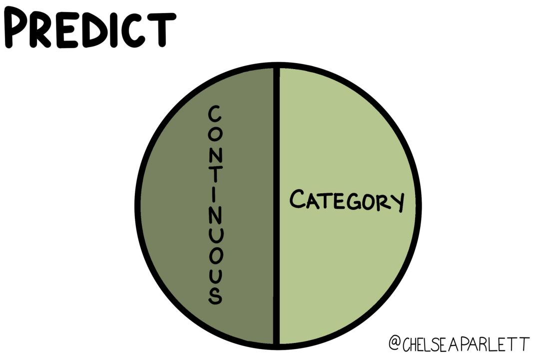


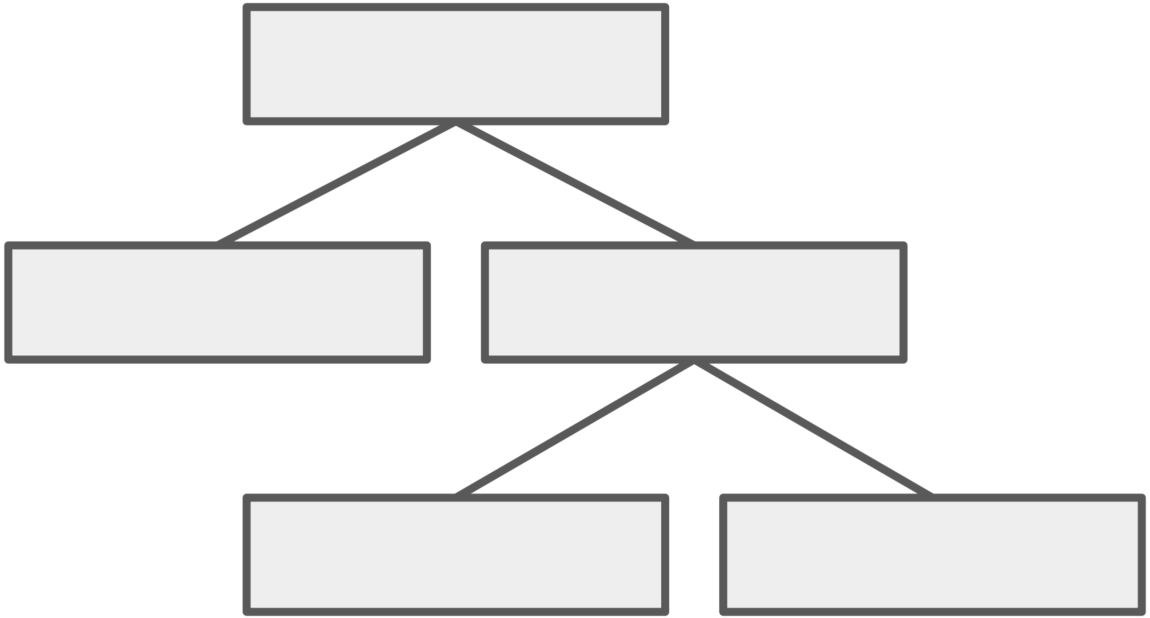
🌲 Based Models

Dr. Chelsea Parlett-Pelleriti

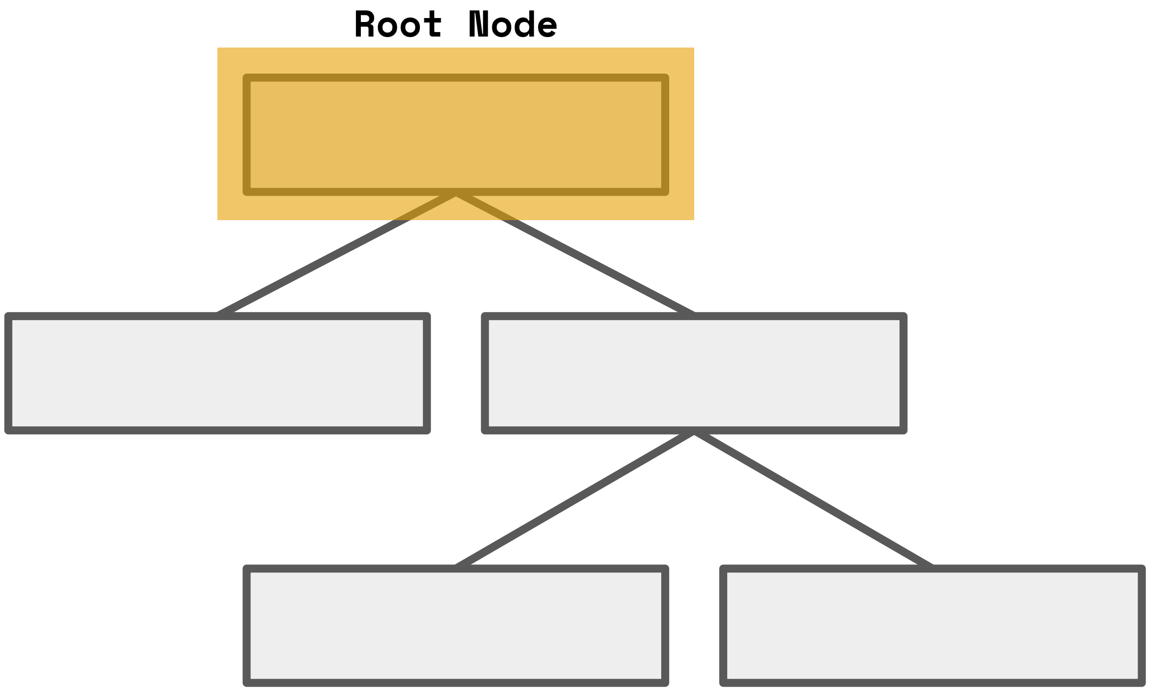
Decision Trees


I BE-LEAF
IN YOU
@CHELSEAPARLETT

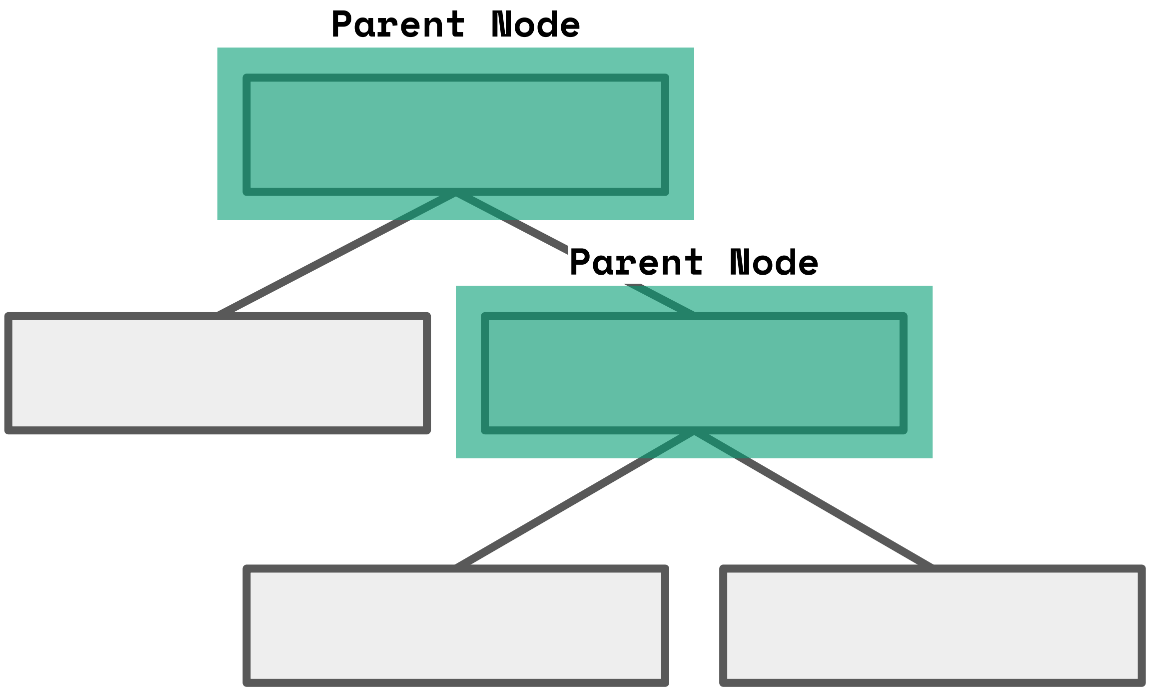
Tree Vocabulary



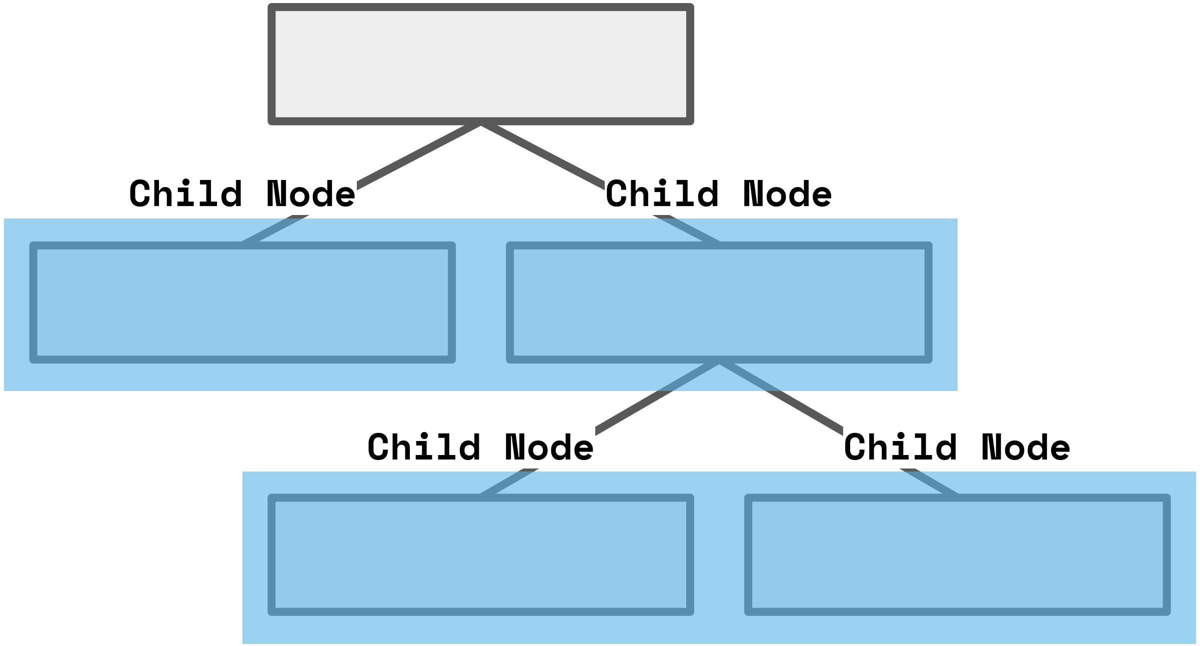
Tree Vocabulary



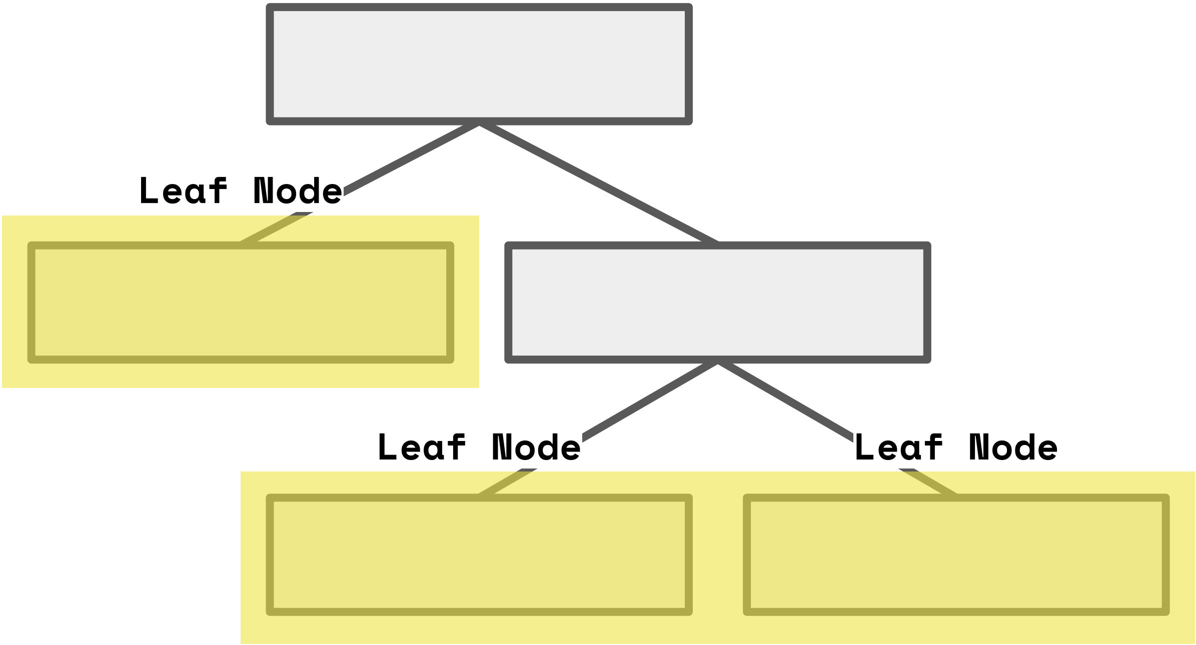
Tree Vocabulary



Tree Vocabulary

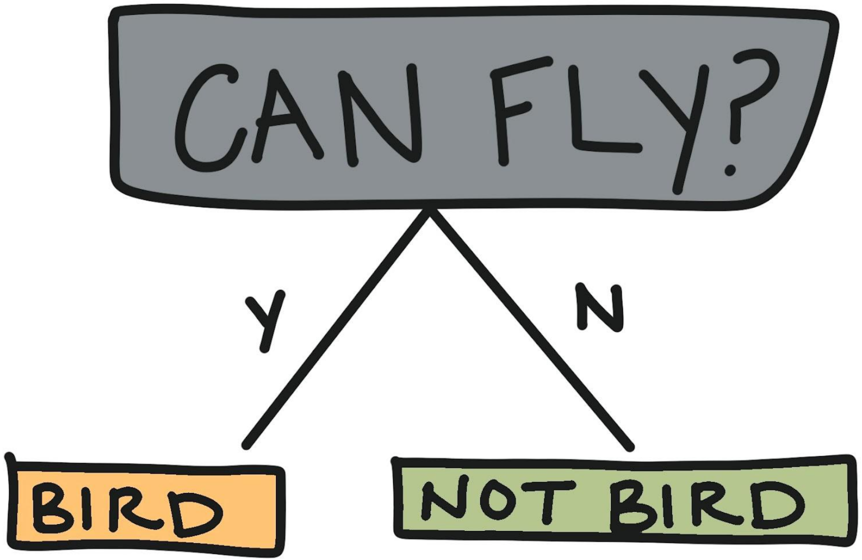


Tree Vocabulary

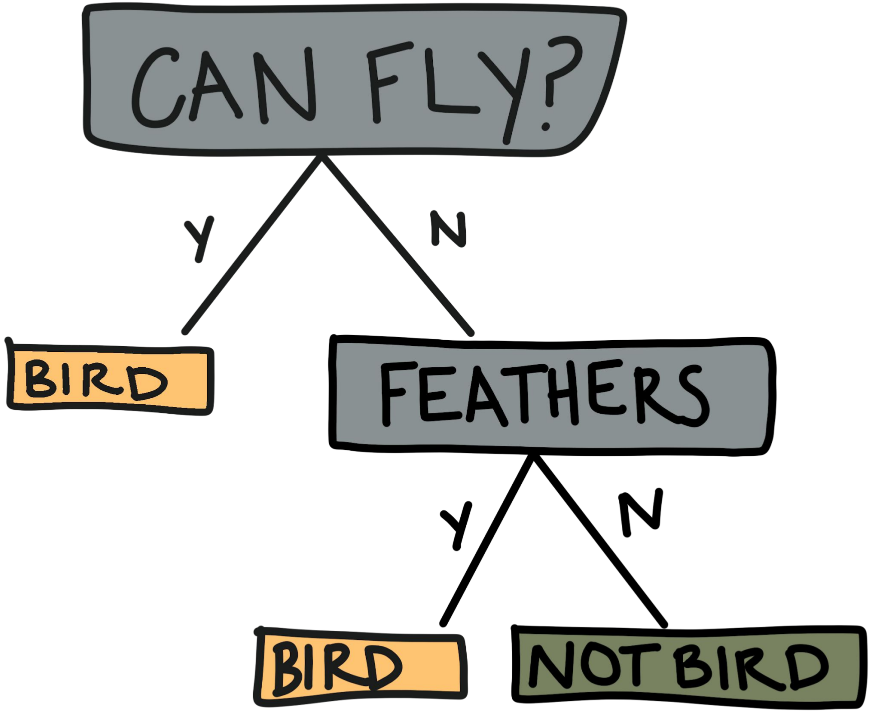


Twenty Questions

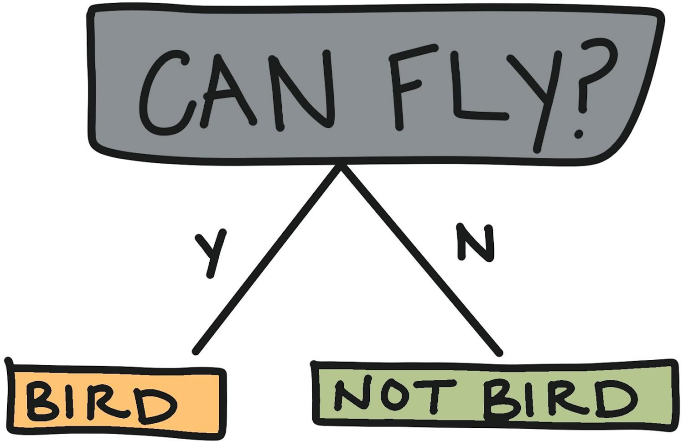
Simple Tree

