CS 5350/6350: Machine Learining Fall 2021

Homework 1

Handed out: 7 Sep, 2021
Due date: 11:59pm, 24 Sep, 2021

- You are welcome to talk to other members of the class about the homework. I am more concerned that you understand the underlying concepts. However, you should write down your own solution. Please keep the class collaboration policy in mind.
- Feel free discuss the homework with the instructor or the TAs.
- Your written solutions should be brief and clear. You need to show your work, not just the final answer, but you do *not* need to write it in gory detail. Your assignment should be **no more than 20 pages**. Not that you do not need to include the problem description. Every extra page will cost a point.
- Handwritten solutions will not be accepted.
- *Your code should run on the CADE machines.* You should include a shell script, `run.sh`, that will execute your code in the CADE environment. Your code should produce similar output to what you include in your report.

  You are responsible for ensuring that the grader can execute the code using only the included script. If you are using an esoteric programming language, you should make sure that its runtime is available on CADE.
- Please do not hand in binary files! We will *not* grade binary submissions.
- The homework is due by **midnight of the due date**. Please submit the homework on Canvas.
- Note the bonus questions are for **both 5350 and 6350** students. If a question is mandatory for 6350, we will highlight it explicitly.

1 Decision Tree [40 points + 10 bonus]

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.

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

First Split Decision

Current entropy: $p = 2/7, n = 5/7$

$H(Y) = -2/7log_2(2/7) - 5/7log_2(5/7) = \mathbf{0.863}$

$x_1 = T$ $p = 1/2, n = 1/2$ $H(x_1 = T) = 1.0$

$x_1 = F$ $p = 1/5, n = 4/5$ $H(x_1 = F) = 0.722$

Expected entropy: $2/7 * 1 + 5/7 * 0.722 = 0.801$

Information gain x_1: $0.863 - 0.801 = 0.062$

$x_2 = T$ $p = 0/4, n = 4/4$ $H(x_2 = T) = 0.0$

$x_2 = F$ $p = 2/3, n = 1/3$ $H(x_2 = F) = 0.918$

Expected entropy: $4/7 * 0 + 3/7 * 0.918 = 0.393$

Information gain x_2: $0.863 - 0.393 = 0.47$

$x_3 = T$ $p = 1/3, n = 2/3$ $H(x_3 = T) = 0.918$

$x_3 = F$ $p = 1/4, n = 3/4$ $H(x_3 = F) = 0.811$

Expected entropy: $3/7 * 0.918 + 4/7 * 0.811 = 0.3857$

Information gain x_3: $0.863 - 0.857 = 0.006$

$x_4 = T$ $p = 2/3, n = 1/3$ $H(x_4 = T) = 0.918$

$x_4 = F$ $p = 0/4, n = 4/4$ $H(x_4 = F) = 0.0$

Expected entropy: $3/7 * 0.918 + 4/7 * 0 = 0.393$

Information gain x_4: $0.863 - 0.393 = 0.47$

From this we can decide to either split on x_2 or x_4 since they have the same level of information gain both being the max. Let us decide to split on x_2. Looking at splitting on x_2 we can see that for all $x_2 = 1$ y resolves to 1. Thus we have come to a leaf node for that branch. Let us look at what happens when $x_2 = 0$. This can be seen in table 2. Looking at this we can see that x_4 will be a perfect split but let us calculate the information gain in order to show this.

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

Table 2: Result of splitting on $x_2 = 0$

Split $x_2 = 0$ Decision

Current entropy: $p = 2/3, n = 1/3$

$$H(Y) = -2/3log_2(2/3) - 1/3log_2(1/3) = \mathbf{0.918}$$

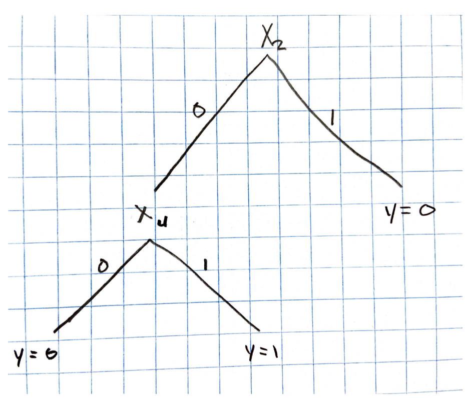
$$x_4 = T \ p = 2/2, n = 0/2 \ H(x_4 = T) = 0.0$$

