İşaret İşleme için Makine Öğrenmesi

Bilge Günsel

HW1

Ömer Bahadır Gökmen 040190030

Iris Veri Kümesinin 4B Vektöründen Ortalama Vektör Elde Edilmesi

&

Kovaryans Matrisinin Elde Edilmesi

Verinin Kullanılacak Duruma Getirilmesi

Kod:

```
def parser(data):
    x = list ()
    for i in range(len(data)):
        temp = data[i].split(',')
        x.append([temp[0],temp[1],temp[2],temp[3]])
    return numpy.array(x).astype(numpy.float64)
def data_processing():
    with open('irisdata.xml') as f:
        irisdata = f.read()
    data = BeautifulSoup(irisdata, 'xml').text.split('\n')[1:-3]
    numpy.copyto(Iris,parser(data))
    numpy.copyto(Iris_setosa,Iris[:50])
    numpy.copyto(Iris_versicolor,Iris[50:100])
    numpy.copyto(Iris_virginica,Iris[100:150])
```

A. Verilerin Ortalama Vektörleri

Iris_setosa Ortalama Vektörü: [5.006 3.418 1.464 0.244]

Iris_versicolor Ortalama Vektörü: [5.936 2.77 4.26 1.326]

Iris_virginica Ortalama Vektörü: [6.588 2.974 5.552 2.026]

Kod:

```
print(numpy.mean(numpy.reshape(Iris_setosa,(Training_number,Feature_number)).astype(numpy.float64),axis = 0))
print(numpy.mean(numpy.reshape(Iris_versicolor,(Training_number,Feature_number)).astype(numpy.float64),axis = 0))
print(numpy.mean(numpy.reshape(Iris_virginica,(Training_number,Feature_number)).astype(numpy.float64),axis = 0))
```

B. Veri Kümelerindeki Sınıfların Kovaryans Matrisleri

Iris_setosa

[[0.12176397 0.09829199 0.015816 0.010336]

[0.09829199 0.14227599 0.011448 0.011208]

[0.015816 0.011448 0.02950401 0.005584]

[0.010336 0.011208 0.005584 0.011264]]

Iris_versicolor

[[0.261104 0.08347998 0.17924003 0.054664]

[0.08347998 0.09649999 0.081 0.04038]

[0.17924003 0.081 0.2164 0.07163999]

[0.054664 0.04038 0.07163999 0.03832399]]

Iris_virginica

[[0.3962559 0.09188799 0.29722396 0.04811199]

[0.09188799 0.10192399 0.06995199 0.04667599]

[0.29722396 0.06995199 0.29849604 0.04784798]

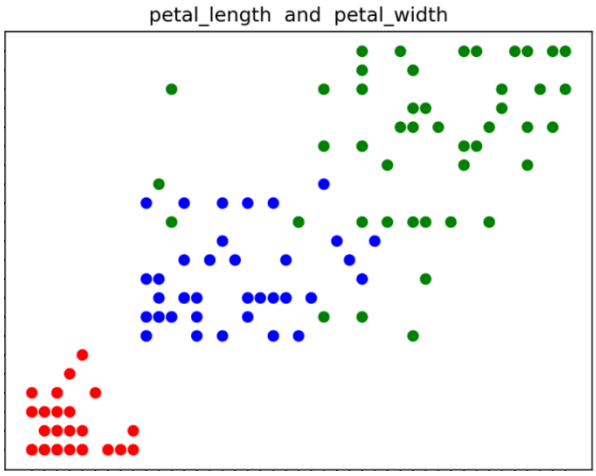
[0.04811199 0.04667599 0.04784798 0.07392401]]

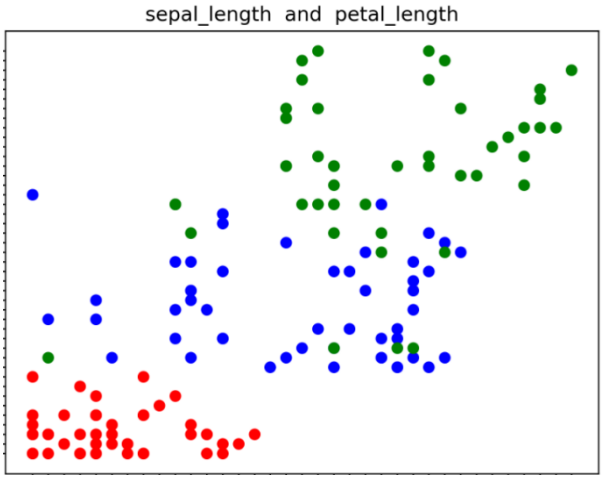
Kod:

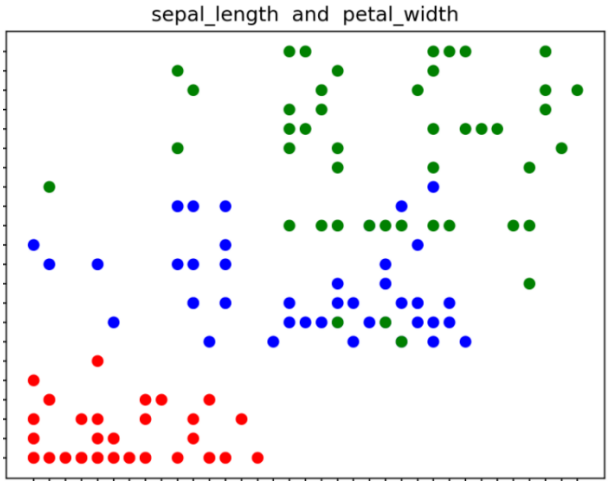
```
def get_covariance_matrix(A):
    A=numpy.array(A, dtype='f')
    mean_vector=numpy.mean(A,axis=0)
    cov_matrix = numpy.reshape(numpy.zeros(Feature_number*Feature_number),
(Feature_number,Feature_number))

    for x in range(Feature_number):
        for y in range(len(A[:,x])):
            A[:,x][y]=float(A[:,x][y])-float(mean_vector[x])

    cov_matrix=(numpy.dot(numpy.transpose(A),A))/Training_number
    print(cov_matrix)
```

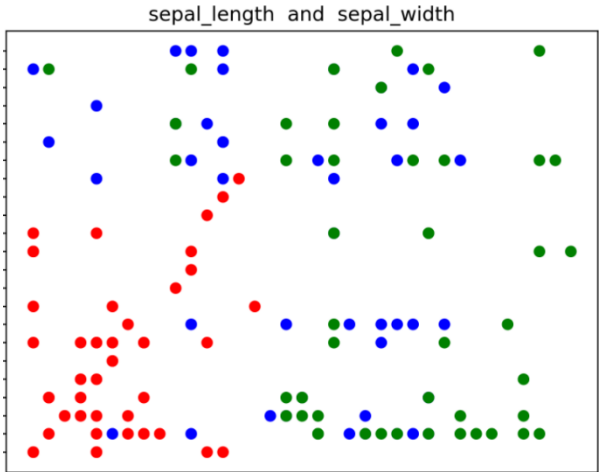
C. Veri kümesinden elde edilen kovaryans matrisinin köşegen elemanları varyans değerleine karşılık gelmektedir.Varyans elemanların birbirleri ile ilişintisini vermektedir.

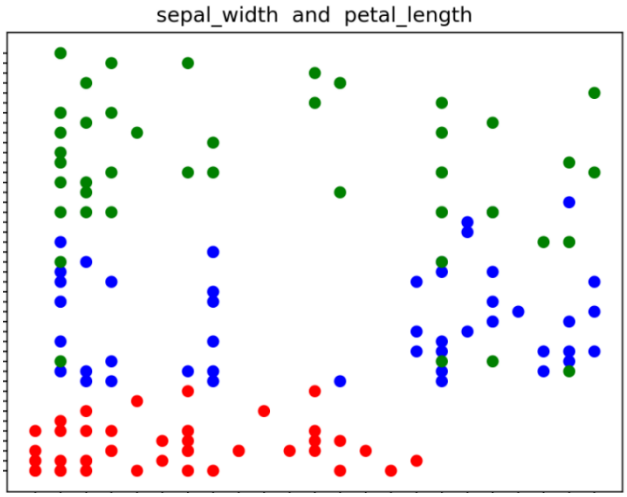
D. Köşegen dışındaki değerler matrisin satır ve sütun değerlerine karşılık gelen vektörlerin birbiri ile ilişintisini vermektedir. Değerin sıfır çıkması o iki vektörün birbiri ile ilişintili olmadığını yani biri diğerini etkilemediğini göstermektedir. Bizim matrislerimizin köşegen dışı elemanları sıfır değildir. Bu gösterir ki elemanlar birbiri ile ilişintilidir.

