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

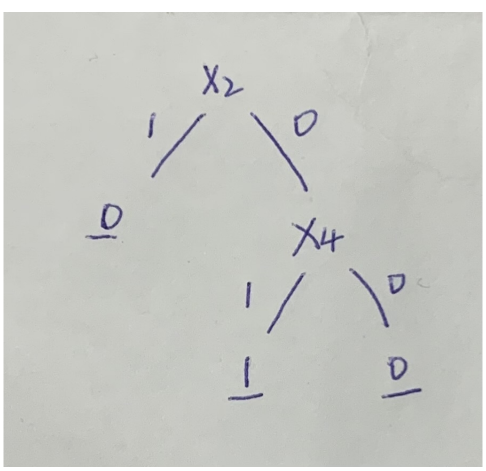
Table 1: Training data for a Boolean classifier

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.

   For ID3(S, $\{x_1, x_2, x_3, x_4\}$, $\{0, 1\}$):
   Entropy(S)=H(y)=$-\frac{2}{7}\log_2\frac{2}{7} - \frac{5}{7}\log_2\frac{5}{7} = 0.863$
   $x_1 = 0$: 5 of 7 examples, $p = \frac{1}{5}$, $n = \frac{4}{5}, H_0 = 0.722$
   $x_1 = 1$: 2 of 7 examples, $p = \frac{1}{2}$, $n = \frac{1}{2}, H_1 = 1$
   $Gain(S, x_1) = 0.863 - (\frac{5}{7} \times 0.722 + \frac{2}{7}) = 0.062$
   $x_2 = 0$: 3 of 7 examples, $p = \frac{2}{3}$, $n = \frac{1}{3}, H_0 = 0.918$
   $x_2 = 1$: 4 of 7 examples, $p = 0$, $n = 1, H_1 = 0$
   $Gain(S, x_2) = 0.863 - (\frac{3}{7} \times 0.918) = 0.470$

$x_3 = 0$: 4 of 7 examples, $p = \frac{1}{4}$, $n = \frac{3}{4}$, $H_0 = 0.811$
$x_3 = 1$: 3 of 7 examples, $p = \frac{1}{3}$, $n = \frac{2}{3}$, $H_1 = 0.918$
$Gain(S, x_3) = 0.863 - (\frac{4}{7} \times 0.811 + \frac{3}{7}) \times 0.918 = 0.006$
$x_4 = 0$: 4 of 7 examples, $p = 0$, $n = 1$, $H_0 = 0$
$x_4 = 1$: 3 of 7 examples, $p = \frac{2}{3}$, $n = \frac{1}{3}$, $H_1 = 0.918$
$Gain(S, x_4) = 0.863 - (\frac{3}{7} \times 0.918) = 0.470$
Since $Gain(S, x_2) = Gain(S, x_4) = 0.470$, we can choose either one as the root node to split. I will use x_2 as root node.
Then create two branches, which corresponding to label 0 and 1, and run ID3 on each branch.
For ID3$(S_{x_2=1}, \{x_1, x_3, x_4\}, \{0, 1\})$**:** all examples have same label, so return a leaf node with label 0.
For ID3$(S_{x_2=0}, \{x_1, x_3, x_4\}, \{0, 1\})$**:**
Entropy$(S_{x_2=0})$=$-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918$
$x_1 = 0$: 2 of 3 examples, $p = \frac{1}{2}$, $n = \frac{1}{2}$, $H_0 = 1$
$x_1 = 1$: 1 of 3 examples, $p = 1$, $n = 0$, $H_1 = 0$
$Gain(S, x_1) = 0.918 - (\frac{2}{3}) = 0.251$
$x_3 = 0$: 1 of 3 examples, $p = 1$, $n = 0$, $H_0 = 0$
$x_3 = 1$: 2 of 3 examples, $p = \frac{1}{2}$, $n = \frac{1}{2}$, $H_1 = 1$
$Gain(S, x_3) = 0.918 - (\frac{2}{3}) = 0.251$
$x_4 = 0$: 1 of 3 examples, $p = 0$, $n = 1$, $H_1 = 0$
$x_4 = 1$: 2 of 3 examples, $p = 1$, $n = 0$, $H_1 = 0$
$Gain(S, x_4) = 0.918 - (0) = 0.918$
Since Gain(S,x_4) is the largest, so we will choose x_4 as the root node to split.
Then create two branches, which corresponding to label 0 and 1, and run run ID3 on each branch.
For ID3$(S_{x_2=0,x_4=1}, \{x_1, x_3\}, \{0, 1\})$**:** all examples have same label, so return a leaf node with label 1.
For ID3$(S_{x_2=0,x_4=0}, \{x_1, x_3\}, \{0, 1\})$**:** all examples have same label, so return a leaf node with label 0.



(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.

Boolean function is:
y=0, when (x_2=1) or (x_2=0 and x_4=0)
y=1, when x_2=0 and x_4=1

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

‘

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.

   For ID3(S, {Outlook,Temperature,Humidity,Wind}, {Yes, No}):
   ME(S)=$\frac{5}{14} = 0.357$
   O-Sunny: 5 of 14, Overcast: 4 of 14, Rainy: 5 of 14
   $Gain(S, O) = 0.357 - (\frac{5}{14} \times \frac{2}{5} + \frac{4}{14} \times 0 + \frac{5}{14} \times \frac{2}{5}) = 0.071$
   T-Hot: 4 of 14, Medium: 6 of 14, Cool: 4 of 14
   $Gain(S, T) = 0.357 - (\frac{4}{14} \times \frac{1}{2} + \frac{6}{14} \times \frac{2}{6} + \frac{4}{14} \times \frac{1}{4}) = 0$
   H-High: 7 of 14, Normal: 7 of 14, Low: 0 of 14
   $Gain(S, H) = 0.357 - (\frac{1}{2} \times \frac{3}{7} + \frac{1}{2} \times \frac{1}{7}) = 0.071$
   W-Strong: 6 of 14, Weak: 8 of 14
   $Gain(S, W) = 0.357 - (\frac{6}{14} \times \frac{1}{2} + \frac{8}{14} \times \frac{1}{4}) = 0$

Since $Gain(S,O) = Gain(S,H) = 0.071$, we can choose either one as the root node to split. I will use Outlook as root node.
Then create three branches, which corresponding to label Sunny, Overcast and Rainy, and run ID3 on each branch.

For ID3($S_{O=Sunny}$, {Temperature,Humidity,Wind}, {Yes, No}):
ME($S_{O=Sunny}$)=$\frac{2}{5} = 0.4$
T-Hot: 2 of 5, Medium: 2 of 5, Cool: 1 of 5
$Gain(S_{O=Sunny}, T) = 0.4 - (\frac{2}{5} \times 0 + \frac{2}{5} \times \frac{1}{2} + \frac{1}{5} \times 0) = 0.2$
H-High: 3 of 5, Normal: 2 of 5, Low: 0 of 5
$Gain(S_{O=Sunny}, H) = 0.4 - (\frac{3}{5} \times 0 + \frac{2}{5} \times 0) = 0.4$
W-Strong: 2 of 5, Weak: 3 of 5
$Gain(S_{O=Sunny}, W) = 0.4 - (\frac{2}{5} \times \frac{1}{2} + \frac{3}{5} \times \frac{1}{3}) = 0$
Since $Gain(S_{O=Sunny}, H)$ is the largest one, we will choose Humidity as root node to split.
Then create three branches, which corresponding to label High, Normal, and Low, and run ID3 on each branch.

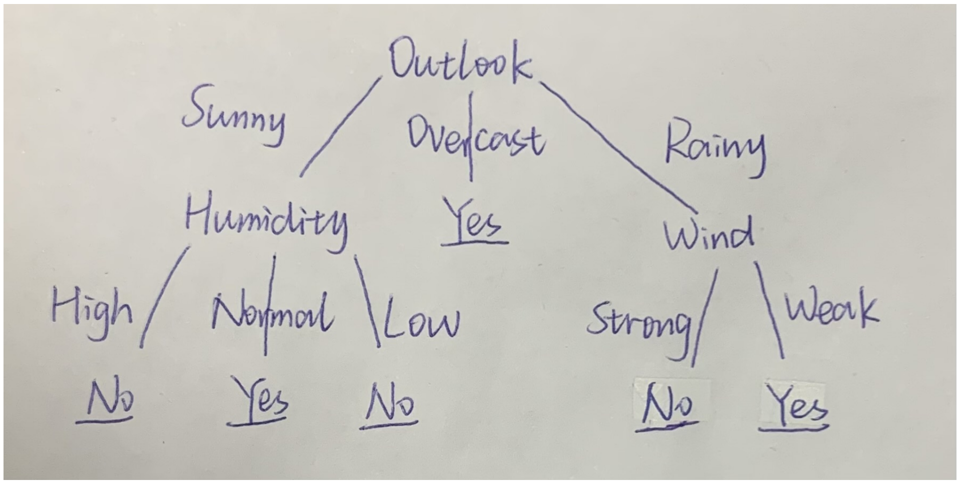
For ID3($S_{O=Sunny,H=High}$, {Temperature,Wind}, {Yes, No}):
all examples have same label, so return a leaf node with label No.
For ID3($S_{O=Sunny,H=Normal}$, {Temperature,Wind}, {Yes, No}):
all examples have same label, so return a leaf node with label Yes.
For ID3($S_{O=Sunny,H=Low}$, {Temperature,Wind}, {Yes, No}):
all examples have same label but attribute is empty, so return a leaf node with most common label, which is No.

For ID3($S_{O=Overcast}$, {Temperature,Humidity,Wind}, {Yes, No}):
all examples have same label, so return a leaf node with label Yes.

For ID3($S_{O=Rainy}$, {Temperature,Humidity,Wind}, {Yes, No}):
ME($S_{O=Rainy}$)=$\frac{2}{5} = 0.4$
T-Hot: 0 of 5, Medium: 3 of 5, Cool: 2 of 5
$Gain(S_{O=Rainy}, T) = 0.4 - (\frac{1}{3} \times \frac{3}{5} + \frac{1}{2} \times \frac{2}{5}) = 0$
H-High: 2 of 5, Normal: 3 of 5, Low: 0 of 5
$Gain(S_{O=Rainy}, H) = 0.4 - (\frac{2}{5} \times \frac{1}{2} + \frac{3}{5} \times \frac{1}{3}) = 0$
W-Strong: 2 of 5, Weak: 3 of 5
$Gain(S_{O=Rainy}, W) = 0.4 - (\frac{2}{5} \times 0 + \frac{3}{5} \times \mathbf{0}) = 0.4$
Since $Gain(S_{O=Rainy}, W)$ is the largest one, we will choose Wind as root node to split.
Then create two branches, which corresponding to label Strong and Weak, and run ID3 on each branch.

For ID3($S_{O=Rainy,W=Strong}$, {Temperature,Humidity}, {Yes, No}):
all examples have same label, so return a leaf node with label No.

For ID3($S_{O=Sunny,W=Weak}$, {Temperature,Humidity}, {Yes, No}):
all examples have same label, so return a leaf node with label Yes.



(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.

For ID3(S, {Outlook,Temperature,Humidity,Wind}, {Yes, No}):

GI(S)=$1-((\frac{5}{14})^2+(\frac{9}{14})^2)=0.459$

Outlook:

$GI(O=Sunny)=1-((\frac{3}{5})^2+(\frac{2}{5})^2)=0.48, GI(O=Overcast)=1-((\frac{5}{5})^2+(\frac{0}{5})^2)=0$, $GI(O=Rainy)=1-((\frac{3}{5})^2+(\frac{2}{5})^2)=0.48$

$Gain(S,Outlook)=0.459-(\frac{5}{14}\times 0.48+\frac{4}{14}\times 0+\frac{5}{14}\times 0.48)=0.116$

Temperature:

$GI(T=Hot)=1-((\frac{1}{2})^2+(\frac{1}{2})^2)=0.5, GI(T=Medium)=1-((\frac{1}{3})^2+(\frac{2}{3})^2)=0.444$, $GI(T=Cool)=1-((\frac{3}{4})^2+(\frac{1}{4})^2)=0.375$

$Gain(S,Outlook)=0.459-(\frac{4}{14}\times 0.5+\frac{6}{14}\times 0.444+\frac{4}{14}\times 0.375)=0.019$

Humidity:

$GI(H=High)=1-((\frac{3}{7})^2+(\frac{4}{7})^2)=0.490, GI(H=Normal)=1-((\frac{6}{7})^2+(\frac{1}{7})^2)=0.245$

$Gain(S,Humidity)=0.459-(\frac{7}{14}\times 0.490+\frac{7}{14}\times 0.245)=0.092$

Wind:

$GI(W=Strong)=1-((\frac{1}{2})^2+(\frac{1}{2})^2)=0.5, GI(W=Weak)=1-((\frac{1}{4})^2+(\frac{3}{4})^2)=0.375$

$Gain(S,Wind)=0.459-(\frac{6}{14}\times 0.5+\frac{8}{14}\times 0.375)=0.030$

Since $Gain(S,Outlook)$ is the largest one, so we will choose Outlook as root node to split.

Then create three branches, which corresponding to label Sunny, Overcast and Rainy, and run ID3 on each branch.

For ID3($S_{O=Sunny}$, {Temperature,Humidity,Wind}, {Yes, No}):

$GI(S_{O=Sunny})=1-((\frac{2}{5})^2+(\frac{3}{5})^2)=0.48$

Temperature:

$GI(T=Hot)=1-((\frac{2}{2})^2)=0, GI(T=Medium)=1-((\frac{1}{2})^2+(\frac{1}{2})^2)=0.5$,

$GI(T = Cool) = 1 - 1 = 0$
$Gain(S_{O=Sunny}, Temperature) = 0.48 - (\frac{2}{5} \times 0.5) = 0.28$
Humidity:
$GI(H = High) = 1 - 1 = 0$,$GI(H = Normal) = 1 - 1 = 0$
$Gain(S_{O=Sunny}, Humidity) = 0.48 - (0) = 0.48$
Wind:
$GI(W = Strong) = 1 - ((\frac{1}{2})^2 + (\frac{1}{2})^2) = 0.5$,$GI(W = Weak) = 1 - ((\frac{1}{3})^2 + (\frac{2}{3})^2) = 0.444$
$Gain(S, Wind) = 0.48 - (\frac{2}{5} \times 0.5 + \frac{3}{5} \times 0.444) = 0.014$
Since $Gain(S_{O=Sunny}, Humidity)$ is the largest one, so we will choose Humidity as root node to split.
Then create three branches, which corresponding to label High, Normal and Low, and run ID3 on each branch.

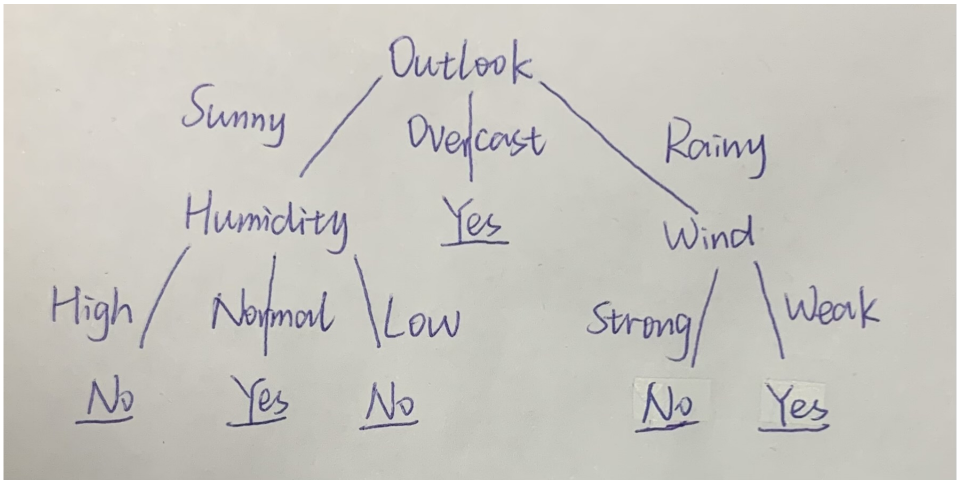
For ID3($S_{O=Sunny,H=High}$, {Temperature,Wind}, {Yes, No}):
all examples have same label, so return a leaf node with label No.
For ID3($S_{O=Sunny,H=Normal}$, {Temperature,Wind}, {Yes, No}):
all examples have same label, so return a leaf node with label Yes.
For ID3($S_{O=Sunny,H=Low}$, {Temperature,Wind}, {Yes, No}):
all examples have same label but attribute is empty, so return a leaf node with most common label, which is No.

For ID3($S_{O=Overcast}$, {Temperature,Humidity,Wind}, {Yes, No}):
all examples have same label, so return a leaf node with label Yes.

For ID3($S_{O=Rainy}$, {Temperature,Humidity,Wind}, {Yes, No}):
$GI(S_{O=Rainy}) = 1 - ((\frac{2}{5})^2 + (\frac{3}{5})^2) = 0.48$
Temperature:
$GI(T = Medium) = 1-((\frac{1}{3})^2+(\frac{2}{3})^2) = 0.444$, $GI(T = Cool) = 1-((\frac{1}{2})^2+(\frac{1}{2})^2) = 0.5$
$Gain(S_{O=Rainy}, Temperature) = 0.48 - (\frac{3}{5} \times 0.444 + \frac{2}{5} \times 0.5) = 0.014$
Humidity:
$GI(H = High) = 1 - ((\frac{1}{2})^2 + (\frac{1}{2})^2) = 0.5$,$GI(H = Normal) = 1 - ((\frac{1}{3})^2 + (\frac{2}{3})^2) = 0.444$
$Gain(S_{O=Rainy}, Humidity) = 0.48 - (\frac{2}{5} \times 0.5 + \frac{3}{5} \times 0.444) = 0.014$
Wind:
$GI(W = Strong) = 1 - 1 = 0$,$GI(W = Weak) = 1 - 1 = 0$
$Gain(S_{O=Rainy}, Wind) = 0.48 - (0) = 0.48$
Since $Gain(S_{O=Rainy}, Wind)$ is the largest one, so we will choose Wind as root node to split.
Then create two branches, which corresponding to label Strong and Weak, and run ID3 on each branch.

For ID3($S_{O=Rainy,W=Strong}$, {Temperature,Humidity}, {Yes, No}):
all examples have same label, so return a leaf node with label No.

For ID3($S_{O=Rainy,W=Weak}$, {Temperature,Humidity}, {Yes, No}):
all examples have same label, so return a leaf node with label Yes.



(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?
There is no difference for all those tree structures. They are the same since they split based on the same attribute, and the reason for that is we are splitting it based on Information Gain. Also there is a tie when choosing the root node using Majority Error, so if I choose Humidity attribute as root node in (a), the tree structure might be different.

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.
The most common value I choose is Sunny.
And I'm using ME to calculate information gain.
ME(S)=$\frac{5}{15} = 0.333$
O-Sunny: 6 of 15, Overcast: 4 of 15, Rainy: 5 of 15
$Gain(S, O) = 0.333 - (\frac{6}{15} \times \frac{1}{2} + \frac{4}{15} \times 0 + \frac{5}{15} \times \frac{2}{5}) = 0$
T-Hot: 4 of 15, Medium: 7 of 15, Cool: 4 of 15
$Gain(S, T) = 0.333 - (\frac{4}{15} \times \frac{1}{2} + \frac{7}{15} \times \frac{2}{7} + \frac{4}{15} \times \frac{1}{4}) = 0$
H-High: 7 of 15, Normal: 8 of 15, Low: 0 of 14
$Gain(S, H) = 0.333 - (\frac{7}{15} \times \frac{3}{7} + \frac{8}{15} \times \frac{1}{8}) = 0.067$
W-Strong: 6 of 15, Weak: 9 of 15
$Gain(S, W) = 0.333 - (\frac{6}{15} \times \frac{1}{2} + \frac{9}{15} \times \frac{2}{9}) = 0$
$Gain(S, H)$ is the largest one. Humidity is the best.

(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.

The most common value I choose is Overcast.

And I'm using ME to calculate information gain.

ME(S)=$\frac{5}{15} = 0.333$

O-Sunny: 5 of 15, Overcast: 5 of 15, Rainy: 5 of 15

$Gain(S, O) = 0.333 - (\frac{5}{15} \times \frac{2}{5} + \frac{5}{15} \times 0 + \frac{5}{15} \times \frac{2}{5}) = 0.067$

T-Hot: 4 of 15, Medium: 7 of 15, Cool: 4 of 15

$Gain(S, T) = 0.333 - (\frac{4}{15} \times \frac{1}{2} + \frac{7}{15} \times \frac{2}{7} + \frac{4}{15} \times \frac{1}{4}) = 0$

H-High: 7 of 15, Normal: 8 of 15, Low: 0 of 14

$Gain(S, H) = 0.333 - (\frac{7}{15} \times \frac{3}{7} + \frac{8}{15} \times \frac{1}{8}) = 0.067$

W-Strong: 6 of 15, Weak: 9 of 15

$Gain(S, W) = 0.333 - (\frac{6}{15} \times \frac{1}{2} + \frac{9}{15} \times \frac{2}{9}) = 0$

$Gain(S, O)$ and $Gain(S, H)$ are the largest one. Either Outlook or Humidity are best.

