**PR Assignment-2**

**G HAREESH**

**18MCMB15**

**M.TECH (IT)**

**1st question**

**def standerdize (data):**

mean = np.mean(data,axis=0)

std = np.std(data,axis=0)

return (data - mean) / std

**def getpca(k,data,get\_eig = False):**

data = standerdize(data)

cov = np.cov(data.T)

eig\_vals , eig\_vecs = np.linalg.eig(cov)

eig\_pairs = [(np.abs(eig\_vals[i]),eig\_vecs[:,i]) for i in range(len(eig\_vals))] eig\_pairs.sort(key = lambda k:k[0] , reverse = True)

mat = np.hstack([eig\_pairs[i][1][:,np.newaxis] for i in range(k)])

if get\_eig == True:

return eigen\_vals , data.dot(mat)

return data.dot(mat)

**def getlda(k,data,datalabels):**

labels = np.unique(datalabels)

data = standerdize(data)

mean\_vecs = []

mean\_labels = []

for label in labels:

mean\_vecs.append(np.mean(data[datalabels==label] , axis=0))

mean\_labels.append(label)

d = data.shape[1]

S\_W = np.zeros((d,d))

for label,mv in zip(labels,mean\_vecs):

class\_scatter = np.zeros((d,d))

for row in data[datalabels==label]:

row, mv = row.reshape(d, 1), mv.reshape(d, 1) class\_scatter += (row-mv).dot((row-mv).T) S\_W += class\_scatter

"""for label,mv in zip(range(1, 4), mean\_vecs):

class\_scatter = np.cov(X\_train\_std[y\_train==label].T)

S\_W += class\_scatter"""

mean\_overall = np.mean(data, axis=0)

mean\_overall = mean\_overall.reshape(d,1)

S\_B = np.zeros((d, d))

for label, mean\_vec in zip(mean\_labels,mean\_vecs):

n = data[datalabels == label].shape[0]

mean\_vec = mean\_vec.reshape(d,1)

S\_B += n \* (mean\_vec - mean\_overall).dot((mean\_vec - mean\_overall).T)

eigen\_vals, eigen\_vecs = np.linalg.eig(np.linalg.inv(S\_W) @ (S\_B))

eigen\_pairs = [(np.abs(eigen\_vals[i]), eigen\_vecs[:,i]) for i in range(len(eigen\_vals))] eigen\_pairs = sorted(eigen\_pairs,key=lambda k: k[0], reverse=True) mat = np.hstack([eigen\_pairs[i][1][:,np.newaxis] for i in range(k)])

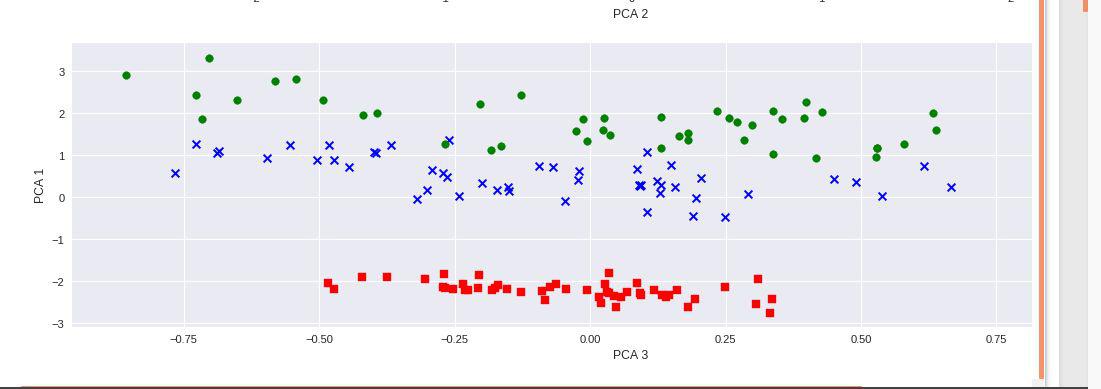
return data.dot(mat)