$$x_4 = F \ p = 0/1, n = 1/1 \ H(x_4 = F) = 0.0$$

Expected entropy: $2/3 * 0.0 + 1/3 * 0.0 = 0.0$

Information gain x_4: $0.918 - 0.0 = 0.918$

Thus because we have the maximum amount of information gain we know this will be the ideal split and result in two leaf nodes. Finally we can construct the tree.



(b) [2 points]

The boolean function that which my decision tree represents is $\neg x_2 \land x_4$. This can be seen in table 3. ‘

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

Table 3: Boolean function for decision tree

2. [17 points] framework.

   (a) [7 points]

First Split Decision

Current Majority Error: $ME(Play?) = 5/14 = 0.357$

$$ME(\text{outlook} = \text{sunny}) = 5/14 * 2/5 = 2/14 = 0.143$$

$$ME(\text{outlook} = \text{overcast}) = 0.0$$

$$ME(\text{outlook} = \text{rainy}) = 5/14 * 2/5 = 2/14 = 0.143$$

$$ME(\text{outlook})) = 0.286$$

Outlook Gain: $0.357 - 0.286 = 0.071$

$$ME(\text{humidity} = \text{high}) = 7/14 * 3/7 = 3/14 = 0.214$$

$$ME(\text{humidity} = \text{normal}) = 7/14 * 1/7 = 1/14 = 0.071$$

$$ME(\text{humidity})) = 0.286$$

Humidity Gain: $0.357 - 0.286 = 0.071$

$$ME(\text{temperature} = \text{hot}) = 4/14 * 2/4 = 2/14 = 0.143$$

$$ME(\text{temperature} = \text{med}) = 6/14 * 2/6 = 2/14 = 0.143$$

$$ME(\text{temperature} = \text{cold}) = 4/14 * 1/4 = 1/14 = 0.071$$

$$ME(\text{temperature})) = 0.357$$

$$\text{Temperature Gain: } 0.357 - 0.357 = 0.0$$

$$ME(\text{wind} = \text{W}) = 8/14 * 2/8 = 2/14 = 0.143$$

$$ME(\text{wind} = \text{S}) = 6/14 * 3/6 = 3/14 = 0.214$$

$$ME(\text{wind})) = 0.357$$

$$\text{Wind Gain: } 0.357 - 0.357 = 0.0$$

From this we can decide to either split on outlook or humidity since they have the same level of gain both being the max. Let us decide to split on outlook. Looking at splitting on out we can see that for all outlook = overcast resolves to +. Thus we have come to a leaf node for that branch. Let us look at what happens when outlook = sunny. This can be seen in *figure 1*. Looking at this we can see that humidity will be a perfect split but let us calculate the gain in order to show this.

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
8	S	M	H	W	-
9	S	C	N	W	+
11	S	M	N	S	+

Figure 1: Result of splitting on outlook=sunny

Split outlook = sunny Decision

$$\text{Current Majority Error: } ME(Play?|outlook = sunny) = 2/5 = 0.4$$

$$ME(\text{humidity} = \text{high}) = 0.0$$

$$ME(\text{humidity} = \text{normal}) = 0.0$$

$$ME(\text{humidity})) = 0.0$$

$$\text{Humidity Gain: } 0.4 - 0.0 = 0.4$$

Thus because we have the maximum amount of gain we know this will be the ideal split and result in three leaf nodes with high going to **-** , normal going to +, and low going to **-**. Now let us look at what happens when outlook = rainy.

	O	T	H	W	Play?
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
10	R	M	N	W	+
14	R	M	H	S	-

Figure 2: Result of splitting on outlook=rainy

This can be seen in *figure 2*. Looking at this we can see that humidity is a perfect split but let us calculate the gain in order to show this.

