Practical-2

Objective:

implement Bayes Classifier using Single Feature.

Description:

1. Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

2. The formula for Bayes' theorem is given as:

$$P(A|B)=P(B|A) * P(A) / P(B)$$

3. P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

4. P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

5. P(B) is Marginal Probability: Probability of Evidence. 6. Suppose we have a dataset of Status of Subject Class and corresponding target variable "Held Status". So using this dataset we need to decide that whether that class happened or not on a particular day. So to solve this problem, we need to follow the below steps: -

Generate Likelihood table by finding the probabilities of given features.
Now, use Bayes theorem to calculate the posterior probability.

Dataset:

1. Have a dataset of Subject Names and corresponding target variable "Yes/No/Canceled" based on class held Status.

2. Have defined three Subject name i.e. Natural Language Processing, Mobile Computing, and Network and Web Security .

3. Have three target values that describes the Status of Class Held : Yes/No/Canceled.

Implementation:

```
#Import libraries
import array
import sys
import math

#Create Feature & Class List
list1 = ["NLP","MC","MC","NLP","NWS","NWS","NLP","MC","MC","NLP"] list2 =
[“Y”,"N","C","Y","Y","N","C","Y","N","C"]

#Calculating Distinct class count
countY=list2.count('Y')
countN=list2.count('N')
countC=list2.count(‘C')

#Calculating Class Probabilities
X=countY+countN+countC
P_Y=countY/X
P_N=countN/X
P_C=countC/X
count_NLP_Y=0
count_NLP_N=0
count_NLP_C=0
count_MC_Y=0
count_MC_C=0
count_MC_N=0
count_NWS_Y=0
count_NWS_C=0
count_NWS_N=0

for i in range(len(list1)):
        if(list1[i] == "NLP" and list2[i] == "Y"):
                count_NLP_Y=count_NLP_Y+1
        elif(list1[i] == "NLP" and list2[i] == "N"):
                count_NLP_N=count_NLP_N+1
        elif(list1[i] == "NLP" and list2[i] == "C"):
                count_NLP_C=count_NLP_C+1
        elif(list1[i] == "MC" and list2[i] == "N"):
                count_MC_N=count_MC_N+1
        elif(list1[i] == "MC" and list2[i] == "Y"):
                count_MC_Y=count_MC_Y+1
        elif(list1[i] == "MC" and list2[i] == "C"):
                count_MC_C=count_MC_C+1
        elif(list1[i] == "NWS" and list2[i] == "N"):
                count_NWS_N=count_NWS_N+1
        elif(list1[i] == "NWS" and list2[i] == "Y"):
```

```
                count_NWS_Y=count_NWS_Y+1
    elif(list1[i] == "NWS" and list2[i] == "C"):
            count_NWS_C=count_NWS_C+1


P_NLP_Y=count_NLP_Y/countY
P_NLP_N=count_NLP_N/countN
P_NLP_C=count_NLP_C/countC
P_MC_Y=count_MC_Y/countY
P_MC_N=count_MC_N/countN
P_MC_C=count_MC_C/countC
P_NWS_Y=count_NWS_Y/countY
P_NWS_N=count_NWS_N/countN
P_NWS_C=count_NWS_C/countC

#Taking Test Value for Prediction
val=input("Enter Value MC/NLP/NWS : “)
if val == "NLP":
        a=P_NLP_Y*P_Y
        b=P_NLP_N*P_N
        c=P_NLP_C*P_C
elif val == "NWS":
        a=P_NWS_Y*P_Y
        b=P_NWS_N*P_N
        c=P_NWS_C*P_C
elif val == "MC":
        a=P_MC_Y*P_Y
        b=P_MC_N*P_N
        c=P_MC_C*P_C


if (a > b) and (a > c):
        predictedOutput="Y"
elif (b > a) and (b > c):
        predictedOutput="N"
elif (c > a) and (c > b):
        predictedOutput=“C"

print("Predicted Output : ",predictedOutput)
```

Output:

practical2_mlir.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[8] print("P(Y) =",P_Y)
    print("P(N) =",P_N)
    print("P(C) =",P_C)
```

```
P(Y) = 0.4
P(N) = 0.3
P(C) = 0.3
```

```
print("NLP/Y",P_NLP_Y)
print("NLP/N",P_NLP_N)
print("NLP/C",P_NLP_C)
print("MC/Y",P_MC_Y)
print("MC/N",P_MC_N)
print("MC/C",P_MC_C)
print("NWS/Y",P_NWS_Y)
print("NWS/N",P_NWS_N)
print("NWS/C",P_NWS_C)
```

```
NLP/Y 0.5
NLP/N 0.0
NLP/C 0.6666666666666666
MC/Y 0.25
MC/N 0.6666666666666666
MC/C 0.3333333333333333
NWS/Y 0.25
NWS/N 0.3333333333333333
NWS/C 0.0
```

✓ 0s complet

practical2_mlir.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[10] #Taking Test Value for Prediction
     val=input("Enter Value MC/NLP/NWS : ")
```

```
Enter Value MC/NLP/NWS : NWS
```

```
[12] if (a > b) and (a > c):
       predictedOutput="Y"
     elif (b > a) and (b > c):
       predictedOutput="N"
     elif (c > a) and (c > b):
       predictedOutput="C"
```

```
print("Predicted Output : ",predictedOutput)
```

```
Predicted Output :  Y
```