**2nd Question:**

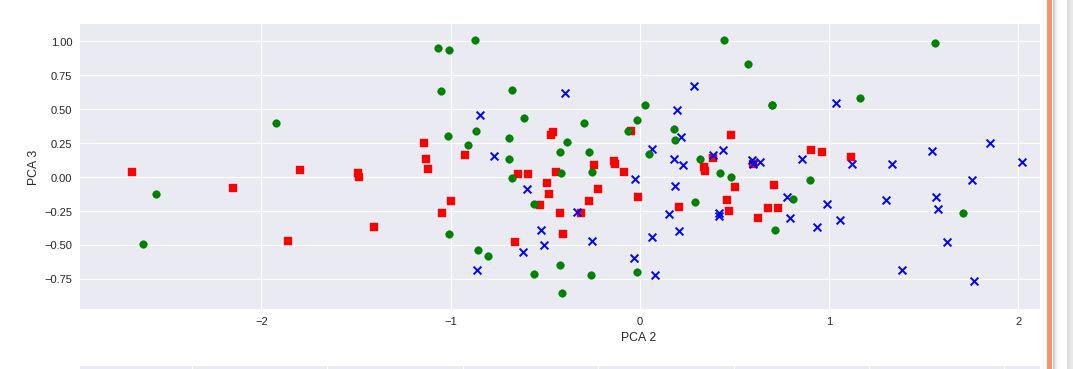
**Results :**

**(Snapshots of the Results TO view them completely Open Notebook)**

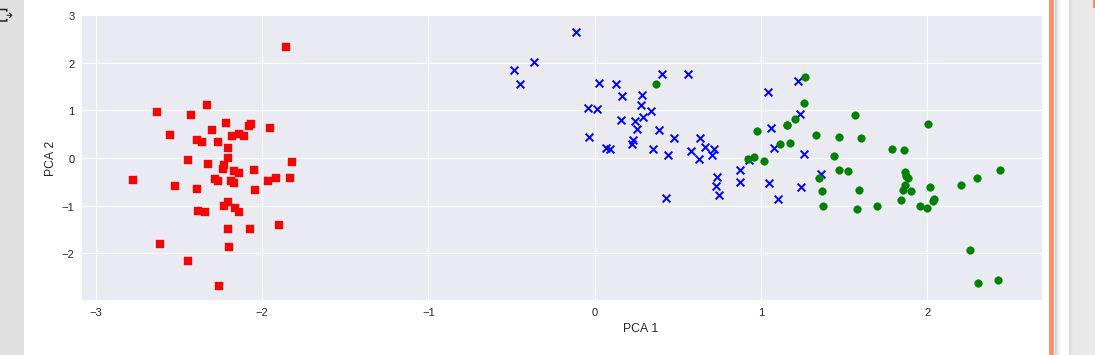
**PCA1 vs PCA3**



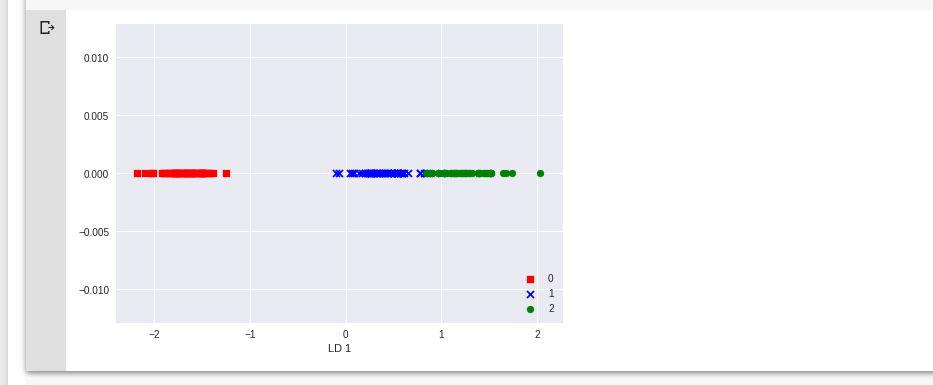
**PCA2 vs PCA3**



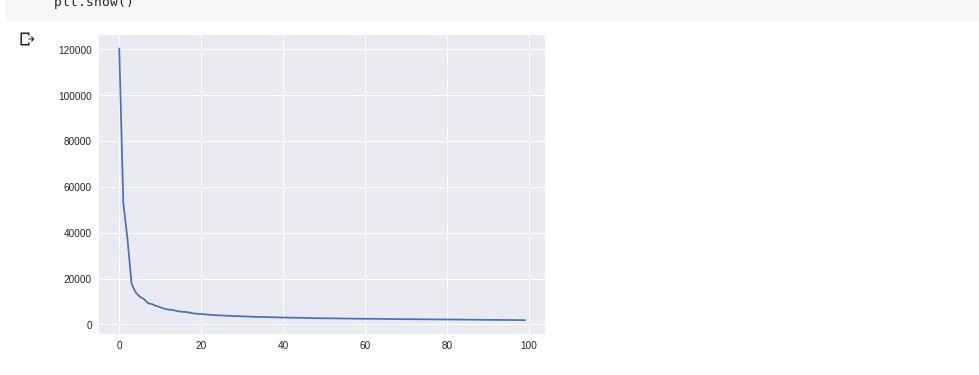
**PCA1 vs PCA2**



**LDA :**



**Arcene :**



**3rd question**

**Libraries Used:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

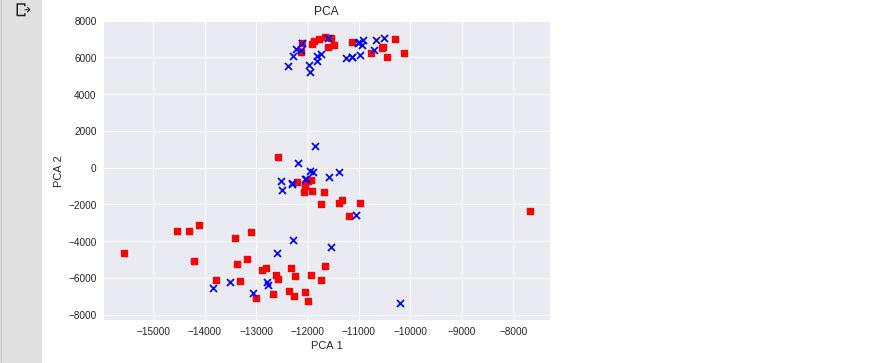
