CSC6850: Machine Learning
Homework – 1:
Mokter Hossain

Q-1. Max Posteriori Hypothesis

When a test for steroids is given to soccer players, 98% of the players taking steroids test positive and 12% of the players are not taking steroids test positive. Suppose that 5% of soccer players take steroids. What is the maximum posteriori hypothesis for a soccer player who tests positive? What are the exact posterior probabilities?

Solution:

$P(+ \mid \neg \text{steroids}) = 0.12 (\text{False Positive}), \quad P(- \mid \neg \text{Steroids}) = 0.88 (\text{True Positive})$

$P(+ \mid \text{Steroids}) = 0.98 \ (\text{True Positive}), \quad P(- \mid \text{Steroids}) = 0.02 \ (\text{False Negative})$

$P(+ \mid \text{Steroids}) * P(\text{Steroids}) = (0.98) * (0.05) = 0.049 \approx 5\%$

$P(+ \mid \neg \text{Steroids}) * P(\neg \text{Steroids}) = (0.12) * (0.95) = 0.114 \approx 11\%$

So, $h_{MAP} = \text{argmax}_{h_i \in H} \{5\%, 11\%\} = 11\%$

⇨ $h_{MAP} = \neg \text{Steroids}$

Q-2. Naive Bayes Classifier

2.1 Consider the hypothesis space defined over these instances (Table 1), in which each hypothesis is represented by a pair of 4-tuples. Using the native Bayes classifier to predict the target value PlayTennis = Yes/No to the following instance.

a) Compute <Sunny, Mild, Normal, Weak>
b) Compute <Rain, Cool, High, Strong>

CSC6850: Machine Learning
Homework – 1:
Mokter Hossain

Table 1: Training examples for the target concept PlayTennis

Day	Outlook	Temperature	Humidity	Wind	Play
D1	Sunny	Hot	Normal	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	Normal	Strong	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Mild	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Rain	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Strong	Yes
D10	Rain	Hot	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Sunny	Mild	Normal	Weak	No
D13	Overcast	Hot	Normal	Weak	Yes
D14	Sunny	Cool	High	Weak	No

Solution:
(A) Overall Distribution Table

Outlook	Temperature	Humidity	Wind	Play
Sunny (6/14) Overcast (3/14) Rain (5/14)	Hot (5/14) Mild (5/14) Cool (4/14)	Normal (10/14) High (4/14)	Weak (7/14) Strong (7/14)	Yes (8/14) No (6/14)

(B) Play/ No Play Distribution Table

Outlook	Temperature	Humidity	Wind
Sunny / Yes (2/8) Sunny / No (4/6)	Hot / Yes (3/8) Hot /No (2/6)	Normal / Yes (7/8) Normal /No (3/6)	Weak / Yes (3/8) Weak /No (4/6)
Overcast / Yes (2/8) Overcast / No (1/6)	Mild / Play (2/8) Mild /No (3/6)	High / Yes (1/8) High / No (3/6)	Strong / Yes (5/8) Strong /No (2/6)
Rain / Yes (3/8) Rain / No (2/6)	Cool / Yes (3/8) Cool / No (1/6)		

We know, **Naive Bayes Classifier [**$P(c|x)$

$P(c \mid x) = (P(x \mid c) * P(c)) / P(x)$

Where, C =Play|¬Play

Here, P(Play) = 8/14, and P(No) 6/14

P(Play | X)=(Play | <Sunny, Mild, Normal, Weak>)
= P(Sunny | Play) *P (Mild | Play) *P (Normal | Play) *P (Weak | Play) *P(Play)
= (2/8) * (2/8) * (7/8) * (3/8) * (8/14)
= 0.0117
≈ 1.17%

P(Not Play | X)=(No Play| <Sunny, Mild, Normal, Weak>)
= P(Sunny | No) *P (Mild | No) *P (Normal | No) *P (Weak | No) *P(No)
= (4/6) * (3/6) * (3/6) * (6/6) * (6/14)
= 0.0476
≈ 4. 76%

Then, (Play | Yes) = 0.0117 / (0.0117 + 0.0476) = 0.0117 / 0.0593
≈ 0.1973 ≈ 20 %

