CS 5350/6350: Machine Learining Fall 2022

Homework 1

Handed out: 6 Sep, 2022
Due date: 11:59pm, 23 Sep, 2022

1 Decision Tree [40 points + 10 bonus]

x_1	x_2	x_3	x_4	y
0	0	1	0	0
0	1	0	0	0
0	0	1	1	1
1	0	0	1	1
0	1	1	0.	0
1	1	0	0	0
0	1	0	1	0

Table 1: Training data for a Boolean classifier

1. [7 points] Decision tree construction.

   (a) [5 points] Use the ID3 algorithm with information gain to learn a decision tree from the training dataset in Table 1. Please list every step in your tree construction, including the data subsets, the attributes, and how you calculate the information gain of each attribute and how you split the dataset according to the selected attribute. Please also give a full structure of the tree. You can manually draw the tree structure, convert the picture into a PDF/EPS/PNG/JPG format and include it in your homework submission; or instead, you can represent the tree with a conjunction of prediction rules as we discussed in the lecture.
   Answer:

   Information Gain:

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

   Current Entropy:

$$p = \frac{2}{7}; n = \frac{2}{7}$$

$$H(Y) = -\frac{2}{7}\log_2\frac{2}{7} - \frac{5}{7}\log_2\frac{5}{7} = 0.8631$$

Attribute: X$_1$

X$_1$ = 0 : 5 out of 7

$$p = \frac{1}{5}; n = \frac{4}{5}$$

$$H(X_1 = 0) = -\frac{1}{5}\log_2\frac{1}{5} - \frac{4}{5}\log_2\frac{4}{5} = 0.7218$$

X$_1$ = 1 : 2 out of 7

$$p = \frac{1}{2}; n = \frac{1}{2}$$

$$H(X_1 = 1) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

Expected Entropy = $\frac{5}{7} * H(X_1 = 0) + \frac{2}{7} * H(X_1 = 1) = 0.8013$
Information Gain(y,X$_1$) = 0.8631 − 0.8031 = 0.06

Attribute: X$_2$

X$_2$ = 0 : 3 out of 7

$$p = \frac{2}{3}; n = \frac{1}{3}$$

$$H(X_2 = 0) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.7218$$

X$_2$ = 1 : 4 out of 7

$$p = 0; n = \frac{4}{4}$$

$$H(X_2 = 1) = 0$$

Expected Entropy = $\frac{3}{7} * H(X_2 = 0) + \frac{4}{7} * H(X_2 = 1) = 0.3934$
Information Gain(y,X$_2$) = 0.8631 − 0.3934 = 0.4697
Attribute: X$_3$

X$_3$ = 0 : 4 out of 7

$$p = \frac{1}{4}; n = \frac{3}{4}$$

$$H(X_3 = 0) = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} = 0.811$$

X$_3$ = 1 : 3 out of 7

$$p = \frac{1}{3}; n = \frac{2}{3}$$

$$H(X_3 = 1) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.918$$

Expected Entropy = $\frac{4}{7} * H(X_3 = 0) + \frac{3}{7} * H(X_3 = 1) = 0.857$
Information Gain(y,X$_3$) = 0.8631 − 0.857 = 0.061

Attribute: X$_4$

$\mathrm{X}_4 = 0:$ 4 out of 7

$$p = 0; n = \frac{4}{4}$$

$$H(X_4 = 0) = 0$$

$\mathrm{X}_4 = 1:$ 3 out of 7

$$n = \frac{1}{3}; p = \frac{2}{3}$$

$$H(X_4 = 1) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918$$

Expected Entropy $= \frac{4}{7} * H(X_4 = 0) + \frac{3}{7} * H(X_4 = 1) = 0.3934$
Information Gain(y,X_4) $= 0.8631 - 0.3934 = 0.4697$

Based on **highest Information gain** we can split on X_4 **or** X_2
Let split on X_2
$\mathrm{X}_2 = 1:$ leaf node $Y = 0$ and $\mathrm{X}_2 = 0:$

$$p = \frac{2}{3}; n = \frac{1}{3}$$

$$H(X_2 = 0) = 0.7218$$

Now , find **highest Information gain** we can split on $X_2 = 0$
Information Gain$(X_2 = 0, X_1) = 0.7218 - \frac{2}{3} * [-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}] - 0 * \frac{1}{3} = 0.0618$

Information Gain$(X_2 = 0, X_3) = 0.7218 - \frac{2}{3} * [-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}] - 0 * \frac{1}{3} = 0.0618$

Information Gain$(X_2 = 0, X_4) = 0.7218 - \frac{2}{3} * 0 - 0 * \frac{1}{3} = 0.7218$

Let split on X_4
$\mathrm{X}_4 = 0:$ leaf node $Y = 0$ and $\mathrm{X}_4 = 1:$ leaf node $Y = 1$
so the tree will be:

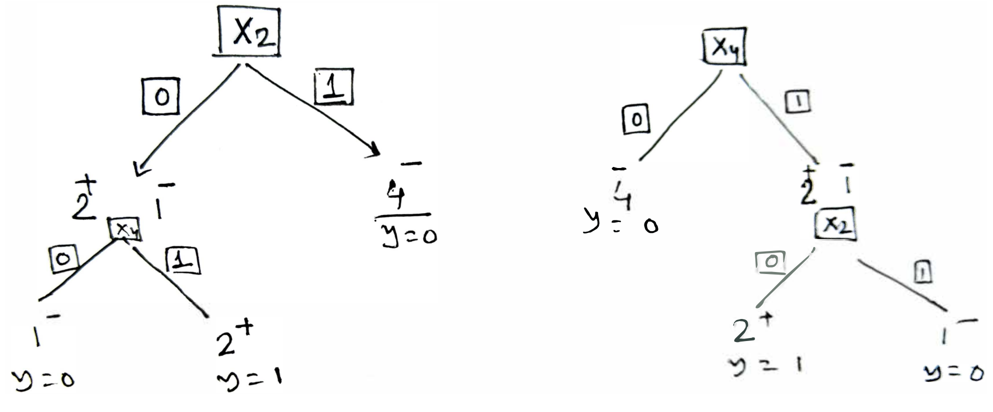

Figure 1: Boolean Tree

(b) [2 points] Write the boolean function which your decision tree represents. Please use a table to describe the function — the columns are the input variables and label, i.e., x_1, x_2, x_3, x_4 and y; the rows are different input and function values.

Answer:

$$Y = \bar{X}_1\bar{X}_2X_3X_4 + X_1\bar{X}_2\bar{X}_3X_4 = \bar{X}_2X_4(X_1 \oplus X_3)$$

2. [17 points] Let us use a training dataset to learn a decision tree about whether to play tennis (**Page 43, Lecture: Decision Tree Learning**, accessible by clicking the link http://www.cs.utah.edu/~zhe/teach/pdf/decision-trees-learning.pdf). In the class, we have shown how to use information gain to construct the tree in ID3 framework.

(a) [7 points] Now, please use majority error (ME) to calculate the gain, and select the best feature to split the data in ID3 framework. As in problem 1, please list every step in your tree construction, the attributes, how you calculate the gain of each attribute and how you split the dataset according to the selected attribute. Please also give a full structure of the tree.
Answer:

$$Gain(S, A) = ME(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} ME(S_v)$$

Step 1 :
ME (S) = $\frac{5}{14}$

Outlook:
Sunny : p = 2, n = 3 , ME (Sunny) = $\frac{2}{5}$
Overcast : p = 4, n = 0 , ME (Overcast) = 0
Rain : p = 3, n = 2 , ME (Rain) = $\frac{2}{5}$

Temperature:
Hot : p = 2, n = 2 , ME (Hot) = $\frac{2}{4}$
Mild : p = 4, n = 2 , ME (Mild) = $\frac{2}{6}$
Cool : p = 3, n = 1 , ME (Cool) = $\frac{1}{4}$

Humidity:
High : p = 3, n = 4 , ME (High) = $\frac{3}{7}$
Normal : p = 6, n = 1 , ME (Normal) = $\frac{1}{7}$
Low : p = n = 0 , ME (Low) = 0

Wind:
Strong : p = 3, n = 3 , ME (Strong) = $\frac{3}{6}$
weak : p = 6, n = 2 , ME (weak) = $\frac{2}{8}$

$$Gain(S, Outlook) = ME(S) - \frac{2}{5} * \frac{5}{14} - 0 - \frac{2}{5} * \frac{5}{14} = 0.071$$

$$Gain(S, Temperature) = ME(S) - \frac{2}{4} * \frac{4}{14} - \frac{1}{4} * \frac{4}{14} - \frac{2}{6} * \frac{6}{14} = 0$$

$$Gain(S, Humidity) = ME(S) - \frac{3}{7} * \frac{7}{14} - \frac{1}{7} * \frac{7}{14} - 0 = 0.071$$

$$Gain(S, Wind) = ME(S) - \frac{3}{6} * \frac{6}{14} - \frac{2}{8} * \frac{8}{14} = 0$$

Based on **highest Gain**, we see a tie between **humidity and Outlook**, lets split on **Outlook**
Overcast has already reached the leaf node, Play/S =Yes
For other attribute, Sunny and Wind, we calculate Gain.
Step 2 :
ME(Sunny) = $\frac{2}{5}$
Attributes are Humidity, Wind and Temperature.

$$Gain(Sunny, Humidity) = ME(Sunny) - \frac{2}{5} * 0 = \frac{2}{5}$$

$$Gain(Sunny, Temperature) = ME(Sunny) - \frac{2}{5} * \frac{1}{2} = \frac{1}{5}$$

$$Gain(Sunny, Wind) = ME(Sunny) - \frac{2}{5} * \frac{1}{2} - \frac{3}{5} * \frac{1}{3} = 0$$

Based on **highest Gain**, we split on **Humidity**
For High, Normal has already reached the leaf node and make low as majority labeled.
Step 3 :

$$Gain(Rain, Humidity) = ME(Rain) - \frac{3}{5} * \frac{1}{3} - \frac{1}{2} * \frac{2}{5} = 0$$

$$Gain(Rain, Wind) = ME(Rain) - \frac{2}{5} * 0 = \frac{2}{5}$$

$$Gain(Rain, Temperature) = ME(Rain) - \frac{3}{5} * \frac{1}{3} - \frac{1}{2} * \frac{2}{5} = 0$$