**Dataset Used:**

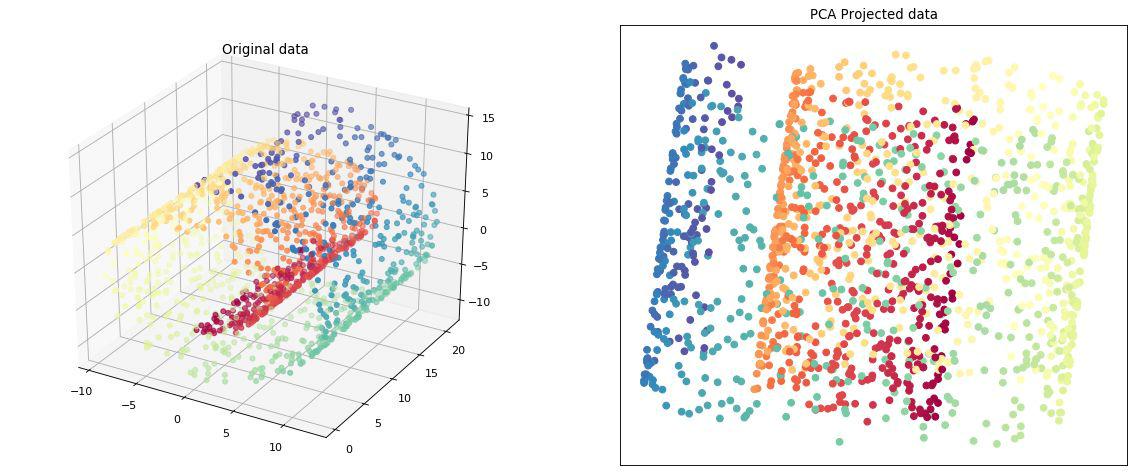
UCI Arcene cancer dataset

**Analysis of components with respect to variance:** No of Components required for getting 0.85 variances are : 61 No of Components required for getting 0.90 variances are : 73 No of Components required for getting 0.95 variances are : 86 No of Components required for getting 0.99 variances are : 97

