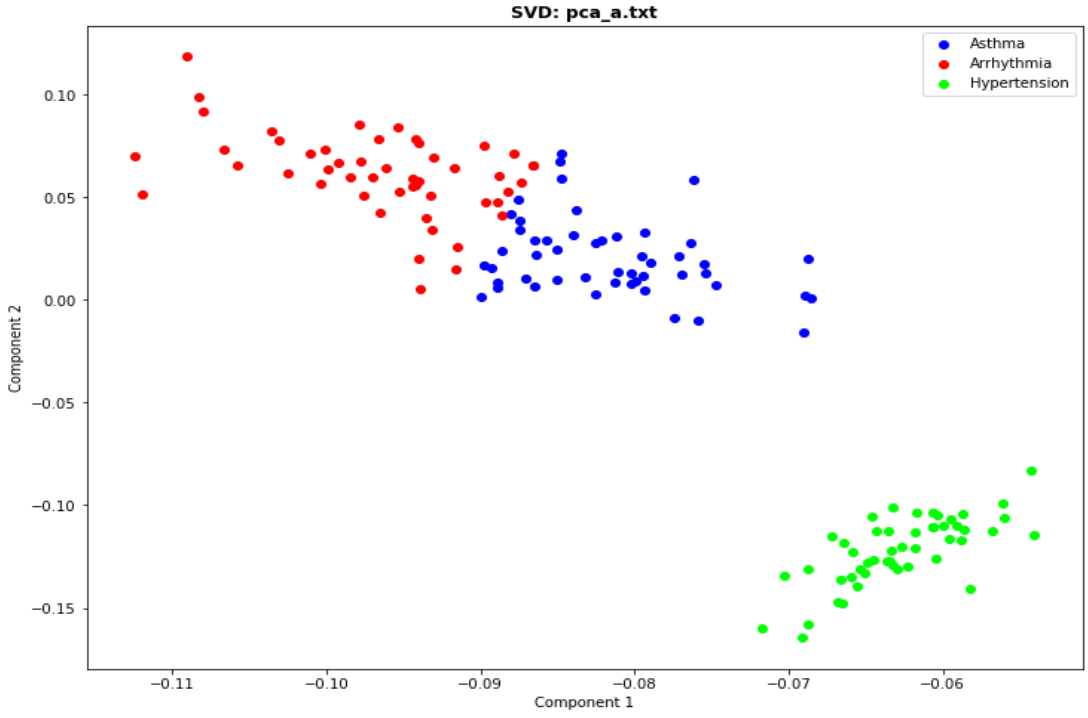
The visualization for the three algorithms is through scatter plots. The color vectors are created for the different labels(diseases) using numpy.linspace() function and colormap is used for coloring the scatter plots.