Based on **highest Gain**, we split on **Wind**
For Strong and Weak, Wind has already reached the leaf node
so the tree will be:

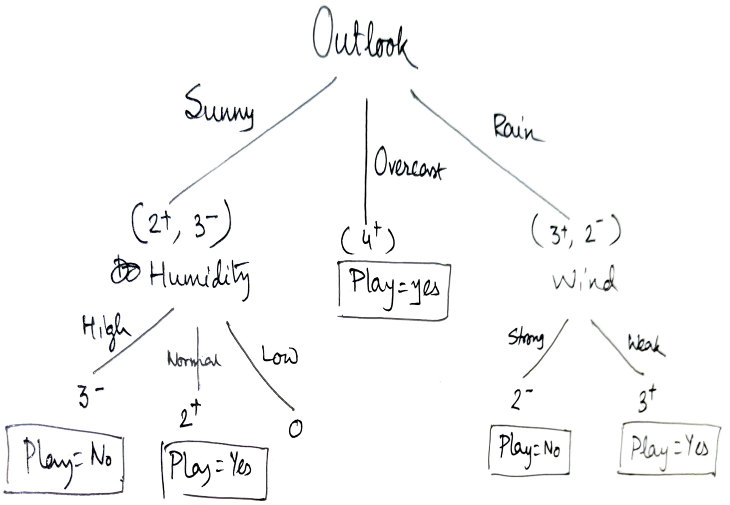

Figure 2: ME dependent:Play Tennis Tree

(b) [7 points] Please use gini index (GI) to calculate the gain, and conduct tree learning with ID3 framework. List every step and the tree structure.

Answer:

$$Gain(S, A) = GI(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} GI(S_v)$$

GI (S) $= 1 - (\frac{5}{14})^2 - (\frac{9}{14})^2 = 0.46$

Step 1:

Outlook:

Sunny : $p_+ = \frac{2}{5}, p_- = \frac{3}{5}$, GI (Sunny) $= 1 - (\frac{2}{5})^2 - (\frac{3}{5})^2 = 0.48$
Overcast : $p_+ = \frac{4}{4}, p_- = 0$, GI (Overcast) $= 0$
Rain : $p_+ = \frac{3}{5}, n = \frac{2}{5}$, GI(Rain) $= 1 - (\frac{2}{5})^2 - (\frac{3}{5})^2 = 0.48$

Temperature:

Hot : $p_+ = \frac{2}{4}, p_- = \frac{2}{4}$, GI (Hot) $= 1 - (\frac{1}{2})^2 - (\frac{1}{2})^2 = 0.5$
Mild : $p_+ = \frac{4}{6}, p_- = \frac{2}{6}$, GI (Mild) $= 1 - (\frac{4}{6})^2 - (\frac{2}{6})^2 = 0.44$
Cool: $p_+ = \frac{3}{4}, p_- = \frac{1}{4}$, GI (Cool) $= 1 - (\frac{3}{4})^2 - (\frac{1}{4})^2 = 0.375$

Humidity:

High : $p_+ = \frac{3}{7}, p_- = \frac{4}{7}$, GI (High) $= 1 - (\frac{3}{7})^2 - (\frac{4}{7})^2 = 0.489$
Normal : $p_+ = \frac{6}{7}, p_- = \frac{1}{7}$, GI (Normal) $= 1 - (\frac{6}{7})^2 - (\frac{1}{7})^2 = 0.245$
Low: $p_+ = 0, p_- = 0$, GI (Low) $= 0$

Wind:

Strong : $p_+ = \frac{3}{6}, p_- = \frac{3}{6}$, GI (Strong) $= 1 - (\frac{1}{2})^2 - (\frac{1}{2})^2 = 0.5$
Weak :$p_+ = \frac{6}{8}, p_- = \frac{2}{8}$, GI (Weak) $= 1 - (\frac{6}{8})^2 - (\frac{2}{8})^2 = 0.375$

$$Gain(S, Outlook) = GI(S) - 0.48 * \frac{5}{14} - 0 - 0.48 * \frac{5}{14} = 0.12$$

$$Gain(S, Temperature) = GI(S) - 0.5 * \frac{4}{14} - 0.44 * \frac{6}{14} - 0.375 * \frac{4}{14} = 0.019$$

$$Gain(S, Humidity) = GI(S) - 0.489 * \frac{7}{14} - 0.245 * \frac{7}{14} - 0 = 0.093$$

$$Gain(S, Wind) = GI(S) - 0.5 * \frac{4}{14} - 0.44 * \frac{6}{14} = 0.031$$

Based on **highest Gain**, we split on **Outlook**
Overcast has already reached the leaf node, Play/S =Yes
For other attribute, Sunny and Wind, we calculate Gain. **Step 2:**

$$Gain(Sunny, Humidity) = GI(Sunny) - 0 = 0.48$$

$$Gain(Sunny, Wind) = GI(Sunny) - [[1 - (\frac{1}{2})^2 - (\frac{1}{2})^2] * \frac{2}{5}] - [[1 - (\frac{1}{3})^2 - (\frac{2}{3})^2] * \frac{3}{5}] = 0.013$$

$$Gain(Sunny, Temperature) = GI(Sunny) - [[1 - (\frac{1}{2})^2 - (\frac{1}{2})^2] * \frac{2}{5}] = 0.28$$

Based on **highest Gain**, we split on **Humidity**
For High, Normal has already reached the leaf node and make low as majority labeled We need to calculate on **Rain**.
Step 3:

$$Gain(Rain, Humidity) = GI(Rain) - [[1 - (\frac{1}{3})^2 - (\frac{2}{3})^2] * \frac{3}{5}] - [[1 - (\frac{1}{2})^2 - (\frac{1}{2})^2] * \frac{2}{5}] = 0.102$$

$$Gain(Rain, Wind) = GI(Rain) - 0 = 0.48$$

$$Gain(Rain, Temperature) = GI(Rain) - [[1 - (\frac{1}{3})^2 - (\frac{2}{3})^2] * \frac{3}{5}] - [[1 - (\frac{1}{2})^2 - (\frac{1}{2})^2] * \frac{2}{5}] = 0.102$$

Based on **highest Gain**, we split on **Wind**
For Strong and Weak, Wind has already reached the leaf node
so the tree will be:

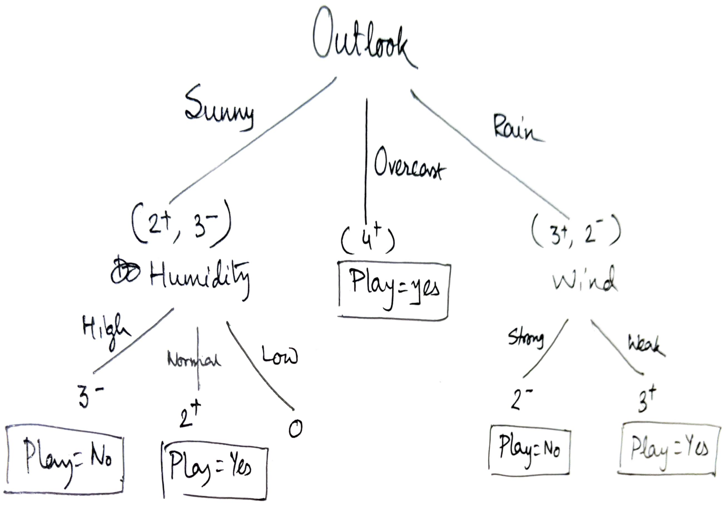

Figure 3: ME dependent:Play Tennis Tree

(c) [3 points] Compare the two trees you just created with the one we built in the class (see Page 62 of the lecture slides). Are there any differences? Why?

Answer: There aren't difference in making decision tree but in attribute selection as ME offers tie between 3 attributes but GI gives accurate measurement. Although the Information gain is conditioned on same attributes and GI and ME measure the purity of the labels, I would prefer GI as measurement of purity for choosing root node .

3. [16 points] Continue with the same training data in Problem 2. Suppose before the tree construction, we receive one more training instance where Outlook's value is missing: {Outlook: Missing, Temperature: Mild, Humidity: Normal, Wind: Weak, Play: Yes}.

(a) [3 points] Use the most common value in the training data as the missing value, and calculate the information gains of the four features. Note that if there is a tie for the most common value, you can choose any value in the tie. Indicate the best feature.
Answer:

$$Gain(S, A) = ME(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} ME(S_v)$$

ME (S) = $\frac{5}{15}$

Taking missing in outlook as Sunny although there's a tie
Outlook:
Sunny : p = 3, n = 3 , ME (Sunny) = $\frac{3}{6}$
Overcast : p = 4, n = 0 , ME (Overcast) = 0
Rain : p = 3, n = 2 , ME (Rain) = $\frac{2}{5}$

Temperature:

Hot : p = 2, n = 2 , ME (Hot) = $\frac{2}{4}$
Mild : p = 5, n = 2 , ME (Mild) = $\frac{2}{7}$
Cool : p = 3, n = 1 , ME (Cool) = $\frac{1}{4}$

Humidity:
High : p = 3, n = 4 , ME (High) = $\frac{3}{7}$
Normal : p = 7, n = 1 , ME (Normal) = $\frac{1}{8}$
Low : p = n = 0 , ME (Low) = 0

Wind:
Strong : p = 3, n = 3 , ME (Strong) = $\frac{3}{6}$
weak : p = 7, n = 2 , ME (weak) = $\frac{2}{9}$

$$Gain(S, Outlook) = ME(S) - \frac{3}{6} * \frac{6}{15} - 0 - \frac{2}{5} * \frac{5}{15} = 0$$
$$Gain(S, Temperature) = ME(S) - \frac{2}{4} * \frac{4}{15} - \frac{1}{4} * \frac{4}{15} - \frac{2}{7} * \frac{7}{15} = 0$$
$$Gain(S, Humidity) = ME(S) - \frac{3}{7} * \frac{7}{15} - \frac{1}{8} * \frac{8}{15} - 0 = 0.067$$
$$Gain(S, Wind) = ME(S) - \frac{3}{6} * \frac{6}{15} - \frac{2}{9} * \frac{9}{15} = 0$$

Based on **highest Gain**, we split on **Humidity**

(b) [3 points] Use the most common value among the training instances with the same label, namely, their attribute "Play" is "Yes", and calculate the information gains of the four features. Again if there is a tie, you can choose any value in the tie. Indicate the best feature.
Answer:
The new feature added here with the most common value with Play: Yes is:
Outlook: Overcast, Temperature: Mild, Humidity: Normal, Wind: Weak, Play: Yes
Outlook:
Sunny : p = 2, n = 3 , ME (Sunny) = $\frac{2}{5}$
Overcast : p = 5, n = 0 , ME (Overcast) = 0
Rain : p = 3, n = 2 , ME (Rain) = $\frac{2}{5}$

Temperature:
Hot : p = 2, n = 2 , ME (Hot) = $\frac{2}{4}$
Mild : p = 5, n = 2 , ME (Mild) = $\frac{2}{7}$
Cool : p = 3, n = 1 , ME (Cool) = $\frac{1}{4}$

Humidity:
High : p = 3, n = 4 , ME (High) = $\frac{3}{7}$

Normal : p = 7, n = 1 , ME (Normal) = $\frac{1}{8}$
Low : p = n = 0 , ME (Low) = 0

Wind:
Strong : p = 3, n = 3 , ME (Strong) = $\frac{3}{6}$
weak : p = 7, n = 2 , ME (weak) = $\frac{2}{9}$

$$Gain(S, Outlook) = ME(S) - \frac{2}{5} * \frac{5}{15} - 0 - \frac{2}{5} * \frac{5}{15} = 0.067$$

$$Gain(S, Temperature) = ME(S) - \frac{2}{4} * \frac{4}{15} - \frac{1}{4} * \frac{4}{15} - \frac{2}{7} * \frac{7}{15} = 0$$

$$Gain(S, Humidity) = ME(S) - \frac{3}{7} * \frac{7}{15} - \frac{1}{8} * \frac{8}{15} - 0 = 0.067$$

$$Gain(S, Wind) = ME(S) - \frac{3}{6} * \frac{6}{15} - \frac{2}{9} * \frac{9}{15} = 0$$

Based on **highest Gain**,now there's a tie and lets split on **Outlook**

(c) [3 points] Use the fractional counts to infer the feature values, and then calculate the information gains of the four features. Indicate the best feature.
Answer:
ME(S) = $\frac{5}{15}$
Using fractional counts of the attribute values in training data:
Outlook:
Size: Sunny : $5 + \frac{5}{14}$ Overcast : $4 + \frac{4}{14}$ Rain : $5 + \frac{5}{14}$
Sunny : p = $2 + \frac{5}{14}$, n = 3 , ME (Sunny) = $\frac{2+\frac{5}{14}}{5+\frac{5}{14}}$
Overcast : p = $4+\frac{4}{14}$, n = 0 , ME (Overcast) = 0
Rain : p = $3 + \frac{5}{14}$, n = 2 , ME (Rain) = $\frac{2}{5+\frac{5}{14}}$
Temperature:
Hot : p = 2, n = 2 , ME (Hot) = $\frac{2}{4}$
Mild : p = 5, n = 2 , ME (Mild) = $\frac{2}{7}$
Cool : p = 3, n = 1 , ME (Cool) = $\frac{1}{4}$

Humidity:
High : p = 3, n = 4 , ME (High) = $\frac{3}{7}$
Normal : p = 7, n = 1 , ME (Normal) = $\frac{1}{8}$
Low : p = n = 0 , ME (Low) = 0

Wind:
Strong : p = 3, n = 3 , ME (Strong) = $\frac{3}{6}$
weak : p = 7, n = 2 , ME (weak) = $\frac{2}{9}$

$$Gain(S, Outlook) = ME(S) - \frac{2 + \frac{5}{14}}{5 + \frac{5}{14}} * \frac{5 + \frac{5}{14}}{15} - 0 - \frac{2}{5 + \frac{5}{14}} * \frac{5 + \frac{5}{14}}{15} = 0.0428$$

$$Gain(S, Temperature) = ME(S) - \frac{2}{4} * \frac{4}{15} - \frac{1}{4} * \frac{4}{15} - \frac{2}{7} * \frac{7}{15} = 0$$

$$Gain(S, Humidity) = ME(S) - \frac{3}{7} * \frac{7}{15} - \frac{1}{8} * \frac{8}{15} - 0 = 0.067$$

$$Gain(S, Wind) = ME(S) - \frac{3}{6} * \frac{6}{15} - \frac{2}{9} * \frac{9}{15} = 0$$

Based on **highest gain**, we can split on **Humidity**

(d) [7 points] Continue with the fractional examples, and build the whole free with information gain. List every step and the final tree structure.

Answer:

As I already solved Step 1 in Problem 3.c , I will start from step 2 and go towards developing decision tree.

Step 1: So, first split on **Humidity** developing decision tree.

Step 2:

Find the attribute that best Splits S again for High.

ME(High) = $\frac{3}{7}$

Current attributes: A = Outlook, Temperature, Wind

$$Gain(High, Outlook) = ME(High) - \frac{3}{7} * 0 - \frac{2}{7} * 0 - \frac{2}{7} * \frac{1}{2} = 0.285$$

$$Gain(High, Temperature) = ME(High) - \frac{3}{7} * \frac{1}{3} - \frac{4}{7} * \frac{2}{4} - 0 = 0$$

$$Gain(High, Wind) = ME(High) - \frac{3}{7} * \frac{1}{3} - \frac{4}{7} * \frac{2}{4} = 0$$

Based on highest information gain, we will split **High Humidity** with attribute of Outlook.

Under **Outlook, Sunny** and **Overcast** reached the same label, a leaf node with label No, Yes respectively.

Step 3:

Find the attribute that best Splits for Rain.

ME(Rain) = $\frac{1}{2} = 0.5$ Current attributes: A = Temperature, Wind

$$Gain(Rain, Temperature) = ME(Rain) - 0 - \frac{2}{2} * \frac{1}{2} - 0 = 0$$

$$Gain(Rain, Wind) = ME(Rain) - 0 - 0 * \frac{1}{2} - 0 * \frac{1}{2} = 0.5$$

Based on highest information gain, we will split **Wind** with attribute of Rain.

Under **Wind**, **Strong and Weak** have the same label, a leaf node with label No and Yes labelled respectively.

Step 4:

Now , back to **Humidity with Normal** for further split.

ME(Normal) = $\frac{1}{8} = 0.125$

Current attributes: A = Outlook, Temperature, Wind

$$Gain(Normal, Outlook) = ME(Normal) - 0 - 0 - \frac{3.357}{8} * \frac{1}{3.357} = 0$$

$$Gain(Normal, Temperature) = ME(Normal) - 0 - 0 - \frac{4}{8} * \frac{1}{4} = 0$$

$$Gain(Normal, Wind) = ME(Normal) - \frac{3}{8} * \frac{1}{3} = 0$$

None of the attributes produce any information gain. Lets split on **Outlook**
Sunny and Overcast have the same label, a leaf node with Yes and Yes labelled respectively.
Step 5: Find the attribute that best Splits on **Rain**.
ME(Rain) = $\frac{1}{3.357}$ = 0.297
Current attributes: A = Temperature, Wind

$$Gain(Rain, Temperature) = ME(Rain) - 0 - 0 - \frac{2}{3.357} * \frac{1}{2} = 0$$

$$Gain(Rain, Wind) = ME(Rain) - 0 = 0.297$$

Wind has the **highest** information gain, split **Rain** with attribute of **Wind**
Strong and Weak have the same label, a leaf node with No and Yes labelled respectively.
Step 6: Low Humidity is null, a leaf node with the most common label is Yes.

so the tree will be:

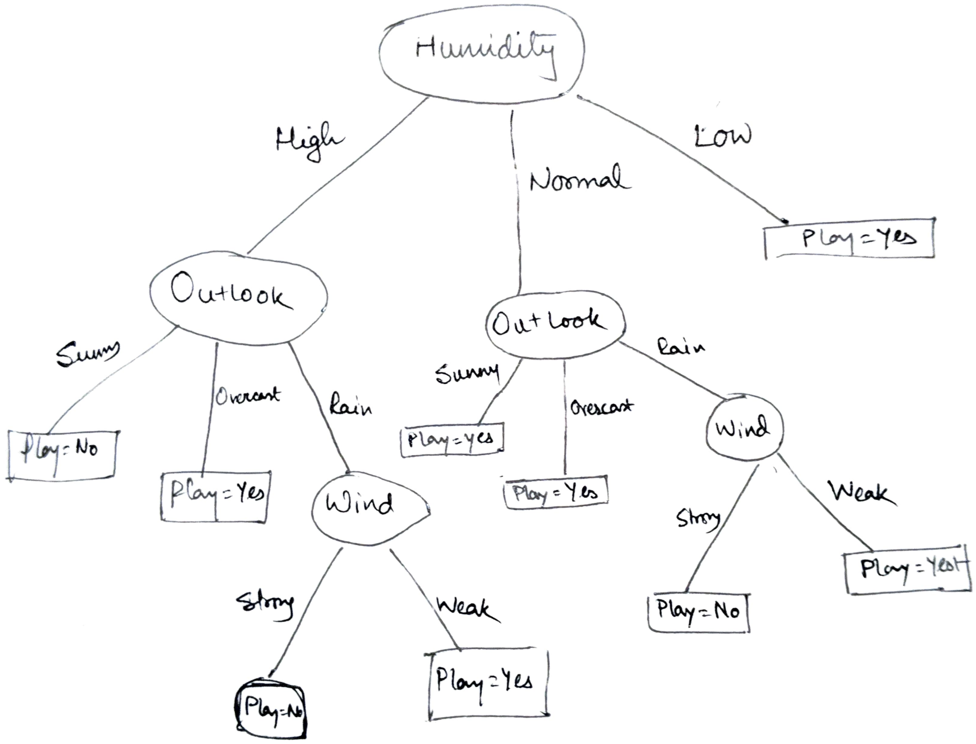

Figure 4: ME dependent:Play Tennis Tree

4. [**Bonus question 1**] [5 points]. Prove that the information gain is always non-negative. That means, as long as we split the data, the purity will never get worse! (Hint: use convexity)

Answer:

Information gain from using attribute A at any node is:

$$IG = I(\frac{p}{p+n}, \frac{n}{p+n}) - \sum_{i=0}^{1}[\frac{p_i+n_i}{p+n} I(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i})]$$

Let

$$f(x) = -x\log_2 x - (1-x)\log_2(1-x)$$
$$f'(x) = -\log_2 x + \log_2(1-x)$$

and

$$f''(x) = -\frac{1}{\ln 2} * \frac{1}{x(1-x)}$$

Since $x \in (0,1)$ we have $f''(x) < 0$ and it's concave.

$$\sum_{i=0}^{1} \frac{p_i+n_i}{p+n} I(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i})$$

$$= \frac{p_0+n_0}{p+n} I(\frac{p_0}{p_0+n_0}, \frac{n_0}{p_0+n_0}) + \frac{p_1+n_1}{p+n} I(\frac{p_1}{p_1+n_1}, \frac{n_1}{p_1+n_1})$$
$$= \frac{p_0+n_0}{p+n} f(\frac{p_0}{p_0+n_0}) + \frac{p_1+n_1}{p+n} f(\frac{p_1}{p_1+n_1})$$

Following Jensen's Inequality:

$$\sum_x p(x)f(x) \leq f(\sum_x p(x)x)$$

where $\sum_x p(x) = 1$, $p(x) \geq 0$ and f(x) is concave.

So,

$$\sum_{i=0}^{1} \frac{p_i+n_i}{p+n} I(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}) \leq f(\frac{p_0+n_0}{p+n}\frac{p_0}{p_0+n_0}) + f(\frac{p_1+n_1}{p+n}\frac{p_1}{p_1+n_1})$$

$$= f(\frac{p_0+p_1}{p+n}) = f(\frac{p}{p+n}) = I(\frac{p}{p+n}, \frac{n}{p+n})$$

So,

$$IG = I(\frac{p}{p+n}, \frac{n}{p+n}) - \sum_{i=0}^{1}[\frac{p_i+n_i}{p+n} I(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i})] \geq 0 \ldots [proved]$$

5. [**Bonus question 2**] [5 points]. We have discussed how to use decision tree for regression (i.e., predict numerical values) — on the leaf node, we simply use the average of the (numerical) labels as the prediction. Now, to construct a regression tree, can you invent a gain to select the best attribute to split data in ID3 framework?

Answer: Decision tree for Classification is developed by 2 measures : Entropy and Information Gain. But since we are predicting continuous variables for regression, we cannot calculate the entropy rather concentrating on how much our predictions deviate from the original value and generally it is calculated as mean square error (MSE).

2 Decision Tree Practice [60 points]

1. [5 Points] Starting from this assignment, we will build a light-weighted machine learning library. To this end, you will first need to create a code repository in Github.com. Please refer to the short introduction in the appendix and the official tutorial to create an account and repository. Please commit a README.md file in your repository, and write one sentence: "This is a machine learning library developed by **Your Name** for CS5350/6350 in University of Utah". You can now create a first folder, "DecisionTree". Please leave the link to your repository in the homework submission. We will check if you have successfully created it.
   Answer:
   `https://github.com/EkataU/CS_6350_Machine_Learning/tree/main/DecisionTree`

2. [30 points] We will implement a decision tree learning algorithm for car evaluation task. The dataset is from UCI repository(`https://archive.ics.uci.edu/ml/datasets/car+evaluation`). Please download the processed dataset (car.zip) from Canvas. In this task, we have 6 car attributes, and the label is the evaluation of the car. The attribute and label values are listed in the file "data-desc.txt". All the attributes are categorical. The training data are stored in the file "train.csv", consisting of $1,000$ examples. The test data are stored in "test.csv", and comprise 728 examples. In both training and test datasets, attribute values are separated by commas; the file "data-desc.txt" lists the attribute names in each column.

   Note: we highly recommend you to use Python for implementation, because it is very convenient to load the data and handle strings. For example, the following snippet reads the CSV file line by line and split the values of the attributes and the label into a list, "terms". You can also use "dictionary" to store the categorical attribute values. In the web are numerous tutorials and examples for Python. if you have issues, just google it!

```
with open(CSVfile, 'r') as f:
     for line in f:
            terms = line.strip().split(',')
            process one training example
```

   (a) [15 points] Implement the ID3 algorithm that supports, information gain, majority error and gini index to select attributes for data splits. Besides, your ID3 should allow users to set the maximum tree depth. Note: you do not need to convert categorical attributes into binary ones and your tree can be wide here.

(b) [10 points] Use your implemented algorithm to learn decision trees from the training data. Vary the maximum tree depth from 1 to 6 — for each setting, run your algorithm to learn a decision tree, and use the tree to predict both the training and test examples. Note that if your tree cannot grow up to 6 levels, you can stop at the maximum level. Report in a table the average prediction errors on each dataset when you use information gain, majority error and gini index heuristics, respectively.

Answer:

$depth$	IG_{train}	ME_{train}	GI_{train}	IG_{test}	ME_{test}	GI_{test}
1	0.3013	0.3013	0.3013	0.297	0.297	0.297
2	0.222	0.3013	0.222	0.223	0.297	0.223
3	0.181	0.263	0.176	0.196	0.285	0.184
4	0.0821	0.176	0.089	0.151	0.256	0.1375
5	0.0270	0.069	0.0270	0.094	0.232	0.094
6	0.0	0.0	1	0.094	0.232	0.094

Table 2: Training data for average prediction error for car dataset

(c) [5 points] What can you conclude by comparing the training errors and the test errors?