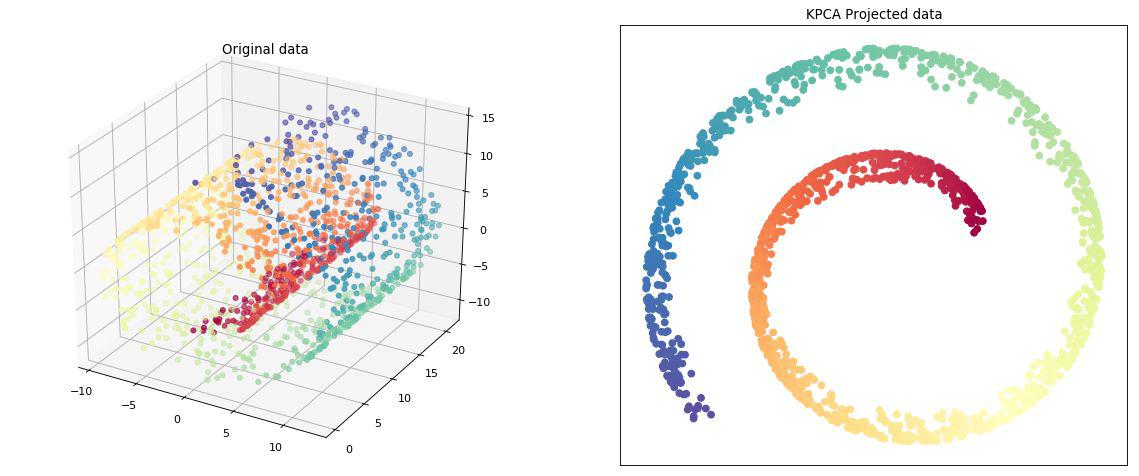
**Results:**

**Pca two principal component**

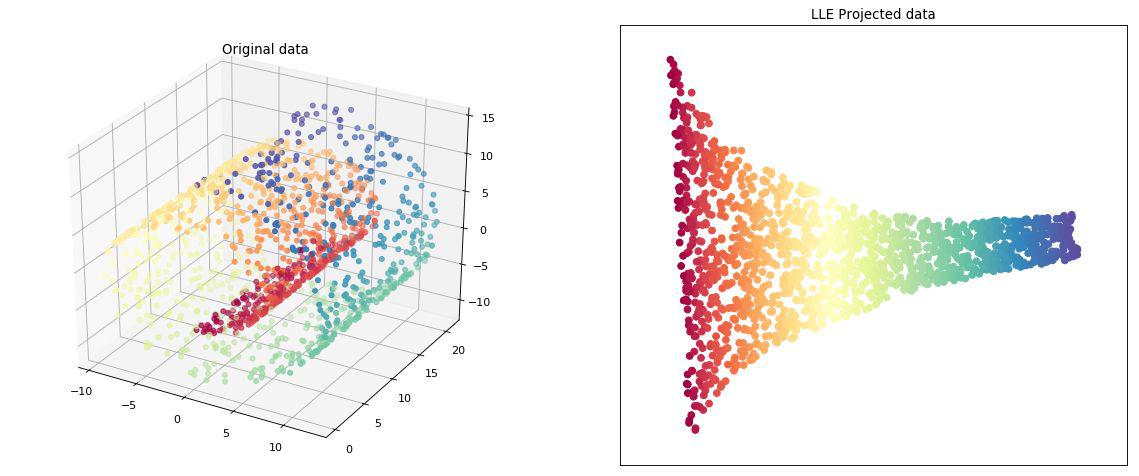


**PCA**

**KPCA**



**LLE**

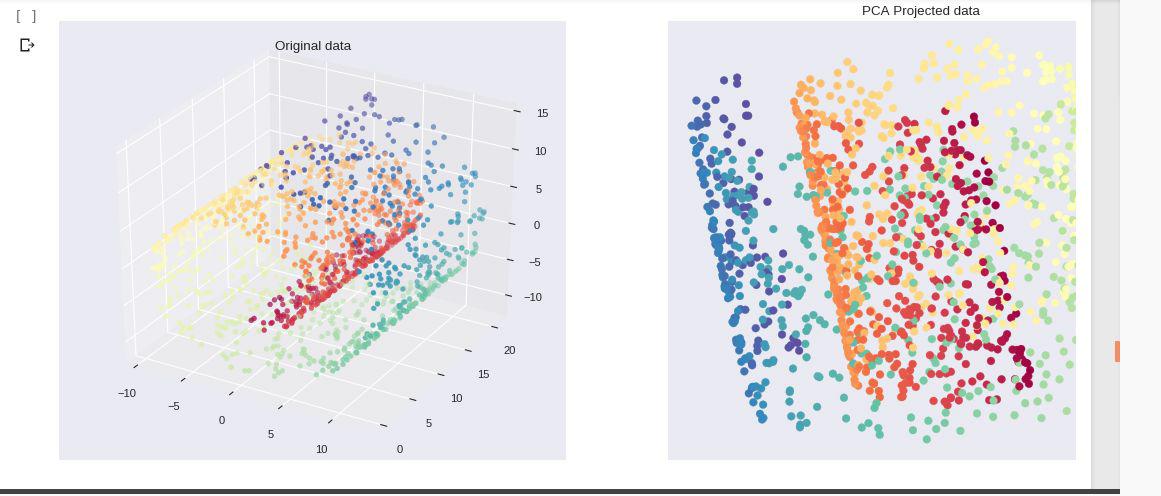


**4th Question**

**Dataset Used:** 3D swissroll dataset

**Results :**

**PCA**



**KPCA**

**LLE**

