CS 584 – Machine Learning

Topic: Decision Trees

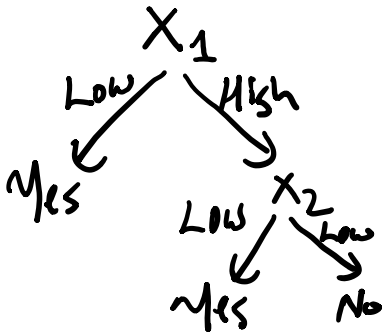
Mustafa Bilgic

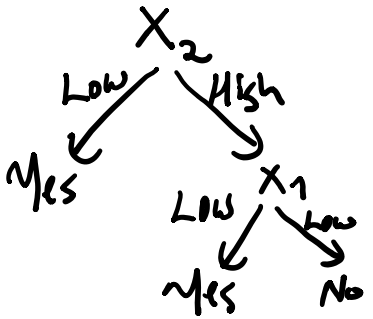
http://www.cs.iit.edu/~mbilgic

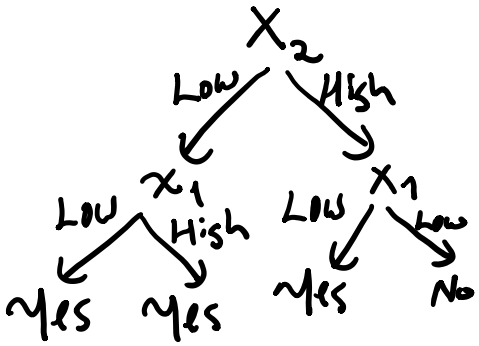
https://twitter.com/bilgicm

Example

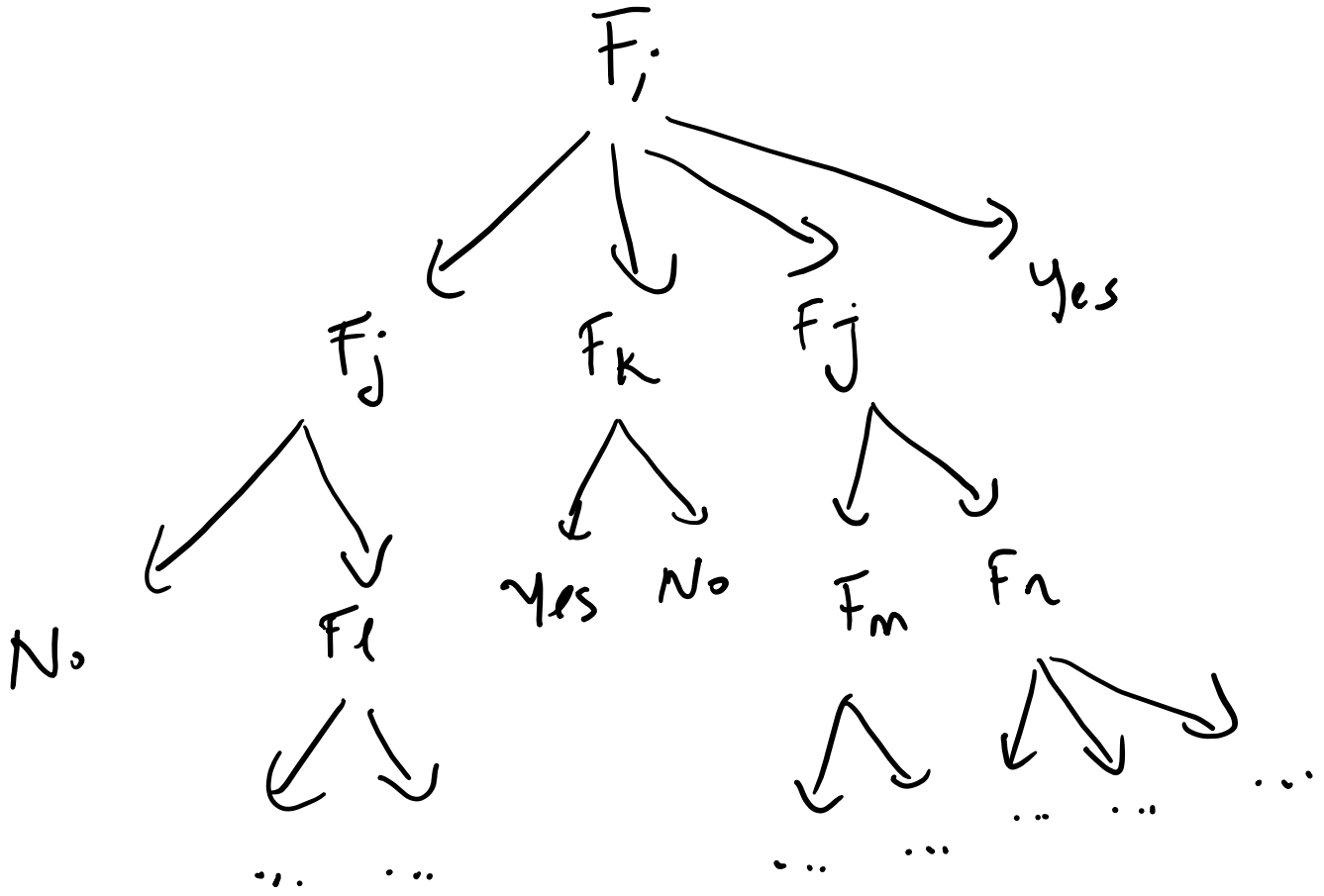
X_1	X_2	Y
Low	Low	Yes
Low	High	Yes
High	Low	Yes
High	High	No