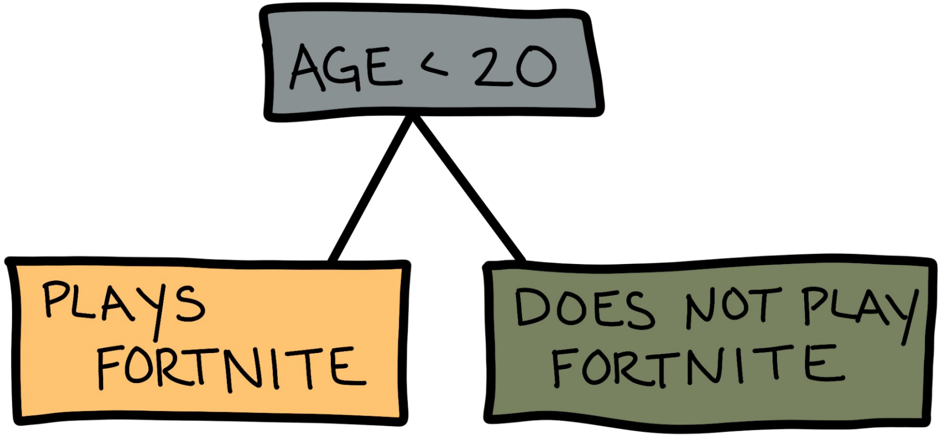


More Complicated Tree



Data Types


CAN FLY?
Y
N
BIRD
NOT BIRD


AGE < 20
PLAYS FORTNITE
DOES NOT PLAY FORTNITE

Gini Impurity and Entropy

$$\mathrm{GI} = 1 - \sum_{i=1}^{n} (p_i)^2$$

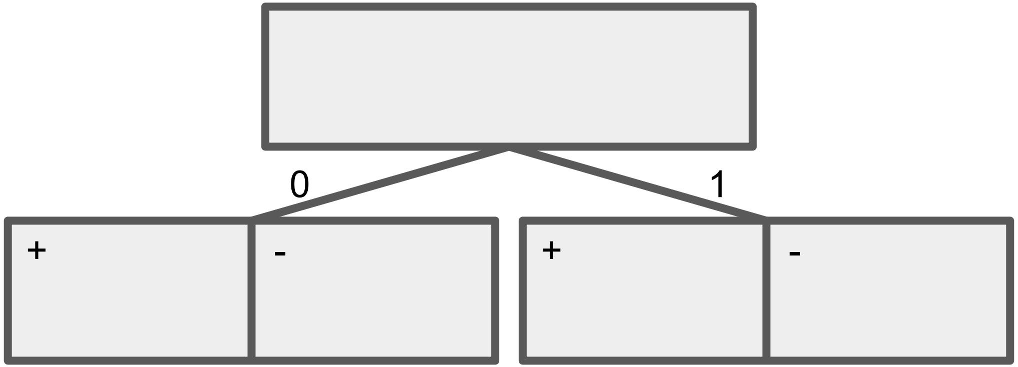
$$\mathrm{E} = - \sum_{i=1}^{n} p_i * log(p_i)$$