(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.

For new data, Outlook=$\{\frac{5}{14}Sunny.\frac{4}{14}Overcast, \frac{5}{14}Rainy\}$

And I'm using ME to calculate information gain.

ME(S)=$\frac{5}{15} = 0.333$

O-Sunny: 5+5/14 of 15, Overcast: 4+4/14 of 15, Rainy: 5+5/14 of 15

$Gain(S, O) = 0.333 - (\frac{5+5/14}{15} \times \frac{2+5/14}{5+5/14} + \frac{4+4/14}{15} \times 0 + \frac{5+5/14}{15} \times \frac{2}{5+5/14}) = 0.043$

T-Hot: 4 of 15, Medium: 7 of 15, Cool: 4 of 15

$Gain(S, T) = 0.333 - (\frac{4}{15} \times \frac{1}{2} + \frac{7}{15} \times \frac{2}{7} + \frac{4}{15} \times \frac{1}{4}) = 0$

H-High: 7 of 15, Normal: 8 of 15, Low: 0 of 14

$Gain(S, H) = 0.333 - (\frac{7}{15} \times \frac{3}{7} + \frac{8}{15} \times \frac{1}{8}) = 0.067$

W-Strong: 6 of 15, Weak: 9 of 15

$Gain(S, W) = 0.333 - (\frac{6}{15} \times \frac{1}{2} + \frac{9}{15} \times \frac{2}{9}) = 0$

$Gain(S, H)$ is the largest one. Humidity is the best.

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

For ID3(S, {Outlook,Temperature,Humidity,Wind}, {Yes, No}):

From (c), we know we can choose Humidity as root node to split.

Then create three branches, which corresponding to label High, Normal and Low, and run ID3 on each branch.

For ID3($S_{H=High}$, {Outlook,Temperature,Wind}, {Yes, No}):

$ME(S_{H=High}) = \frac{3}{7} = 0.429$

O-Sunny: 3 of 7, Overcast: 2 of 7, Rainy: 2 of 7

$Gain(S_{H=High}, O) = 0.429 - (\frac{3}{7} \times 0 + \frac{2}{7} \times 0 + \frac{2}{7} \times \frac{1}{2}) = 0.286$

T-Hot: 3 of 7, Medium: 4 of 7, Cool: 0 of 7

$Gain(S_{H=High}, T) = 0.429 - (\frac{3}{7} \times \frac{1}{3} + \frac{4}{7} \times \frac{1}{2}) = 0$

W-Strong: 4 of 7, Weak: 3 of 7

$Gain(S_{H=High}, W) = 0.429 - (\frac{4}{7} \times \frac{1}{2} + \frac{3}{7} \times \frac{1}{3}) = 0$
Since$Gain(S_{H=High}, O)$ is the largest one, we will choose Outlook as root node to split.

For ID3($S_{H=High,O=Sunny}$, {Temperature,Wind}, {Yes, No}):
all examples have same label, so return a leaf node with label No.
For ID3($S_{H=High,O=Overcast}$, {Temperature,Wind}, {Yes, No}):
all examples have same label, so return a leaf node with label Yes.
For ID3($S_{H=High,O=Rainy}$, {Temperature,Wind}, {Yes, No}):

For ID3($S_{H=High,O=Rainy}$, {Temperature,Wind}, {Yes, No}):
$ME(S_{H=High,O=Rainy}) = \frac{1}{2} = 0.5$
$Gain(S_{H=High,O=Rainy}, T) = 0.5 - 0.5 = 0$ $Gain(S_{H=High,O=Rainy}, W) = 0.5 - 0 = 0.5$ Since $Gain(S_{H=High,O=Rainy}, W)$ is the largest one, I will choose Wind as root node to split.

For ID3($S_{H=High,O=Rainy,W=Strong}$, {Temperature}, {Yes, No}):
all examples have same label, so return a leaf node with label No.
For ID3($S_{H=High,O=Rainy,W=Weak}$, {Temperature}, {Yes, No}):
all examples have same label, so return a leaf node with label Yes.

