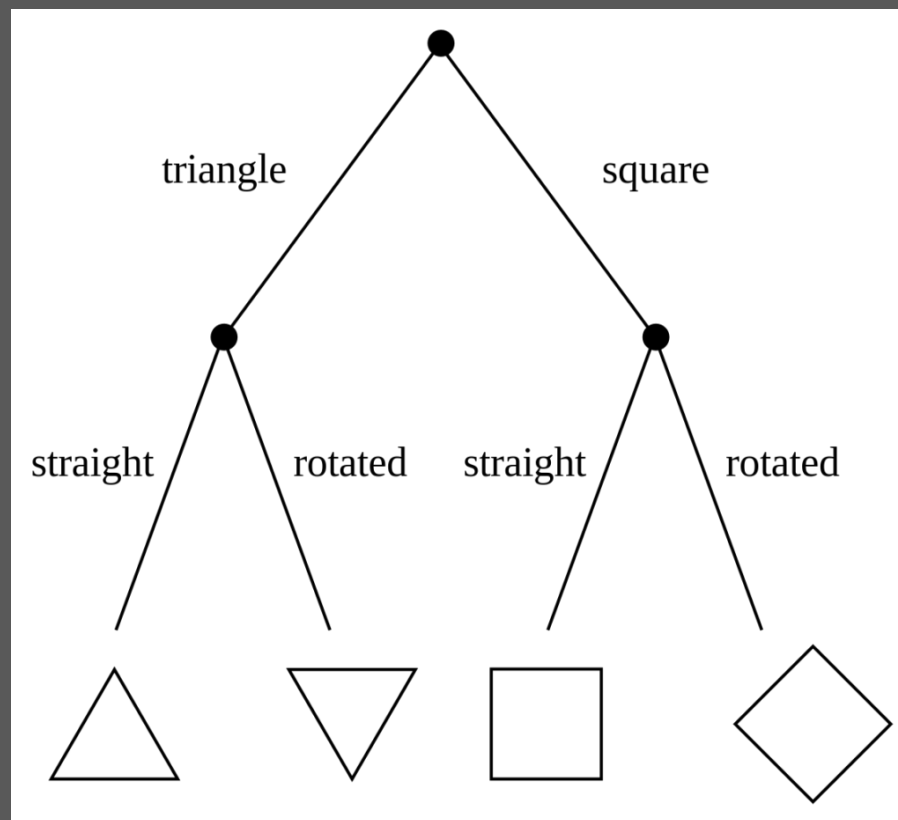


Which Sci-Fi



Discussion Decision Tree





Day	Outlook	Temperature	Humidity	Wind	Go outside
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Example: Information Gain

Algorithm:

1. Calculate entropy of target class
2. Calculate entropy of each feature's values
3. Calculate information gain of each feature
4. Split at the maximum IG
5. Repeat #1 until no further class

Example: Information Gain

Day	Outlook	Temperature	Humidity	Wind	Go outside
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Example: Information Gain

After computing information gain of every features, we have this:

$$IG(S, Wind) = 0.048$$

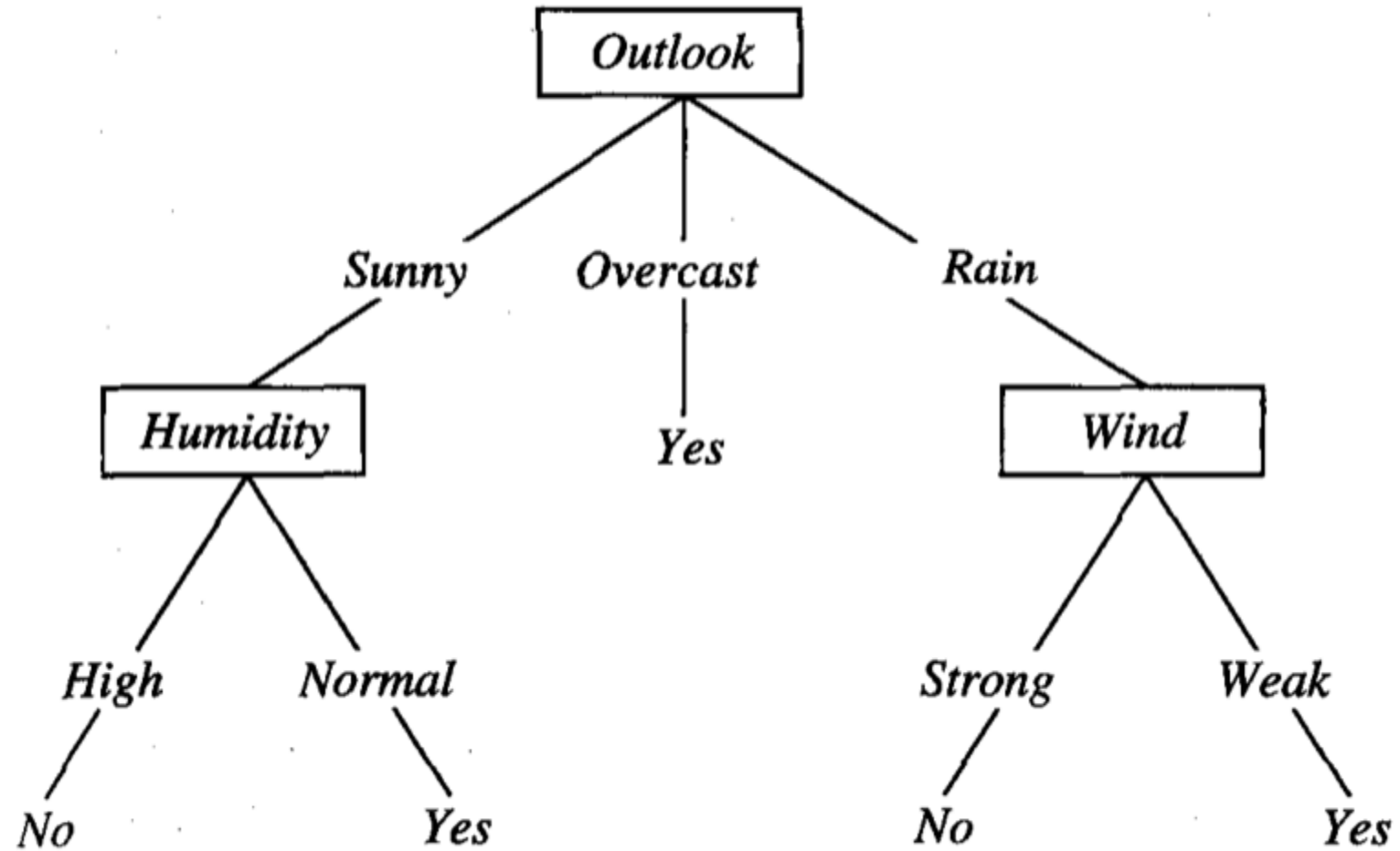
$$IG(S, Outlook) = 0.246$$

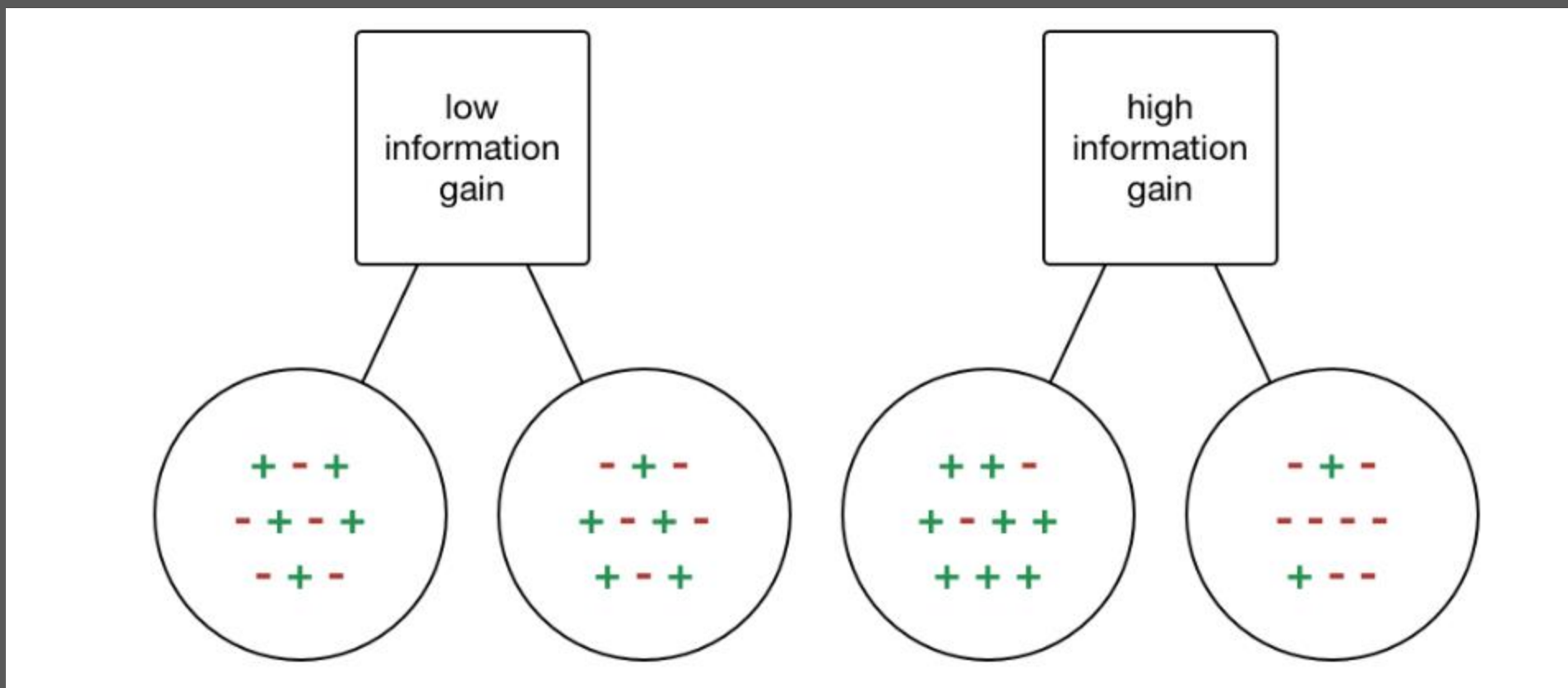
$$IG(S, Temperature) = 0.029$$

$$IG(S, Humidity) = 0.151$$

Maximum information gain when splitting at Outlook, so we split at Outlook. Then keep repeating the process in each node

Day	Outlook	Temperature	Humidity	Wind	Go outside
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No





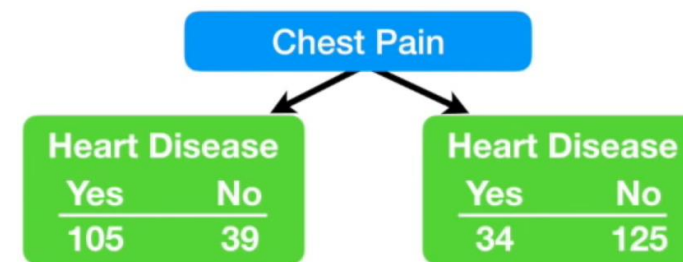
Another decision tree algorithm: Gini



Shrek Smith - the new genie

StatQuest: Decision Trees

(<https://www.youtube.com/watch?v=7VeUPuFGJHk>)