Split outlook = rainy Decision

Current Majority Error: $ME(Play?|outlook = rainy) = 2/5 = 0.4$

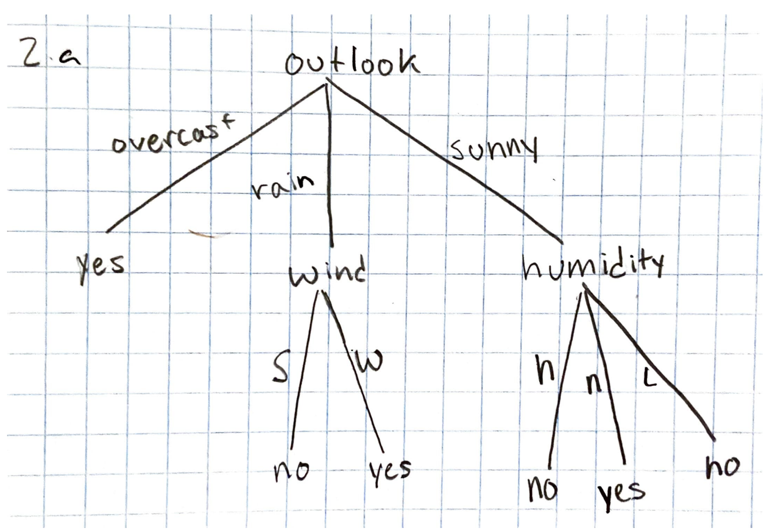
$$ME(\text{wind = W}) = 0.0$$

$$ME(\text{wind = S}) = 0.0$$

$$ME(\text{wind})) = 0.0$$

$$\text{Wind Gain: } 0.357 - 0.0 = 0.357$$

Thus because we have the maximum amount of gain we know this will be the ideal split and result in two leaf nodes with strong winds going to **-** and weak winds going to $+$. Finally we may construct the decision tree.



(b) [7 points]

First Split Decision

$$\text{Current Gini Index: } GI(Play?) = 1 - (5/14)^2 - (9/14)^2 = 0.459$$

$$GI(\text{outlook} = \text{sunny}) = (1 - (2/5)^2 - (3/5)^2) * 5/14 = 0.171$$
$$GI(\text{outlook} = \text{overcast}) = 0.0$$
$$GI(\text{outlook} = \text{rainy}) = (1 - (3/5)^2 - (2/5)^2) * 5/14 = 0.171$$
$$GI(\text{outlook})) = 0.343$$
$$\text{Outlook Gain: } 0.459 - 0.343 = 0.116$$

$$GI(\text{humidity} = \text{high}) = (1 - (3/7)^2 - (4/7)^2) * 7/14 = 0.245$$
$$GI(\text{humidity} = \text{normal}) = (1 - (6/7)^2 - (1/7)^2) * 7/14 = 0.122$$
$$GI(\text{humidity})) = 0.286$$
$$\text{Humidity Gain: } 0.459 - 0.367 = 0.092$$

$$GI(\text{temperature} = \text{hot}) = (1 - (2/4)^2 - (2/4)^2) * 4/14 = 0.143$$
$$GI(\text{temperature} = \text{med}) = (1 - (2/6)^2 - (4/6)^2) * 6/14 = 0.19$$
$$GI(\text{temperature} = \text{cold}) = (1 - (1/4)^2 - (3/4)^2) * 4/14 = 0.107$$
$$GI(\text{temperature})) = 0.44$$
$$\text{Temperature Gain: } 0.469 - 0.44 = 0.019$$

$$GI(\text{wind} = \text{W}) = (1 - (2/8)^2 - (6/8)^2) * 8/14 = 0.214$$
$$GI(\text{wind} = \text{S}) = (1 - (3/6)^2 - (3/6)^2) * 6/14 = 0.214$$
$$GI(\text{wind})) = 0.428$$
$$\text{Wind Gain: } 0.459 - 0.428 = 0.031$$

From this we must split on outlook due to it having the most gain.Looking at splitting on out we can see that for all outlook = overcast resolves to +. Thus we have come to a leaf node for that branch. Let us look at what happens when outlook = sunny. This can be seen in *figure 1*. Looking at this we can see that humidity will be a perfect split but let us calculate the gain in order to show this.

