

```

import pandas as pd
import numpy as np
import random

def main():

    df1=pd.read_csv("SPECT.csv")
    df1=df1.values

    np.random.shuffle(df1)#randomly shuffling the training examples
    X=df1[:,1:]
    Y=df1[:,0]
    nx=X.shape[1]
    m=X.shape[0]
    Y=Y.reshape((m,1))

    print("Number Of features : ",nx,"\nNumber of training examples : ",m)
    NB(X,Y)

def NB(X,Y):
    nx=X.shape[1]
    m=X.shape[0]
    Y=Y.reshape((m,1))
    test_size=m//10

    tp,tn,fp,fn=0,0,0,0

    for fold in range(10):
        X_test = X[(fold*test_size):((fold+1)*test_size),:]
        X_train = np.concatenate((X[0:(fold*test_size),:],X[(fold+1)*test_size:m,:])
, axis=0)

        Y_test = Y[(fold*test_size):((fold+1)*test_size),:]
        Y_train = np.concatenate((Y[0:(fold*test_size),:],Y[(fold+1)*test_size:m,:])
, axis=0)

        m_test,m_train = X_test.shape[0],X_train.shape[0]

        total_yes=0
        total_no=0
        for i in range(m_train):
            if(Y_train[i][0]=='Yes'):
                total_yes+=1
            else:
                total_no+=1

        prob_yes = total_yes/m_train
        prob_no = total_no/m_train

        prob_yes_1,prob_no_1,prob_yes_0,prob_no_0 = [0]*nx,[0]*nx,[0]*nx,[0]*nx

        for i in range(m_train):

            if Y_train[i]=='Yes':
                for k in range(nx):
                    if(X_train[i][k]==1):
                        prob_yes_1[k]+=1
                    else:
                        prob_yes_0[k]+=1

            else:
                for k in range(nx):
                    if(X_train[i][k]==1):
                        prob_no_1[k]+=1
                    else:

```

```
prob_no_0[k] += 1

for k in range(nx):
    prob_yes_1[k] = prob_yes_1[k]/prob_yes
    prob_yes_0[k] = prob_yes_0[k]/prob_yes
    prob_no_1[k] = prob_no_1[k]/prob_no
    prob_no_0[k] = prob_no_0[k]/prob_no

pred_classes=[]
for i in range(m_test):
    prob_yes_given_features=prob_yes
    prob_no_given_features=prob_no

    for k in range(nx):
        if(X_test[i][k]==1):
            prob_yes_given_features *= prob_yes_1[k]
            prob_no_given_features *= prob_no_1[k]
        else:
            prob_yes_given_features *= prob_yes_0[k]
            prob_no_given_features *= prob_no_0[k]

    pred_class = 'Yes'
    if(prob_yes_given_features<prob_no_given_features):
        pred_class='No'
    pred_classes.append(pred_class)

for i in range(m_test):

    if(Y_test[i]=='Yes' and pred_classes[i]=='Yes'):
        tp+=1
    elif(Y_test[i]=='Yes' and pred_classes[i]=='No'):
        fn+=1
    elif(Y_test[i]=='No' and pred_classes[i]=='No'):
        tn+=1
    else:
        fp+=1

accuracy = (tp+tn)/(tp+tn+fp+fn)
precision = tp/(tp + fp)
recall = tp/(tp + fn)

print("Accuracy : ",accuracy)
print("Recall : ",recall)
print("Precision : ",precision)

if __name__=='__main__':
    main()
```