```
# -*- coding: utf-8 -*-
"""
Created on Thu May  2 08:23:01 2019
@author: Sandeep Kumar
"""
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("mean of class1\n",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("mean of class2\n",md2)
x, y = md2.T
plt.scatter(x,y)
d=np.concatenate((d1,d2),axis=0)
print("all data points:\n",d)

x, y = d1.T
plt.scatter(x,y)
x1, y1 = d2.T
plt.scatter(x1,y1)
#mean of data
u=np.mean(d, axis=0)
print("mean of the whole data",u)
u1=np.mean(d1,axis=0)
print("mean of first class:\n")
print(u1)
u2=np.mean(d2,axis=0)
print("mean of the second class :\n",u2)
u1u=np.subtract(u1,u)
print("u1u is",u1u)
u1u= np.array([u1u])
u1ut=u1u.T
print("transpose",u1ut)
m1=np.matmul(u1ut,[u1u])
m1=m1*7
print("value of m1\n",m1)
u2u=np.subtract(u2,u)
```

```
print(u2u)
u2u= np.array([u2u])
u2ut=u2u.T
m2=np.matmul(u2ut,[u2u])
m2=m2*7
print("value of m2\n",m2)
SB=np.add(m1,m2)
print("VALUE OF SB:",SB)
print("process for the SW\n")
xu1=np.subtract(d1,u1)
xu1t=np.transpose(xu1)
w1=np.matmul(xu1t,xu1)
print("xu1\n",xu1)
print("transpose xu1\n",xu1t)
print("w1",w1)
xu2=np.subtract(d2,u2)
xu2t=np.transpose(xu2)
w2=np.matmul(xu2t,xu2)

print("xu2\n",xu2)
print("transpose xu2\n",xu2t)
print("w2",w2)
SW=np.add(w1,w2)
SW=SW/7
print("SW\n",SW)

print("process for finding the eigen value:\n")

a= np.array(SW)

b = np.array(SW)
b=np.linalg.inv(b)
print("inverse matrix of the SW\n:",b)
SBSWINV=np.matmul(b,SB)
print("multiplication\n",SBSWINV)
# calculation for the eigen value of the SW^-1 SB
c=[[]]
for i in SBSWINV :
        c=i

print("value of matrix c:\n",c)
a=np.array(c)
eigenvalue,eigenvector=linalg.eig(a)
```

```
print("eigenvalue\n",eigenvalue,"\neigenvector",eigenvector)

print("first eigen vector is direction of 45 degree and second is 135 degree")
x=[-5,12]
y=[-3.5,14]
plt.plot(x,y)
x=[-5,12]
y=[12,-5]
plt.plot(x,y)
plt.show()
if(eigenvalue[0]>eigenvalue[1]):
	print("after projection of the data point in first eigen vector data will be more seperable than second eigen vector because eigen value of first eigen vector is more than first :\n")
else:
	print("after projection of the data point in second eigen vector data will be more seperable than first eigen vector:\n")


'''
swin = np.linalg.inv(a)
print(swin)
mat=np.matmul(swin,SB)
print(mat)
A = np.array(mat)
w, v = linalg.eig(A)
print(w)
print(v)
'''
```