For this leaf, the Gini impurity = $1 - (\text{the probability of "yes"})^2 - (\text{the probability of "no"})^2$

$$= 1 - \left(\frac{105}{105 + 39}\right)^2 - \left(\frac{39}{105 + 39}\right)^2$$

$$= 0.395$$

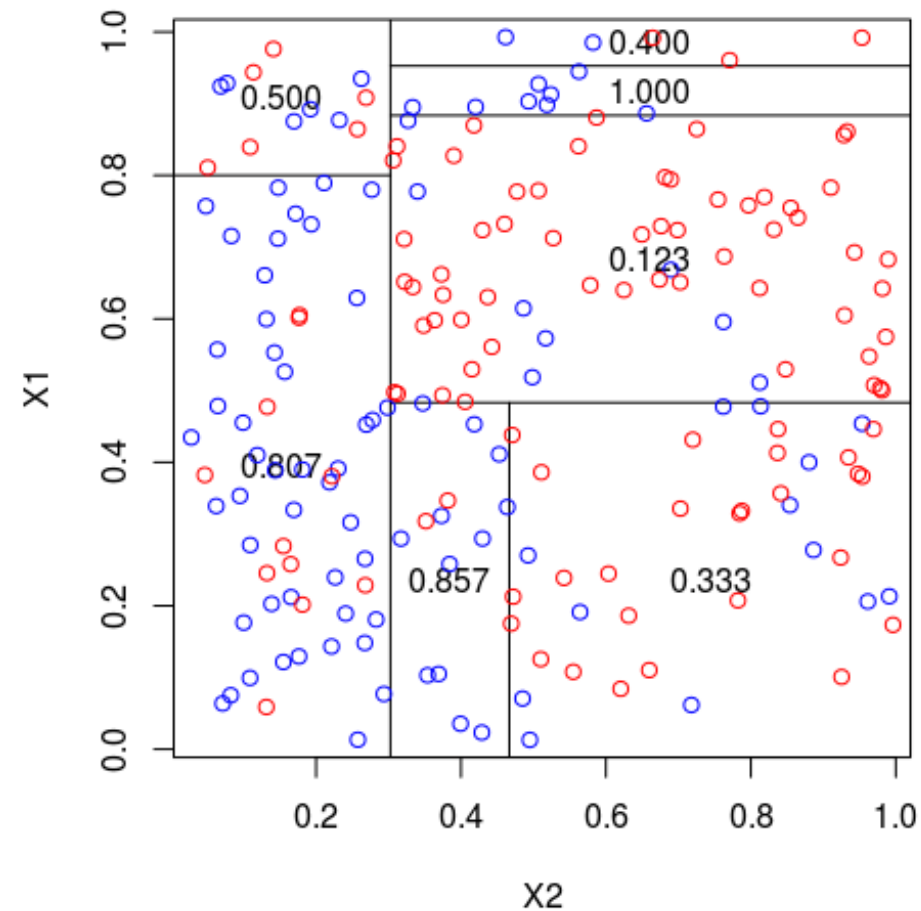
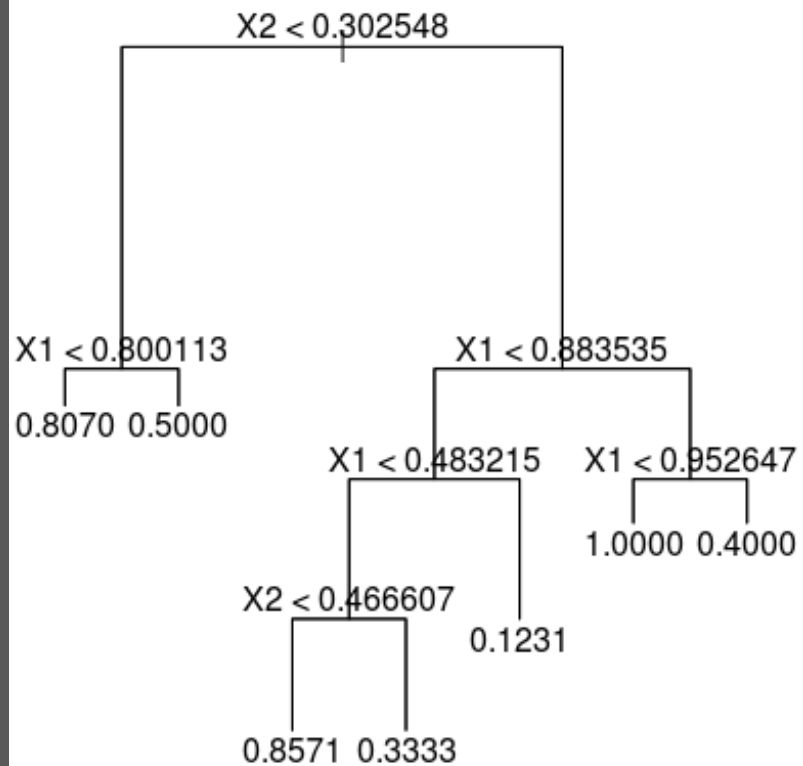


Continuous Data (number, and not categorical)

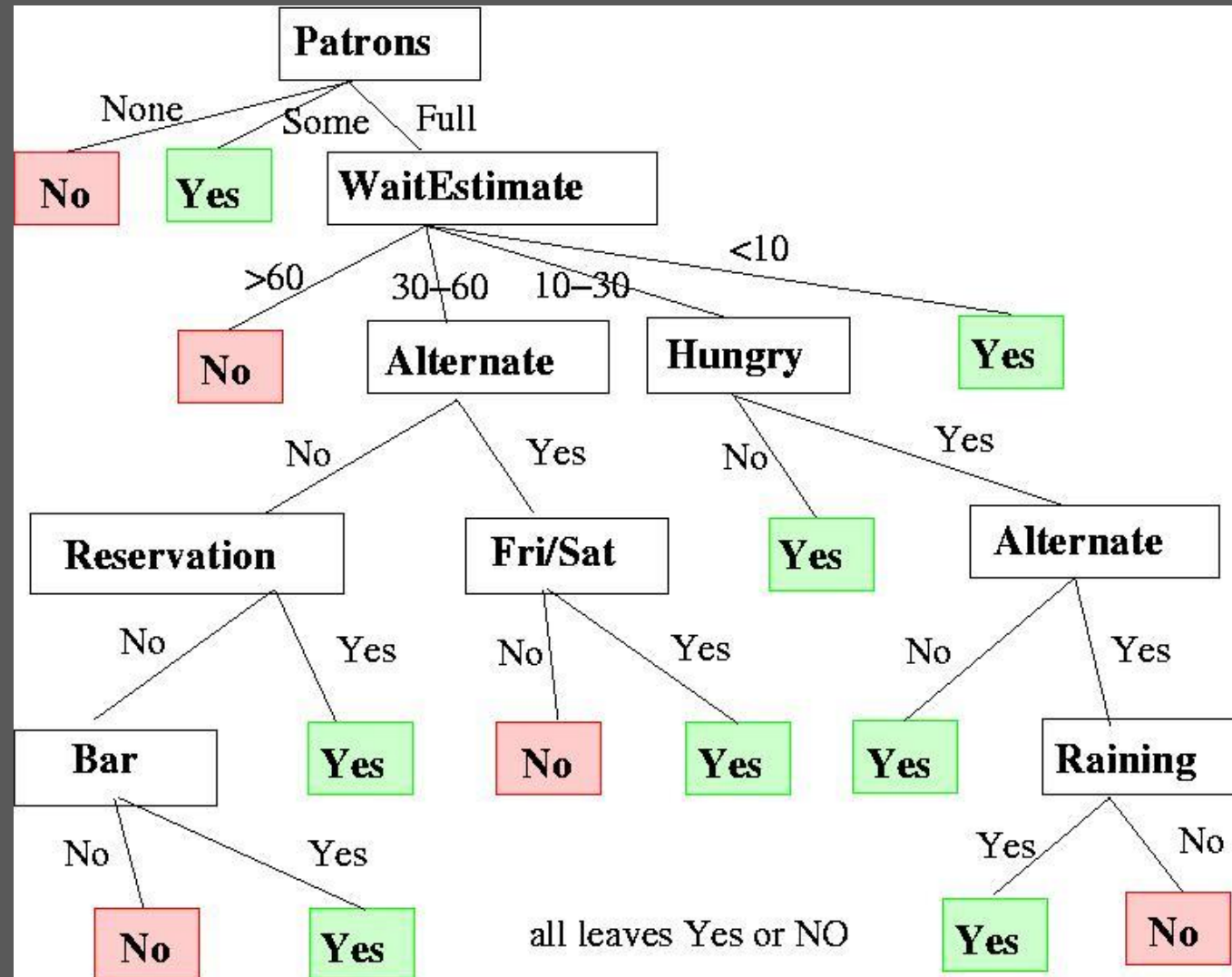
If the data is continuous data, you then sort, average for each adjacent row, and do the splitting algorithm at each average data



