```
# -*- coding: utf-8 -*-
"""
Created on Thu May  2 05:05:57 2019
@author: Sandeep Kumar MCA 4 th sem JNU
"""
from scipy import linalg
import numpy as np
import matplotlib.pyplot as plt

d1 = np.array([[1,1],[2,2],[3,3],[4,5],[5,4],[6,6],[8,7]])
md1=np.mean(d1, axis=0)
print(md1)
x, y = md1.T
plt.scatter(x,y)
d2= np.array( [[-2,3],[-1,4],[1,5],[3,6],[4,7],[2,8],[5,9]])
md2=np.mean(d2, axis=0)
print(md2)
x, y = md2.T
plt.scatter(x,y)
xd1=np.array([1,2,3,4,5,6,8])
print("x values of first class:",xd1)
yd1=np.array([1,2,3,5,4,6,7])
print("y values of first class:",yd1)
xd2=np.array([-2,-1,1,3,4,2,5])
print("x values of second class:",xd2)
yd2=np.array([3,4,5,6,7,8,9])
print("y values of second class:",yd2)
# all x values
xtotal=np.concatenate((xd1,xd2),axis=0)
print("all x values:",xtotal)
#all y values
ytotal=np.concatenate((yd1,yd2),axis=0)
print("all y value:",ytotal)
d=np.concatenate((d1,d2),axis=0)
print("all data points:",d)

x, y = d1.T
plt.scatter(x,y)
x1, y1 = d2.T
plt.scatter(x1,y1)
#mean of data
meand=np.mean(d, axis=0)
print(meand)
```

```
x, y = meand.T
plt.scatter(x,y)
#mean substracted data
msd= d - d.mean(axis = 0)
print(msd)
#covariance of matrix
cov=np.cov(msd, rowvar=0)
print(cov)
#eigen value of the  matrix
A = np.array(cov)
l, v = linalg.eig(A)
print(l)
print(v)
if(l[0]>l[1]):
        print("first principal component is good for making clustering:\n")
        print("we will project the data point in first principal component:\n")
else:
        print("second principal component is good for making clustering:\n")
```