

```
In [1]: import numpy as np
import math
import csv
import pdb
def read_data(filename):

    with open(filename, 'r') as csvfile:
        datareader = csv.reader(csvfile)
        metadata = next(datareader)
        traindata=[]
        for row in datareader:
            traindata.append(row)

    return (metadata, traindata)

def splitDataset(dataset, splitRatio):
    trainSize = int(len(dataset) * splitRatio)
    trainSet = []
    testset = list(dataset)
    i=0
    while len(trainSet) < trainSize:
        trainSet.append(testset.pop(i))
    return [trainSet, testset]
def classify(data, test):

    total_size = data.shape[0]
    print("\n")
    print("training data size=", total_size)
    print("test data size=", test.shape[0])

    countYes = 0
    countNo = 0
    probYes = 0
    probNo = 0
    print("\n")
    print("target count probability")

    for x in range(data.shape[0]):
        if data[x, data.shape[1]-1] == 'yes':
            countYes +=1
```

```

        if data[x,data.shape[1]-1] == 'no':
            countNo +=1
    probYes=countYes/total_size
    probNo= countNo / total_size

    print('Yes',"\\t",countYes,"\\t",probYes)
    print('No',"\\t",countNo,"\\t",probNo)

    prob0 =np.zeros((test.shape[1]-1))
    prob1 =np.zeros((test.shape[1]-1))
    accuracy=0
    print("\\n")
    print("instance prediction target")

    for t in range(test.shape[0]):
        for k in range (test.shape[1]-1):
            count1=count0=0
            for j in range (data.shape[0]):
                #how many times appeared with no
                if test[t,k] == data[j,k] and data[j,data.shape[1]-1]=='no':
                    count0+=1
                #how many times appeared with yes
                if test[t,k]==data[j,k] and data[j,data.shape[1]-1]=='yes':
                    count1+=1
            prob0[k]=count0/countNo
            prob1[k]=count1/countYes

    probno=probNo
    probyes=probYes
    for i in range(test.shape[1]-1):
        probno=probno*prob0[i]
        probyes=probyes*prob1[i]
    if probno>probyes:
        predict='no'
    else:
        predict='yes'

    print(t+1,"\\t",predict,"\\t ",test[t,test.shape[1]-1])
    if predict == test[t,test.shape[1]-1]:
        accuracy+=1
    final_accuracy=(accuracy/test.shape[0])*100

```

```

    print("accuracy",final_accuracy,"%")
    return

metadata,traindata= read_data("tennis1.csv")
splitRatio=0.6
trainingset, testset=splitDataset(traindata, splitRatio)
training=np.array(trainingset)
print("\n The Training data set are:")
for x in trainingset:
    print(x)

testing=np.array(testset)
print("\n The Test data set are:")
for x in testing:
    print(x)
classify(training,testing)

```

The Training data set are:

```

['sunny', 'hot', 'high', 'weak', 'no']
['sunny', 'hot', 'high', 'strong', 'no']
['overcast', 'hot', 'high', 'weak', 'yes']
['rainy', 'mild', 'high', 'weak', 'yes']
['rainy', 'cool', 'normal', 'weak', 'yes']
['rainy', 'cool', 'normal', 'strong', 'no']
['overcast', 'cool', 'normal', 'strong', 'yes']
['sunny', 'mild', 'high', 'weak', 'no']

```

The Test data set are:

```

['sunny' 'cool' 'normal' 'weak' 'yes']
['rainy' 'mild' 'normal' 'weak' 'yes']
['sunny' 'mild' 'normal' 'strong' 'yes']
['overcast' 'mild' 'high' 'strong' 'yes']
['overcast' 'hot' 'normal' 'weak' 'yes']
['rainy' 'mild' 'high' 'strong' 'no']

```

training data size= 8

test data size= 6

target count probability

Yes	4	0.5
No	4	0.5

```
instance prediction target
1          no          yes
2          yes          yes
3          no          yes
4          yes          yes
5          yes          yes
6          no          no
accuracy 66.66666666666666 %
```

In []: