**6.** **Write a program to implement the naïve Bayesian classifier for a sample training**

**data set stored as a .CSV file. Compute the accuracy of the classifier, considering**

**few test data sets.**

THEORY: **Naive bayes classifier**

It is supervised learning algorithm used for classification based on Baye s

Theorem.

NBC is not just an algorithm, but a collection of many algorithms

that work on the same concept, the BayesTheorem.

**Bayes theorem**

NBS works only on the bass theorem. Let’s see what the bayes theorem is:

                                    P(H/E) =  P(E/H) P(H)/P(E)

H- Hypothesis  , E-Event / Evidence ,P(H) - It is said priori (A prior

probability), Probability of H before E is happen.P(H/E) - Posterior

probability, Probability of E  after event E is true.

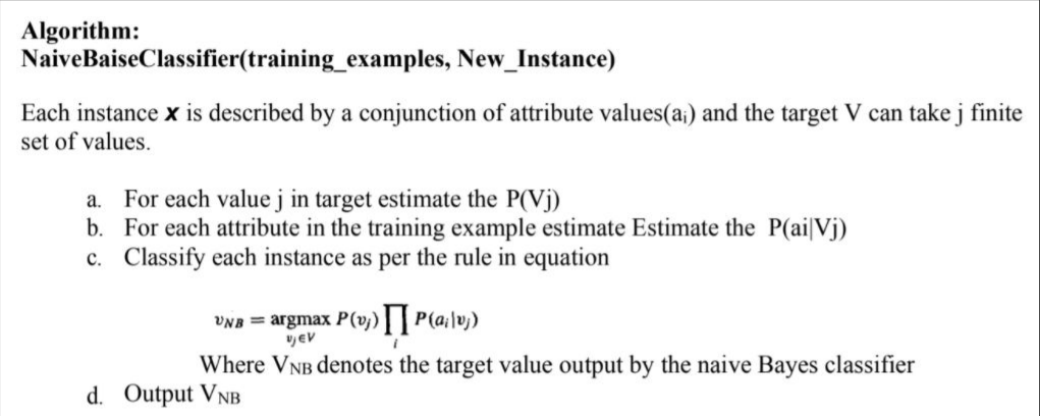
Bayes Theorem works on conditional probability.We have been given that

if the event has happened or the event is true, then we have to calculate the

probability of Hypothesis on this even.Means the chances of happening H when

the event E is happened.

**Naive Bayesian Classifier Algorithm :**



**PROCEDURE/PROGRAM:**

import numpy as np

import math

import csv

**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("training data size=",total\_size)

    print("test data size=",test.shape[0])

    target=np.unique(data[:,-1])

    count = np.zeros((target.shape[0]), dtype=np.int32)

    prob = np.zeros((target.shape[0]), dtype=np.float32)

    print("target count probability")

    for y in range(target.shape[0]):

        for x in range(data.shape[0]):

            if data[x,data.shape[1]-1] == target[y]:

                count[y] += 1

        prob[y]=count[y]/total\_size *# comptes the probability of target*

        print(target[y],"\t",count[y],"\t",prob[y])

    prob0 = np.zeros((test.shape[1]-1), dtype=np.float32)

    prob1 = np.zeros((test.shape[1]-1), dtype=np.float32)

    accuracy=0

    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]):

                if test[t,k]== data[j,k] and data[j,data.shape[1]-1]== target[0]:

                    count0+=1

                elif test[t,k]== data[j,k] and data[j,data.shape[1]-1]== target[1]:

                    count1+=1

            prob0[k]= count0/count[0]

            prob1[k]= count1/count[1]

        probno=prob[0]

        probyes=prob[1]

        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("PlayTennis.csv")

splitRatio = 0.6

trainingset, testset = splitDataset(traindata, splitRatio)

training=np.array(trainingset)

testing=np.array(testset)

print("------------------Training Data ------------------ ")

print(trainingset)

print("-------------------Test Data ------------------ ")

print(testset)

classify(training,testing)

------------------Training Data ------------------

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

-------------------Test Data ------------------

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

training data size= 8

test data size= 6

target count probability

no 4 0.5

yes 4 0.5

Instance prediction target

1 no yes

accuracy 0.0 %

2 yes yes

accuracy 16.666666666666664 %

3 no yes

accuracy 16.666666666666664 %

4 yes yes

accuracy 33.33333333333333 %

5 yes yes

accuracy 50.0 %

6 no no

accuracy 66.66666666666666 %