**SCATTER POLTS:**

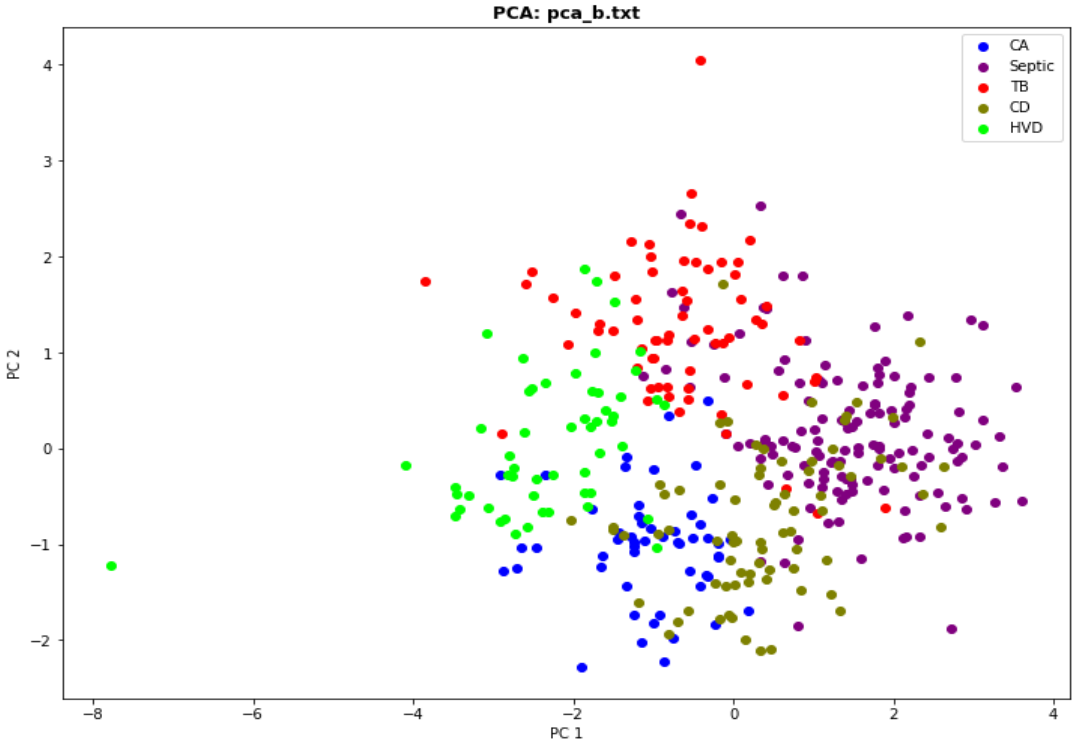
**a) pca_a.txt**

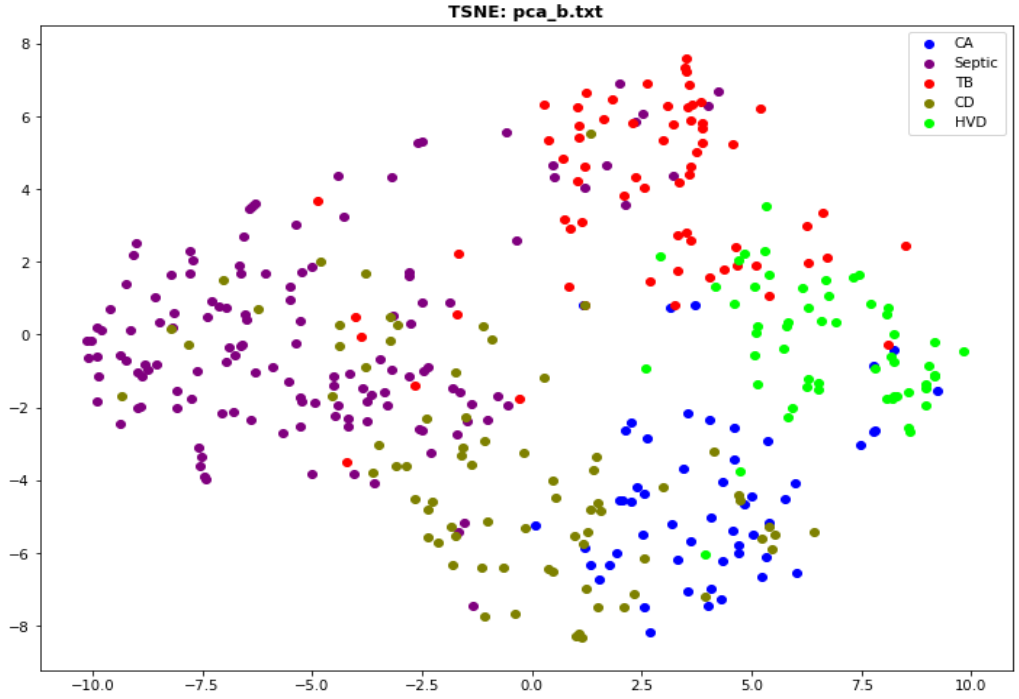


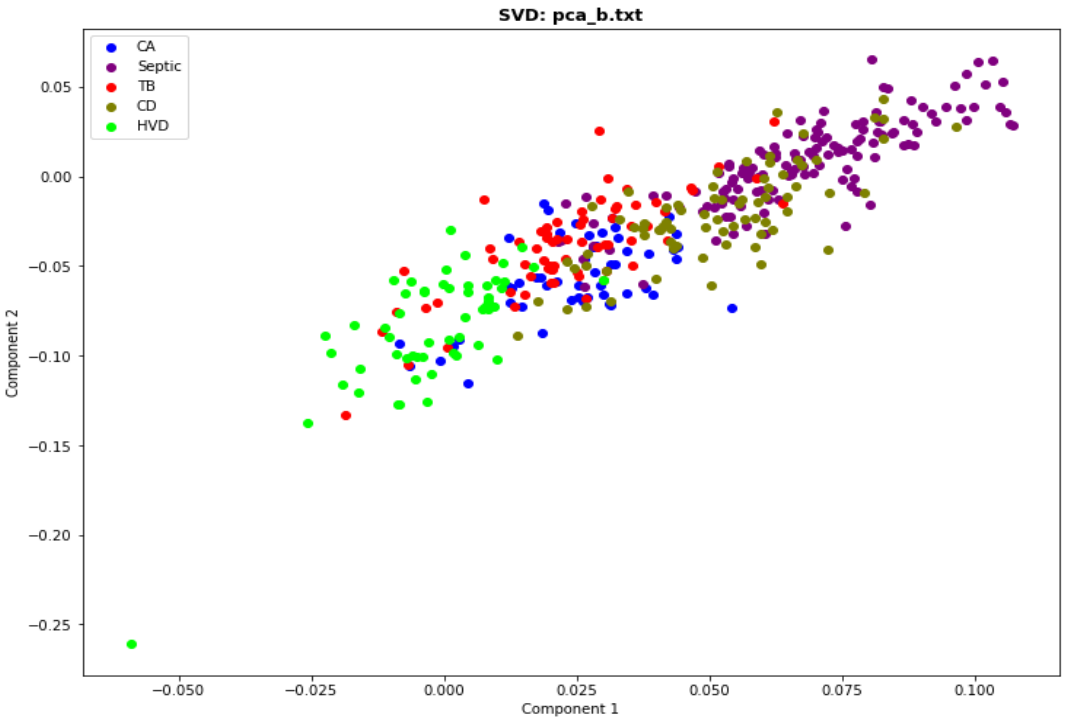

TSNE: pca_a.txt
Asthma
Arrhythmia
Hypertension
4
2
0
−2
−4
−6
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5


SVD: pca_a.txt
Asthma
Arrhythmia
Hypertension
0.10
0.05
0.00
−0.05
−0.10
−0.15
Component 2
−0.11
−0.10
−0.09
−0.08
−0.07
−0.06
Component 1