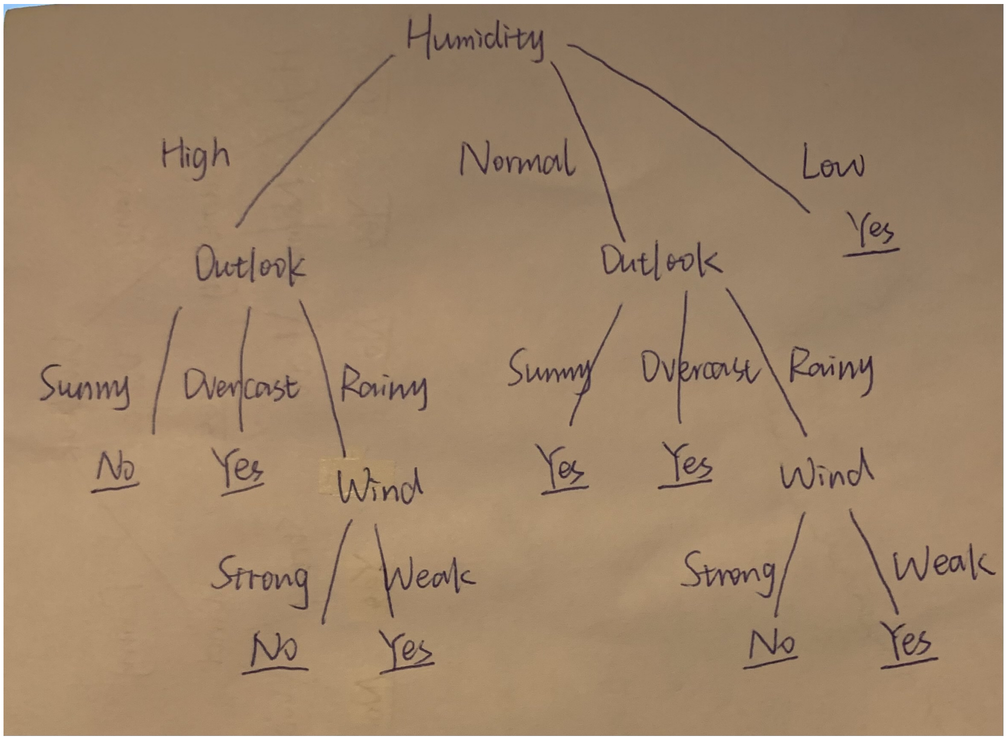
For ID3($S_{H=Normal}$, {Outlook,Temperature,Wind}, {Yes, No}):
$ME(S_{H=Normal}) = \frac{1}{8} = 0.125$
O-Sunny: 2+5/14 of 8, Overcast: 2+4/14 of 8, Rainy: 3+5/14 of 8
$Gain(S, O) = 0.125 - (\frac{2+5/14}{8} \times 0 + \frac{2+4/14}{8} \times 0 + \frac{3+5/14}{8} \times \frac{1}{3+5/14}) = 0$
T-Hot: 1 of 8, Medium: 3 of 8, Cool: 4 of 8
$Gain(S, T) = 0.125 - (\frac{1}{8} \times 0 + \frac{3}{8} \times 0 + \frac{4}{8} \times \frac{1}{4}) = 0$
W-Strong: 3 of 8, Weak: 5 of 8
$Gain(S, W) = 0.125 - (\frac{3}{8} \times \frac{1}{3} + \frac{5}{8} \times 0) = 0$
Since it is a tie for information gain, we will choose Outlook as root node to split.

For ID3($S_{H=Normal,O=Sunny}$, {Temperature,Wind}, {Yes, No}):
all examples have same label, so return a leaf node with label Yes.
For ID3($S_{H=Normal,O=Overcast}$, {Temperature,Wind}, {Yes, No}):
all examples have same label, so return a leaf node with label Yes.
For ID3($S_{H=Normal,O=Rainy}$, {Temperature,Wind}, {Yes, No}):
$ME(S_{H=Normal,O=Rainy}) = \frac{1}{3+5/14} = 0.298$
$Gain(S_{H=Normal,O=Rainy}, T) = 0.298 - 0.298 = 0$ $Gain(S_{H=Normal,O=Rainy}, W) = 0.298 - 0 = 0.298$ Since $Gain(S_{H=High,O=Rainy}, W)$ is the largest one, I will choose Wind as root node to split.

For ID3($S_{H=Normal,O=Rainy,W=Strong}$, {Temperature}, {Yes, No}):
all examples have same label, so return a leaf node with label No.
For ID3($S_{H=Normal,O=Rainy,W=Weak}$, {Temperature}, {Yes, No}):

all examples have same label, so return a leaf node with label Yes.

For ID3($S_{H=Low}$, {Outlook,Temperature,Wind}, {Yes, No}):
attribute set is empty, so return a leaf node with most common label, which is Yes.



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.
https://github.com/Ashley00/CS5350

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.

```
CarTraingPredict:
Max Depth is  1 :
Entropy:  0.30200000000000005  MajorityError:  0.30200000000000005  GiniIndex:  0.30200000000000005
Max Depth is  2 :
Entropy:  0.22199999999999998  MajorityError:  0.30100000000000005  GiniIndex:  0.22199999999999998
Max Depth is  3 :
Entropy:  0.18100000000000005  MajorityError:  0.18899999999999995  GiniIndex:  0.17600000000000005
Max Depth is  4 :
Entropy:  0.08199999999999996  MajorityError:  0.09599999999999997  GiniIndex:  0.08899999999999997
Max Depth is  5 :
Entropy:  0.027000000000000024  MajorityError:  0.028000000000000025  GiniIndex:  0.027000000000000024
Max Depth is  6 :
Entropy:  0.0  MajorityError:  0.0  GiniIndex:  0.0
CarTestingPredict:
Max Depth is  1 :
Entropy:  0.29670329670329665  MajorityError:  0.29670329670329665  GiniIndex:  0.29670329670329665
Max Depth is  2 :
Entropy:  0.22115384615384615  MajorityError:  0.2857142857142857  GiniIndex:  0.22115384615384615
Max Depth is  3 :
Entropy:  0.16620879120879117  MajorityError:  0.19368131868131866  GiniIndex:  0.16620879120879117
Max Depth is  4 :
Entropy:  0.08104395604395609  MajorityError:  0.0892857142857143  GiniIndex:  0.08104395604395609
Max Depth is  5 :
Entropy:  0.019230769230769273  MajorityError:  0.019230769230769273  GiniIndex:  0.019230769230769273
Max Depth is  6 :
Entropy:  0.0  MajorityError:  0.0  GiniIndex:  0.0
```

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

The training error drops more quickly than testing error when depth increases. When depth is the same, training error is slightly larger than testing error. And when the tree depth becomes larger, both training error and testing error will become smaller. Also, the training error and testing error can eventually become zero.