Then, (Play | No) = 0.047 / (0.0117 + 0.0476) = 0.0476 / 0.0593
≈ 0.8026 ≈ 80 %

Since (Play | Yes) < (Play | No) => No Play with this condition.

b) Compute <Rain, Cool, High, Strong>

P(Play | X)=(Play | <Rain, Cool, High, Strong>)
= P(Rain | Play) *P (Cool | Play) *P (High | Play) *P (Strong | Play) *P(Play)
= (3/8) * (3/8) * (1/8) * (5/8) * (8/14)
= (3*3*1*5*8)/(8*8*8**814)

= 360 / 7168 = 0.006277
≈ 0.63%

P(Not Play | X)=(No Play| <Rain, Cool, High, Strong>)
= P(Rain | No) *P (Cool | No) *P (High | No) *P (Strong | No) *P(No)
= (2/6) * (1/6) * (3/6) * (2/6) * (6/14)
= (2*1*3*2*6)/ (6*6*6*6*14)
= 72 / 18144
= 0.00396
≈ 0.4%

Then, (Play | No) = 0.00627 / (0.00627 + 0.00396) = 0.00627 / 0.01023
≈ 0.61290 ≈ 61 %

Then, (Not Play | No) = 0.00396 / (0.00627 + 0.00396) = 0.00396 / 0.01023
≈ 0.38709 ≈ 39 %

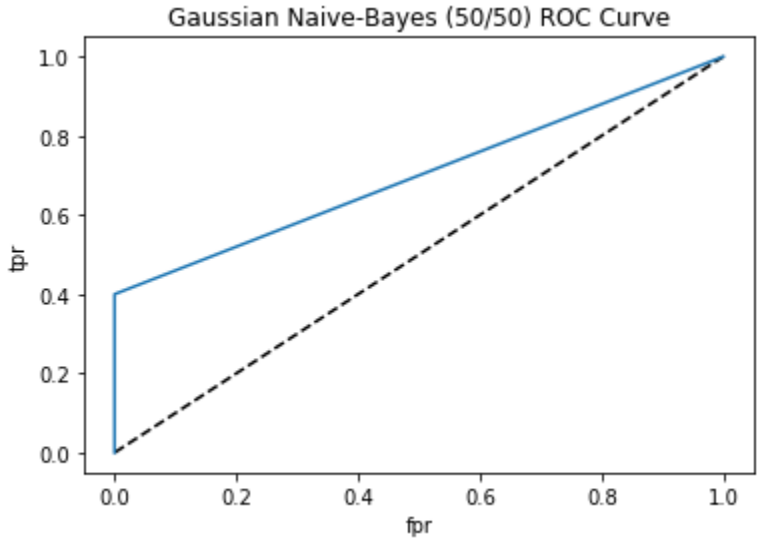
Since (Play | No) > (Not Play | No) => Play with this condition.

Q-2.2: Consider the following example and calculate the accuracy of the classifier with precision, recall, F1-score, specificity and ROC curve using Python.

[Please see Program codes in the attached file]

Output in Brief:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	0
1	1.00	0.33	0.50	3
accuracy			0.33	3
macro avg	0.50	0.17	0.25	3
weighted avg	1.00	0.33	0.50	3



Q-3. KNN (20 Points + 10 Points)

3.1 Table 2 is a gene expression microarray data, where each row represents a sample for a person, each column represents a gene, and each entry represents gene expression value of a gene over a sample. The output of each sample indicates whether a specific disease exists in this sample. Please use Euclidean Distance-Weighted KNN (K=3) to predict the output of the sample 11 and sample 12.

Sample	G1	G2	G3	G4	G5	G6	G7	Output
1	1.0	2.3	5.2	1.2	5.3	2.6	2.3	Yes
2	2.0	3.6	1.8	2.3	1.6	2.1	1.5	No
3	1.5	1.5	4.1	1.3	1.2	3.1	1.6	Yes
4	2.2	1.9	9.5	1.5	1.5	4.2	1.4	No
5	3.9	2.4	5.3	1.7	1.6	2.5	2.9	Yes
6	5.1	3.6	2.7	2.6	1.7	2.8	3.4	Yes
7	1.8	4.2	3.6	3.5	1.6	3.4	1.3	No
8	2.3	1.5	7.2	4.1	7.1	3.1	1.8	No
9	4.2	2.4	6.2	2.9	2.5	3.3	2.5	Yes
10	3.6	5.6	1.9	3.2	2.6	5.2	2.7	No

Table 2: Gene expression data

Sample	G1	G2	G3	G4	G5	G6	G7	Output
11	2.1	2.2	3.2	1.4	5.1	2.4	1.4	?
12	2.4	2.3	3.4	3.8	2.3	5.7	5.2	?