Overfit and Pruning



To fix overfitting problem, you can indicate what is the maximum tree's depth and stop there. (Pruning)



sklearn.tree.DecisionTreeClassifier

```
class sklearn.tree.DecisionTreeClassifier(*, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, ccp_alpha=0.0)
```

[\[source\]](#)

A decision tree classifier.

Read more in the [User Guide](#).

Parameters:

criterion : {"gini", "entropy"}, default="gini"

The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "entropy" for the information gain.

splitter : {"best", "random"}, default="best"

The strategy used to choose the split at each node. Supported strategies are "best" to choose the best split and "random" to choose the best random split.

max_depth : int, default=None

The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

min_samples_split : int or float, default=2

The minimum number of samples required to split an internal node:

- If int, then consider `min_samples_split` as the minimum number.
- If float, then `min_samples_split` is a fraction and `ceil(min_samples_split * n_samples)` are the minimum number of samples for each split.

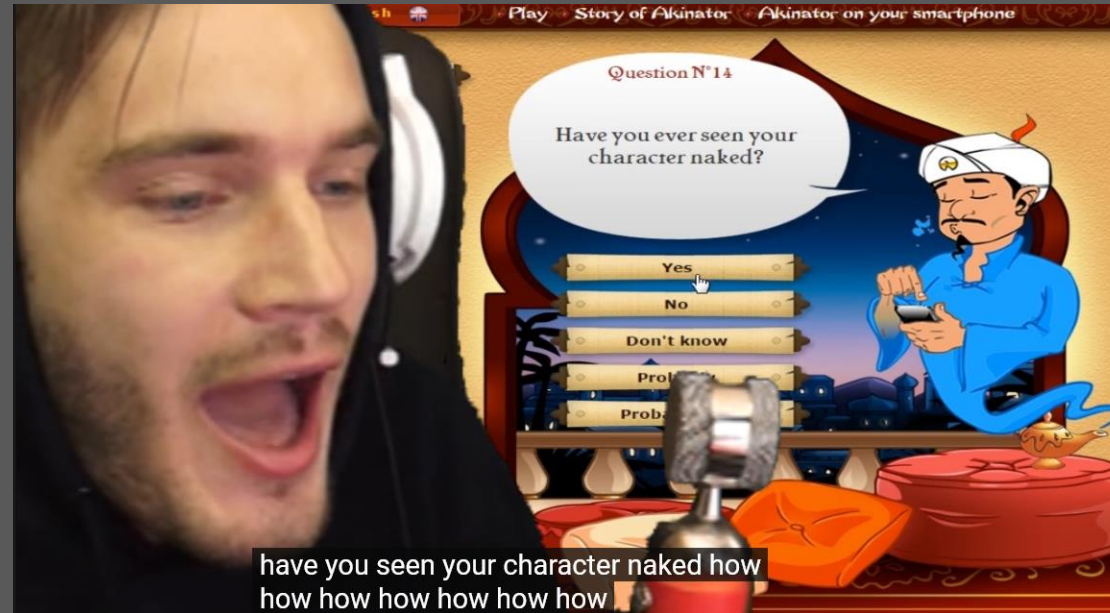
Gentle Introduction to Random Forest

If interested, check out Statquest: <https://youtu.be/7VeUPuFGJHk>



Might not do this

But you know, I learned something today



- Decision Tree is used for classification by “step into” child nodes until reaching leaf node
- To prevent overfitting, pruning the tree