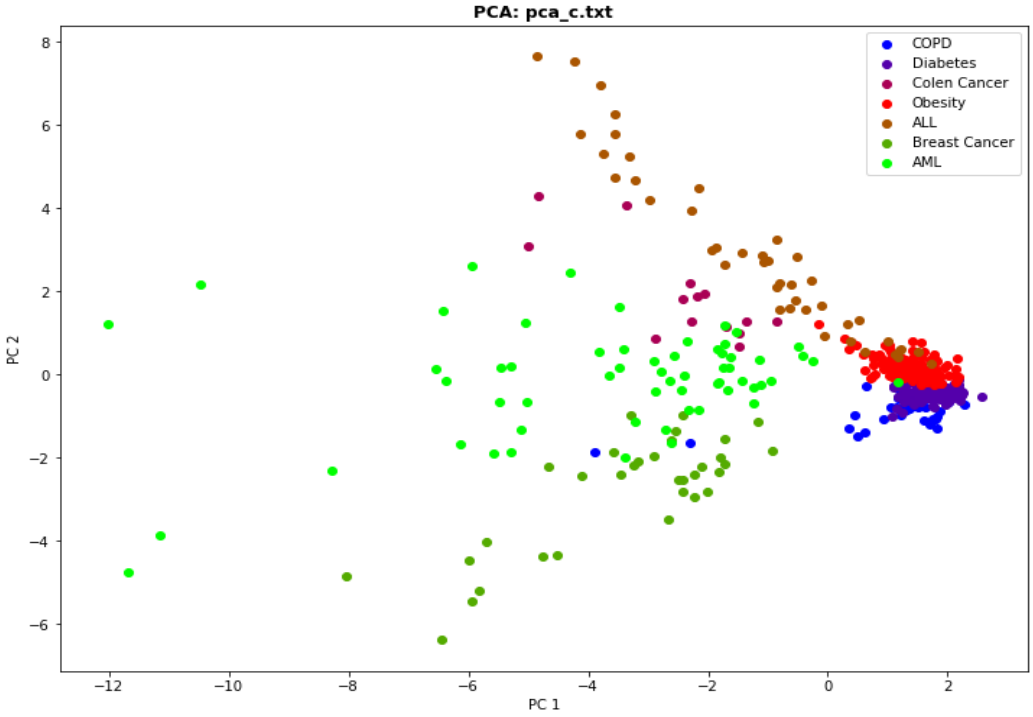
**b) pca_b.txt**

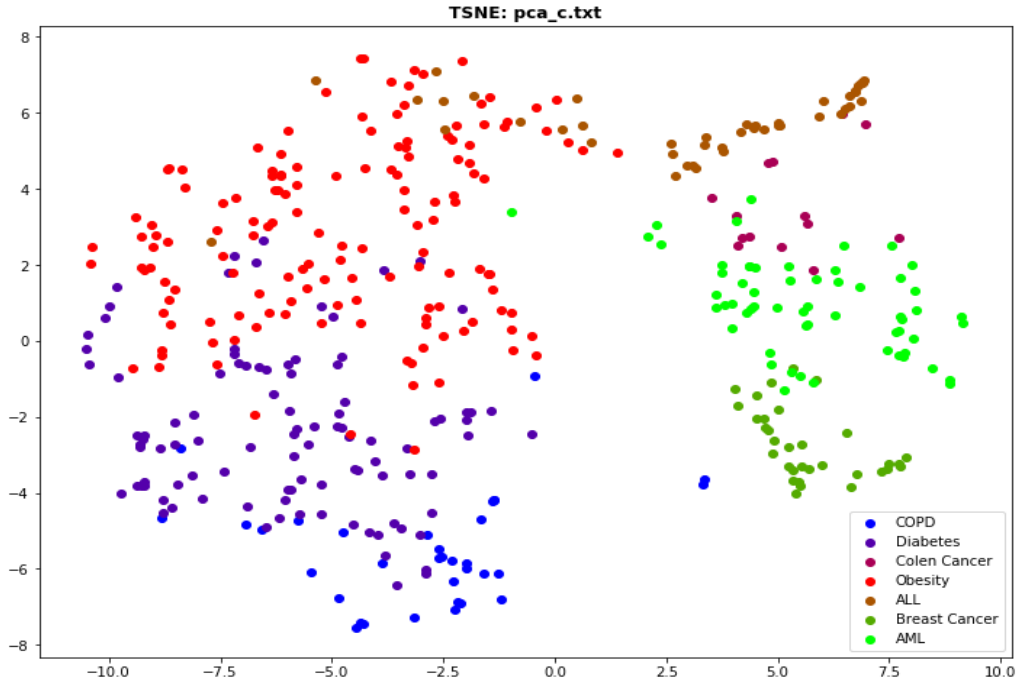

PCA: pca_b.txt
CA
Septic
TB
CD
HVD
PC 2
PC 1
4
3
2
1
0
−1
−2
−8
−6
−4
−2
0
2
4

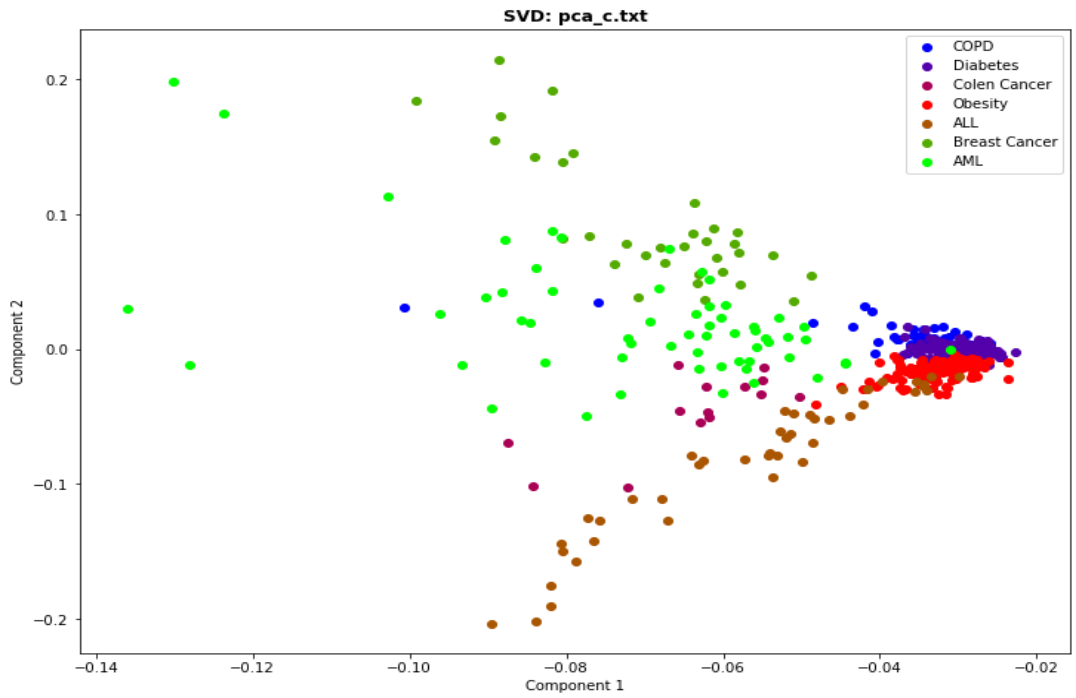

TSNE: pca_b.txt
CA
Septic
TB
CD
HVD
8
6
4
2
0
−2
−4
−6
−8
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


SVD: pca_b.txt
CA
Septic
TB
CD
HVD
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
Component 2
−0.050
−0.025
0.000
0.025
0.050
0.075
0.100
Component 1