Solution:

We know that the Euclidean Distance [$D(x,y)$$D(x,y)$**]**

$$D(x,y)= \sqrt{[\ (x_1-y_1)^2+(x_2-y_2)^2+\ldots+(x_n-y_n)^2\ ]}$$

Let's suppose:

x = sample <G1, G2, G3, G4, G5, G6, G7>
y1 = sample$_{11}$<2.1, 2.1, 3.2, 1.4, 5.1, 2.4, 1.4>
y2 = sample$_{12}$<2.4, 2.3, 3.4, 3.8, 2.3, 5.7, 5.2>

For the KNN Distance: K =

$$D(1, 11) = \sqrt{[\ (<1,G1>-<11,G1>)^2+(<1,G2>-<11,G2>)^2+\ldots+(<1,G7>-<11,G7>)^2]}$$
$$\approx 2.489$$

$$D(2, 11)= \sqrt{[(<2, G1>-<11,G1>)^2 + (<2,G2>-<11,G2>)^2+\ldots+(<2,G7>-<11,G7>)^2\ ]}\approx 4.134$$

Similarly,
D(3, 11) ≈ 4.172
D(4, 11) ≈ 7.476
D(5, 11) ≈ 4.723
D(6, 11) ≈ 5.33
D(7, 11) ≈ 4.681
D(8, 11) ≈ 5.336
D(9, 11) ≈ 4.958
D(10, 11) ≈ 5.875

For K = 3,

3 Nearest Neighbors of sample-11 = <sample1, sample2, sample3>

Which as Yes, No, and Yes, respectively.

- ⇨ Sample11 Output should be Yes

Similarly, for the given Sample12,

$D(1, 12) = \sqrt{[(<1,G1> - <11,G1>)^2 + (<1,G2> - <11,G2>)^2 + \ldots + (<1,G7> - <11,G7>)^2]} \approx 6.2436$

$D(2, 12) = \sqrt{[(<2,G1> - <11,G1>)^2 + (<2,G2> - <11,G2>)^2 + \ldots + (<2,G7> - <11,G7>)^2]} \approx 5.8129$

Similarly,
D(3, 12) ≈ 5.395
D(4, 12) ≈ 7.747
D(5, 12) ≈ 5.129
D(6, 12) ≈ 4.789
D(7, 12) ≈ 4.008
D(8, 12) ≈ 5.518
D(9, 12) ≈ 4.998
D(10, 12) ≈ 4.604

For K = 3,

3 Nearest Neighbors of the Sample-12 = < Sample-6, Sample-7, sample-10>

Which as Yes, No, and No, respectively.

- ⇨ Sample12 Output should be No

I found same output using program. Please see in the attached program file.

Q-3.2: Consider the following example and calculate the accuracy of the classifier with precision, recall, F1-score, specificity and ROC curve using Python.

[Please see Program codes in the attached file]

Output in Brief:

Train Accuracy Score: 0.91

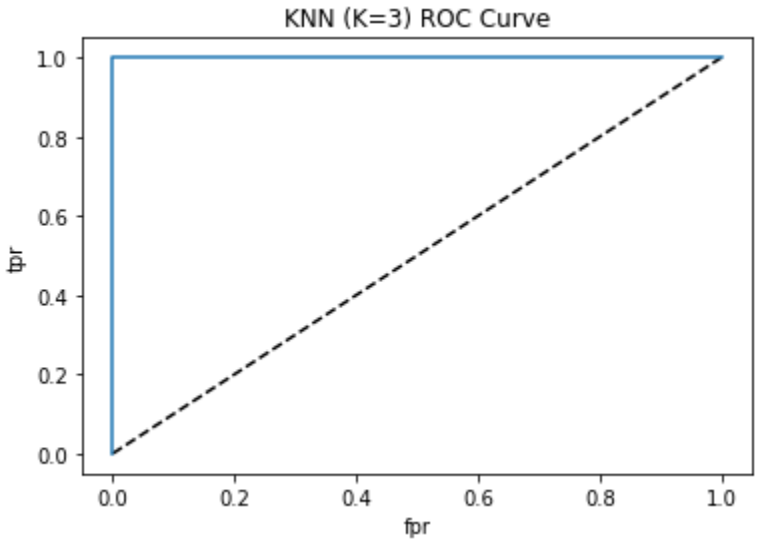
Test Accuracy Score: 0.3333333333333333

Accuracy Score 0.3333333333333333

Precision Score 1.0

Recall Score 0.3333333333333333

F1 Score 0.5



Q-4.1. Consider the hypothesis space defined over these instances (Table 3), in which each hypothesis is represented by a pair of 5-tuples. Please provide a hand trace of the ID3 algorithm to build a Decision Tree Classifier.

CSC6850: Machine Learning
Homework – 1:
Mokter Hossain

Table 3: Training examples for the target concept PlayTennis

D	Outlook	Temperature	Humidity	Wind	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	Yes
D3	Rain	Hot	High	Weak	Yes
D4	Rain	Mild	Normal	Strong	No
D5	Rain	Cool	High	Weak	No
D6	Rain	Mild	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Cool	High	Weak	No
D9	Sunny	Cool	High	Strong	Yes
D10	Rain	Mild	High	Strong	Yes

Solution:
(A) Overall Distribution Table

Outlook	Temperature	Humidity	Wind	Play
Sunny (4/10) Overcast (1/10) Rain (5/10)	Hot (3/10) Mild (3/10) Cool (4/10)	Normal (3/10) High (7/10)	Weak (4/10) Strong (6/10)	Yes (5/10) No (5/10)

(B) Play/ No Play Distribution Table

Outlook	Temperature	Humidity	Wind
Sunny / Yes (2/5) Sunny / No (2/5)	Hot / Yes (2/5) Hot /No (1/5)	Normal / Yes (1/5) Normal /No (2/5)	Weak / Yes (1/5) Weak /No (3/5)
Overcast / Yes (1/5) Overcast / No (0/5)	Mild / Play (1/5) Mild /No (2/5)	High / Yes (4/5) High / No (3/5)	Strong / Yes (4/5) Strong /No (2/5)
Rain / Yes (2/5) Rain / No (3/5)	Cool / Yes (2/5) Cool / No (2/5)		

Entropy (Single Attribute):
$E(S) = -\sum p_i \log p_i$
Entropy (Multi Attributes):
$E(T,X) = \sum_{c \in X} P(c) E(c) = P(X_1) \cdot c_1 \log_2 c_1 + P(X_2) \cdot c_2 \log_2 c_2 + \ldots + P(X_n) \cdot c_n \log_2 c_n$

Information Gain

$G(T,X)=E(T)-E(T,X)$

Best Split Point:
$SplitInfo_A(D)=\sum |D_j|/|D| \cdot \log_2(|D_j|/|D|)$

Gain Ratio:
$GainRatio(A)=Gain(A)/SplitInfo(A)$

Outlook	Temperature	Humidity	Wind	Play
Sunny (4/10) Overcast (1/10) Rain (5/10)	Hot (3/10) Mild (3/10) Cool (4/10)	Normal (3/10) High (7/10)	Weak (4/10) Strong (6/10)	Yes (5/10) No (5/10)

Outlook	Temperature	Humidity	Wind
Sunny / Yes (2/5) Sunny / No (2/5)	Hot / Yes (2/5) Hot /No (1/5)	Normal / Yes (1/5) Normal /No (2/5)	Weak / Yes (1/5) Weak /No (3/5)
Overcast / Yes (1/5) Overcast / No (0/5)	Mild / Play (1/5) Mild /No (2/5)	High / Yes (4/5) High / No (3/5)	Strong / Yes (4/5) Strong /No (2/5)
Rain / Yes (2/5) Rain / No (3/5)	Cool / Yes (2/5) Cool / No (2/5)		