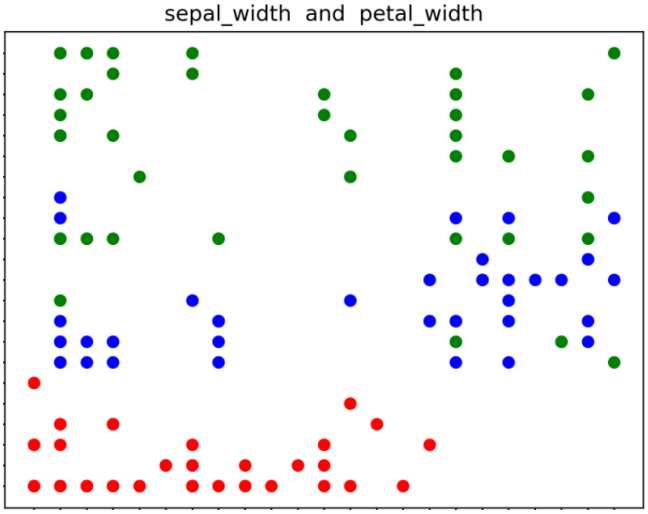

petal_length and petal_width


sepal_length and petal_length


sepal_length and petal_width


sepal_length and sepal_width


sepal_width and petal_length


sepal_width and petal_width

KOD:

```
import numpy
import csv
import matplotlib.pyplot as plt
from bs4 import BeautifulSoup
label=["sepal_length", "sepal_width", "petal_length", "petal_width", "class"]
Feature_number=4
Training_number=50
Iris_setosa= numpy.empty((50,4))
Iris_versicolor=numpy.empty((50,4))
Iris_virginica=numpy.empty((50,4))
Iris=numpy.empty((150,4))
def get_covariance_matrix(A):
    A=numpy.array(A, dtype='f')
    mean_vector=numpy.mean(A,axis=0)
    cov_matrix = numpy.reshape(numpy.zeros(Feature_number*Feature_number),
(Feature_number,Feature_number))

    for x in range(Feature_number):
        for y in range(len(A[:,x])):
            A[:,x][y]=float(A[:,x][y])-float(mean_vector[x])

    cov_matrix=(numpy.dot(numpy.transpose(A),A))/Training_number
    print(cov_matrix)

def parser(data):
    x = list()
    for i in range(len(data)):
        temp = data[i].split(',')
        x.append([temp[0],temp[1],temp[2],temp[3]])
    return numpy.array(x).astype(numpy.float64)
def data_processing():
    with open('irisdata.xml') as f:
        irisdata = f.read()
    data = BeautifulSoup(irisdata, 'xml').text.split('\n')[1:-3]
    numpy.copyto(Iris,parser(data))
    numpy.copyto(Iris_setosa,Iris[:50])
    numpy.copyto(Iris_versicolor,Iris[50:100])
    numpy.copyto(Iris_virginica,Iris[100:150])


def draw():
    for m in range(Feature_number):
        for n in range(Feature_number):
            if m < n:
```

KOD:

```
                fn=open("irisdata.txt","r")
                for row in csv.DictReader(fn, label):
                    #plt.xlim(0,10)
                    #plt.ylim(0,10)
                    plt.xlabel(label[m])
                    plt.ylabel(label[n])
                    plt.title(label[m]+"  and  "+label[n])
                    x = row[label[m]]
                    y = row[label[n]]
                    if row["class"]=="Iris-setosa":
                        plt.plot(x,y,"ro")
                    elif row["class"]=="Iris-versicolor":
                        plt.plot(x,y,"bo")
                    else:
                        plt.plot(x,y,"go")
                plt.savefig(label[m]+"_and_"+label[n]+".png",dpi=300,format="png"
)
                plt.close()
                fn.close()



data_processing()
print("Iris_setosa\n")
get_covariance_matrix(Iris_setosa)
print("Iris_versicolor\n")
get_covariance_matrix(Iris_versicolor)
print("Iris_virginica\n")
print(get_covariance_matrix(Iris_virginica))

print("Iris_setosa mean vector\n")
print(numpy.mean(numpy.reshape(Iris_setosa,(Training_number,Feature_number)).asty
pe(numpy.float64),axis = 0))
print("Iris_versicolor mean vector\n")
print(numpy.mean(numpy.reshape(Iris_versicolor,(Training_number,Feature_number)).
astype(numpy.float64),axis = 0))
print("Iris_virginica mean vector\n")
print(numpy.mean(numpy.reshape(Iris_virginica,(Training_number,Feature_number)).a
stype(numpy.float64),axis = 0))
#draw()
```