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.

```
3a-BankTraingPredict:
Max Depth is  1 :
Entropy:  0.11919999999999997  MajorityError:  0.10880000000000001  GiniIndex:  0.10880000000000001
Max Depth is  2 :
Entropy:  0.10599999999999998  MajorityError:  0.10419999999999996  GiniIndex:  0.10419999999999996
Max Depth is  3 :
Entropy:  0.10060000000000002  MajorityError:  0.09599999999999997  GiniIndex:  0.09340000000000004
Max Depth is  4 :
Entropy:  0.07920000000000005  MajorityError:  0.0826  GiniIndex:  0.07479999999999998
Max Depth is  5 :
Entropy:  0.06120000000000003  MajorityError:  0.06840000000000002  GiniIndex:  0.059799999999999964
Max Depth is  6 :
Entropy:  0.04720000000000002  MajorityError:  0.05840000000000001  GiniIndex:  0.04679999999999995
Max Depth is  7 :
Entropy:  0.03480000000000005  MajorityError:  0.04820000000000002  GiniIndex:  0.034599999999999964
Max Depth is  8 :
Entropy:  0.02859999999999996  MajorityError:  0.038799999999999946  GiniIndex:  0.026800000000000046
Max Depth is  9 :
Entropy:  0.02300000000000002  MajorityError:  0.03059999999999996  GiniIndex:  0.021199999999999997
Max Depth is  10 :
Entropy:  0.017000000000000015  MajorityError:  0.025399999999999978  GiniIndex:  0.017000000000000015
Max Depth is  11 :
Entropy:  0.014399999999999968  MajorityError:  0.02059999999999995  GiniIndex:  0.014599999999999946
Max Depth is  12 :
Entropy:  0.013599999999999945  MajorityError:  0.017800000000000038  GiniIndex:  0.013800000000000034
Max Depth is  13 :
Entropy:  0.013599999999999945  MajorityError:  0.016000000000000014  GiniIndex:  0.013599999999999945
Max Depth is  14 :
Entropy:  0.013599999999999945  MajorityError:  0.013599999999999945  GiniIndex:  0.013599999999999945
Max Depth is  15 :
Entropy:  0.013599999999999945  MajorityError:  0.013599999999999945  GiniIndex:  0.013599999999999945
Max Depth is  16 :
Entropy:  0.013599999999999945  MajorityError:  0.013599999999999945  GiniIndex:  0.013599999999999945
```

```
3a-BankTestingPredict:
Max Depth is  1 :
Entropy:  0.12480000000000002  MajorityError:  0.11660000000000004  GiniIndex:  0.11660000000000004
Max Depth is  2 :
Entropy:  0.11480000000000001  MajorityError:  0.1078  GiniIndex:  0.1078
Max Depth is  3 :
Entropy:  0.09619999999999995  MajorityError:  0.0928  GiniIndex:  0.09340000000000004
Max Depth is  4 :
Entropy:  0.08020000000000005  MajorityError:  0.07920000000000005  GiniIndex:  0.07679999999999998
Max Depth is  5 :
Entropy:  0.06279999999999997  MajorityError:  0.06779999999999997  GiniIndex:  0.05840000000000001
Max Depth is  6 :
Entropy:  0.04700000000000004  MajorityError:  0.05779999999999996  GiniIndex:  0.04420000000000002
Max Depth is  7 :
Entropy:  0.03300000000000003  MajorityError:  0.04800000000000004  GiniIndex:  0.03259999999999996
Max Depth is  8 :
Entropy:  0.024599999999999955  MajorityError:  0.03859999999999997  GiniIndex:  0.024800000000000044
Max Depth is  9 :
Entropy:  0.018000000000000016  MajorityError:  0.029000000000000026  GiniIndex:  0.018199999999999994
Max Depth is  10 :
Entropy:  0.01319999999999999  MajorityError:  0.02200000000000002  GiniIndex:  0.014000000000000012
Max Depth is  11 :
Entropy:  0.011399999999999966  MajorityError:  0.016599999999999948  GiniIndex:  0.011600000000000055
Max Depth is  12 :
Entropy:  0.011199999999999988  MajorityError:  0.014399999999999968  GiniIndex:  0.011199999999999988
Max Depth is  13 :
Entropy:  0.011199999999999988  MajorityError:  0.012800000000000034  GiniIndex:  0.011199999999999988
Max Depth is  14 :
Entropy:  0.011199999999999988  MajorityError:  0.011199999999999988  GiniIndex:  0.011199999999999988
Max Depth is  15 :
Entropy:  0.011199999999999988  MajorityError:  0.011199999999999988  GiniIndex:  0.011199999999999988
Max Depth is  16 :
Entropy:  0.011199999999999988  MajorityError:  0.011199999999999988  GiniIndex:  0.011199999999999988
```

