Decision Trees:
- Built from training data.
  - Each leaf has a label (YES/NO, T/F).
  - Non-leaf node has a feature (ex. patrons?, hungry?, type?).
- Each edge has a possible value of the feature of its parent node (ex. none, some, full).

How to use a tree to classify new example?
ex. Patrons: Full, Hungry = Yes, Type = French.

To classify new example with features $f_1, f_2, \ldots, f_n$:
- Start at root.
- Traverse edges corresponding to $f_1, f_2, \ldots, f_n$ until at a leaf.
- Set example's label to label at leaf.

Best Decision Tree?
Smallest tree that correctly classifies all training examples.

Problem is NP-hard so want to find a small tree.

How do we construct a small decision tree?
How do we construct any decision tree?
Construct (Node n) : "High-level"

1. Check base cases.

2. Select a feature f, for node at current level.

3. For each possible value of f:
   subChildren[k] = Partition examples based on their value for f.

4. construct (subChildren[1],)
   construct (subChildren[2],)
   construct (subChildren[k],)

Base Cases:

1. If all remaining examples at n have same label, output that label.

2. If no more examples to partition at node n, output parent's majority label (i.e. True if # true exs > # false exs o/w False).

3. If no more features, output majority label.
Suppose we chose features: Patrons, Hungry, Type.
(will discuss later how to choose features)

```
Patrons?
  None  Some  Full
  0 True 4 True
  2 False 0 False

Base Case 1
Output False
```

Hungry?
F

```
Type
  1  2  3
     French Italian Thai Burger
     (later)

BC. #1
Output True
```

No examples match (BC #2).
Use parent's majority label.
Parent is Type.
Up to Type: Patrons = Full, Hungry: T.
3 True, 2 False
Output: True
Example of BC #3:

Fri/Sat? ≤ last feature

- T
  - 4 True
    - 5 False
  - False
- F
  - 5 True
    - 3 False
  - True