Split outlook = sunny Decision

$$\text{Current Majority Error: } GI(Play?|outlook = sunny) = 0.48$$

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
8	S	M	H	W	-
9	S	C	N	W	+
11	S	M	N	S	+

Figure 3: Result of splitting on outlook=sunny

$$GI(\text{humidity} = \text{high}) = 0.0$$

$$GI(\text{humidity} = \text{normal}) = 0.0$$

$$GI(\text{humidity})) = 0.0$$

$$\text{Humidity Gain: } 0.48 - 0.0 = 0.48$$

Thus because we have the maximum amount of gain we know this will be the ideal split and result in three leaf nodes with high going to **-** , normal going to +, and low going to **-**. Now let us look at what happens when outlook = rainy. This can be seen in *figure 2*. Looking at this we can see that wind is a perfect split but let us calculate the gain in order to show this.

	O	T	H	W	Play?
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
10	R	M	N	W	+
14	R	M	H	S	-

Figure 4: Result of splitting on outlook=rainy

Split outlook = rainy Decision

Current Majority Error: $GI(Play?|outlook = rainy) = 0.48$

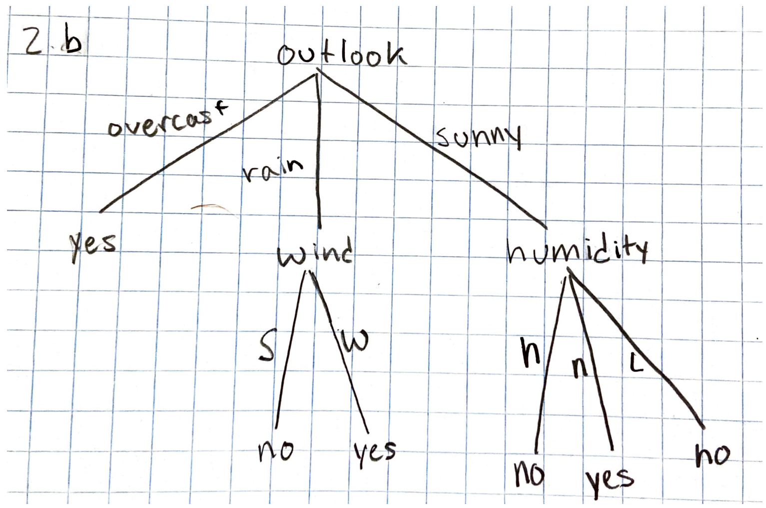
$$GI(\text{wind} = \text{W}) = 0.0$$

$$GI(\text{wind} = \text{S}) = 0.0$$

$$GI(\text{wind})) = 0.0$$

$$\text{Wind Gain: } 0.48 - 0.0 = 0.48$$

Thus because we have the maximum amount of gain we know this will be the ideal split and result in two leaf nodes with strong winds going to - and weak winds going to +. Finally we may construct the decision 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?

The trees constructed in both part a and b are identical to the one created in class. For part a the tree could've been different as the gain for outlook and humidity were the same, but due to choosing to split on outlook it resulted in the same tree as in class. I believe that all the trees are the same because due to it being such a small data set it was very likely for the attribute with the most information gain would also have a high gain when using ME and GI.

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]

There are an equal amount of rainy and sunny thus we decided to consider the missing value as rainy.