(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.

```
3b-BankTraingPredict:
Max Depth is  1 :
Entropy:  0.11919999999999997  MajorityError:  0.10880000000000001  GiniIndex:  0.10880000000000001
Max Depth is  2 :
Entropy:  0.10599999999999998  MajorityError:  0.10499999999999998  GiniIndex:  0.10519999999999996
Max Depth is  3 :
Entropy:  0.10219999999999996  MajorityError:  0.09760000000000002  GiniIndex:  0.10099999999999998
Max Depth is  4 :
Entropy:  0.08679999999999999  MajorityError:  0.08640000000000003  GiniIndex:  0.08760000000000001
Max Depth is  5 :
Entropy:  0.07140000000000002  MajorityError:  0.07720000000000005  GiniIndex:  0.07379999999999998
Max Depth is  6 :
Entropy:  0.05679999999999996  MajorityError:  0.06720000000000004  GiniIndex:  0.05720000000000003
Max Depth is  7 :
Entropy:  0.04520000000000002  MajorityError:  0.05900000000000005  GiniIndex:  0.04500000000000004
Max Depth is  8 :
Entropy:  0.03859999999999997  MajorityError:  0.052200000000000024  GiniIndex:  0.036800000000000055
Max Depth is  9 :
Entropy:  0.03200000000000003  MajorityError:  0.043200000000000016  GiniIndex:  0.02939999999999998
Max Depth is  10 :
Entropy:  0.02639999999999998  MajorityError:  0.03620000000000001  GiniIndex:  0.024800000000000044
Max Depth is  11 :
Entropy:  0.023399999999999976  MajorityError:  0.029200000000000004  GiniIndex:  0.022399999999999975
Max Depth is  12 :
Entropy:  0.022199999999999998  MajorityError:  0.02639999999999998  GiniIndex:  0.02200000000000002
Max Depth is  13 :
Entropy:  0.02200000000000002  MajorityError:  0.024800000000000044  GiniIndex:  0.02200000000000002
Max Depth is  14 :
Entropy:  0.02200000000000002  MajorityError:  0.02200000000000002  GiniIndex:  0.02200000000000002
Max Depth is  15 :
Entropy:  0.02200000000000002  MajorityError:  0.02200000000000002  GiniIndex:  0.02200000000000002
Max Depth is  16 :
Entropy:  0.02200000000000002  MajorityError:  0.02200000000000002  GiniIndex:  0.02200000000000002
```

```
3b-BankTestingPredict:
Max Depth is  1 :
Entropy:  0.12480000000000002  MajorityError:  0.11660000000000004  GiniIndex:  0.11660000000000004
Max Depth is  2 :
Entropy:  0.11480000000000001  MajorityError:  0.10799999999999998  GiniIndex:  0.10799999999999998
Max Depth is  3 :
Entropy:  0.09619999999999995  MajorityError:  0.09819999999999995  GiniIndex:  0.09919999999999995
Max Depth is  4 :
Entropy:  0.08320000000000005  MajorityError:  0.08819999999999995  GiniIndex:  0.08399999999999996
Max Depth is  5 :
Entropy:  0.06940000000000002  MajorityError:  0.07520000000000004  GiniIndex:  0.06999999999999995
Max Depth is  6 :
Entropy:  0.05479999999999996  MajorityError:  0.06740000000000002  GiniIndex:  0.05820000000000003
Max Depth is  7 :
Entropy:  0.04259999999999997  MajorityError:  0.05940000000000001  GiniIndex:  0.04479999999999995
Max Depth is  8 :
Entropy:  0.03359999999999996  MajorityError:  0.050000000000000044  GiniIndex:  0.03639999999999999
Max Depth is  9 :
Entropy:  0.027599999999999958  MajorityError:  0.04139999999999999  GiniIndex:  0.02939999999999998
Max Depth is  10 :
Entropy:  0.0232  MajorityError:  0.03359999999999996  GiniIndex:  0.023599999999999954
Max Depth is  11 :
Entropy:  0.020199999999999996  MajorityError:  0.02739999999999998  GiniIndex:  0.020000000000000018
Max Depth is  12 :
Entropy:  0.019199999999999995  MajorityError:  0.024599999999999955  GiniIndex:  0.01859999999999995
Max Depth is  13 :
Entropy:  0.01859999999999995  MajorityError:  0.02100000000000002  GiniIndex:  0.01859999999999995
Max Depth is  14 :
Entropy:  0.01859999999999995  MajorityError:  0.01859999999999995  GiniIndex:  0.01859999999999995
Max Depth is  15 :
Entropy:  0.01859999999999995  MajorityError:  0.01859999999999995  GiniIndex:  0.01859999999999995
Max Depth is  16 :
Entropy:  0.01859999999999995  MajorityError:  0.01859999999999995  GiniIndex:  0.01859999999999995
```

(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?

The training error is slightly larger than testing error, but the difference is subtle. When depth becomes larger, both training error and testing error will become smaller.

Although we use different ways to handle the unknown value, there is not too much difference regarding the error. The performance for treating "unknown" as a particular attribute value is slightly better than replacing it with majority value.