EHB 328 - ODEV 1 - Guz 22

Ogrenci: Erkan Giray Arat

Ogretmen: Prof. Bilge Gunsel

Asistan: Ars.Gor. Elif Akbaba

"irisdata.xml" Dosyasindan Veri Almak

In [1]:
```
import numpy as np
import re
from bs4 import BeautifulSoup
```

"numpy" kutuphanesi array islemleri icin, "regex", "re" kutuphanesi string islemleri icin, "BeautifulSoup" kutuphanesi "irisdata.xml" dosyasini isleyebilmek icin import edildi.

In [2]:
```
with open('irisdata.xml') as fp:
    soup = BeautifulSoup(fp, "xml")

obj=soup.find('para')
text=obj.getText()
print('\nThe first item in the document is:\n',text,'\n')
```

```
The first item in the document is:
 5.1,3.5,1.4,0.2,Iris-setosa
```

"soup" degiskeni icerisine "irisdata.xml" verileri islendi. BeautifulSoup tarafindan olusturulan soup objesi bir veri agaci. Agacin 'para' tag'li ilk elemanini bulduk ve obj'u ona isaret ettirdik.

"text" degiskeni icerisine "obj"un isaret ettigi elemanin ozelligi olan getText() fonksiyonuyla, ilk veriyi yazdirdik. Kontrol etmek amacli "text" degiskenini yazdirdik.

In [3]:
```
iris_setosa_holder=np.zeros((50,4))
iris_versicolor_holder=np.zeros((50,4))
iris_virginica_holder=np.zeros((50,4))

word='Iris-setosa'
a=re.search(word,text)
if a==None:
    print(a,'. No match for ', word, ' in', text,'.\n')
first_index=0
```

