IE 4903 Assignment 4

Bahar Atay-2188092

**Initialization**

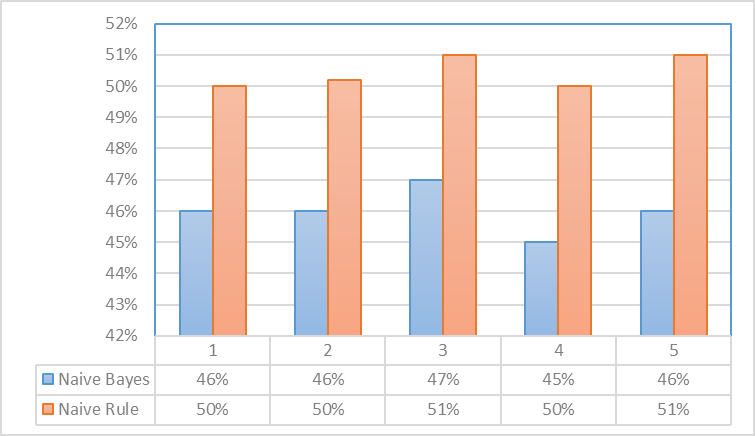
Data is partitioned into Train and Test using random sampling, 80% and 20% respectively.

**PART 1**

Using the data collected from 42183 accidents, Naive Bayes rule is implemented to classify the accidents on whether they result in no injuries, injuries or fatalities. According to Naive Bayes’ characteristic, independence assumption on the conditional probabilities are realized.

To benchmark the error, Naive Rule is also used on the data. The program is runned for 5 times for Navie Bayes and Naive Rule. The misclassification errors of the methods on same partitions can be seen in Table 1. The outcomes show that Naive Bayes classification has lower error rate. On the assumption that all attributes are indicators of the predicted attribute, Naive Rule gives the upper bound error. However, not always the attributes are meaningful in predicting another attribute. In such a case, Naive Rule can give a lower error rate than Naive Bayes.

*Table 1 Error Outcomes of Naive Bayes and Naive Rule Classifiers*



**PART 2**

Using the data collected from 131 wines, Naive Bayes rule is implemented to classify the wines according to their types. However, because the data is numerical, in order to categoritize the data, discretization is performed. Each attribute is divided into equisized bins. Number of bins, n, is decided by plotting histograms of the attributes using various bin sizes. The bin size which results in a histogram that indicates the distribution of the dataset is chosen. In this case, n is chosen to be 5.

The histograms of the first four attributes for bin size n=4 are given in Figure 1. It can be seen that n=4 is a reasonable bin size which indicates the pattern of the distribution of data. Remaining attributes have similar histograms.

Attribute 1



Attribute 2



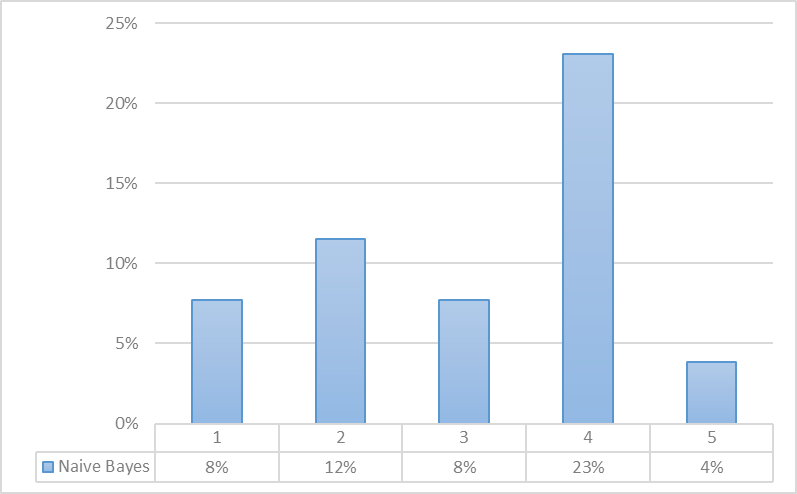
Attribute 4



Attribute 3

*Figure 1 Histograms of First Four Attributes at n=4*

The misclassification errors for five runs can be seen in Table 2. The errors in average are smaller than the errors in Part 1. The reasons for this might be the characteristics of the data, since they are different. Or data size can be another factor. Data size of the Wine data is much smaller than the Accidents data.



*Table 2 Error Outcomes of Naive Bayes on Wine Data*

**MATLAB CODE**

**PART 1**

[alldata]=xlsread('Accidents.xls','Data');

[r,c]=size(alldata);

%%Train and Test Data Partitioning

rIndex=randperm(r);

sizeTrain=ceil(r\*0.8);

sizeTest=r-sizeTrain;

ctr=0;

cte=0;

for i=1:r

if i<=sizeTrain

ctr=ctr+1;

train(ctr,:)=alldata(rIndex(i),:);

else

cte=cte+1;

test(cte,:)=alldata(rIndex(i),:);

end

end

%NAIVE RULE

for i=1:cte

if test(i,14)==1

abc(i,1)=1;

else

abc(i,1)=0;

end

end

true=0;

for i=1:cte

if abc(i,1)==1

true=true+1;

else

true=true;

end

end

errornrule=1-(true/cte);

%END OF NAIVE RULE

num(1)=sum(train(:,14)==0);

num(2)=sum(train(:,14)==1);

num(3)=sum(train(:,14)==2);

new1=zeros(num(1),14);

new2=zeros(num(2),14);

new3=zeros(num(3),14);

k=1;

for i=1:ctr

if train(i,14)==0

new1(k,:)=train(i,:);

k=k+1;

end

end

t=1;

for i=1:ctr

if train(i,14)==1

new2(t,:)=train(i,:);

t=t+1;

end

end

l=1;

for i=1:ctr

if train(i,14)==2

new3(l,:)=train(i,:);

l=l+1;

end

end

for i=1:cte

prob(i,1,1)=num(1)/33747;

for j=1:13

prob(i,1,1)=prob(i,1,1)\*(sum(new1(:,j)==test(i,j))/num(1));

end

end

for i=1:cte

prob(i,1,2)=num(2)/33747;

for j=1:13

prob(i,1,2)=prob(i,1,2)\*(sum(new2(:,j)==test(i,j))/num(2));

end

end

for i=1:cte

prob(i,1,3)=num(3)/33747;

for j=1:13

prob(i,1,3)=prob(i,1,3)\*(sum(new3(:,j)==test(i,j))/num(3));

end

end

for i=1:cte

if prob(i,1,1)>=prob(i,1,2) && prob(i,1,1)>=prob(i,1,3)

pred(i,1)=0;

else if prob(i,1,2)>prob(i,1,1) && prob(i,1,2)>prob(i,1,3)

pred(i,1)=1;

else if prob(i,1,3)>=prob(i,1,1) && prob(i,1,3)>=prob(i,1,2)

pred(i,1)=2;

end

end

end

end

for i=1:cte

if pred(i,1)==test(i,14)

valid(i,1)=1;

else

valid(i,1)=0;

end

end

errornbayes=sum(valid(:,1)==0)/cte;

**PART 2**

[winedata]=xlsread('Wine.xls','Data');

[r,c]=size(winedata);

for j=1:13

incr(j)=(max(winedata(:,j))-min(winedata(:,j)))/4;

end

% for i=1:13

% x=winedata(:,i);

% histogram(x,4);

% figure(i);

% end

for j=1:13

for i=1:r

a=min(winedata(:,j));

for k=1:4

if a<=winedata(i,j) && winedata(i,j)<a+incr(j)

new(i,j)=k;

a=a+incr(j);

else

a=a+incr(j);

end

end

end

end

%%Train and Test Data Partitioning

rIndex=randperm(r);

sizeTrain=ceil(r\*0.8);

sizeTest=r-sizeTrain;

ctr=0;

cte=0;

for i=1:r

if i<=sizeTrain

ctr=ctr+1;

train(ctr,:)=new(rIndex(i),:);

else

cte=cte+1;

test(cte,:)=new(rIndex(i),:);

end

end

k=1;

for i=1:ctr

if rIndex(i)<=60

new1(k,:)=train(i,:);

k=k+1;

end

end

l=1;

for i=1:ctr

if rIndex(i)>60

new2(l,:)=train(i,:);

l=l+1;

end

end

[m,n]=size(new1);

[p,q]=size(new2);

for i=1:cte

prob(i,1,1)=m/sizeTrain;

for j=1:13

prob(i,1,1)=prob(i,1,1)\*(sum(new1(:,j)==test(i,j))/m);

end

end

for i=1:cte

prob(i,1,2)=p/sizeTrain;

for j=1:13

prob(i,1,2)=prob(i,1,2)\*(sum(new2(:,j)==test(i,j))/p);

end

end

for i=1:cte

if prob(i,1,1)>=prob(i,1,2)

pred(i,1)=1;

else

pred(i,1)=2;

end

end

for i=105:130

if rIndex(i)<=60

actual(i-104,1)=1;

else

actual(i-104,1)=2;

end

end

for i=1:cte

if actual(i,1)==pred(i,1)

result(i,1)=1;

else

result(i,1)=0;

end

end

error= sum(result(:,1)==0)/cte;