Answer:
We can see that training error is gradually decreasing with the depth level however test data set are showing best result till certain depth level and from the table we can reach to optimal level. On the other note, if we allow more than certain level then over-fitting may incur.

3. [25 points] Next, modify your implementation a little bit to support numerical attributes. We will use a simple approach to convert a numerical feature to a binary one. We choose the media (NOT the average) of the attribute values (in the training set) as the threshold, and examine if the feature is bigger (or less) than the threshold. We will use another real dataset from UCI repository(`https://archive.ics.uci.edu/ml/datasets/Bank+Marketing`). This dataset contains 16 attributes, including both numerical and categorical ones. Please download the processed dataset from Canvas (bank.zip). The attribute and label values are listed in the file "data-desc.txt". The training set is the file "train.csv", consisting of $5,000$ examples, and the test "test.csv" with $5,000$ examples as well. In both training and test datasets, attribute values are separated by commas; the file "data-desc.txt" lists the attribute names in each column.

   (a) [10 points] Let us consider "unknown" as a particular attribute value, and hence we do not have any missing attributes for both training and test. Vary the maximum tree depth from 1 to 16 — for each setting, run your algorithm to learn a decision tree, and use the tree to predict both the training and test examples. Again, if your tree cannot grow up to 16 levels, stop at the maximum level. Report

in a table the average prediction errors on each dataset when you use information gain, majority error and gini index heuristics, respectively.
Answer:

depth	IG_{train}	ME_{train}	GI_{train}	IG_{test}	ME_{test}	GI_{test}
1	0.119	0.1088	0.1088	0.1248	0.1166	0.1166
2	0.106	0.1042	0.1042	0.1114	0.1088	0.1088
3	0.1006	0.096	0.0936	0.1074	0.1134	0.1162
4	0.0812	0.085	0.0762	0.1212	0.117	0.1242
5	0.0636	0.0736	0.0616	0.1318	0.12	0.1346
6	0.049	0.069	0.0494	0.1368	0.122	0.1438
7	0.038	0.0642	0.0384	0.1444	0.1244	0.1520
8	0.0316	0.058	0.0312	0.1514	0.1284	0.155
9	0.0258	0.0546	0.0272	0.1546	0.3106	0.159
10	0.023	0.0516	0.0228	0.1588	0.1320	0.1616
11	0.022	0.0482	0.022	0.1636	0.1346	0.1660
12	0.0218	0.0418	0.0218	0.1632	0.1450	0.1656
13	0.0218	0.037	0.0218	0.1632	0.1502	0.1656
14	0.0218	0.0306	0.0218	0.1632	0.1566	0.1656
15	0.0218	0.0258	0.0218	0.1632	0.1638	0.1656
16	0.0218	0.0218	0.0218	0.1632	0.16723	0.1656

Table 3: Training data for average prediction error for bank dataset with unknown as value

(b) [10 points] Let us consider ”unknown” as attribute value missing. Here we simply complete it with the majority of other values of the same attribute in the training set. Vary the maximum tree depth from 1 to 16 — for each setting, run your algorithm to learn a decision tree, and use the tree to predict both the training and test examples. Report in a table the average prediction errors on each dataset when you use information gain, majority error and gini index heuristics, respectively.
Answer:

$depth$	IG_{train}	ME_{train}	GI_{train}	IG_{test}	ME_{test}	GI_{test}
1	0.119	0.1088	0.1088	0.1248	0.1166	0.1166
2	0.1062	0.1050	0.1052	0.1142	0.1102	0.1104
3	0.1022	0.0976	0.101	0.1078	0.1156	0.1084
4	0.0876	0.0868	0.088	0.1238	0.1148	0.1212
5	0.0704	0.0776	0.0722	0.1314	0.1168	0.1298
6	0.056	0.718	0.566	0.1416	0.1192	0.1380
7	0.046	0.069	0.0458	0.1468	0.1236	0.1480
8	0.0394	0.066	0.0390	0.1508	0.1254	0.1522
9	0.0336	0.0608	0.0332	0.1558	0.1282	0.1574
10	0.0296	0.0578	0.0292	0.1598	0.1306	0.1588
11	0.0286	0.0530	0.0284	0.1616	0.1362	0.1608
12	0.0284	0.0488	0.0284	0.1616	0.1388	0.1616
13	0.0282	0.0440	0.0282	0.1628	0.1442	0.1628
14	0.0282	0.037	0.0282	0.1628	0.1542	0.1628
15	0.0282	0.0316	0.0282	0.1628	0.1578	0.1628
16	0.0282	0.0282	0.0282	0.1628	0.1636	0.1628

Table 4: Training data for average prediction error for bank dataset with unknown as missing value

(c) [5 points] What can you conclude by comparing the training errors and the test errors, with different tree depths, as well as different ways to deal with "unknown" attribute values?

Answer:

When unknown values assigned to majority label gain is increased a little bit and converged early.