

CancerTesting

- A feladat, hogy páciensek adatai alapján egy egyszer? jóslatot adjunk arról, hogy az illet? beteg-e, vagy nem.
- Minden betegnek van sok különböz? attribútuma.
- Ezeket egy sokoszlopos file-ban tároljuk.
- Egy páciens akkor nyilvánítunk betegnek, ha minimum 5 attribútumának értéke az adott attribútum átlaga feletti.
- A file-okban 1 + 10 + 1 oszlop szerepel.
- Az els? oszlop a beteg ID-ja, a következ? 10 a különböz? attribútumokat jelöli, az utolsó pedig az "http://wiki.math.bme.huigazság"http://wiki.math.bme.hu, 1/0 érték arról, hogy a páciens valóban beteg-e.
- Számoljuk ki a "http://wiki.math.bme.hu/cancerTrainData.txt"http://wiki.math.bme.hu adatai alapján az egyes attribútumok átlagait.
- Ezek után jósoljuk meg, hogy a "http://wiki.math.bme.hu/cancerTestData.txt"http://wiki.math.bme.hu-ben rögzített páciensek betegek-e!
- Vessük össze predikciónkat a valósággal (órán részletesen).
- File-ok:
- cancerTrainData
- cancerTestData

```
import sys
import math

def reader(fileName):
    dataSet =
    open(fileName, 'r')
    for line in f:
        strippedLine = line.
        split("L" line.
        [] data =
    for l in L:
        append(float(l)) data.
        append(dataSet.
    return dataSet

def learn(trainingData):
    averages =
    for i in range(0,10):
        append(averages.
    for user in trainingData:
        for i in range(0,10):
            [i]+=user[i+1] averages
    for i in range(10):
        [i] = averages[i] / float(len(trainingData))
    return averages

def predict(averages,testingData):
    predictions =
    for test in testingData:
        aboveAverage = 0
        for i in range(0,10):
            if(test[i+1]>averages[i]):
                aboveAverage+=1
    id = test[0]
    int(test[1:11]) =
    if aboveAverage > len(averages)*0.5:
        append([id, predictions.
```

```

else:
    append([id, tp, tn, fp, fn], predictions)
return predictions

def accuracy(predictions):
    acc = 0
    for prediction in predictions:
        if prediction[1] == prediction[2]:
            acc += 1
    return acc/float(len(predictions))

def learnAndPredict():
    trainingData = sys.argv[1]
    testingData = sys.argv[2]
    averages = train(trainingData)
    predictions = (averages, testingData)
print "Accuracy: ", accuracy(predictions)

learnAndPredict()

```