Goal: choose split so that GI (or entropy) is minimized

$$1 - \sum_{i=1}^{n} (p_i)^2$$

Categorical

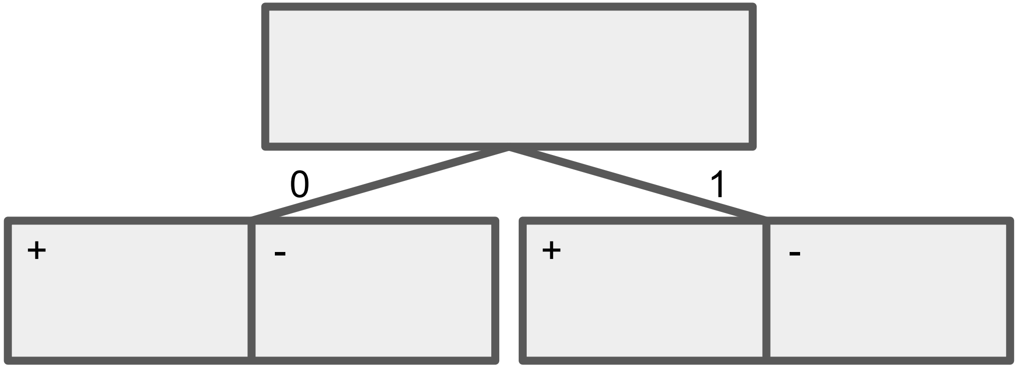
cats	pet	wfh	children	income
1	0	1	1	34
0	1	0	1	58.3
1	1	1	0	71.5
0	0	0	1	74.9
0	0	0	1	75.3
1	0	0	1	75.6
0	0	0	1	81
1	1	1	0	82.3
1	1	1	0	85.6
1	1	1	1	95.4



