Experiment No: 4a

BAYE'S THEOREM

Aim: Write a program to illustrate the Baye's theorem •

Introduction : Applications of the theorem are widespread and not limited to the financial realm. As an example, Bayes' theorem can be used to determine !he accuracy of medical test results by taking into consideration how likely anygiven person is to have a disease and the general accuracy of the test. Bayes' theorem relies on incorporating prior probability distributions in order to generate posterior probabilities. Prior probability, in Bayesian statistical inference, is the probability of an event before new data is collected. This is the best rational assessment of the probability of an outcome based on the current knowledge before an experiment is performed. Posterior probability is the revised probability of an event occurring after taking into consideration new information. Posterior probability is calculated by updating the prior probability by using Bayes' theorem. In statistical terms, the posterior probability is the probability of event A occurring given that event B has occurred.

Bayes' theorem thus gives the probability ·of an event based on new information that is, or may be related, to that event. The formula can also be used to see how h probability of an event occurring is affected by hypothetical new information, supposing the new information will turn out to be true.

Formula:

The Bayes' theorem is expressed in the following formula:

P(C;/X)=(P(X/C;) *P/p(x)

Where:

• P(Ci|x) - the posterior probability of event Ci occurring,given eventx has occurred

. P(XICi) -the class conditional probability of eventX occurring, given event Ci has occurred

• P(C;) - the apriori probability of event Ci

• P(X) - the total probability of event c

Example:

Data 'D' is divided into two classes called INDIAN and CHINESE.

CALSS 1=INDIAN

COLOR HEIGHT COUNT
Dark Tall 90 Fair Short 20
Dark Short 50 Fair Tall 60
=220

CLASS 2=CHINESE

COLOR HEIGHT COUNT
Dark Tall 10 Fair Short 90
Dark Short 40 Fair Tall 40
=180

Given x=(Dark,short)

What is the probability of x whether it belongs to class1 or class2

P(X/INDIAN)=P((Dark,short)/INDAN)=50/220

P(INDIAN)=220/400

P(X/CHINESE)=P((Dark,short)/CHINESE)=40/180

P(CHINESE)=180/400

P(X)= P(X/INDAIN)"' P(INDII\N)+ P(X/CHINESE) P(CHINESE)

50/220 * 220/400+40/180 * 180/400=9/40

P(INDAIN/X)= (P(X/INDIAN)*P(INDIAN))/P(X)=S/9

P(CHINESE/X)= (P(X/CHINESE)*P(CHINESE))/P(X)=4/9

MAX{S/9,4/9}=5/9

Hence, X belongs to CLASS1 i.e.,INDIAN CLASS

PROGRAM:

```
#include<stdio.h>
#include<string.h>
void class(int ,int);
char
cls[10][20],titems[50][20][20],attr[10][20];
int pcount[20],count[10],fc=0,c=0;
float p[10],prob[20],pre[10],result[10];
int main()
{
char tup[15][20];
int i,j,n,tuples,k,ans=0,t=0;
printf("enter no of attributes:");
scanf("%d",&n);
printf("enter no of tuples:");
scanf("%d",&tuples);
printf("enter %d attributes\n",n);
for(i=0;i<n;i++)
scanf("%s",attr[i]);
for(i=0;i<tuples;i++)
{
printf("enter tuple%d\n",i+1);
for(j=0;j<n;j++)
scanf("%s",titems[i][j]);
}
printf("enter test tuple\n");
for(i=0;i<n-1;i++)
scanf("%s",tup[i]);
class(n,tuples);
for(i=0;i<fc;i++)
p[i]=count[i]/(float)tuples;
for(i=0;i<fc;i++)
{
for(j=1;j<n-1;j++)
{
pcount[j]=0; for(k=0;k<tuples;k++)
{
if(strcmp(titems[k][j],tup[j])==0 &&
strcmp(cls[i],titems[k][n-1])==0)
pcount[j]+=1;
}
if(pcount[j]!=0 && t==0)
prob[c++]=pcount[j]/(float)count[i]; else
{
t=1;
prob[c++]=(pcount[j]+1)/(float)count[i];
}
}
}
j=0;
for(i=0;i<fc;i++)
{
pre[i]=1.0;
for( ;j<((i+1)*(c/fc));j++) pre[i]*=prob[j];
}
for(i=0;i<fc;i++)
{
result[i]=pre[i]*p[i];
if(i>0 && result[i]>result[i-1]) ans=i;
}
printf("The test tuple belongs to %s
class",cls[ans]);
}
void class(int p,int q)
{
int i=0,k,t=0; strcpy(cls[fc++],titems[0][p-
1]);
for(i=1;i<q;i++)
{
t=0;
for(k=0;k<fc;k++)
{
if(strcmp(titems[i][p-1],cls[k])==0)
{
t=-1;
break;
}
}
if(t!=-1) strcpy(cls[fc++],titems[i][p-1]);
}
for(i=0;i<fc;i++)
{
count[i]=0;
for(k=0;k<q;k++)
{
if(strcmp(titems[k][p-1],cls[i])==0)
count[i]+=1; } } }
```

OUTPUT:

OUT PUT:

```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip  0, Program:    TC
enter no of attributes:4
enter no of tuples:5
enter 4 attributes
color type origin class
enter tuple1
red sports dom yes
enter tuple2
red sports dom no
enter tuple3
yello sports dom yew
enter tuple4
yello sports dom no
enter tuple5
green sports dom yes
enter test tuple
yello sports dom yes
The test tuple belongs to yes classenter no of attributes:4
enter no of tuples:1
enter 4 attributes
color type orign class
enter tuple1
blue sports dom yes
enter test tuple
blue sports dom yes
The test tuple belongs to yes classenter no of attributes:
```

Experiment No: 4b

NEAREST NEIGHBOUR CLASSIFICATION

Aim : Write a program to illustrate Nearest neighbor classification Algorithm.