$$\text{Current Entropy: } H(Play?) = 0.918$$

$$H(\text{outlook} = \text{sunny}) = 0.971$$

$$H(\text{outlook} = \text{overcast}) = 0.0$$

$$H(\text{outlook} = \text{rainy}) = 0.918$$

$$H(\text{outlook})) = 5/15 * 0.971 + 4/15 * 0 + 6/15 * 0.918 = 0.691$$

$$\text{Outlook Gain: } 0.918 - 0.691 = 0.227$$

$$H(\text{humidity} = \text{high}) = 0.985$$
$$H(\text{humidity} = \text{normal}) = 0.544$$
$$H(\text{humidity})) = 7/15 * 0.985 + 8/15 * 0.544 = 0.75$$
$$\text{Humidity Gain: } 0.918 - 0.75 = 0.168$$

$$H(\text{temperature} = \text{hot}) = 1.0$$
$$H(\text{temperature} = \text{med}) = 0.485$$
$$H(\text{temperature} = \text{cold}) = 0.811$$
$$H(\text{temperature})) = 4/15 * 1.0 + 7/15 * 0.485 + 4/15 * 0.811 = 0.709$$
$$\text{Temperature Gain: } 0.918 - 0.709 = 0.209$$

$$H(\text{wind} = \text{W}) = 0.764$$
$$H(\text{wind} = \text{S}) = 1.0$$
$$H(\text{wind})) = 9/15 * 0.764 + 6/15 * 1.0 = 0.858$$
$$\text{Wind Gain: } 0.918 - 0.858 = 0.06$$

Thus the best feature is outlook having the most information gain at 0.227.

(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.

We will consider the missing value as overcast, this is because all values with outlook go to a positive label. Due to this change not affecting any of the other info gain values, I will not be re-calculating these and using the information from the previous answer.

$$\text{Current Entropy: } H(Play?) = 0.918$$

$$H(\text{outlook} = \text{sunny}) = 0.971$$
$$H(\text{outlook} = \text{overcast}) = 0.0$$
$$H(\text{outlook} = \text{rainy}) = 0.971$$
$$H(\text{outlook})) = 5/15 * 0.971 + 5/15 * 0 + 5/15 * 0.971 = 0.647$$
$$\text{Outlook Gain: } 0.918 - 0.647 = 0.271$$

Thus the best feature is outlook having the most information gain at 0.271.

(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.

Due to this change not affecting any of the other info gain values, I will not be re-calculating these and using the information from the previous answer.

$$\text{Current Entropy: } H(Play?) = 0.918$$

$$H(\text{outlook} = \text{sunny}) = -(\frac{2+5/14}{5+5/14}) \log_2(\frac{2+5/14}{5+5/14}) - (\frac{3}{5+5/14}) \log_2(\frac{3}{5+5/14}) = 0.99$$

$$H(\text{outlook} = \text{overcast}) = 0.0$$

$$H(\text{outlook} = \text{rainy}) = -(\frac{2+5/14}{5+5/14}) \log_2(\frac{2+5/14}{5+5/14}) - (\frac{3}{5+5/14}) \log_2(\frac{3}{5+5/14}) = 0.99$$

$$H(\text{outlook})) = \frac{5+5/14}{15} * 0.99 + \frac{4+4/14}{15} * 0 + \frac{5+5/14}{15} * 0.99 = 0.707$$

$$\text{Outlook Gain: } 0.918 - 0.707 = 0.211$$

Thus the best feature is outlook having the most information gain at 0.211.

(d) [7 points] Continue with the fractional examples, and build the whole free with information gain. List every step and the final tree structure. From the previous problem we must split on outlook, due to it having the max information gain. Looking at splitting on out we can see that for all outlook = overcast resolves to +. Thus we have come to a leaf node for that branch. Let us look at what happens when outlook = sunny. This can be seen in *figure 3*. Looking at this we can see that humidity will be a perfect split but let us calculate the gain in order to show this.

	O	T	H	W	Play?	
1	S	H	H	W	-	
2	S	H	H	S	-	
8	S	M	H	W	-	
9	S	C	N	W	+	
11	S	M	N	S	+	
15	S	M	N	W	+	* 5/14

Figure 5: Result of splitting on outlook=sunny

Split outlook = sunny Decision

$$\text{Current Entropy: } H(Play?|outlook = sunny) = 0.99$$

$$H(\text{humidity = high}) = -1 * log_2(1) - 0 * log2(0) = 0.0$$

$$H(\text{humidity = normal}) = -1 * \log_2(1) - 0 * \log_2(0) = 0.0$$

$$H(\text{humidity})) = \frac{3}{5 + 5/14} * 0 + \frac{2 + 5/14}{5 + 5/14} * 0 = 0.0$$

$$\text{Humidity Gain: } 0.99 - 0.0 = 0.99$$

Thus because we have the maximum amount of gain we know this will be the ideal split and result in three leaf nodes with high going to **-** , normal going to $+$, and low going to **-**. Now let us look at what happens when outlook = rainy. This can be seen in *figure 2*. Looking at this we can see that wind is a perfect split but let us calculate the gain in order to show this.

	O	T	H	W	Play?
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
10	R	M	N	W	+
14	R	M	H	S	-
15	R	M	N	W	+ *5/14

Figure 6: Result of splitting on outlook=rainy

Split outlook = rainy Decision

Current Entropy: $H(Play?|outlook = rainy) = 0.99$

$$H(\text{wind = W}) = -1 * log_2(1) - 0 * log2(0) = 0.0$$

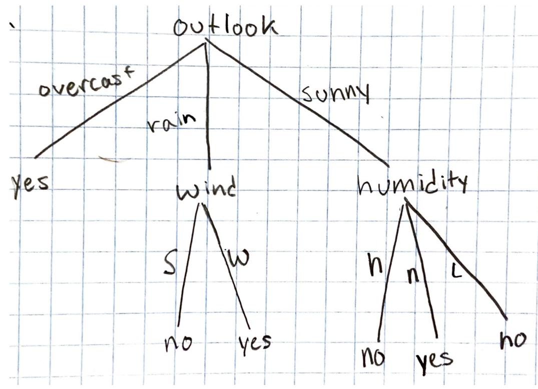
$$H(\text{wind = S}) = -1 * \log_2(1) - 0 * \log_2(0) = 0.0$$

$$H(\text{wind})) = \frac{2}{5 + 5/14} * 0 + \frac{3 + 5/14}{5 + 5/14} * 0 = 0.0$$

$$\text{Wind Gain: } 0.99 - 0.0 = 0.99$$

Thus because we have the maximum amount of gain we know this will be the ideal split and result in two leaf nodes with strong winds going to **-** and weak winds going to $+$. Finally we may construct the decision 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)



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?

   In order to construct a regression tree we would use the variance as the way to select the best attribute to split on. This will give use values that are closer together instead of simply just taking the mean.

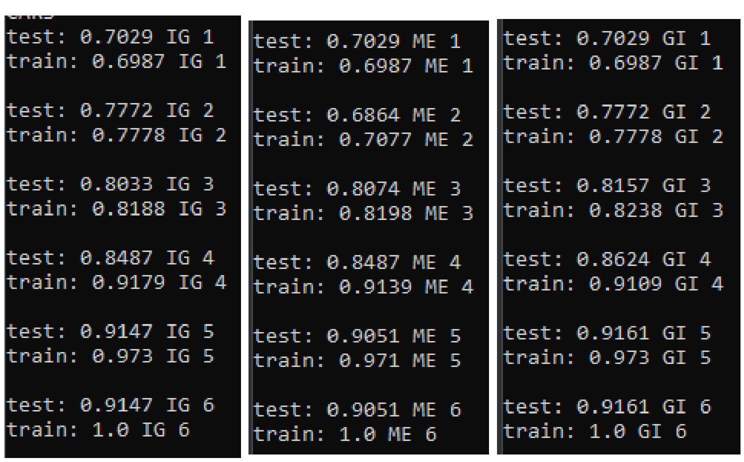
2 Decision Tree Practice [60 points]

1. [5 Points] `https://github.com/ericodonoghue/machine_learning_uofu`

2. [30 points]

   (a) [15 points]

   (b) [10 points]



```
test: 0.7029 IG 1
train: 0.6987 IG 1

test: 0.7772 IG 2
train: 0.7778 IG 2

test: 0.8033 IG 3
train: 0.8188 IG 3

test: 0.8487 IG 4
train: 0.9179 IG 4

test: 0.9147 IG 5
train: 0.973 IG 5

test: 0.9147 IG 6
train: 1.0 IG 6
```

```
test: 0.7029 ME 1
train: 0.6987 ME 1

test: 0.6864 ME 2
train: 0.7077 ME 2

test: 0.8074 ME 3
train: 0.8198 ME 3

test: 0.8487 ME 4
train: 0.9139 ME 4

test: 0.9051 ME 5
train: 0.971 ME 5

test: 0.9051 ME 6
train: 1.0 ME 6
```

```
test: 0.7029 GI 1
train: 0.6987 GI 1

test: 0.7772 GI 2
train: 0.7778 GI 2

test: 0.8157 GI 3
train: 0.8238 GI 3

test: 0.8624 GI 4
train: 0.9109 GI 4

test: 0.9161 GI 5
train: 0.973 GI 5

test: 0.9161 GI 6
train: 1.0 GI 6
```

Figure 7: 2.2.b

(c) [5 points]

Through comparing our training errors and test errors we can see that for each heuristic at a certain depth the test prediction will stop improving while the train prediction will keep improving. This is because building decision trees with id3 will cause over fitting on the train data when the depth isn't limited. Thus in order to get the best prediction for any testing we must limit the depth.

3. [25 points]

(a) [10 points]

```
test: 0.8752 IG 1
train: 0.8808 IG 1

test: 0.8886 IG 2
train: 0.894 IG 2

test: 0.893 IG 3
train: 0.8994 IG 3

test: 0.8824 IG 4
train: 0.9164 IG 4

test: 0.8704 IG 5
train: 0.9302 IG 5

test: 0.8612 IG 6
train: 0.9404 IG 6

test: 0.855 IG 7
train: 0.9468 IG 7

test: 0.8482 IG 8
train: 0.9496 IG 8

test: 0.8416 IG 9
train: 0.95 IG 9

test: 0.837 IG 10
train: 0.9508 IG 10

test: 0.8342 IG 11
train: 0.9538 IG 11

test: 0.834 IG 12
train: 0.9534 IG 12

test: 0.8324 IG 13
train: 0.9544 IG 13

test: 0.8324 IG 14
train: 0.9544 IG 14

test: 0.8324 IG 15
train: 0.9544 IG 15

test: 0.8324 IG 16
train: 0.9544 IG 16
```

```
test: 0.8834 ME 1
train: 0.8912 ME 1

test: 0.8866 ME 2
train: 0.8934 ME 2

test: 0.884 ME 3
train: 0.8992 ME 3

test: 0.8764 ME 4
train: 0.9078 ME 4

test: 0.8704 ME 5
train: 0.92 ME 5

test: 0.859 ME 6
train: 0.9302 ME 6

test: 0.8526 ME 7
train: 0.9372 ME 7

test: 0.8478 ME 8
train: 0.9402 ME 8

test: 0.8438 ME 9
train: 0.9424 ME 9

test: 0.8406 ME 10
train: 0.9446 ME 10

test: 0.8382 ME 11
train: 0.9458 ME 11

test: 0.8356 ME 12
train: 0.9468 ME 12

test: 0.8318 ME 13
train: 0.9488 ME 13

test: 0.8294 ME 14
train: 0.9496 ME 14

test: 0.8294 ME 15
train: 0.9496 ME 15

test: 0.8294 ME 16
train: 0.9496 ME 16
```

```
test: 0.8834 GI 1
train: 0.8912 GI 1

test: 0.8858 GI 2
train: 0.893 GI 2

test: 0.877 GI 3
train: 0.9052 GI 3

test: 0.8738 GI 4
train: 0.9186 GI 4

test: 0.8636 GI 5
train: 0.9266 GI 5

test: 0.8526 GI 6
train: 0.9352 GI 6

test: 0.8444 GI 7
train: 0.9436 GI 7

test: 0.8388 GI 8
train: 0.9474 GI 8

test: 0.837 GI 9
train: 0.9486 GI 9

test: 0.8308 GI 10
train: 0.949 GI 10

test: 0.8274 GI 11
train: 0.95 GI 11

test: 0.8258 GI 12
train: 0.9504 GI 12

test: 0.825 GI 13
train: 0.9504 GI 13

test: 0.825 GI 14
train: 0.9502 GI 14

test: 0.825 GI 15
train: 0.9502 GI 15

test: 0.825 GI 16
train: 0.9502 GI 16
```

Figure 8: 2.3.a

(b) [10 points]

```
test: 0.8752 IG 1
train: 0.8808 IG 1

test: 0.8886 IG 2
train: 0.894 IG 2

test: 0.8874 IG 3
train: 0.8978 IG 3

test: 0.8502 IG 4
train: 0.8796 IG 4

test: 0.8514 IG 5
train: 0.8858 IG 5

test: 0.848 IG 6
train: 0.8942 IG 6

test: 0.8452 IG 7
train: 0.8824 IG 7

test: 0.8424 IG 8
train: 0.8872 IG 8

test: 0.8424 IG 9
train: 0.8898 IG 9

test: 0.8418 IG 10
train: 0.889 IG 10

test: 0.8422 IG 11
train: 0.8906 IG 11

test: 0.8422 IG 12
train: 0.8906 IG 12

test: 0.8422 IG 13
train: 0.8908 IG 13

test: 0.8422 IG 14
train: 0.8908 IG 14

test: 0.8422 IG 15
train: 0.8908 IG 15

test: 0.8422 IG 16
train: 0.8908 IG 16
```

```
test: 0.8834 ME 1
train: 0.8912 ME 1

test: 0.8848 ME 2
train: 0.8926 ME 2

test: 0.8826 ME 3
train: 0.8994 ME 3

test: 0.8792 ME 4
train: 0.9004 ME 4

test: 0.8748 ME 5
train: 0.885 ME 5

test: 0.8758 ME 6
train: 0.8684 ME 6

test: 0.8728 ME 7
train: 0.8616 ME 7

test: 0.8726 ME 8
train: 0.864 ME 8

test: 0.872 ME 9
train: 0.866 ME 9

test: 0.8706 ME 10
train: 0.866 ME 10

test: 0.8698 ME 11
train: 0.8674 ME 11

test: 0.8698 ME 12
train: 0.8678 ME 12

test: 0.8696 ME 13
train: 0.8692 ME 13

test: 0.8692 ME 14
train: 0.87 ME 14

test: 0.8692 ME 15
train: 0.87 ME 15

test: 0.8692 ME 16
train: 0.87 ME 16
```

```
test: 0.8834 GI 1
train: 0.8912 GI 1

test: 0.8842 GI 2
train: 0.892 GI 2

test: 0.8794 GI 3
train: 0.8992 GI 3

test: 0.8788 GI 4
train: 0.9084 GI 4

test: 0.8762 GI 5
train: 0.8886 GI 5

test: 0.8734 GI 6
train: 0.896 GI 6

test: 0.871 GI 7
train: 0.8784 GI 7

test: 0.8678 GI 8
train: 0.8802 GI 8

test: 0.8672 GI 9
train: 0.8786 GI 9

test: 0.8672 GI 10
train: 0.8802 GI 10

test: 0.8674 GI 11
train: 0.8806 GI 11

test: 0.8674 GI 12
train: 0.8814 GI 12

test: 0.8674 GI 13
train: 0.8814 GI 13

test: 0.8674 GI 14
train: 0.8814 GI 14

test: 0.8674 GI 15
train: 0.8814 GI 15

test: 0.8674 GI 16
train: 0.8814 GI 16
```

Figure 9: 2.3.b

(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?

Looking at this we can conclude that after a certain depth the decision tree begins to overfit to the train data and causes a decrease in prediction accuracy on the test data. We can see this happen with every heuristic where at first both train and test predictions are increasing then after a certain depth the test accuracy begins to decrease. When dealing with unknown values we can see that simply treating it as a value the we get better results for train and worse results for test as opposed to filling it in with the most common value. Thus we can come to the conclusion that filling in unknown improves our test accuracy and using a even better calculation (fractional counts) we could further increase our test prediction accuracy.