PRINCIPAL COMPONENT ANALYSIS

Scatter Plots obtained from the three algorithms are as follows:

1. PCA implementation on pca\_a.txt dataset:



1. PCA implementation on pca\_b.txt dataset:



1. PCA implementation on pca\_c.txt dataset:



PCA Algorithm:

Principal Component Analysis algorithm provides a way to reduce the number of dimensions in a given dataset by preserving the data that seems to be important for analysis. New and arguably smaller set of dimensions are obtained from the existing feature space that explain most of the variance in the data. These dimensions are called principal components.

Steps involved in implementing PCA:

1. The dataset is read into a data frame and feature vectors and the target distinguishable column containing different classes are separated.
2. Mean of the feature vectors is calculated and thereby, covariance is computed using the following function:
   1. S = (1/n) \* (X-X’)\*(X-X’)T

Where X’ is the mean of matrix X.

1. Alternatively, covariance can also be computed using cov() function in the numpy package in python.
2. Once the covariance matrix is obtained, compute eigen vectors and eigen values using numpy’s linalg.eig() method. We can also assert that the eigen values and eigen vectors are correct by using the following formula:
   1. 
3. Append the eigen vectos and eigen values in to a single array and sort the array based on the eigen values.
4. Take the top 2 eigen vectors, as we have to reduce the dataset to 2-dimensional dataset here.
5. Compute the reduced axes using the mean centered feature matrix obtained in step 2 as follows:
   1. (X-X’) \* eig\_vec
6. 2 dimensional orthogonal axes is obtained using this procedure, now scatter plots are generated using these 2 axes and the target vector in the dataset.