$$1 - \sum_{i=1}^{n} (p_i)^2$$

Categorical

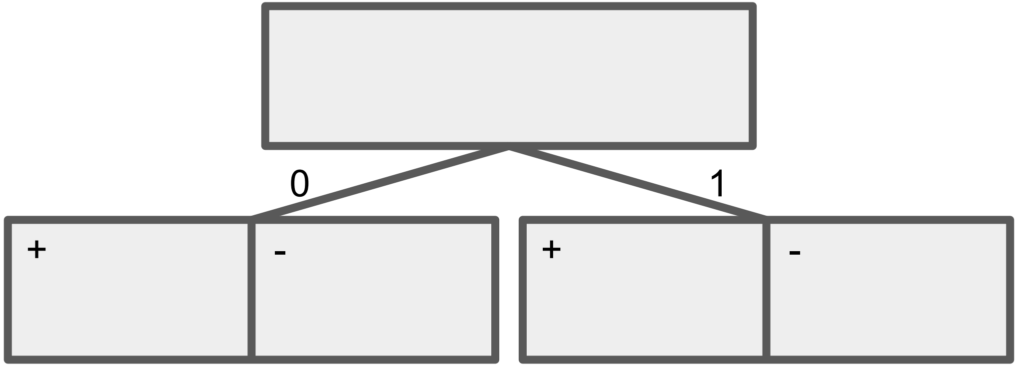
cats	pet	wfh	children	income
1	0	1	1	34
0	1	0	1	58.3
1	1	1	0	71.5
0	0	0	1	74.9
0	0	0	1	75.3
1	0	0	1	75.6
0	0	0	1	81
1	1	1	0	82.3
1	1	1	0	85.6
1	1	1	1	95.4



$$1 - \sum_{i=1}^{n} (p_i)^2$$

Continuous

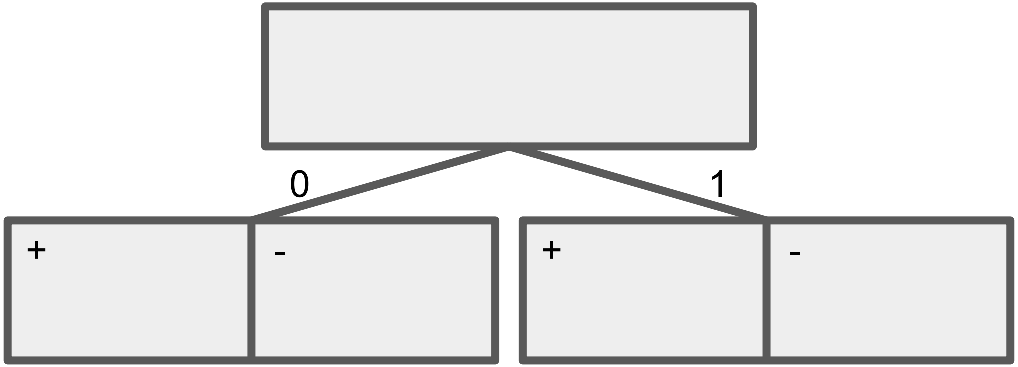
cats	pet	wfh	children	income
1	0	1	1	34
0	1	0	1	58.3
1	1	1	0	71.5
0	0	0	1	74.9
0	0	0	1	75.3
1	0	0	1	75.6
0	0	0	1	81
1	1	1	0	82.3
1	1	1	0	85.6
1	1	1	1	95.4



Continuous

$$1 - \sum_{i=1}^{n} (p_i)^2$$

cats	pet	wfh	children	income
1	0	1	1	34
0	1	0	1	58.3
1	1	1	0	71.5
0	0	0	1	74.9
0	0	0	1	75.3
1	0	0	1	75.6
0	0	0	1	81
1	1	1	0	82.3
1	1	1	0	85.6
1	1	1	1	95.4



Basic Steps

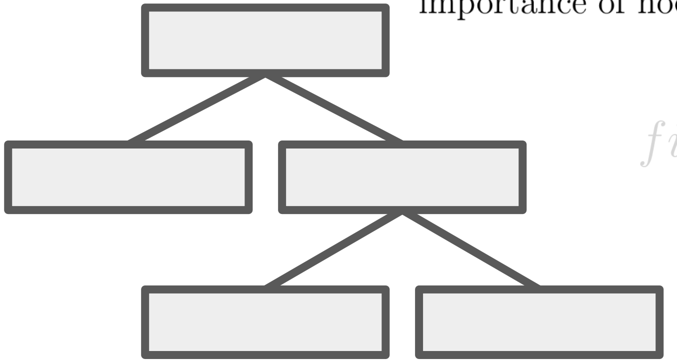
1. Calculate Gini Impurity (or Entropy/Information Gain) for each node
2. Choose Node with lowest score
3. If the parent node has the lowest score, it is a leaf.

Example

Variable Importance

1. How much does this feature reduce node impurity?
2. If we shuffle the values of this feature, how much does it reduce the performance?

$$\underbrace{\mathrm{imp}_j}_{\text{importance of node } j} = \overbrace{w_j C_j}^{\text{weighted parent node impurity}} - \underbrace{(w_{\mathrm{left}_j} C_{\mathrm{left}_j} + w_{\mathrm{right}_j} C_{\mathrm{right}_j})}$$

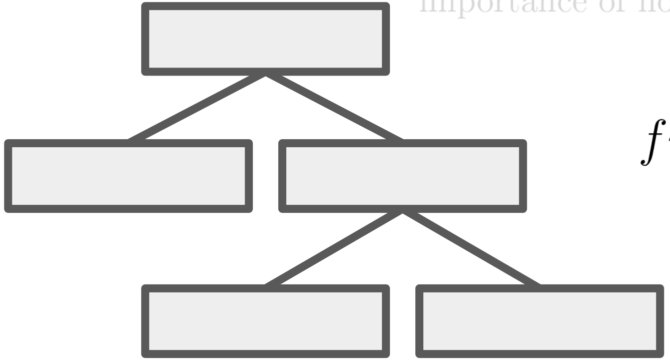


$$fi_i = \frac{\sum_{j \in S_i} \mathrm{imp}_j}{\sum_{k \in S_{all}} \mathrm{imp}_k}; S_i \text{ is set of all nodes that split on feature}_i$$

Variable Importance

1. How much does this feature reduce node impurity?
2. If we shuffle the values of this feature, how much does it reduce the performance?