In [4]:
```
while a!=None:
    print('Matched:')

    for i in range(0,4):
                iris_setosa_holder[first_index,i]=float(text[(0+i*4):(3+i*4)])


    print(first_index+1,'. array of iris_setosa_holder is: ', iris_setosa_holder[first_index])
```

```
    first_index+=1

    #move to next item
    obj=obj.findNext('para')

    #update text variable
    text=obj.getText()

    #search for iris-setosa in next item
    a=re.search(word,text)

    #then loop starts again for new item
```

```
Matched:
1 . array of iris_setosa_holder is:  [5.1 3.5 1.4 0.2]
Matched:
2 . array of iris_setosa_holder is:  [4.9 3.  1.4 0.2]
Matched:
3 . array of iris_setosa_holder is:  [4.7 3.2 1.3 0.2]
Matched:
4 . array of iris_setosa_holder is:  [4.6 3.1 1.5 0.2]
Matched:
5 . array of iris_setosa_holder is:  [5.  3.6 1.4 0.2]
Matched:
6 . array of iris_setosa_holder is:  [5.4 3.9 1.7 0.4]
Matched:
7 . array of iris_setosa_holder is:  [4.6 3.4 1.4 0.3]
Matched:
8 . array of iris_setosa_holder is:  [5.  3.4 1.5 0.2]
Matched:
9 . array of iris_setosa_holder is:  [4.4 2.9 1.4 0.2]
Matched:
10 . array of iris_setosa_holder is:  [4.9 3.1 1.5 0.1]
Matched:
11 . array of iris_setosa_holder is:  [5.4 3.7 1.5 0.2]
Matched:
12 . array of iris_setosa_holder is:  [4.8 3.4 1.6 0.2]
Matched:
13 . array of iris_setosa_holder is:  [4.8 3.  1.4 0.1]
Matched:
14 . array of iris_setosa_holder is:  [4.3 3.  1.1 0.1]
Matched:
15 . array of iris_setosa_holder is:  [5.8 4.  1.2 0.2]
Matched:
16 . array of iris_setosa_holder is:  [5.7 4.4 1.5 0.4]
Matched:
17 . array of iris_setosa_holder is:  [5.4 3.9 1.3 0.4]
Matched:
18 . array of iris_setosa_holder is:  [5.1 3.5 1.4 0.3]
Matched:
19 . array of iris_setosa_holder is:  [5.7 3.8 1.7 0.3]
Matched:
20 . array of iris_setosa_holder is:  [5.1 3.8 1.5 0.3]
Matched:
21 . array of iris_setosa_holder is:  [5.4 3.4 1.7 0.2]
Matched:
22 . array of iris_setosa_holder is:  [5.1 3.7 1.5 0.4]
Matched:
23 . array of iris_setosa_holder is:  [4.6 3.6 1.  0.2]
Matched:
24 . array of iris_setosa_holder is:  [5.1 3.3 1.7 0.5]
Matched:
25 . array of iris_setosa_holder is:  [4.8 3.4 1.9 0.2]
Matched:
26 . array of iris_setosa_holder is:  [5.  3.  1.6 0.2]
Matched:
27 . array of iris_setosa_holder is:  [5.  3.4 1.6 0.4]
Matched:
28 . array of iris_setosa_holder is:  [5.2 3.5 1.5 0.2]
Matched:
29 . array of iris_setosa_holder is:  [5.2 3.4 1.4 0.2]
Matched:
30 . array of iris_setosa_holder is:  [4.7 3.2 1.6 0.2]
Matched:
31 . array of iris_setosa_holder is:  [4.8 3.1 1.6 0.2]
```

```
Matched:
32 . array of iris_setosa_holder is:  [5.4 3.4 1.5 0.4]
Matched:
33 . array of iris_setosa_holder is:  [5.2 4.1 1.5 0.1]
Matched:
34 . array of iris_setosa_holder is:  [5.5 4.2 1.4 0.2]
Matched:
35 . array of iris_setosa_holder is:  [4.9 3.1 1.5 0.1]
Matched:
36 . array of iris_setosa_holder is:  [5.  3.2 1.2 0.2]
Matched:
37 . array of iris_setosa_holder is:  [5.5 3.5 1.3 0.2]
Matched:
38 . array of iris_setosa_holder is:  [4.9 3.1 1.5 0.1]
Matched:
39 . array of iris_setosa_holder is:  [4.4 3.  1.3 0.2]
Matched:
40 . array of iris_setosa_holder is:  [5.1 3.4 1.5 0.2]
Matched:
41 . array of iris_setosa_holder is:  [5.  3.5 1.3 0.3]
Matched:
42 . array of iris_setosa_holder is:  [4.5 2.3 1.3 0.3]
Matched:
43 . array of iris_setosa_holder is:  [4.4 3.2 1.3 0.2]
Matched:
44 . array of iris_setosa_holder is:  [5.  3.5 1.6 0.6]
Matched:
45 . array of iris_setosa_holder is:  [5.1 3.8 1.9 0.4]
Matched:
46 . array of iris_setosa_holder is:  [4.8 3.  1.4 0.3]
Matched:
47 . array of iris_setosa_holder is:  [5.1 3.8 1.6 0.2]
Matched:
48 . array of iris_setosa_holder is:  [4.6 3.2 1.4 0.2]
Matched:
49 . array of iris_setosa_holder is:  [5.3 3.7 1.5 0.2]
Matched:
50 . array of iris_setosa_holder is:  [5.  3.3 1.4 0.2]
```

"irisdata.xml" icerisindekileri tutmak icin 50'ye 3'luk elemanlari sifir olan matrisler olusturduk. Odev icin sunulan veri icin programlanmistir. Veri sayisi ve bicimi degisirse kod kendisini ona gore uyduramaz. Bunun icin array olusturmak yerine dinamik veri yapilari kullanmak gerekirdi.

"regex" kutuphanesi kullanarak daha once "text" verisi icerisine yazdigimiz veride, ilk cicek sinifi olan Iris Setosa kelimesini aradik. Eger eslesme varsa text icindeki string veri biciminde olan sayilari, float veri bicimine cevirerek daha once "numpy.zeros()" ile olusturdugumuz array'e yazmasini istedik.

"soup" bir agac oldugu icin "obj" isaretcisini ".findNext()" fonksiyonuyla agactaki bir sonraki 'para' tag'ine sahip elemana ilerlettik. Isaret edilen yeni elemanin icerdigi veriyi yine ".getText()" ile aldik, ve karsilastirmayi tekrar yaptik. Bunlarin hepsinden once ".zeros()" array'imizin icerisine veri yazmak icin kullandigimiz "first_index" degiskeninin degerini bir yukselttik. Bu sekilde dongu baslatabiliriz ve dongu butun iris-setosa'lari bulana kadar devam edebilir.

