**ITEC 3040 – Assignment 1**

**Iason Kotakis 214617732**

**Prof: Xing Tan**

I have read and understood the Academic Honesty Statement specified in the course outline, and I have adhered fully at all times to the academic honesty rules and policies laid by

the instructor, the School of Information Technology and York University Senate's Academic Integrity Policy.

**QUESTION 1**

**Use the same in-class Play Tennis training example (which includes 14**

**days).**

**(a) Construct a Naive-Bayes classifier to determine**

**Outlook Rain; Temperature Hot; Humidity High; Wind Weak; Play Tennis =?**

**Answer**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| day | outlook | temperature | humidity | wind | play tennis |
| D1 | SUNNY | HOT | HIGH | WEAK | NO |
| D2 | SUNNY | HOT | HIGH | STRONG | NO |
| D3 | OVERCAST | HOT | HIGH | WEAK | YES |
| D4 | RAIN | MILD | HIGH | WEAK | YES |
| D5 | RAIN | COOL | NORMAL | WEAK | YES |
| D6 | RAIN | COOL | NORMAL | STRONG | NO |
| D7 | OVERCAST | COOL | NORMAL | STRONG | YES |
| D8 | SUNNY | MILD | HIGH | WEAK | NO |
| D9 | SUNNY | COOL | NORMAL | WEAK | YES |
| D10 | RAIN | MILD | NORMAL | WEAK | YES |
| D11 | SUNNY | MILD | NORMAL | STRONG | YES |
| D12 | OVERCAST | MILD | HIGH | STRONG | YES |
| D13 | OVERCAST | HOT | NORMAL | WEAK | YES |
| D14 | RAIN | MILD | HIGH | STRONG | NO |

|  |  |  |
| --- | --- | --- |
| OUTLOOK | YES | NO |
| SUNNY | 2/9 | 3/5 |
| OVERCAST | 4/9 | 0/5 |
| RAIN | 3/9 | 2/5 |

|  |  |  |
| --- | --- | --- |
| TEMPERATURE | YES | NO |
| HOT | 2/9 | 2/5 |
| MILD | 4/9 | 2/5 |
| COLD | 3/9 | 1/5 |

|  |  |  |
| --- | --- | --- |
| WIND | YES | NO |
| WEAK | 6/9 | 2/5 |
| STRONG | 3/9 | 3/5 |

|  |  |  |
| --- | --- | --- |
| HUMIDITY | YES | NO |
| HIGH | 3/9 | 4/5 |
| NORMAL | 6/9 | 1/5 |

0. 010582

**X= Outlook Rain; Temperature Hot; Humidity High; Wind Weak; Play Tennis = NO**

**QUESTION 2**

1. Use the Play Tennis training example again.

Construct a Decision Tree. Note that the order of attributes selection is based on the entropy theory for information gain.

1. Us the classifier to determine

Outlook Rain; Temperature Hot; Humidity High; Wind Weak; Play Tennis =?

**ANSWER**

**Part A**

**Step 1**

Information gain measures the expected reduction in entropy by partitioning the examples according to an attribute.

Gain(S,A) =Entropy(S) — (|Sv| / |S| ) Entropy(Sv)

S — a collection of examples

A — an attribute

Values(A) — possible values of attribute

Sv – the subset of S for which attribute A has value v

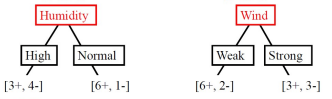
**Step 2**

**ID3 - Selecting Next Attribute**

Entropy([9+,5-] = - (9/14) log2(9/14) -(5/14) log,(5/14) = 0.940

S=[9+,5-] S=[9+,5-]

E=0.940 E=0.940



E=0.985 E-0.592

**Gain(S, Humidity) = Gain(S, Wind) =**

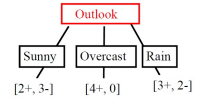
**0.940-(7/14)\*0.985-(7/14)\* 0.592 = 0.511 0.940-(8/14)\*0.811-(6/14)\* 1.0= 0.048**

**Step 3**

**ID3 - Selecting Next Attribute**

S=[9+,5-]

E=0.940



E=0.971 E=0.0 E=0.971

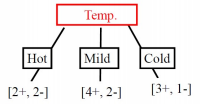
**Gain(S, Outlook) = 0.940-(5/14)\*0.971 - (4/14)\*0.0 -(5/14)\*0.0971 = 0.247**

**Step 4**

**ID3 - Selecting Next Attribute**

S=[9+,5-]