Now, Let's draw the ID3 Algorithm

Play, Outlook
$E(Play, Outlook)=P(Outlook)*E(Outlook_{,\neg p})$

$=P(Sunny) * E(Sunny_p, Sunny_{\neg p}) + P(Overcast) * E(Overcast_p, Overcast_{\neg p}) + P(Rain)* E(Rain_p, Rain_{\neg p})$
$=(4/10) * E(2/5, 2/5) + (1/10) * E(1/5, 0/5) + (5/10) * E(2/5, 3/5)$

$=(0.4)*(1) + (0.1)*0 + (.5)*(0.97095) \approx 0.8855$

$G(Play, Outlook) = E(Play)-E(Play, Outlook)$
$= E(5/10, 5/10) - 0.8855$
$= 1-0.8855 \approx 0.1145 \approx 11\%$

Play, Temperature

E(Play, Temperature)=P(Temperature)*E(Temperature$_{,\neg p}$)

=P(Hot) * E(Hot$_p$, Hot$_{\neg p}$) + P(Mild) * E(Mild$_p$, Mild$_{\neg p}$) + P(Cool)* E(Cool$_p$, Cool$_{\neg p}$)

=(3/10) * E(2/5, 1/5) + (3/10) * E(1/5, 2/5) + (4/10) * E(2/5, 2/5)

=(0.3)(0.9932)+(0.3)(0.9932)+(.4)(1) ≈ 0.8

G(Play, Temperature) = E(Play)−E (Play, Temperature)
= E (5/10, 5/10) − 0.8
= 1−0.8 ≈ 0.2 ≈ 20%

Play, Humidity

E(Play, Humidity)=P(Humidity)*E(Humidity$_{,\neg p}$)

=P(Normal) * E(Normal$_p$, Normal$_{\neg p}$) + P(High) * E(High$_p$, High$_{\neg p}$)

=(3/10) * E(1/5, 2/5) + (7/10) * E(4/5, 3/5)

=(0.3)(0.9932)+(0.7)(0.6997)
= 0.2979 + 0.4898 ≈ 0.7877

G(Play, Temperature) = E(Play)−E (Play, Humidity)
= E (5/10, 5/10) − 0.7877
= 1−0.8 ≈ 0.2123 ≈ 21%

Play, Wind

E(Play, Wind)=P(Wind)*E(Wind$_{,\neg p}$)

=P(Weak) * E(Weak$_p$, Weak$_{\neg p}$) + P(Strong) * E(Strong$_p$, Strong$_{\neg p}$)

=(4/10) * E(1/5, 3/5) + (6/10) * E(4/5, 2/5)

=(0.4)(0.9066)+(0.6)(0.7863)
= 0.3626 + 0.4718 ≈ 0.8344

G(Play, Wind) = E(Play)−E (Play, Wind)
= E (5/10, 5/10) − 0. 8344
= 1−0.8344 ≈ 0.1655 ≈ 17%

Thus, the largestInfoGain is for the Wind attribute. So Wind will be the root node.

For Wind = Strong

Outlook	Temperature	Humidity	Play
Sunny	Hot	High	Yes
Sunny	Cool	High	Yes
Rain	Mild	High	Yes
Rain	Mild	Normal	No
Rain	Mild	Normal	No
Overcast	Cool	Normal	Yes

Play, Outlook
E(Play, Outlook)=P(Outlook)*E($\text{Outlook}_{,\neg p}$)

=P(Sunny) * E(Sunny_p, $\text{Sunny}_{\neg p}$) + P(Overcast) * E(Overcast_p, $\text{Overcast}_{\neg p}$) + P(Rain)* E(Rain_p, $\text{Rain}_{\neg p}$)
=(2/6) * E(2/4, 0/2) + (1/6) * E(1/4, 0/2) + (3/6) * E(1/3, 2/3)

=(0.3333)*(0) + (0.1667)*0 + (.5)*(0.9193) ≈ 0.46

G(Play, Outlook) = E(Play)−E (Play, Outlook)
= E (4/6, 2/6) − 0.46
= 0.9193 1− 0.46 ≈ 0. 4593 ≈ 46%

Play, Temperature
E(Play, Temperature)=P(Temperature)*E($\text{Temperature}_{,\neg p}$)

=P(Hot) * E(Hot$_p$, Hot$_{\neg p}$) + P(Cool) * E(Cool$_p$, Cool$_{\neg p}$) + P(Mild)* E(Mild$_p$, Mild$_{\neg p}$)
=(2/6) * E(2/4, 0/2) + (1/6) * E(1/4, 0/2) + (3/6) * E(1/4, 2/2)
=(0.3333)*(0) + (0.1667)*0 + (0.5)*(0.5) ≈ 0.25

G(Play, Temperature) = E(Play)−E (Play, Temperature)
= E (4/6, 2/6) − 0.25
= 0.9193 1− 0.25 ≈ 0. 6693 ≈ 67%

Play, Humidity
E(Play, Humidity)=P(Humidity)*E(Humidity$_{,\neg p}$)

=P(High) * E(High, High $_{\neg p}$) + P(Normal) * E(Normal, Normal $_{\neg p}$)
=(3/6) * E(3/4, 0/2) + (3/6) * E(1/3, 2/3)
=(0.5)*(0) + (0.5)*(0.9183) ≈ 0.46

G(Play, Humidity) = E(Play)−E (Play, Humidity)
= E (4/6, 2/6) − 0.46
= 0.9193 1− 0.46 ≈ 0. 46 ≈ 46%

Thus, the largestInfoGain is for the Wind = Strong attribute is the Temperature. So the Temperature will be used for the tree's second level

So, the next nodes will be Humidity => Outlook

Now for (Wind = Strong) and (Humidity = Normal)

Normal => Rain => No

Normal => Overcast => Yes

And for (Wind = Strong) and (Humidity = High) => Yes

For Wind = Weak

Wind = Weak

Outlook	Temperature	Humidity	Play
Sunny	Cool	High	No
Sunny	Hot	High	No
Rain	Hot	High	Yes
Rain	Cool	High	No

Play, Outlook

$E(Play, Outlook)=P(Outlook)*E(Outlook_{,\neg p})$

$=P(Sunny) * E(Sunny_p, Sunny_{\neg p}) + P(Rain)* E(Rain_p, Rain_{\neg p})$
$=(2/4) * E(0/1, 2/2) + (2/4) * E(1/1, 1/3)$

$=(0.5)*(0) + (0.5)*0.5283 \approx 0.264$

$G(Play, Outlook) = E(Play) - E(Play, Outlook)$
$= E(1/4, 3/4) - 0.264$
$= 0.8112 - 0.264 \approx 0.5470 \approx 54\%$

Play, Temperature

$E(Play, Temperature)=P(Temperature)*E(Temperature_{,\neg p})$

$=P(Hot) * E(Hot_p, Hot_{\neg p}) + P(Cool) * E(Cool_p, Cool_{\neg p})$
$=(2/4) * E(1/1, 1/3) + (2/4) * E(0/1, 2/3)$
$=(0.5)*(0.5283) + (0.5)*0 \approx 0.264$

$G(Play, Temperature) = E(Play) - E(Play, Temperature)$
$= E(4/6, 2/6) - 0.264$
$= 0.9193\ 1 - 0.264 \approx 0.655 \approx 65\%$

Play, Humidity

$E(Play, Humidity)=P(Humidity)*E(Humidity_{,\neg p})$

$=P(High) * E(High, High_{\neg p})$

$=(4/4) * E(1/4, 3/4)$

$=(1)*(0.8112) \approx 0.8112$

$G(\text{Play, Humidity}) = E(\text{Play}) - E(\text{Play, Humidity})$
$= E(1/4, 3/4) - 0.8112$
$= 0.8112 - 0.8112 \approx 0.0 \approx 0\%$

Thus, the largestInfoGain is for the Wind = Weak attribute is the Temperature. So the Temperature will be used for the tree's second level

So, the next nodes will be Outlook and Temperature

Now for (Wind = Weak) and (Outlook = Sunny)

Sunny => Play = No

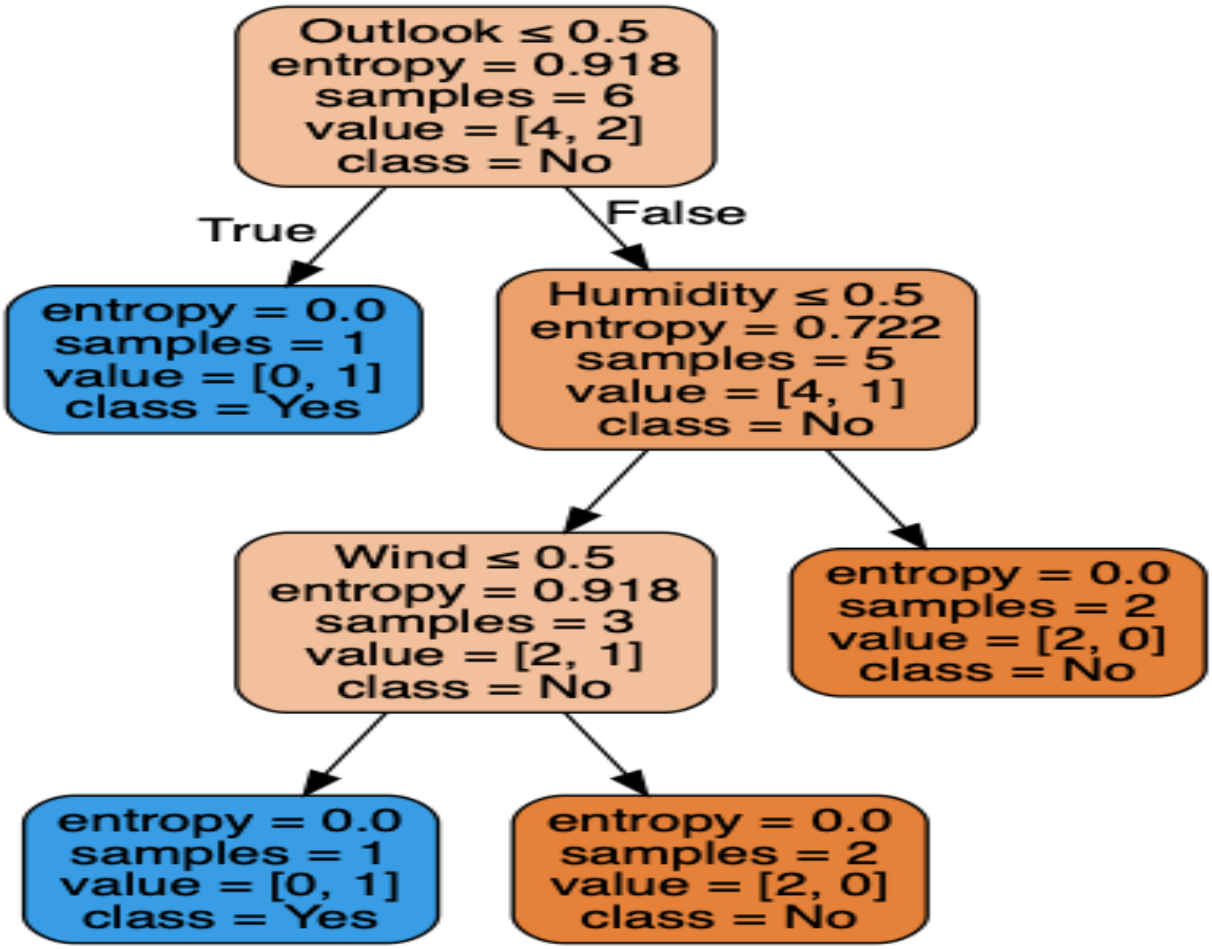
Normal => Overcast => Yes

And for (Wind = Weak) and (Outlook = Rain)

Rain => Hot => Play = Yes

Rain => Cool => Play = No

Major Output:



4.2 Consider the following example and calculate the accuracy of the classifier with precision, recall, F1-score, sensitivity, specificity and ROC curve using Python.

[Please see Program codes in the attached file]

Output in Brief:

Train Accuracy Score: 1.0

Test Accuracy Score: 0.5

Accuracy Score 1.0

Precision Score 1.0

Recall Score 1.0

F1 Score 1.0