In [5]:
```
word='Iris-versicolor'
a=re.search(word,text)
if a==None:
    print(a,'. No match for ', word, ' in', text,'.\n')
first_index=0
```

In [6]:
```
while a!=None:
    #print('Matched:')

    for i in range(0,4):
                iris_versicolor_holder[first_index,i]=float(text[(0+i*4):(3+i*4)])


    #print(first_index+1,'. array of iris_versicolor_holder is:\n ', iris_versicolor_holder[first

    first_index+=1

    #move to next item
    obj=obj.findNext('para')

    #update text variable
    text=obj.getText()

    #search for iris-versicolor in next item
    a=re.search(word,text)

    #then loop starts again for new item
```

In [7]:
```
word='Iris-virginica'
a=re.search(word,text)
if a==None:
    print(a,'. No match for ', word, ' in', text,'.\n')
first_index=0
```

In [8]:
```
while a!=None:
    #print('Matched:')

    for i in range(0,4):
                iris_virginica_holder[first_index,i]=float(text[(0+i*4):(3+i*4)])


    #print(first_index+1,'. array of iris_virginica_holder is:\n ', iris_virginica_holder[first_

    first_index+=1

    #move to next item
    obj=obj.findNext('para')

    #update text variable
    text=obj.getText()

    #search for iris-virginica in next item
    a=re.search(word,text)

    #then loop starts again for new item
```

Ortalama Deger Vektorlerinin Hesaplanmasi

Butun veriyi array'lere aktardiktan sonra islemler yapilabilir.

In [9]:
```
def calculate_averages(array):
    average=0

    # sum all the data of a feature
    for i in array:
```

```
        average=average+i

    # divide the result by number of elements
    average=average/50

    return average
```

In [10]:
```
average_virginica=calculate_averages(iris_virginica_holder)
print('Averages of the features of Iris Virginica are:')
print(average_virginica)
print('\nAv. of first feature:',average_virginica[0])
print('Av. of second feature:',average_virginica[1])
print('Av. of third feature:',average_virginica[2])
print('Av. of fourth feature:',average_virginica[3])
```

```
Averages of the features of Iris Virginica are:
[6.588 2.974 5.552 2.026]

Av. of first feature: 6.587999999999998
Av. of second feature: 2.9739999999999998
Av. of third feature: 5.552
Av. of fourth feature: 2.026
```

In [11]:
```
average_versicolor=calculate_averages(iris_versicolor_holder)
print('Averages of the features of Iris Versicolor are:')
print(average_versicolor)
print('\nAv. of first feature:',average_versicolor[0])
print('Av. of second feature:',average_versicolor[1])
print('Av. of third feature:',average_versicolor[2])
print('Av. of fourth feature:',average_versicolor[3])
```

```
Averages of the features of Iris Versicolor are:
[5.936 2.77  4.26  1.326]

Av. of first feature: 5.936
Av. of second feature: 2.7700000000000005
Av. of third feature: 4.26
Av. of fourth feature: 1.3259999999999998
```

In [12]:
```
average_setosa=calculate_averages(iris_setosa_holder)
print('Averages of the features of Iris Setosa are:')
print(average_setosa)
print('\nAv. of first feature:',average_setosa[0])
print('Av. of second feature:',average_setosa[1])
print('Av. of third feature:',average_setosa[2])
print('Av. of fourth feature:',average_setosa[3])
```

```
Averages of the features of Iris Setosa are:
[5.006 3.418 1.464 0.244]

Av. of first feature: 5.005999999999999
Av. of second feature: 3.4180000000000006
Av. of third feature: 1.464
Av. of fourth feature: 0.2439999999999999
```

Tanimlanan fonksiyon icerisinde for loop yapilarak her sinifin her rastgele degiskeninin ortalama degeri hesaplanir.

Kovaryans Matrisinin Hesaplanmasi

Dort tane rastgele degisken oldugu icin her birinin kendiyle ve diger degiskenlerle kovaryansi, dorde dortluk

bir matris olusturur.

Kovaryans Matrisini olusturan fonksiyon tanimlanir.

