Decision Tree Help Slides

CS0160 2021

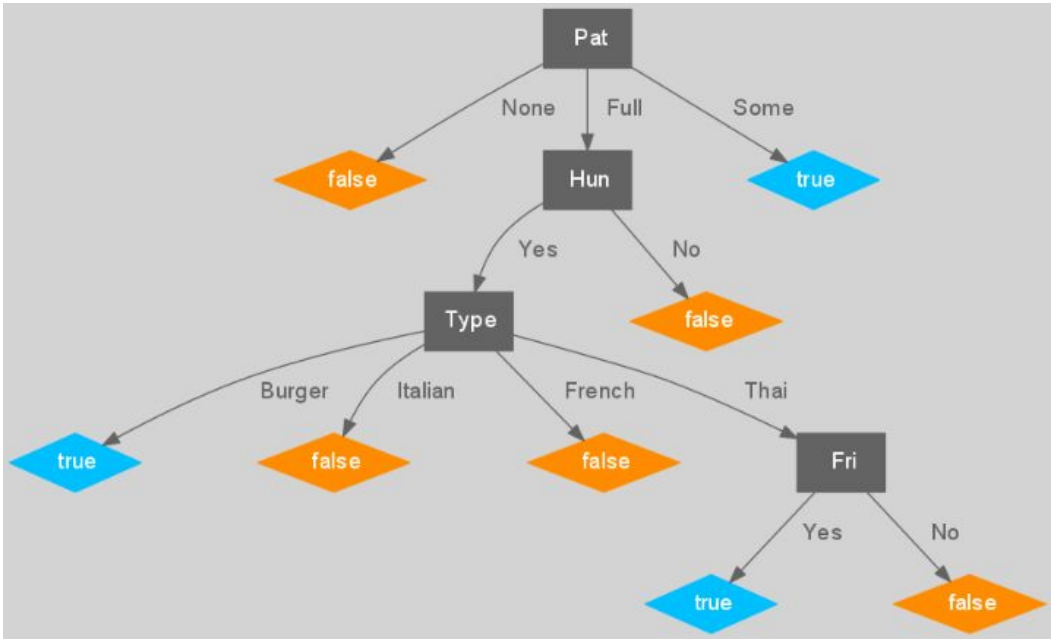
Tree w/ Labels

Nodes:

- Internal: Attribute names
- External: Classification

Branches:

- Values of parent node attribute
- Connect to a node that is created recursively



Data

Ex.	Input Attributes										Classif.
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	
1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	Yes
5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	Yes
7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	No
8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	Yes
9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	No
11	No	No	No	No	None	$	No	No	Thai	0-10	No
12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes

Example 12's attribute values

Example 12's classification

Finding Entropy

In order to find entropy, you might want to think about creating a function that takes care of the math! Make sure to look in the handout for the equation.

Remainder Pseudocode

```
function remainder(data, attr):
    reset remainder
    for each value of attr:
        find number of positive and negative ex that map to
        value
        if pos ex plus neg ex is 0:
            continue
        find the proportion of the sum of pos and neg ex to the
        total number of examples
        increment remainder by the product of the proportion
        calculation and the entropy of positive and negative
        examples
    return remainder
```

Look at section 7.1 of the handout for an example calculation!

Tips and Reminders: Debugging

- The easiest way to print an array in Java is to import java.util.Arrays and use `Arrays.deepToString(yourArray)` to print it
- If you get to the point where your program is producing a tree, but it looks wrong/different from the demo, first check for the common errors listed in section 9 (Important Notes and Reminders) of the handout

Tips and Reminders

- All logarithms in this algorithm are base 2
- Remember that the algorithm treats $\log_2(0)$ as 0, **not** -∞
- Do not compare strings in Java with “==”. Compare with “.equals”:
  - **No:** st1 == str2
  - **Yes:** str1.equals(str2)
  - **Why:** using “==” will not cause a compiler error or an exception, but it will almost always evaluate to false. This is because “==” checks that two Strings are the same instance of String, while “.equals” checks if they have the same sequence of characters.
  - String s1 = “tree”; String s2 = “tree”; s1 == s2 -> will be false! s1.equals(s2) -> will be true!

Tips and Reminders

- Floating point arithmetic
  - When an int is divided by an int in java, the result is “floored” to the closest integer. In other words, the remainder is truncated.
    - `5 / 2 → 2`
    - `int / int → int`
  - Casting this result to a double or float will not recover this remainder:
    - `(double) (5 / 2) → 2.0`
  - To return a double or float with the remainder preserved, at least one of the inputs must be a float or double:
    - `((double) 5) / 2 → 2.5`

Tips and Reminders

- It is not necessary to have anything in your MyID3 constructor. Your trigger method is where you need to run your algorithm.
- **Carefully read** the javadocs for which methods are provided to avoid doing extra work!
- Remember that this pseudocode describes the algorithm, not an implementation. Your code might not (and very likely will not) match this pseudocode or the lecture pseudocode precisely.

What is DecisionTreeData?

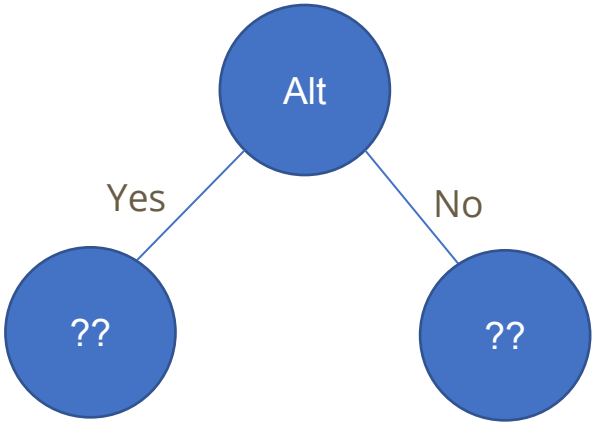
- This is a support class that models the data we need as one object, rather than the separate components you see in the lecture pseudocode.
- You should create a new instance of DecisionTreeData every time your algorithm recurs
    - Do not try to directly modify the examples array of the attribute list of the current examples set. Create new instances of the arrays and of DecisionTreeData.
- Check out methods: getAttributeList() and getClassifications()
- getExamples() returns a String[][]. Each row represents an example. (Row major!)
    - The last column is the example's classification and the first row is the first example, **not** the attribute names.
    - For instance, the classification of the first example (the top right corner of the 2D array) in the array can be found at this index:
        - examples[0][examples[0].length - 1]

What attribute should we split on first?

Ex	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	Yes
5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	Yes
7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	No
8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	Yes
9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	No
11	No	No	No	No	None	$	No	No	Thai	0-10	No
12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes

Let's start with the first attribute and visualize what the tree would look like.

How well does "Alternative" split the data?



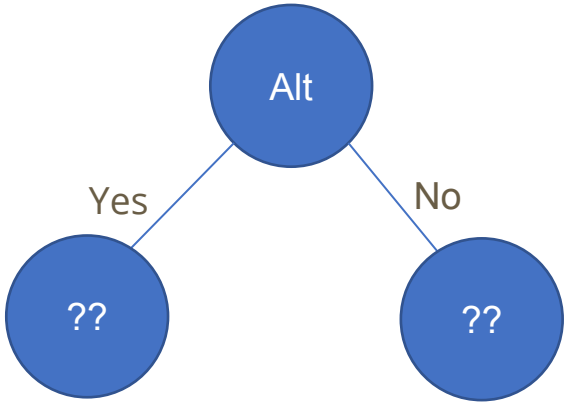
In other words, does having an alternative restaurant option impact someone's decision to wait for a table?

Ex	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	Yes
5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	Yes
7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	No
8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	Yes
9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	No
11	No	No	No	No	None	$	No	No	Thai	0-10	No
12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes

How much entropy does each subset have, and how probable is it that an example will be in that subset?

The "Yes" alt subset	
pK = 3	nK = 3
Entropy = 1	
Proportion = 6/12	

pK is the number of positive examples in the subset with "Yes" alt



The "No" alt subset	
pK = 3	nK = 3
Entropy = 1	
Proportion = 6/12	

Ex	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	Yes
5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	No
12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes

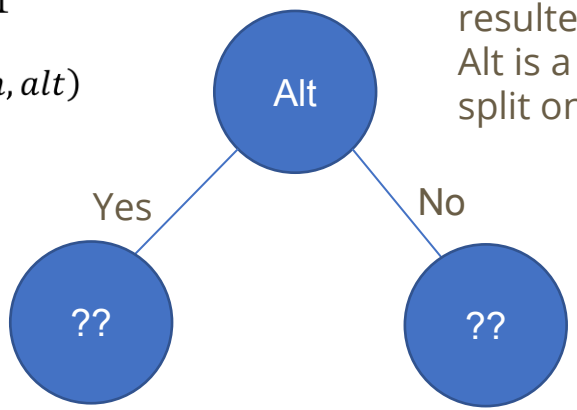
Ex	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	Yes
7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	No
8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	Yes
9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
11	No	No	No	No	None	$	No	No	Thai	0-10	No

How much entropy does each subset have, and how probable is it that an example will be in that subset?

$$Remainder(p,n,alt) = \frac{6}{12}(1) + \frac{6}{12}(1) = 1$$

$$Gain(p,n,alt) = Entropy(p,n) - Remainder(p,n,alt)$$
$$= 1 - 1 = 0$$

We didn't gain any information, since the split resulted in no loss of entropy. Alt is a terrible attribute to split on :(



The "Yes" alt subset	
pK = 3	nK = 3
Entropy = 1	
Proportion = 6/12	

pK is the number of positive examples in the subset with "Yes" alt

The "No" alt subset	
pK = 3	nK = 3
Entropy = 1	
Proportion = 6/12	

Ex	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	Yes
5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	No
12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes

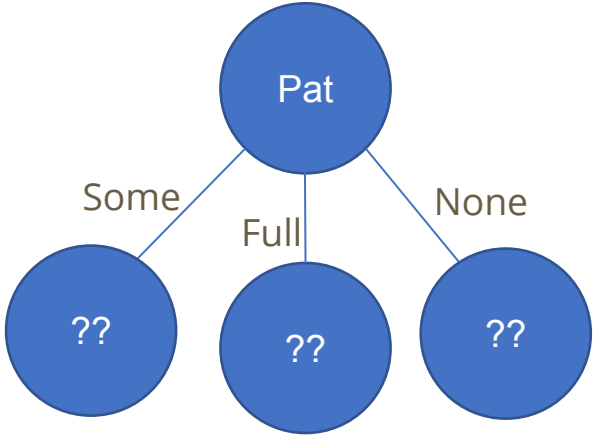
Ex	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	Yes
7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	No
8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	Yes
9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
11	No	No	No	No	None	$	No	No	Thai	0-10	No

Let's start over with another attribute.
Remember the full dataset has entropy of 1

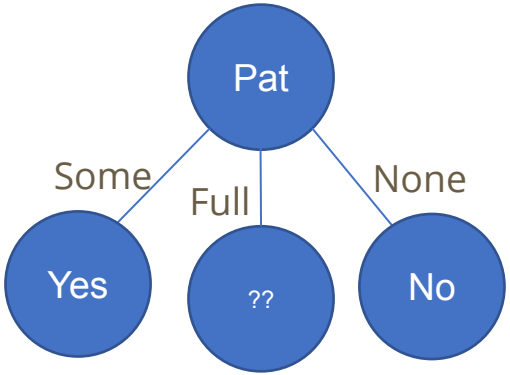
The current data	
p = 6	n = 6
Entropy(p, n) = 1	

Ex	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	Yes
5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	Yes
7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	No
8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	Yes
9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	No
11	No	No	No	No	None	$	No	No	Thai	0-10	No
12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes

Spoiler alert “Patrons” is the best attribute to split on. Let’s see how much it reduces entropy compared to “Alternative”



Ex	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	Yes
5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	Yes
7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	No
8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	Yes
9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	No
11	No	No	No	No	None	$	No	No	Thai	0-10	No
12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes



"None" patrons	
pK = 0	nK = 2
Entropy(pk, nK) = 0	
Proportion = 2/12	

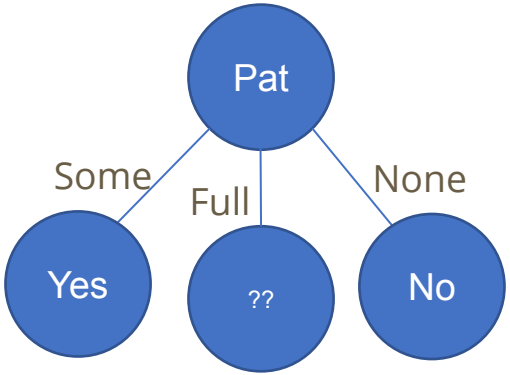
"Some" patrons	
pK = 4	nK = 0
Entropy(pK, nK) = 0	
Proportion = 4/12	

"Full" patrons	
pK = 2	nK = 4
Entropy(pK, nK) = 0.92	
Proportion = 6/12	

Ex.	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	No
11	No	No	No	No	None	$	No	No	Thai	0-10	No

Ex.	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	Yes
8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	Yes

Ex	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	Yes
5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	No
12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes



"None" patrons	
pK = 0	nK = 2
Entropy(pk, nK) = 0	
Proportion = 2/12	

"Some" patrons	
pK = 4	nK = 0
Entropy(pK, nK) = 0	
Proportion = 4/12	

"Full" patrons	
pK = 2	nK = 4
Entropy(pK, nK) = 0.92	
Proportion = 6/12	

Ex.	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	No
11	No	No	No	No	None	$	No	No	Thai	0-10	No

Ex.	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	Yes
8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	Yes

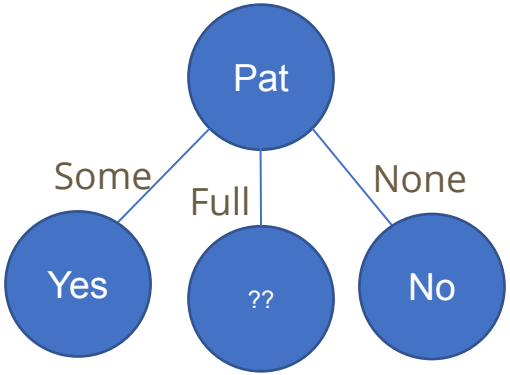
$$Remainder(p, n, pat) = \frac{2}{12}(0) + \frac{4}{12}(0) + \frac{6}{12}(0.92) = 0.46$$

$$Gain(p, n, pat) = Entropy(p, n) - Remainder(p, n, alt)$$
$$= 1 - 0.46 = 0.54$$

Ex	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	Yes
5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	No
12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes

We were able to greatly reduce the amount of entropy in the data set! This is a good attribute to split on.

To be sure it is the best, we have to examine every attribute and pick the one with the most information gain.



"None" patrons	
pK = 0	nK = 2
Entropy(pk, nK) = 0	
Proportion = 2/12	

"Some" patrons	
pK = 4	nK = 0
Entropy(pK, nK) = 0	
Proportion = 4/12	

"Full" patrons	
pK = 2	nK = 4
Entropy(pK, nK) = 0.92	
Proportion = 6/12	

Ex.	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	No
11	No	No	No	No	None	$	No	No	Thai	0-10	No

Ex.	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	Yes
8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	Yes

$$Remainder(p,n,pat) = \frac{2}{12}(0) + \frac{4}{12}(0) + \frac{6}{12}(0.92) = 0.46$$

$$Gain(p,n,pat) = Entropy(p,n) - Remainder(p,n,alt)$$
$$= 1 - 0.46 = 0.54$$

Ex	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	Yes
5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	No
12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes

The subsets with "None" and "Some" patrons have hit a base case since all their examples have the same classification, so the node those branches point to are leaves.

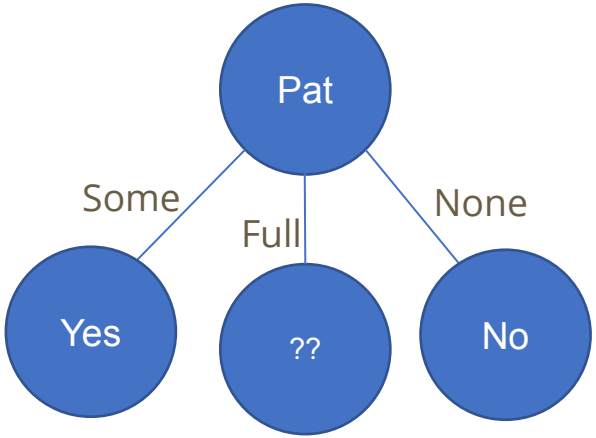
The "Full" subset is still unsorted and there are still attributes to consider, so we will continue with general case.

The subtree

Ex.	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	Yes
5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	No
12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes

Now we have to pick a new attribute to split on

Ex.	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Cls.
2	Yes	No	No	Yes	*Full*	$	No	No	Thai	30-60	No
4	Yes	No	Yes	Yes	*Full*	$	Yes	No	Thai	10-30	Yes
5	Yes	No	Yes	No	*Full*	$$$	No	Yes	French	>60	No
9	No	Yes	Yes	No	*Full*	$	Yes	No	Burger	>60	No
10	Yes	Yes	Yes	Yes	*Full*	$$$	No	Yes	Italian	10-30	No
12	Yes	Yes	Yes	Yes	*Full*	$	No	No	Burger	30-60	Yes



In this branch, we are only considering the examples with “Full” patrons. We know that every example in this branch has the same value for this attribute, so we don’t consider it in our calculations for the rest of this subtree.

It’s important to make sure you remove “Pat” from the list of available attributes, since checking if we’ve run out of attributes is a base case.

Now, we repeat the whole process with this smaller data set!