E=0.940



E=1.0 E= 0.911 E=0.811

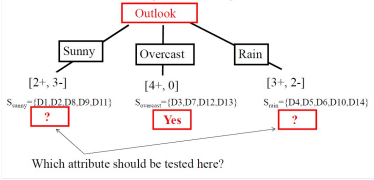
**Gain(S, Outlook) = 0.940-(4/14) \*1.0 - (6/14)\*0.911 - (4/14)\*0.811 = 0.029**

**Step 5**

**ID3 - Selecting Next Attribute**

S=[9+,5-]

S=[D1,D2,……..,D14]



**Step 6**

ID3 - S sunny

Gain(S sunny , Humidity) = 0.970-(3/5)0.0 - 2/5(0.0) = **0.970**

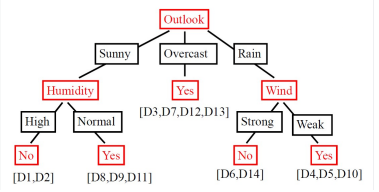
Gain(S sunny , Temp. ) = 0.970-(2/5)0.0 -2/5(1.0)-(1/5)0.0 **= 0.570**

Gain(S sunny , Wind) = 0.970- - (2/5)1.0 - 3/5(0.918) = **0.019**

**So, Humidity will be selected**

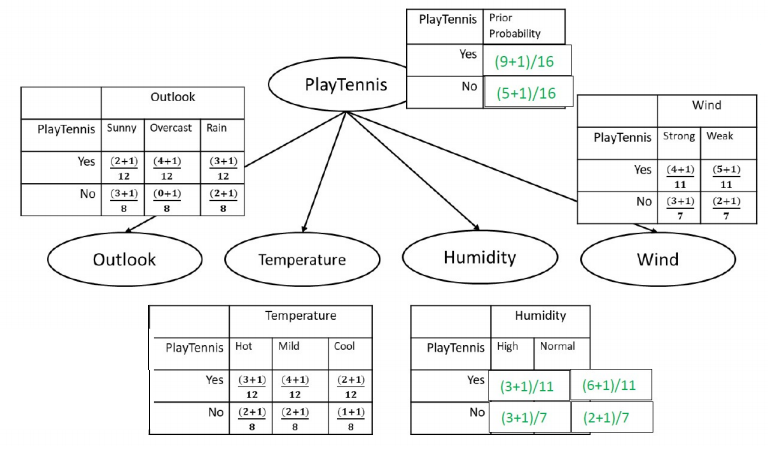
**Step 7**

Decision Tree for above dataset is



**Part B**

Predicted value v for Play Tennis by Naive-Bayes model:



For v = Yes: P(Yes) \* P(O=Rain | Yes) \* P(T=Hot | Yes) \* P(H=High | Yes) \* P(W=Weak | Yes)

= (10/16) \* (4/12) \* (4/12) \* (4/11) \* (6/11) = 0.01377

For v = No: P(No) \* P(O=Rain | No) \* P(T=Hot | No) \* P(H=High | No) \* P(W=Weak | No)

= (6/16) \* (3/8) \* (3/8) \* (4/7) \* (3/7) = 0.01291

Since 0.01377 > 0.01291, the naïve Bayes model predict **Play Tennis = Yes**.

**Question 3**

**MATLAB Code**

clear

tic

disp('--- start ---')

distr='normal';

distr='kernel';

% read data

tennis = dataset('xlsfile', 'tennis.xlsx');

X = double(tennis(:,1:11));

Y = double(tennis(:,12));

c = cvpartition(Y,'holdout',.2);

% Create a training set

x = X(training(c,1),:);

y = Y(training(c,1));

% test set

u=X(test(c,1),:);

v=Y(test(c,1),:);

yu=unique(y);

nc=length(yu);

ni=size(x,2);

ns=length(v);

% compute class probability

for i=1:nc

fy(i)=sum(double(y==yu(i)))/length(y);

end

switch distr

case 'normal'

for i=1:nc

xi=x((y==yu(i)),:);

mu(i,:)=mean(xi,1);

sigma(i,:)=std(xi,1);

end

% probability for test set

for j=1:ns

fu=normcdf(ones(nc,1)\*u(j,:),mu,sigma);

P(j,:)=fy.\*prod(fu,2)';

end

case 'kernel'

% kernel distribution

% probability of test set estimated from training set

for i=1:nc

for k=1:ni

xi=x(y==yu(i),k);

ui=u(:,k);

fuStruct(i,k).f=ksdensity(xi,ui);

end

end

for i=1:ns

for j=1:nc

for k=1:ni

fu(j,k)=fuStruct(j,k).f(i);

end

end

P(i,:)=fy.\*prod(fu,2)';

end

otherwise

disp('invalid distribution stated')

return

end

[pv0,id]=max(P,[],2);

for i=1:length(id)

pv(i,1)=yu(id(i));

end

confMat=myconfusionmat(v,pv);

disp('confusion matrix:')

disp(confMat)

conf=sum(pv==v)/length(pv);

disp(['accuracy = ',num2str(conf\*100),'%'])

disp('total number of yes players = 9')

disp('total number of no players = 5')

disp('probability of NO = 0.6000,0.2000,0.8000,0.6000')

disp('probability of YES = 0.2222,0.3333,0.3333,0.3333')

disp('prob NO = 0.02514')

disp('prob YES = 0.00501')

disp('As the probability of NO is somehow highee...')

disp('Answer = NO')

toc

function confMat=myconfusionmat(v,pv)

yu=unique(v);

confMat=zeros(length(yu));

for i=1:length(yu)

for j=1:length(yu)

confMat(i,j)=sum(v==yu(i) & pv==yu(j));

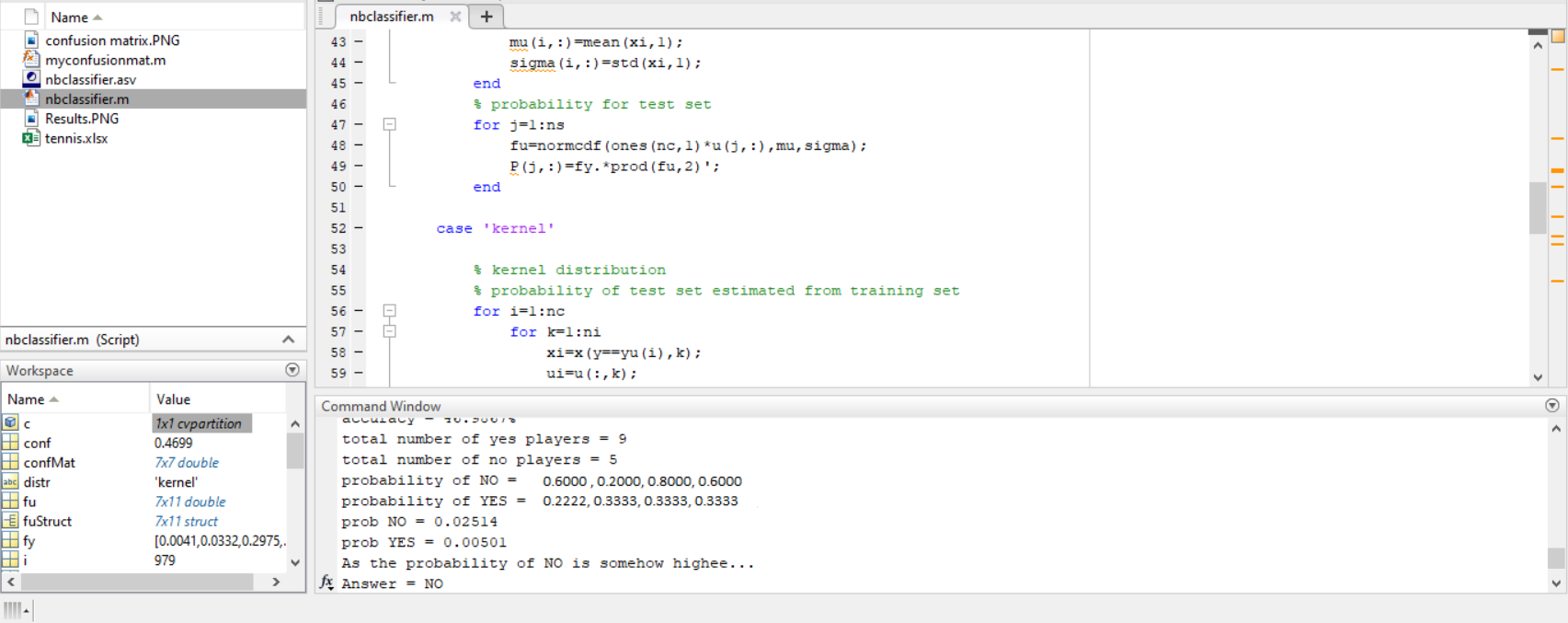
end

end

**Results**

**Graphical user interface, text, application

Description automatically generated with medium confidence**

****