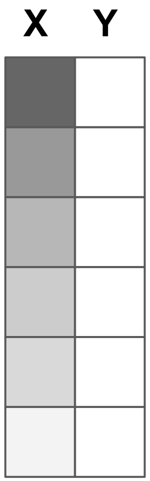
$$\underbrace{\mathrm{imp}_j}_{\text{importance of node } j} = \overbrace{w_j C_j}^{\text{weighted parent node impurity}} - \underbrace{(w_{\mathrm{left}_j} C_{\mathrm{left}_j} + w_{\mathrm{right}_j} C_{\mathrm{right}_j})}$$



$$fi_i = \frac{\sum_{j \in S_i} \mathrm{imp}_j}{\sum_{k \in S_{all}} \mathrm{imp}_k}; S_i \text{ is set of all nodes that split on feature}_i$$

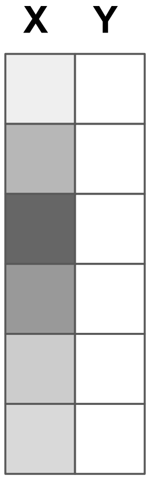
Variable Importance

1. How much does this feature reduce node impurity?
2. If we shuffle the values of this feature, how much does it reduce the performance?



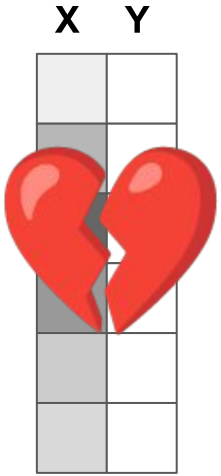
Variable Importance

1. How much does this feature reduce node impurity?
2. If we shuffle the values of this feature, how much does it reduce the performance?



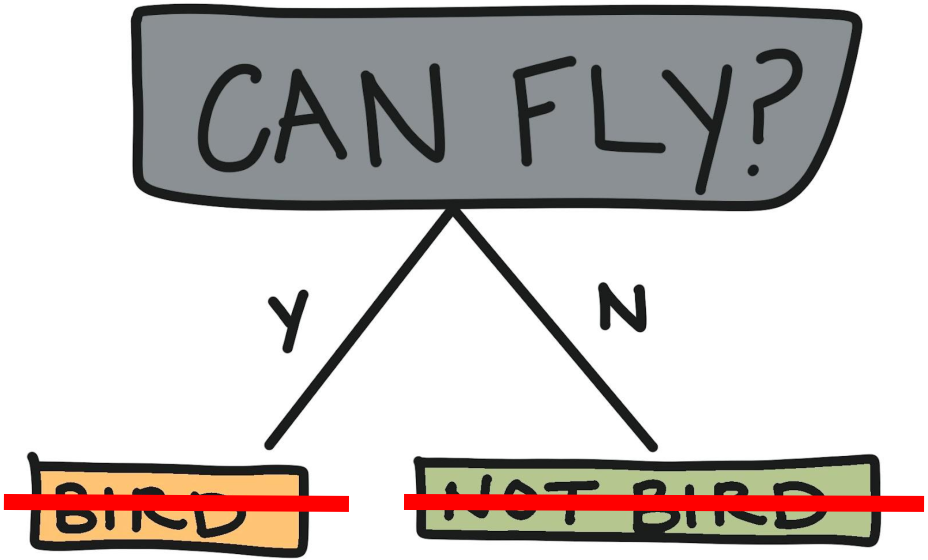
Variable Importance

1. How much does this feature reduce node impurity?
2. If we shuffle the values of this feature, how much does it reduce the performance?