INTRODUCTION:

The nearest neighbour rule is one of the oldest and simplest methods for pattern classification. Nevertheless, it often yields competitive results, and in certain domains, when cleverly combined with prior knowledge, it has significantly advanced the state-of-the• art. The NNC rule classifies each unlabelled example by the majoritylabel among its k-nearest neighbours the training set. Its performance thus depends crucially o he distance metric used to identify nearest neighbours.

Steps:

1. Store the entire training set (database).
2. Given a query pattern ''Q', find the distances from all patterns in the database or using any distance measure.
3. Sort the distances in ascending order.
4. Assign the class label for the 'Q' to the nearest neighbour.

Applications of KNN:

The following are some of the areas in which KNN can be applied successfully -

Banking System

KNN can be used in banking system to predict weather an individual is fit for loan approval? Does that individual have the characteristics similar to the defaulters one?

Calculating Credit Ratings

KNN algorithms can be used to find an individual's credit rating by comparing with the persons having similar traits.

Example:

Let a data base having the following attributes,

PID	Height in inches	Weight in kg's	TyPE of person
P1	5.2	70	obesic
P2	5.4	75	obesic
P3	5.6	78	obesic
P4	6.2	90	fit
PS	6.1	92	fit
P6	6.0	91	fit

query(Q)=(5.2,71)

Euclidean distance between points:((x1-x2)2+ (y1-y 2) 2)1 / 2

D(P1,Q)=D((S.2,70),(5.2,71))=1

D(P2,Q)=D((5.4,75),(5.2,71))=4.004

D(P3,Q)=D((5.6,78),(5.2,71))=7.011

D(P4,Q)=D((6.2,90),(5.2,71))=19.02

D(P5,Q)=D((6.1,92),(5.2,71))=20.01

D(P6,Q)=D((6.0,91),(5.2,71))=21.01

'Q' is assigned to the class of obesic person

PROGRAM:

```
#include <iostream.h>
#include<string.h>
int main()
{
int i, j, l;
int tsc = 12;
char gen[12] = { 'F','M', 'F','F','F','M','F','M','M','F','M','F' };
float h[12] = {1.6f,2.0f,1.9f,1.88f,1.7f,1.85f,1.6f,1.7f,2.2f,1.8f,1.95f,1.9f};
char op[12][10] = {"short","tall","medium","medium","short","medium","short","short","tall","medium","medium","medium"};
cout << "\n Initial Set:";
cout << "\nGender\tHeight\tOutput";
for (i = 0; i < 12; i++) {
cout << "\n" << gen[i] << "\t" << h[i] << "\t" << op[i];
}
float nh;
char ng;
cout << "\n Enter tuple to be processed (Height,Gender) :";
cin >> nh >> ng;
int t;
cout << "\n Enter threshold:";
cin >> t;
float d[12][2], k;
//calculating distance to each value in training set
for (i = 0; i < 12; i++) {
d[i][0] = i;
k = h[i] - nh;
if (k < 0) {
d[i][1] = -k;
} else {
d[i][1] = k;
}
}
//Sorting
for (i = 0; i < 11; i++) {
for (j = 0; j < 11; j++) {
if (d[j][1] > d[j + 1][1]) {
k = d[j][1];
d[j][1] = d[j + 1][1];
d[j + 1][1] = k;
l = d[j][0];
d[j][0] = d[j + 1][0];
d[j + 1][0] = l;
} } }
int nos = 0; //no of shorts
int nom = 0; //no of mediums
int nota = 0; //no of talls
cout << "\nGender\tHeight\tOutput\n";
for (i = 0; i < t; i++) {
l = d[i][0];
cout << gen[l] << "\t" << h[l] << "\t" << op[l] << "\n";
if (strcmp(op[l], "short") == 0) {
nos++;
}
if (strcmp(op[l], "medium") == 0) {
nom++;
}
if (strcmp(op[l], "tall") == 0) {
nota++;
}
}
cout << "\n No of shorts:" << nos;
cout << "\n No of medium:" << nom;
cout << "\n No of tall:" << nota;
if (nos > nom && nos > nota) {
cout << "\n New Tuple is classified as Short";
}
if (nom > nos && nom > nota) {
cout << "\n New Tuple is classified as Medium";
}

if (nota > nom && nota > nos) {
cout << "\n New Tuple is classified as Tall";
}

}
```

OUTPUT:

```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip  0, Program:     TC
F       1.6     short
F       1.6     short
F       1.6     short
F       1.6     short
F       1.6     short

 No of shorts:2253
 No of medium:13
 No of tall:12
 New Tuple is classified as Short
 Initial Set:
Gender  Height  Output
F       1.6     short
M       2       tall
F       1.9     medium
F       1.88    medium
F       1.7     short
M       1.85    medium
F       1.6     short
M       1.7     short
M       2.2     tall
F       1.8     medium
M       1.95    medium
F       1.9     medium
 Enter tuple to be processed (Height,Gender) :
```