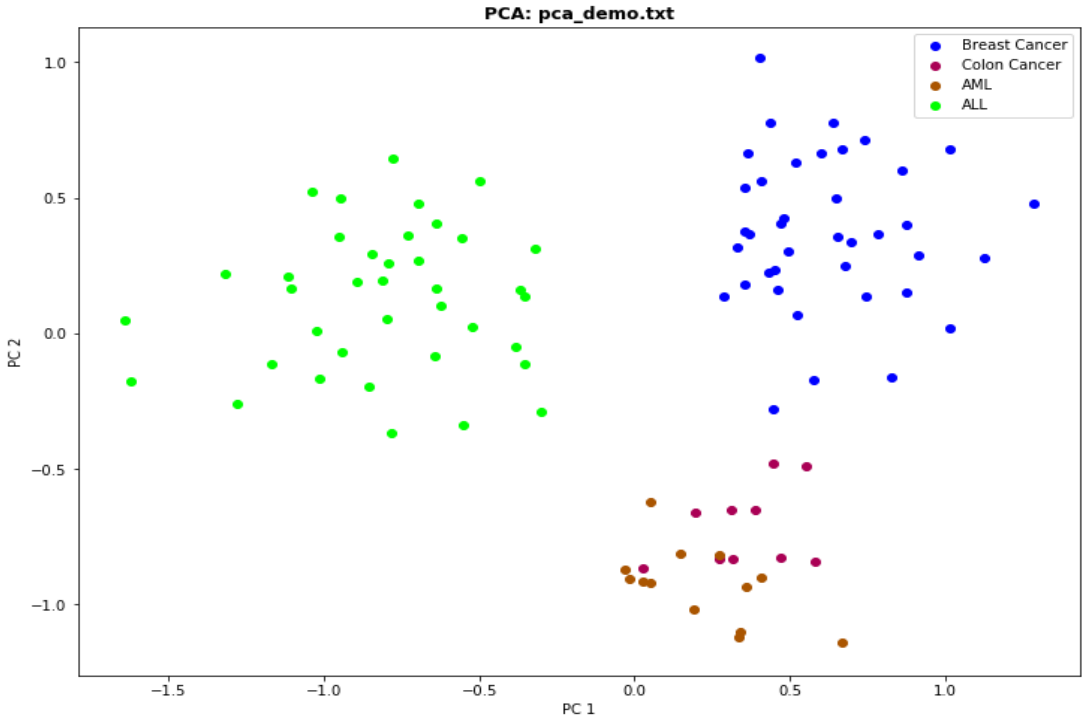
c) **pca_c.txt**

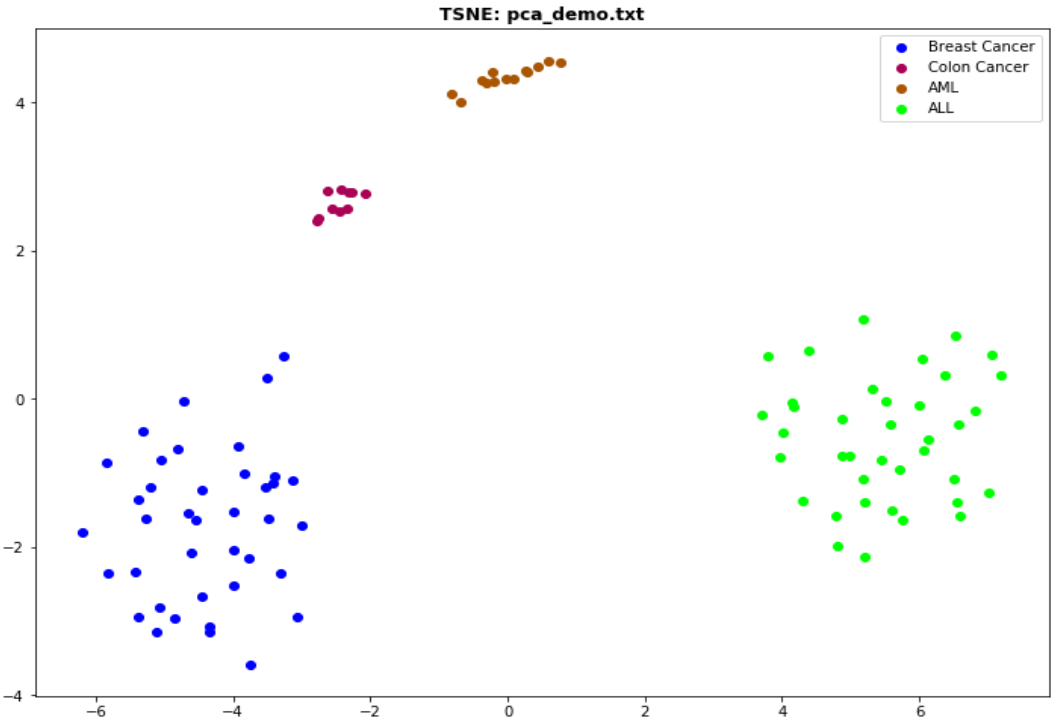

PCA: pca_c.txt
COPD
Diabetes
Colen Cancer
Obesity
ALL
Breast Cancer
AML
PC 2
PC 1
8
6
4
2
0
−2
−4
−6
−12
−10
−8
−6
−4
−2
0
2