Regression Trees

Regression Trees



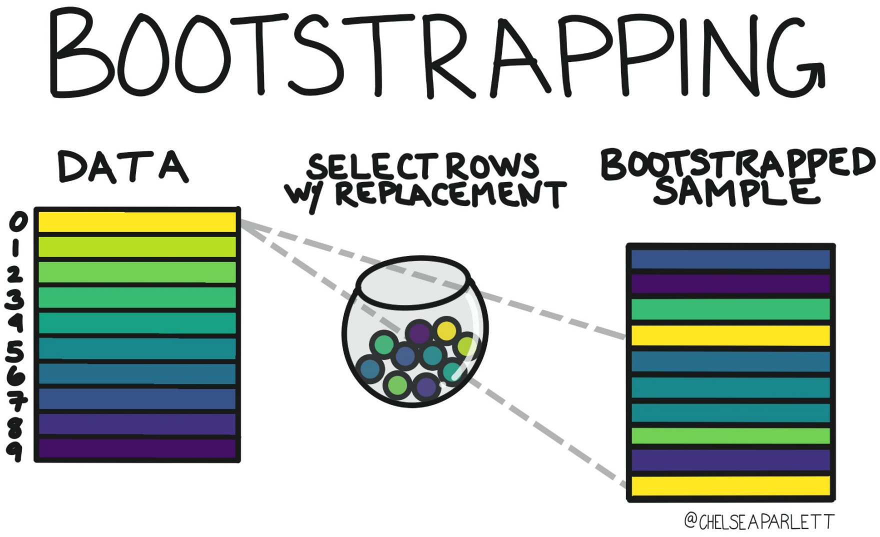
Random Forests

ONE vs. MANY

Random Forests

- Bootstrap Aggregating (**Bagging**)
- **Random Feature Selection**

Random Forests



Random Forests

- Bootstrap Aggregating (**Bagging**)
- **Random Feature Selection**

Random Forests

Random Forests

Random Forests

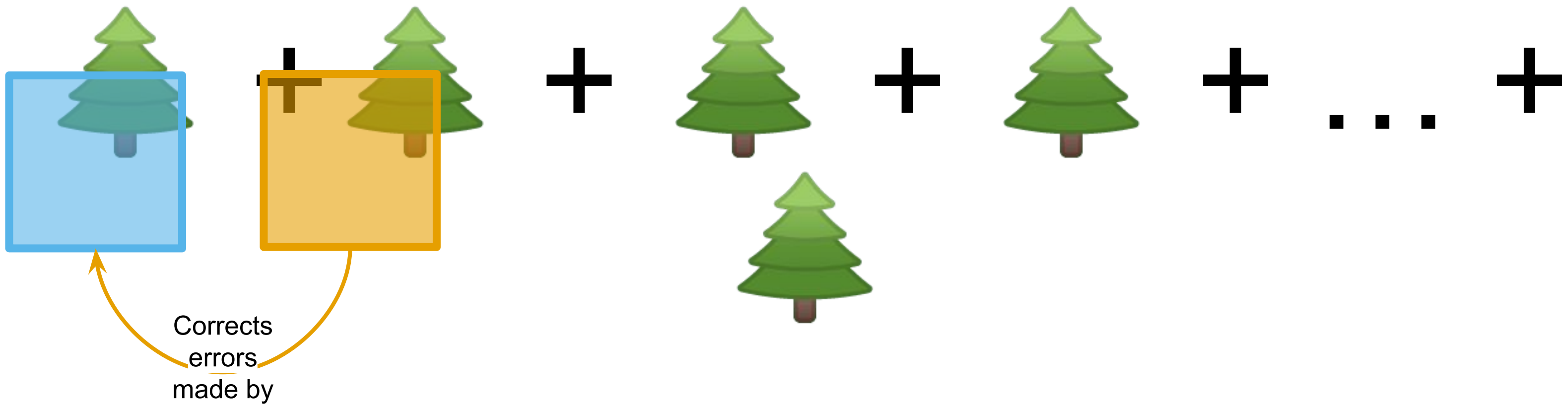
Important Hyperparameters

- # of trees
- # of features per tree

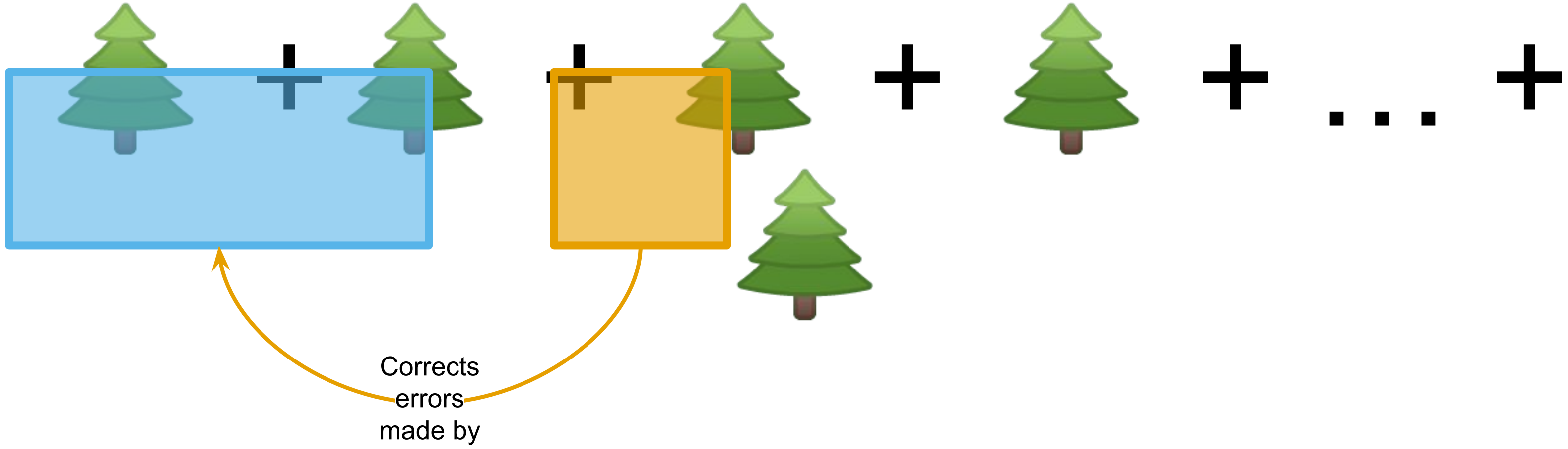
Gradient Boosting Trees

Gradient Boosting Trees

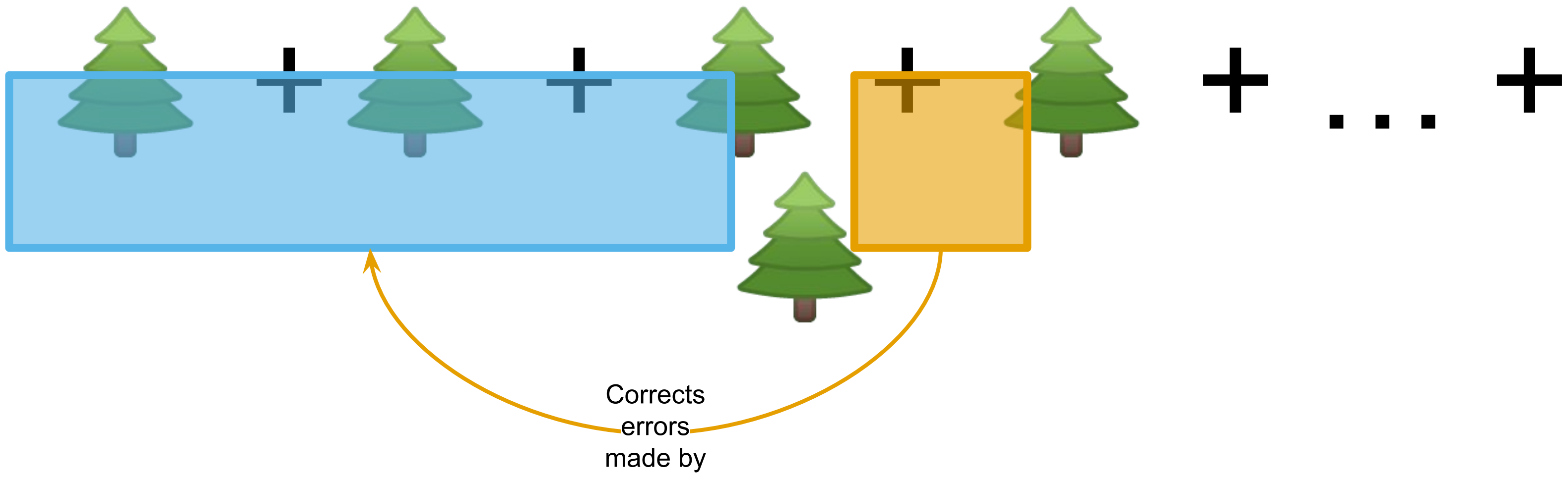
Gradient Boosting Trees



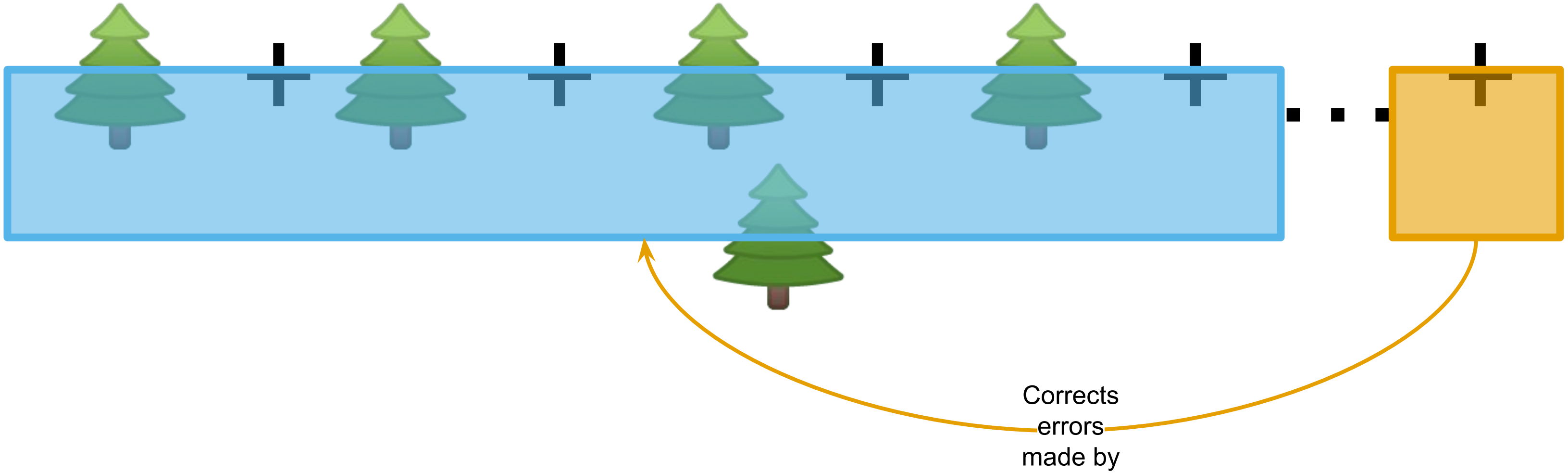
Gradient Boosting Trees



Gradient Boosting Trees



Gradient Boosting Trees



Gradient Boosting Tree

	Age	Initial Guess	Residual
Person 1	20		
Person 2	19		
Person 3	21		
Person 4	20		

Gradient Boosting Tree

	Age	Initial Guess	Residual
Person 1	20	20	
Person 2	19	20	
Person 3	21	20	
Person 4	20	20	

Gradient Boosting Tree

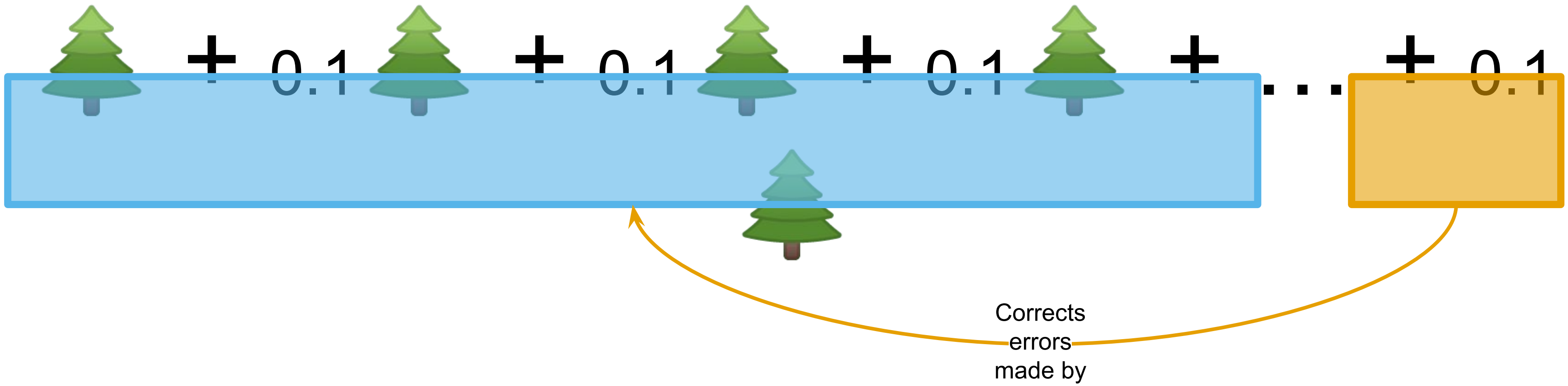
	Age	Initial Guess	Residual
Person 1	20	20	0
Person 2	19	20	-1
Person 3	21	20	1
Person 4	20	20	0

Gradient Boosting Tree

Actual Value = Prediction + Residual

	Age	Initial Guess	Residual
Person 1	20	20	0
Person 2	19	20	-1
Person 3	21	20	1
Person 4	20	20	0

Gradient Boosting Trees



In class question

- Which is more parallelizable?
- At inference, which is more parallelizable?