In [13]:
```
def calculate_covariances(array,mean):

    # subtract the mean from each sample of the random variable
    subtracted=array-mean

    # create arrays to hold the sums
    a_holder=np.zeros((1,4))
    b_holder=np.zeros((1,4))
    c_holder=np.zeros((1,4))
    d_holder=np.zeros((1,4))

    # multiply each element with other elements
    # manual, this can be done better
    # need to figure out element wise multiplication
    for i in subtracted:
        for x in range(4):
            a_holder[0,x]=a_holder[0,x]+i[0]*i[x]
            b_holder[0,x]=b_holder[0,x]+i[1]*i[x]
            c_holder[0,x]=c_holder[0,x]+i[2]*i[x]
            d_holder[0,x]=d_holder[0,x]+i[3]*i[x]

    # divide each sum by one less than sample size
    covarianceMatrix=np.zeros((4,4))
    covarianceMatrix[0]=a_holder/(50-1)
    covarianceMatrix[1]=b_holder/(50-1)
    covarianceMatrix[2]=c_holder/(50-1)
    covarianceMatrix[3]=d_holder/(50-1)

    return covarianceMatrix
```

In [14]:
```
covarianceMatrix_IrisSetosa = calculate_covariances(iris_setosa_holder,average_setosa)
print('The Covariance Matrix of Iris Setosa is:')
print(covarianceMatrix_IrisSetosa)

covarianceMatrix_IrisVersicolor = calculate_covariances(iris_versicolor_holder,average_versicolo
print('\nThe Covariance Matrix of Iris Versicolor is:')
print(covarianceMatrix_IrisVersicolor)

covarianceMatrix_IrisVirginica = calculate_covariances(iris_virginica_holder,average_virginica)
print('\nThe Covariance Matrix of Iris Virginica is:')
print(covarianceMatrix_IrisVirginica)
```

```
The Covariance Matrix of Iris Setosa is:
[[0.12424898 0.10029796 0.01613878 0.01054694]
 [0.10029796 0.14517959 0.01168163 0.01143673]
 [0.01613878 0.01168163 0.03010612 0.00569796]
 [0.01054694 0.01143673 0.00569796 0.01149388]]

The Covariance Matrix of Iris Versicolor is:
[[0.26643265 0.08518367 0.18289796 0.05577959]
 [0.08518367 0.09846939 0.08265306 0.04120408]
 [0.18289796 0.08265306 0.22081633 0.07310204]
 [0.05577959 0.04120408 0.07310204 0.03910612]]

The Covariance Matrix of Iris Virginica is:
[[0.40434286 0.09376327 0.3032898  0.04909388]
 [0.09376327 0.10400408 0.07137959 0.04762857]
 [0.3032898  0.07137959 0.30458776 0.04882449]
 [0.04909388 0.04762857 0.04882449 0.07543265]]
```

Her sinif icin ayri kovaryans matrisi olusturulur. Her oznitelige butun orneklemelerinden, oznitelige ait elde edilen ortalama cikartilir.Sonrasinda her oznitelige her orneklemesi sirasiyla diger oznitelikler ve kendisiyle carpilir. Her carpim birbirine eklenerek her oznitelige ait 1x4 array olusturulur.

Sirasiyla her bir elemani birbiriyle carpip yeni 4x4 bir matriste tutmak suan zor geldigi icin hepsini teker teker ayri arraylerde tuttum. Ileride bu islem de tek bir satirda yapilabilir.

Kovaryans Matrisinin Kosegen Elemanlarinin Anlami

Kovaryans matrisindeki butun kosegen elemanlar, rastgele degisken olan oznitelikler varyanslaridir.

Varyansi az olan oznitelikler sisteme kendini tekrarlattigi, "redundancy" getirdigi soylenebilir. Bu sebeple degeri dusuk olan kosegen elemanlarin karsilik geldigi oznitelikler duruma gore modelde odak noktasindan cikartilabilir.

Kovaryans Matrisinin Kosegen Disindaki Elemanlari

Soru: Kovaryans matrisinin kosegen disindaki elemanlari sifir mi? Hayir degil. Bu demektir ki oznitelikler birbirleri arasinda korelasyona sahip. Yani degisimleri birbirleriyle baglantili.

Veri setimizdeki ciceklerin oznitelikleri arasinda korelasyon var. Ciceklerin oznitelikleri arasinda kovaryanslari sifirdan farkli. Bundan dolayi kovaryans matrisin kosegen disindaki elemanlari sifirdan farklidir.

In [ ]: