1. [7 points] Problem 1.1 (Training Data for Boolean Classifier)

   (a) [5 points] Problem 1.1a (Boolean Decision Tree)

      i. ID3(S,Attributes = $\{x_1, x_2, x_3, x_4\}$,Label = $y \epsilon \{0, 1\}$):

         A. First Root Node - Check if all base cases are met: No.

         B. Otherwise - Create a root node using the "best attribute":

         - Entropy of S, H(S):

$$p = 2$$
$$n = 5$$
$$H(S) = -(2/7)\log_2(2/7) - (5/7)\log_2(5/7) = \mathbf{0.863120569}$$

For attribute x_1:

$x_1 = 0$ (5)

$$p = 1$$
$$n = 4$$
$$H = 0.721928095$$

$x_1 = 1$ (2)

$$p = 1$$
$$n = 1$$
$$H = 1$$

Expected Entropy = (5/7)(0.721928095) + (2/7)(1) = 0.801377211
Information Gain = 0.863120569 - 0.801377211 = **0.061743358**

For attribute x_2:

$x_2 = 0$ (3)

$$p = 2$$
$$n = 1$$
$$H = 0.918295834$$

$x_2 = 1$ (4)

$$p = 0$$
$$n = 4$$
$$H = 0$$

Expected Entropy = (3/7)(0.918295834) + (4/7)(0) = 0.393555357
Information Gain = 0.863120569 - 0.393555357 = **0.469565212**

For attribute x_3:

$x_3 = 0$ (4)

$$p = 1$$
$$n = 3$$
$$H = 0.811278124$$

$x_3 = 1$ (3)

$$p = 1$$
$$n = 2$$
$$H = 0.918295834$$

Expected Entropy = (4/7)(0.811278124) + (3/7)(0.918295834) = 0.857142857
Information Gain = 0.863120569 - 0.857142857 = **0.0.005977712**

For attribute x_4:

$x_4 = 0$ (4)

$$p = 0$$
$$n = 4$$
$$H = 0$$

$x_4 = 1$ (3)

$$p = 2$$
$$n = 1$$
$$H = 0.918295834$$

Expected Entropy = (4/7)(0) + (3/7)(0.918295834) = 0.393555357
Information Gain = 0.863120569 - 0.393555357 = **0.469565212**

The best attributes are x_2 or x_4 (both tie for highest information gain). Work is shown for x_2, but both tree diagrams are shown. For attribute x_2, two branches are created with values 0 and 1. Each value has its own subset, S_v.

ii. ID3(S_v where $x_2 = 1$, Attributes = $\{x_1, x_3, x_4\}$, Label = {0, 1}):

Return leaf node with label "$y = 0$" (i.e, if $x_2 = 1$ then $y = 0$).

iii. ID3(S_v where $x_2 = 0$, Attributes = $\{x_1, x_3, x_4\}$, Label = {0, 1}):

A. Second Node - Check if all base cases are met:No.

B. Otherwise - Create a node using the "best attribute":

- Entropy of S, H(S_v where $x_2 = 0$):

$$p = 2$$
$$n = 1$$
$$H(S_v) = -(2/3)\log_2(2/3) - (1/3)\log_2(1/3) = 0.918295834$$

For attribute x_1:

$x_1 = 0$ (2) $x_1 = 1$ (2)

$p = 1$, $n = 1$, $H(x_1 = 0) = 1$ | $p = 1$, $n = 0$, $H = 0$

Expected Entropy = (2/3)(1) + (1/3)(0) = 2/3
Information Gain = 0.918295834 - (2/3) = **0.251629167**

For attribute x_3:

$x_3 = 0$ (1) $x_3 = 1$ (2)

$p = 0$, $n = 1$, $H = 0$ | $p = 1$, $n = 1$, $H = 1$

Expected Entropy = (1/3)(0) + (2/3)(1) = 2/3
Information Gain = 0.918295834 - (2/3) = **0.251629167**

For attribute x_4:

$x_4 = 0$ (1) $x_4 = 1$ (2)

$p = 0$, $n = 1$, $H = 0$ | $p = 2$, $n = 0$, $H = 0$

Expected Entropy = (1/3)(0) + (2/3)(0) = 0
Information Gain = 0.918295834 - 0 = **0.918295834**

The best attribute is x_4 (for the highest information gain). For attribute x_4, two branches are created with values 0 and 1. Each value has its own subset, S_v.

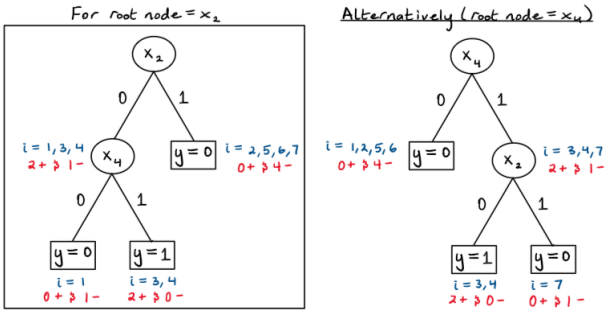
iv. ID3(S_v where $x_2 = 0$ and $x_4 = 1$, Attributes = $\{x_1, x_3\}$, Label = $\{0, 1\}$):

Return leaf node with label "$y = 1$" (i.e, if $x_2 = 0$ and $x_4 = 1$ then $y = 1$).

v. ID3(S_v where $x_2 = 0$ and $x_4 = 0$, Attributes = $\{x_1, x_3\}$, Label = $\{0, 1\}$):

Return leaf node with label "$y = 1$" (i.e, if $x_2 = 0$ and $x_4 = 0$ then $y = 0$).

vi. **We end with the following decision tree:**



(b) [2 points] Problem 1.1b (Boolean Function Representation)

Inputs				*Output*	Inputs				*Output*
x_1	x_2	x_3	x_4	y	x_1	x_2	x_3	x_4	y
0	0	0	0	0	1	0	0	0	0
0	0	0	1	1	1	0	0	1	1
0	0	1	0	0	1	0	1	0	0
0	0	1	1	1	1	0	1	1	1
0	1	0	0	0	1	1	0	0	0
0	1	0	1	0	1	1	0	1	0
0	1	1	0	0	1	1	1	0	0
0	1	1	1	0	1	1	1	1	0

2. [17 points] Problem 1.2 (Tennis Training Dataset)

(a) [7 points] Problem 1.2a (Majority Error Tree)

i. ID3(S,Attributes = {O, T, H, W}, Labels = $\{+, -\}$):

A. First Root Node - Check if all base cases are met: No.

B. Otherwise - Create a root node using the "best attribute":

- Majority Error of S, ME(S):

$$p = 9$$
$$n = 5$$
$$ME(S) = 5/14$$

For attribute Outlook:

Sunny (5)

$p = 2$
$n = 3$
$ME = 2/5$

Overcast (4)

$p = 4$
$n = 0$
$ME = 0$

Rainy (5)

$p = 3$
$n = 2$
$ME = 2/5$

Expected = (5/14)(2/5) + (4/14)(0) + (5/14)(2/5) = 2/7
Gain = 5/14 - 2/7 = 1/14 = **0.061743358**

For attribute Temperature:

Hot (4)

$p = 2$
$n = 2$
$ME = 1/2$

Medium (6)

$p = 4$
$n = 2$
$ME = 1/3$

Cold (4)

$p = 3$
$n = 1$
$ME = 1/4$

Expected = (4/14)(1/2) + (6/14)(1/3) + (4/14)(1/4) = 5/14
Gain = 5/14 - 5/14 = **0**

For attribute Humidity:

High (7)

$p = 3$
$n = 4$
$ME = 3/7$

Normal (7)

$p = 6$
$n = 1$
$ME = 1/7$

Low (0)

$p = 0$
$n = 0$
$ME = 0$

Expected = (7/14)(3/7) + (7/14)(1/7) + (0)(0) = 2/7
Gain = 5/14 - 2/7 = 1/14 = **0.061743358**

For attribute Wind:

Strong (6)

$p = 3$
$n = 3$
$ME = 1/2$

Weak (8)

$p = 6$
$n = 2$
$ME = 1/4$

Expected = (6/14)(1/2) + (8/14)(1/4) = 5/14
Gain = 5/14 - 5/14 = **0**

The best attributes are Outlook or Humidity (both tie for highest gain). Work is shown for Outlook as the root node. For the Outlook attribute, three branches are created with values S, O, and R. Each value has its own subset, S_v.

ii. ID3(S_v where Outlook = S, Attributes = {T, H, W}, Label = {+, -}):
   A. Node - Check if all base cases are met: No.
   B. Otherwise - Create a node using the "best attribute":
      - Majority Error of S_v, ME(S_v):

$p = 2$
$n = 3$
$ME(S) = 2/5$

For attribute Temperature:

Hot (2)

$p = 0$
$n = 2$
$ME = 0$

Medium (2)

$p = 1$
$n = 1$
$ME = 1/2$

Cold (1)

$p = 1$
$n = 0$
$ME = 0$

Expected = (2/5)(0/2) + (2/5)(1/2) + (1/5)(0/1) = 1/5
Gain = 2/5 - 1/5 = 1/5 = **0.20**

For attribute Humidity:

High (3)

$p = 0$
$n = 3$
$ME = 0$

Normal (2)

$p = 2$
$n = 0$
$ME = 0$

Low (0)

$p = 0$
$n = 0$
$ME = 0$

Expected = (3/5)(0) + (2/5)(0) + (0)(0) = 0
Gain = 2/5 - 0 = 2/5 = **0.40**

For attribute Wind:

Strong (3)

$p = 1$
$n = 2$
$ME = 1/3$

Weak (2)

$p = 1$
$n = 1$
$ME = 1/2$

Expected Entropy = (3/5)(1/3) + (2/5)(1/2) = 2/5
Gain = 2/5 - 2/5 = **0**

The best attribute is Humidity for Outlook = S (for the highest gain). For attribute Humidity, two branches are created with values H, N, and L. Each value has its own subset, S_v.

iii. ID3(S_v where Outlook = O, Attributes = {T, H, W}, Label = {+, -}): **Return leaf node with la**

iv. ID3(S_v where Outlook = R, Attributes = {T, H, W}, Label = {+, -}):

A. Node - Check if all base cases are met: No.

B. Otherwise - Create a node using the "best attribute":

- Majority Error of S_v, ME(S_v):

$p = 2$
$n = 3$
$ME(S) = 2/5$

For attribute Temperature:

Hot (0)

$p = 0$
$n = 0$
$ME = 0$

Medium (3)

$p = 2$
$n = 1$
$ME = 1/3$

Cold (2)

$p = 1$
$n = 1$
$ME = 1/2$

Expected = (0)(0) + (3/5)(1/3) + (2/5)(1/2) = 2/5
Gain = 2/5 - 2/5 = **0**

For attribute Humidity:

High (2)

$p = 1$
$n = 1$
$ME = 1/2$

Normal (3)

$p = 2$
$n = 1$
$ME = 1/3$

Low (0)

$p = 0$
$n = 0$
$ME = 0$

Expected = (2/5)(1/2) + (3/5)(1/3) + (0)(0) = 2/5
Gain = 2/5 - 2/5 = **0**

For attribute Wind:

Strong (2)

$p = 0$
$n = 2$
$ME = 0$

Weak (3)

$p = 3$
$n = 0$
$ME = 0$

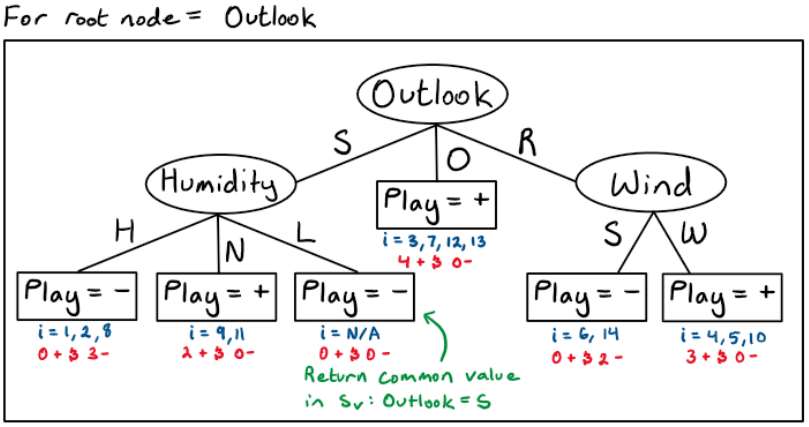
Expected Entropy = (2/5)(0) + (3/5)(0) = 0
Gain = 2/5 - 0 = 2/5 = **0.40**

The best attribute is Humidity for Outlook = S (for the highest gain). For attribute Humidity, two branches are created with values H, N, and L. Each value has its own subset, S_v.

v. Continuing the recursive algorithm, the rest of the leaf nodes are as follows:

If Outlook=S & Humidity=H then Play=-.
If Outlook=S & Humidity=N then Play=+.
If Outlook=S & Humidity=L then Play=-. (Note, the most common value in S_v: Outlook=S was returned.)
If Outlook=R & Wind=S then Play=-.
If Outlook=R & Wind=W then Play=+.

vi. **We end with the following decision tree:**



(b) [7 points] Problem 1.2b (Gini Index Tree)

i. ID3(S, Attributes = {O, T, H, W},Label ={+, −}):

A. First Root Node - Check if all base cases are met: No.

B. Otherwise - Create a root node using the "best attribute":

- Gini Index of S, GI(S):

$$p = 9$$
$$n = 5$$
$$GI(S) = 1 - (5/14)^2 - (9/14)^2 = 45/98$$

For attribute Outlook:

Sunny (5)

$p = 2$
$n = 3$
$GI = 12/25$

Overcast (4)

$p = 4$
$n = 0$
$GI = 0$

Rainy (5)

$p = 3$
$n = 2$
$GI = 12/25$

Expected = (5/14)(12/25) + (4/14)(0) + (5/14)(12/25) = 12/35
Gain = 45/98 - 12/35 = 57/490 = **0.116326531**

For attribute Temperature:

Hot (4)

$p = 2$
$n = 2$
$GI = 1/2$

Medium (6)

$p = 4$
$n = 2$
$GI = 4/9$

Cold (4)

$p = 3$
$n = 1$
$GI = 3/8$

Expected = (4/14)(1/2) + (6/14)(4/9) + (4/14)(3/8) = 37/84
Gain = 45/98 - 37/84 = 11/588 =**0.018707483**

For attribute Humidity:

High (7)

$p = 3$
$n = 4$
$GI = 24/49$

Normal (7)

$p = 6$
$n = 1$
$GI = 12/49$

Low (0)

$p = 0$
$n = 0$
$GI = 0$

Expected = (7/14)(24/49) + (7/14)(12/49) + (0)(0) = 18/49
Gain = 45/98 - 18/49 = 9/98 = **0.091836735**

For attribute Wind:

Strong (6)

$p = 3$
$n = 3$
$GI = 1/2$

Weak (8)

$p = 6$
$n = 2$
$GI = 3/8$

Expected = (6/14)(1/2) + (8/14)(3/8) = 3/7
Gain = 45/98 - 3/7 = 3/98 = **0.030612245**

The best attribute is Outlook (for highest gain). For the Outlook attribute, three branches are created with values S, O, and R. Each value has its own subset, S_v.

ii. ID3(S_v where Outlook = S, Attributes = {T, H, W}, Label = {+, -}):

A. Node - Check if all base cases are met: No.
B. Otherwise - Create a node using the "best attribute":
- Gini Index of S_v, GI(S_v):

$$p = 2$$
$$n = 3$$
$$GI(S) = 1 - (2/5)^2 - (3/5)^2 = 12/25$$

For attribute Temperature:

Hot (2)

$p = 0$
$n = 2$
$GI = 0$

Medium (2)

$p = 1$
$n = 1$
$GI = 1/2$

Cold (1)

$p = 1$
$n = 0$
$GI = 0$

Expected = (2/5)(0) + (2/5)(1/2) + (1/5)(0) = 1/5
Gain = 12/25 - 1/5 = 7/25 = **0.28**

For attribute Humidity:

High (3)

$p = 0$
$n = 3$
$GI = 0$

Normal (2)

$p = 2$
$n = 0$
$GI = 0$

Low (0)

$p = 0$
$n = 0$
$GI = 0$

Expected = (3/5)(0) + (2/5)(0) + (0)(0) = 0
Gain = 12/25 - 0 = 12/25 = **0.48**

For attribute Wind:

Strong (3)

$p = 1$
$n = 2$
$GI = 4/9$

Weak (2)

$p = 1$
$n = 1$
$GI = 1/2$

Expected Entropy = (3/5)(4/9) + (2/5)(1/2) = 7/15
Gain = 12/25 - 7/15 = 1/75 = **0.013333333**

The best attribute is Humidity for Outlook = S (for the highest gain). For attribute Humidity, two branches are created with values H, N, and L. Each value has its own subset, S_v.

iii. ID3(S_v where Outlook = O, Attributes = {T, H, W}, Label = {+, -}): **Return leaf node with la**

iv. ID3(S_v where Outlook = R, Attributes = {T, H, W}, Label = {+, -}):
A. Node - Check if all base cases are met: No.
B. Otherwise - Create a node using the "best attribute":
- Gini Index of S_v, GI(S_v):

$$p = 2$$
$$n = 3$$
$$GI(S) = 1 - (2/5)^2 - (3/5)^2 = 12/25$$

For attribute Temperature:

Hot (0)

$p = 0$
$n = 0$
$GI = 0$

Medium (3)

$p = 2$
$n = 1$
$GI = 4/9$

Cold (2)

$p = 1$
$n = 1$
$GI = 1/2$

Expected = (0)(0) + (3/5)(4/9) + (2/5)(1/2) = 7/15
Gain = 12/25 - 7/15 = 1/75 = **0.013333333**

For attribute Humidity:

High (2)

$p = 1$
$n = 1$
$GI = 1/2$

Normal (3)

$p = 2$
$n = 1$
$GI = 4/9$

Low (0)

$p = 0$
$n = 0$
$GI = 0$

Expected = (2/5)(1/2) + (3/5)(4/9) + (0)(0) = 7/15
Gain = 12/25 - 7/15 = 1/75 = **0.013333333**

For attribute Wind:

Strong (2)

$p = 0$
$n = 2$
$ME = 0$

Weak (3)

$p = 3$
$n = 0$
$ME = 0$

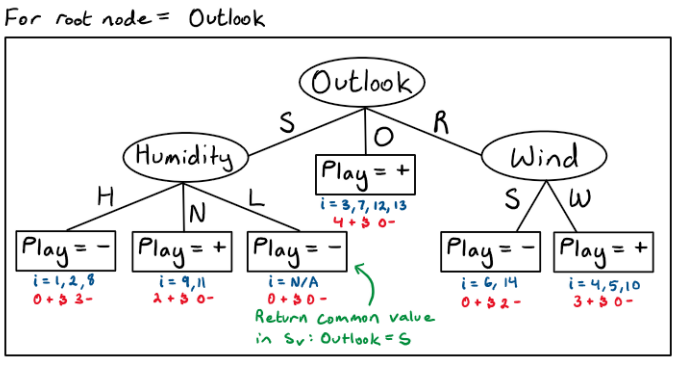
Expected Entropy = (2/5)(0) + (3/5)(0) = 0
Gain = 12/25 - 0 = 12/25 = **0.48**

The best attribute is Wind for Outlook = R (for the highest gain). For attribute Wind, two branches are created with values S and W. Each value has its own subset, S_v.

v. Continuing the recursive algorithm, the rest of the leaf nodes are as follows:

If Outlook=S & Humidity=H then Play=-.
If Outlook=S & Humidity=N then Play=+.
If Outlook=S & Humidity=L then Play=-. Note, the most common value in S_v: Outlook=S was returned.
If Outlook=R & Wind=S then Play=-.
If Outlook=R & Wind=W then Play=+.

vi. **We end with the following decision tree:**



(c) [3 points] Problem 1.2c (In-class/HW Comparisons)

There is no difference between the tree derived in class and the trees from problems 1.2.a and 1.2.b. Do note that in problem 1.2.a, it is possible to start the root node with Humidity attribute instead of Outlook since they both tie for the highest gain. The method in which we compute the gain will impact the nodes of the tree, but because they all share the same goal of "purifying/simplfying" the tree, they will end with the same or similar results.

3. [16 points] Problem 1.3 (Tennis Training Dataset - Missing Attribute)

(a) [3 points] Problem 1.3a (Common Label - Best Attribute)

i. ID3(S,Attributes = {O,T,W,H},Label ={+, −}):

A. First Root Node - Check if all base cases are met: Attribute empty.

The Outlook attribute is missing for the new training instance. The most common value in the Outlook attribute in the set S is either S or R. I will set $Outlook_{15} = S$ in this case.

B. Otherwise - Create a root node using the "best attribute":

- Entropy of S, H(S):

$p = 10$
$n = 5$
$H(S) = -(10/15)\log_2(10/15) - (5/15)\log_2(5/15) = \mathbf{0.918295834}$

For attribute Outlook:

Sunny (6)	Overcast (4)	Rainy (5)
$p = 3$	$p = 4$	$p = 3$
$n = 3$	$n = 0$	$n = 2$
$H = 1$	$H = 0$	$H = 0.970950594$

Expected = (6/15)(1) + (4/15)(0) + (5/15)(0.970950594) = 0.723650198
Gain = 0.918295834 - 0.723650198 = **0.194645636**

For attribute Temperature:

Hot (4)	Medium (7)	Cold (4)
$p = 2$	$p = 5$	$p = 3$
$n = 2$	$n = 2$	$n = 1$
$H = 1$	$H = 0.863120569$	$H = 0.811278124$

Expected = (4/15)(1) + (7/15)(0.863120569) + (4/15)(0.811278124) = 0.885797099
Gain = 0.918295834 - 0.885797099 = **0.032498735**

For attribute Humidity:

High (7)	Normal (8)	Low (0)
$p = 3$	$p = 7$	$p = 0$
$n = 4$	$n = 1$	$n = 0$
$H = 0.811278124$	$H = 0.543564443$	$H = 0$

Expected = (7/15)(0.811278124) + (8/15)(0.543564443) + (0)(0) = 0.668497494
Gain = 0.918295834 - 0.668497494 = **0.24979834**

For attribute Wind:

Strong (6)

$p = 3$
$n = 3$
$GI = 1$

Weak (9)

$p = 7$
$n = 2$
$GI = 0.764204507$

Expected = (6/15)(1) + (9/15)(0.764204507) = 0.858522704
Gain = 0.918295834 - 0.858522704 = **0.05977313**

The best attribute is Humidity (for highest gain). This is for when we set the missing value $Outlook_{15} = S$.

(b) [3 points] Problem 1.3b (Common Value w/ Same Label - Best Attribute)

i. ID3(S,Attributes = {O, T, H, W},Label= {+, -}):

A. First Root Node - Check if all base cases are met: Attribute empty.

The Outlook attribute is missing for the new training instance. The new training instance (index 15) has label=+. In the set S with label=+, the Outlook values are as follows:
- 2 instances of Outlook = S
- 4 instances of Outlook=O
- 3 instances of Outlook=R
I will set $Outlook_{15} = O$ because it is the most common value within the subset label=+.

B. Otherwise - Create a root node using the "best attribute":

- Entropy of S, H(S):

$p = 10$
$n = 5$
$H(S) = -(10/15)\log_2(10/15) - (5/15)\log_2(5/15) = \mathbf{0.918295834}$

For attribute Outlook:

Sunny (5)

$p = 2$
$n = 3$
$H = 0.970950594$

Overcast (5)

$p = 5$
$n = 0$
$H = 0$

Rainy (5)

$p = 3$
$n = 2$
$H = 0.970950594$

Expected = (5/15)(0.970950594) + (5/15)(0) + (5/15)(0.970950594) = 0.647300396
Gain = 0.918295834 - 0.647300396 = **0.270995438**

For attribute Temperature:

Hot (4)

$p = 2$
$n = 2$
$H = 1$

Medium (7)

$p = 5$
$n = 2$
$H = 0.863120569$

Cold (4)

$p = 3$
$n = 1$
$H = 0.811278124$

Expected = (4/15)(1) + (7/15)(0.863120569) + (4/15)(0.811278124) = 0.885797099
Gain = 0.918295834 - 0.885797099 = **0.032498735**

For attribute Humidity:

High (7)	Normal (8)	Low (0)
$p = 3$	$p = 7$	$p = 0$
$n = 4$	$n = 1$	$n = 0$
$H = 0.811278124$	$H = 0.543564443$	$H = 0$

Expected = (7/15)(0.811278124) + (8/15)(0.543564443) + (0)(0) = 0.668497494
Gain = 0.918295834 - 0.668497494 = **0.24979834**

For attribute Wind:

Strong (6)	Weak (9)
$p = 3$	$p = 7$
$n = 3$	$n = 2$
$GI = 1$	$GI = 0.764204507$

Expected = (6/15)(1) + (9/15)(0.764204507) = 0.858522704
Gain = 0.918295834 - 0.858522704 = **0.05977313**

The best attribute is Outlook (for highest gain). This is for when we set the missing value $Outlook_{15} = O$.

(c) [3 points] Problem 1.3c (Fractional Counts - Best Attribute)

i. ID3(S,Attributes= {O,T,H,W},Label={+, -}):

A. First Root Node - Check if all base cases are met:

The Outlook attribute is missing for the new training instance. The new training instance can be split as follows:
14/14 (ith. ? M N W +) =
5/14 (15 S M N W +)
4/14 (16 O M N W +)
5/14 (17 R M N W +)

B. Otherwise - Create a root node using the "best attribute":

- Entropy of S, H(S): Same work as done in problem 1.3b.

$$H(S) = -(10/15)\log_2(10/15) - (5/15)\log_2(5/15) = \mathbf{0.918295834}$$

For attribute Outlook:

Sunny (5 + 5/14)	Overcast (4 + 4/14)	Rainy (5 + 5/14)
$p = 2 + 5/14$	$p = 4 + 4/14$	$p = 3 + 5/14$
$n = 3$	$n = 0$	$n = 2$
$H = 0.989587521$	$H = 0$	$H = 0.953197173$

Expected = $(\frac{5+\frac{5}{14}}{15})(0.989587521) + (0) + (\frac{5+\frac{5}{14}}{15})(0.953197173) = 0.693851676$ Gain = 0.918295834 - 0.693851676 = **0.224444158**

For attribute Temperature, Humidity, and Wind:
Same work as done in problem 1.3b. (Only Outlook attribute changes.)

Temperature: Gain = 0.918295834 - 0.885797099 = **0.032498735**
Humidity: Gain = 0.918295834 - 0.668497494 = **0.24979834**
Wind: Gain = 0.918295834 - 0.858522704 = **0.05977313**

The best attribute is Humidity (for highest gain). This is for when we split the missing value $Outlook_{15}$ into fractional counts between S, O, and R.

(d) [7 points] Problem 1.3d (Fractional Counts - Tree)

i. ID3(S_v where Humidity = H, Attributes = {O, T, W}, Label = {+, -}):

A. Node - Check if all base cases are met: No.

B. Otherwise - Create a node using the "best attribute":

- Entropy of S_v, H(S_v):

$$p = 3$$
$$n = 4$$
$$H(S) = -(3/7)\log_2(3/7) - (4/7)\log_2(4/7) = \mathbf{0.985228136}$$

For attribute Outlook:

Sunny (3): $p = 0$, $n = 3$, $H = 0$

Overcast (2): $p = 2$, $n = 0$, $H = 0$

Rainy (2): $p = 1$, $n = 1$, $H = 1$

Expected = $0 + 0 + (2/7)(1) = 2/7$
Gain = 0.985228136 - 2/7 = **0.69951385**

For attribute Temperature:

Hot (3): $p = 1$, $n = 2$, $H = 0.918295834$

Medium (4): $p = 2$, $n = 2$, $H = 1$

Cold (0): $p = 0$, $n = 0$, $HGI = 0$

Expected = $(3/7)(0.918295834) + (4/7)(1) + (0) = 0.964983929$
Gain = 0.985228136 - 0.964983929 = **0.020244207**

For attribute Wind:

Strong (3): $p = 1$, $n = 2$, $H = 0.918295834$

Weak (4): $p = 2$, $n = 2$, $H = 1$

Expected Entropy = $(3/7)(0.918295834) + (4/7)(1) = 0.964983929$
Gain = 0.985228136 - 0.964983929 = **0.020244207**

The best attribute is Outlook for Humidity = H (for the highest gain). For attribute Outlook, three branches are created with values S, O, and R. Each value has its own subset, S_v.

ii. ID3(S_v where Humidity = N, Attributes = {O, T, W}, Label = {+, -}):

A. Node - Check if all base cases are met: No.
B. Otherwise - Create a node using the "best attribute":
- Entropy of S_v, H(S_v):

$p = 7$
$n = 1$
$H(S) = -(7/8)\log_2(7/8) - (1/8)\log_2(1/8) = \mathbf{0.543564443}$

For attribute Outlook:

Sunny (2 + 5/14)

$p = 2 + 5/14$
$n = 0$
$H = 0$

Overcast (2 + 4/14)

$p = 2 + 4/14$
$n = 0$
$H = 0$

Rainy (3 + 5/14)

$p = 2 + 5/14$
$n = 1$
$H = 0.878674493$

Expected $= 0 + 0 + (\frac{3+\frac{5}{14}}{8})(0.878674493) = 0.368729475$
Gain = 0.543564443 - 0.878674493 = **0.174834968**

For attribute Temperature:

Hot (1)

$p = 1$
$n = 0$
$H = 0$

Medium (3)

$p = 3$
$n = 0$
$H = 0$

Cold (4)

$p = 3$
$n = 1$
$H = 0.811278124$

Expected = (1/7)(0) + (3/7)(0) + (4/7)(0.811278124) = 0.463587499
Gain = 0.543564443 - 0.463587499 = **0.079976944**

For attribute Wind:

Strong (3)

$p = 2$
$n = 1$
$H = 0.918295834$

Weak (5)

$p = 5$
$n = 0$
$H = 0$

Expected Entropy = (3/8)(0.918295834) + (5/8)(0) = 0.344360938
Gain = 0.543564443 - 0.344360938 = **0.199203505**

The best attribute is Wind for Humidity = N (for the highest gain). For attribute Wind, two branches are created with values S and W. Each value has its own subset, S_v.

iii. ID3(S_v where Humidity = L, Attributes = {O, T, W}, Label = {+, -}):

Return leaf node with la
Note, the most common

iv. ID3(S_v where Humidity=H & Outlook=R, Attributes = {T, W}, Label = {+, -}):
A. Node - Check if all base cases are met: No.
B. Otherwise - Create a node using the "best attribute":
- Entropy of S_v, H(S_v):

$p = 1$
$n = 1$
$H(S) = -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) = \mathbf{1}$

For attribute Temperature:

Hot (0)

$p = 0$
$n = 0$
$H = 0$

Medium (2)

$p = 1$
$n = 1$
$H = 1$

Cold (0)

$p = 0$
$n = 0$
$H = 0$

Expected = (0) + (2/2)(1) + (0) = 1
Gain = 1 - 1 = **0**

For attribute Wind:

Strong (1)

$p = 0$
$n = 1$
$H = 0$

Weak (1)

$p = 1$
$n = 0$
$H = 0$

Expected Entropy = (1/2)(0) + (1/2)(0) = 0
Gain = 1 - 0 = **1**

The best attribute is Wind for Humidity=H & Outlook=R (for the highest gain). For attribute Wind, two branches are created with values S and W. Each value has its own subset, S_v.

v. ID3(S_v where Humidity=N & Wind=S, Attributes = {O, T}, Label = {+, -}):

A. Node - Check if all base cases are met: No.

B. Otherwise - Create a node using the "best attribute":

- Entropy of S_v, H(S_v):

$p = 2$
$n = 1$
$H(S) = -(2/3)\log_2(2/3) - (1/3)\log_2(1/3) =$ **0.918295834**

For attribute Outlook:

Sunny (1)

$p = 1$
$n = 0$
$H = 0$

Overcast (1)

$p = 1$
$n = 0$
$H = 0$

Rainy (1)

$p = 0$
$n = 1$
$H = 0$

Expected = (0) + (0) + (0) = 0
Gain = 1 - 0 = **1**

For attribute Temperature:

Hot (0)

$p = 0$
$n = 0$
$H = 0$

Medium (1)

$p = 1$
$n = 0$
$H = 0$

Cold (2)

$p = 1$
$n = 1$
$H = 1$

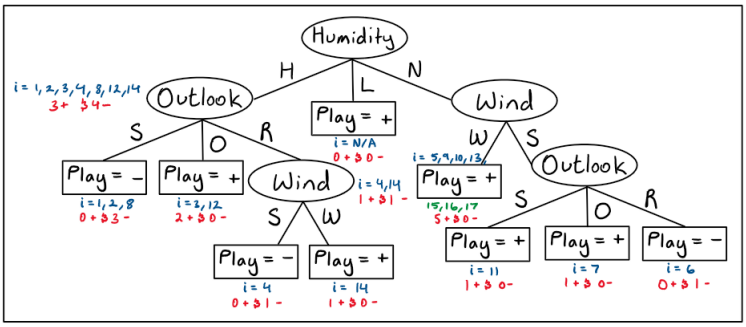
Expected = (0) + (1/3)(0) + (2/3)(1) = 2/3
Gain = 1 - 2/3 = 1/3 = **0.33333333**

The best attribute is Outlook for Humidity=N & Wind=S (for the highest gain). For attribute Outlook, three branches are created with values S, O, and R. Each value has its own subset, S_v.

vi. Continuing the recursive algorithm, the rest of the leaf nodes are as follows:

If Humidity=H & Outlook=S then Play=-.
If Humidity=H & Outlook=O then Play=+.
If Outlook=N & Wind=W then Play=+.
If Humidity=H & Outlook=R & Wind = S then Play=-.
If Humidity=H & Outlook=R & Wind = W then Play=+.
If Humidity=N & Wind=S & Outlook =S then Play=+.
If Humidity=N & Wind=S & Outlook =O then Play=+.
If Humidity=N & Wind=S & Outlook=R then Play=-.

vii. **We end with the following decision tree:**



4. [**Bonus question 1**] [5 points]. Problem 1.4 (Prove non-negative information gain)

Prove $-\log(x)$ is convex:

- Let $g(x) = \log(x)$
- Then $g'(x) = \frac{1}{x}$ and $g"(x) = -\frac{1}{x^2}$
- By theorem, $g"(x) = -\frac{1}{x^2} \rightarrow$ concave $\implies f(x) = -log(x) = convex$✓✓

Using $-\log(x)$ is convex, prove that information gain is always non-negative:

$$Gain(S,A) = Entropy(S) - \sum_{vevalues(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$= -p_+ \log_2(p_+) - p_- \log_2(p_-) - \sum_{i=1}^{k} p_i \log_2(p_i)$$

$$= -\sum p_j \log_2(p_j) - \sum p_i \log_2(p_i)$$

$$\geq -\log_2(\sum p_i \sum p_j) \qquad (Jensen's Inequality)$$

$$Gain(S,A) \geq -\log_2(1) = 0 \qquad (where \sum p_i j = 1)$$

Gain(S,A) ≥ 0 ✓✓

5. [**Bonus question 2**] [5 points]. Problem 1.5 (Invent a gain)

You can create an improved form of the information gain (IG) by normalizing it as follows:

$$Gain(S,A) = \frac{IG(A,S)}{\sum_{values(A)} \frac{|S_v|}{S} * \log_2 \frac{|S_v|}{S}}$$

1. [5 Points] Problem 2.1 (Github Repository)

   Link to Repository: https://github.com/DaisyQuach/Machine-Learning-Fall2024

2. [30 points] Problem 2.2 (Car Evaluation Task)

   (a) [15 points] Problem 2.2a (Implement the ID3 Algorithm)

   Also in submission zip: https://github.com/DaisyQuach/Machine-Learning-Fall2024

   (b) [10 points] Problem 2.2b (Apply Algorithm & Tables)

TreeDepth	*Method*	*Training*	*Test*
1	Information Gain	0.671	0.6607142857142857
2	Information Gain	0.444	0.44505494505494503
3	Information Gain	0.444	0.44505494505494503
4	Information Gain	0.408	0.3942307692307692
5	Information Gain	0.299	0.20054945054945056
6	Information Gain	0.0	0.20054945054945056
1	Majority Error	0.671	0.6607142857142857
2	Majority Error	0.671	0.6607142857142857
3	Majority Error	0.537	0.5480769230769231
4	Majority Error	0.523	0.5013736263736264
5	Majority Error	0.16	0.19917582417582416
6	Majority Error	0.0	0.19917582417582416
1	Gini Index	0.671	0.6607142857142857
2	Gini Index	0.444	0.44505494505494503
3	Gini Index	0.444	0.44505494505494503
4	Gini Index	0.243	0.22115384615384615
5	Gini Index	0.16	0.15796703296703296
6	Gini Index	0.0	0.15796703296703296

   (c) [5 points] Problem 2.2c (Car Evaluation Conclusion)

   As discussed in class, we do not want to use our training data like as the test data because we want to test if the learning algorithm is learning and not memorizing. From the table above, we can see how there is no error when we run the training dataset throught the 6-tier/complete decision tree. Although using test data will result in error, it actually indicates how well the algorithm performs. Any error from the training data below a max tree depth of 6 is due to the tree's cutoff.

3. [25 points] Problem 2.3 (Bank Marketing - Numerical Attributes)

   (a) [10 points] Problem 2.3a (No missing Attributes & Tables)

   Code: https://github.com/DaisyQuach/Machine-Learning-Fall2024/tree/main/DecisionTree
   (Also in Canvas submission zip file)

TreeDepth	*Method*	*Training*	*Test*
1	Information Gain	1.0	1.0
2	Information Gain	1.0	0.9992
3	Information Gain	0.9382	0.9428
4	Information Gain	0.8404	0.8468
5	Information Gain	0.8404	0.8468
6	Information Gain	0.8404	0.8422
7	Information Gain	0.7768	0.785
8	Information Gain	0.624	0.6228
9	Information Gain	0.55	0.514
10	Information Gain	0.2832	0.3066
11	Information Gain	0.1688	0.2294
12	Information Gain	0.163	0.2246
13	Information Gain	0.1054	0.1814
14	Information Gain	0.0818	0.1672
15	Information Gain	0.0538	0.1518
16	Information Gain	N/A	N/A
1	Majority Error	1.0	1.0
2	Majority Error	0.9876	0.9876
3	Majority Error	0.9668	0.9684
4	Majority Error	0.935	0.9038
5	Majority Error	0.6796	0.6784
6	Majority Error	0.5214	0.5568
7	Majority Error	0.5078	0.5458
8	Majority Error	0.3444	0.3908
9	Majority Error	0.2696	0.3366
10	Majority Error	0.2274	0.3004
11	Majority Error	0.169	0.264
12	Majority Error	0.1086	0.214
13	Majority Error	0.0566	0.1764
14	Majority Error	0.035	0.1672
15	Majority Error	0.035	0.1672
16	Majority Error	N/A	N/A
1	Gini Index	1.0	1.0
2	Gini Index	1.0	0.9992
3	Gini Index	0.9382	0.9442
4	Gini Index	0.9382	0.9442
5	Gini Index	0.9382	0.9404
6	Gini Index	0.8482	0.858
7	Gini Index	0.7362	0.7162
8	Gini Index	0.427	0.4402
9	Gini Index	0.343	0.3604
10	Gini Index	0.2316	0.267
11	Gini Index	0.1686	0.2338
12	Gini Index	0.1646	0.2284
13	Gini Index	0.1064	0.189
14	Gini Index	0.1784	0.0876
15	Gini Index	0.0566	0.157
16	Gini Index	N/A	N/A

(b) [10 points] Problem 2.3b (Common Value Missing Attribute & Tables)
Code: `https://github.com/DaisyQuach/Machine-Learning-Fall2024/tree/main/DecisionTree`

(Also in Canvas submission zip file)

TreeDepth	*Method*	*Training*	*Test*
1	Information Gain	1.0	1.0
2	Information Gain	1.0	0.9992
3	Information Gain	0.9382	0.9428
4	Information Gain	0.8404	0.8468
5	Information Gain	0.8404	0.8468
6	Information Gain	0.8404	0.8428
7	Information Gain	0.7534	0.7576
8	Information Gain	0.707	0.7144
9	Information Gain	0.5944	0.554
10	Information Gain	0.3298	0.337
11	Information Gain	0.2078	0.2464
12	Information Gain	0.1472	0.2084
13	Information Gain	0.1442	0.2034
14	Information Gain	0.0908	0.1674
15	Information Gain	0.0742	0.1574
16	Information Gain	N/A	N/A
1	Majority Error	1.0	1.0
2	Majority Error	0.9876	0.9876
3	Majority Error	0.9668	0.9684
4	Majority Error	0.935	0.9052
5	Majority Error	0.6834	0.689
6	Majority Error	0.5446	0.5794
7	Majority Error	0.531	0.569
8	Majority Error	0.3604	0.4056
9	Majority Error	0.286	0.3486
10	Majority Error	0.2414	0.3138
11	Majority Error	0.2264	0.3064
12	Majority Error	0.145	0.235
13	Majority Error	0.0766	0.1846
14	Majority Error	0.0502	0.1724
15	Majority Error	0.0502	0.1724
16	Majority Error	N/A	N/A
1	Gini Index	1.0	1.0
2	Gini Index	1.0	0.9992
3	Gini Index	0.9382	0.9442
4	Gini Index	0.9382	0.9442
5	Gini Index	0.9382	0.9432
6	Gini Index	0.8486	0.855
7	Gini Index	0.6972	0.705
8	Gini Index	0.5722	0.5438
9	Gini Index	0.2994	0.3198
10	Gini Index	0.2306	0.2788
11	Gini Index	0.2258	0.2732
12	Gini Index	0.1532	0.2154
13	Gini Index	0.1278	0.2004
14	Gini Index	0.1176	0.1936
15	Gini Index	0.0766	0.1642
16	Gini Index	N/A	N/A

(c) [5 points] Problem 2.3c (Conclusion - Comparing Errors & Unknown Attribute Values)

Same as before, the using the training reaches a smaller error sooner than the testing data, and it has "perfect prediction" with the right enough depth to represent the data. However, the test error is more true to how well the algorithm performed. As a whole, increasing the tree depth improves the error as there are more rules it covers naturally. When dealing with unknown attributes, using the method from part a and part b are both similar in terms of performance. The part a method did perform slightly better, and it is possible that the unknown attributes have correlation to each other. However, the part b method would have a larger advantage if the unknown attribute had no correlation.