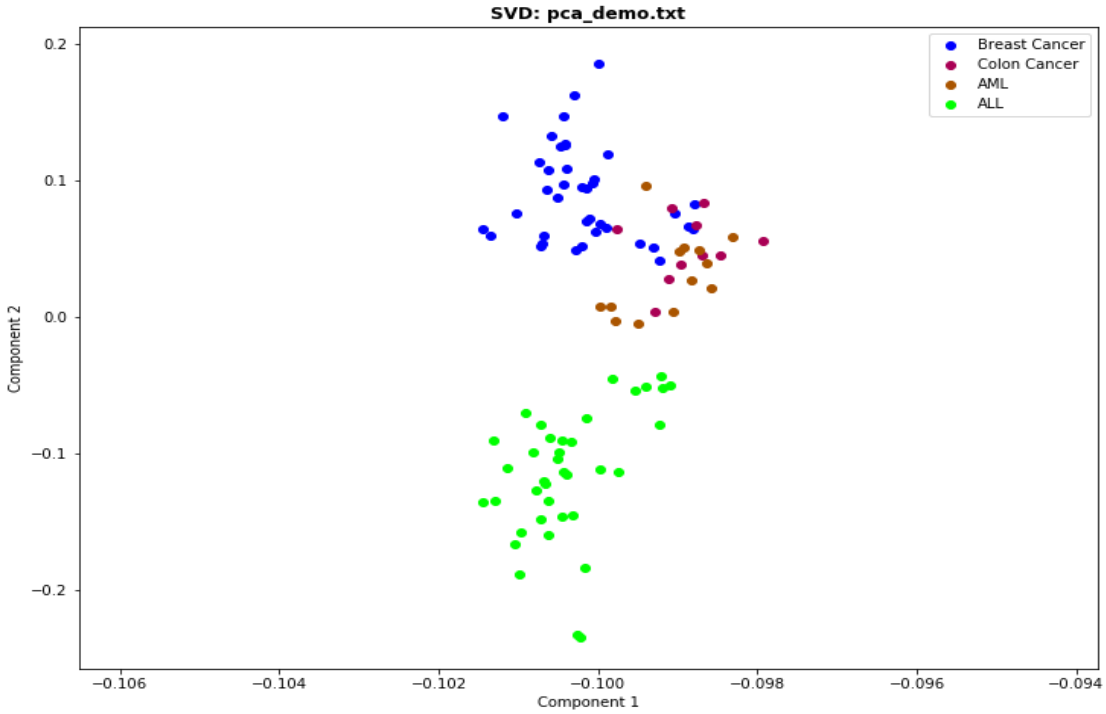

TSNE: pca_c.txt
COPD
Diabetes
Colen Cancer
Obesity
ALL
Breast Cancer
AML
8
6
4
2
0
−2
−4
−6
−8
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0




**d) pca_demo.txt**




TSNE: pca_demo.txt
Breast Cancer
Colon Cancer
AML
ALL
4
2
0
−2
−4
−6
−4
−2
0
2
4
6


SVD: pca_demo.txt
Breast Cancer
Colon Cancer
AML
ALL
Component 2
0.2
0.1
0.0
−0.1
−0.2
−0.106
−0.104
−0.102
−0.100
−0.098
−0.096
−0.094
Component 1

**RESULTS:**

Principal component analysis (PCA) is a dimensionality reduction method which uses orthogonal transformation to convert a set of correlated dimensions into a set of uncorrelated dimensions called principal components. It can also be performed by singular value decomposition (SVD) of the data matrix.

- PCA and SVD tend to give similar results in any test case, because the approach is similar i.e. reduce high dimensional data into low dimensional data. But as the number of dimensions increase, the data points are not clearly distinct.
- t-SNE is another technique for dimensionality reduction and is mostly suited for the visualization of high-dimensional datasets. Unlike PCA, t-SNE has a clear distinction of data points for higher dimensional data.
- The main disadvantage with t-SNE is that it leads to huge unnecessary computations and memory consumption.
- From the scatter plots, we see that PCA and SVD tend to give almost identical results on the scatter plot. But, t-SNE tends to give different results than the PCA and SVD and performs better in case of pca_c.txt which has more dimensions by giving a much clearer distinction of the data points.

**REFERENCES:**


[1] https://matplotlib.org/api/_as_gen/matplotlib.pyplot.scatter.html

[2] https://docs.scipy.org/doc/numpy/reference/generated/numpy.linalg.svd.html

[3] http://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html

[4] https://pythonspot.com/matplotlib-scatterplot/

[5] https://stackoverflow.com/questions/43949395/looping-scatter-plot-colors

[6] https://stats.stackexchange.com/questions/235882/pca-in-numpy-and-sklearn-produces-different-results