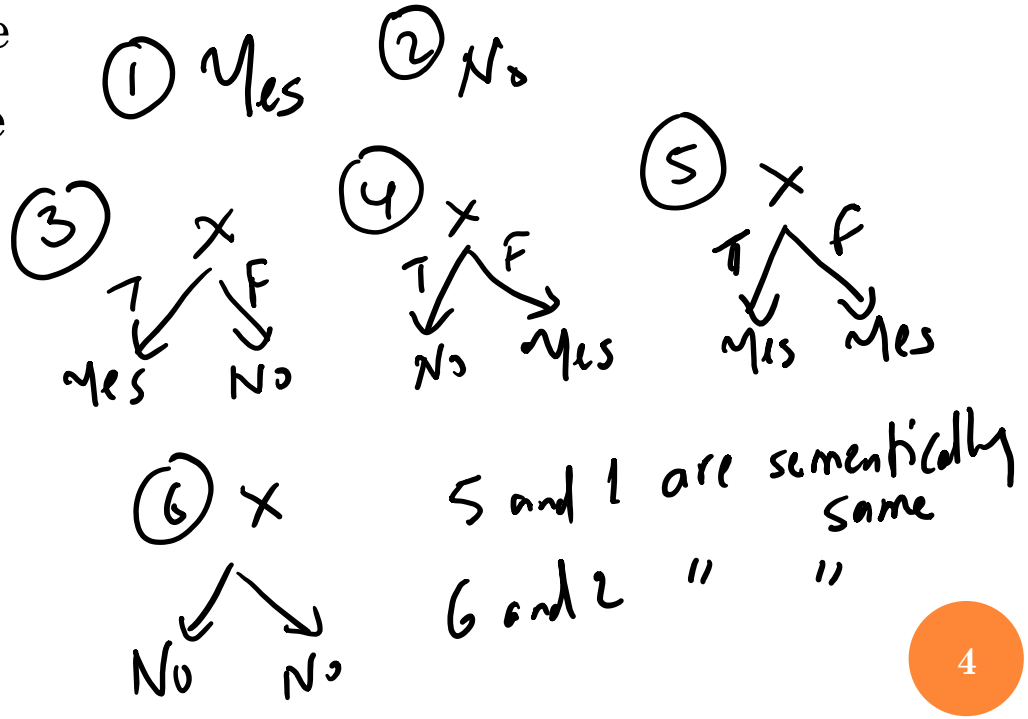
Decision Tree



Of Possible Decision Trees

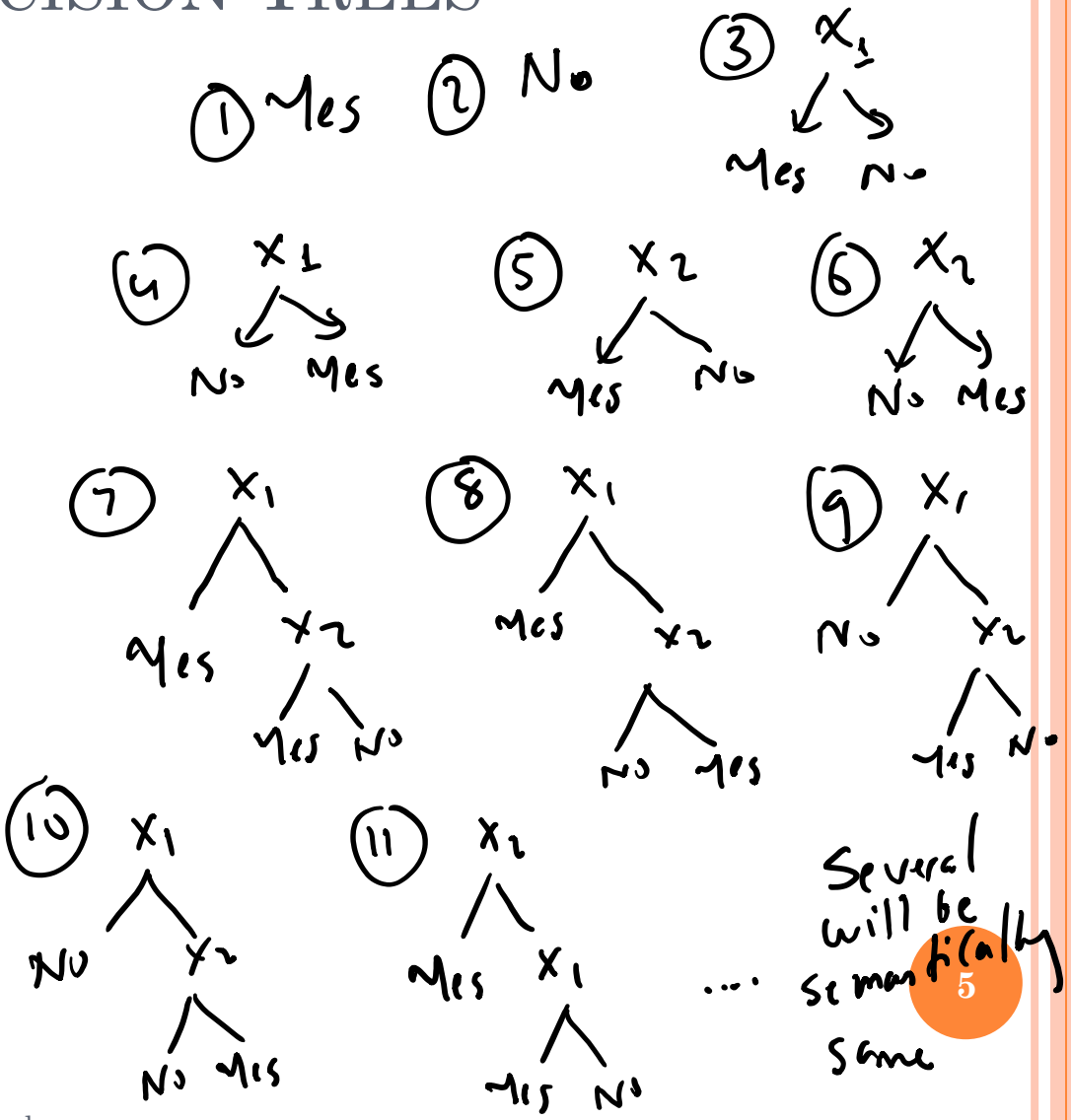
- Given one binary feature X (True/False) and a binary class Y (Yes/No), what is the
  - Size of the instance space
  - Size of the function space
  - Number of syntactically different decision trees?

X	f1	f2	f3	f4
T	Yes	Yes	No	No
F	Yes	No	Yes	No



Of Possible Decision Trees

- Given two binary features X_1 and X_2 and a binary class Y, the
  - Size of the instance space
    - $2x2 = 4$
  - Size of the function space
  - $2^4 = 16$
  - Number of syntactically different decision trees?



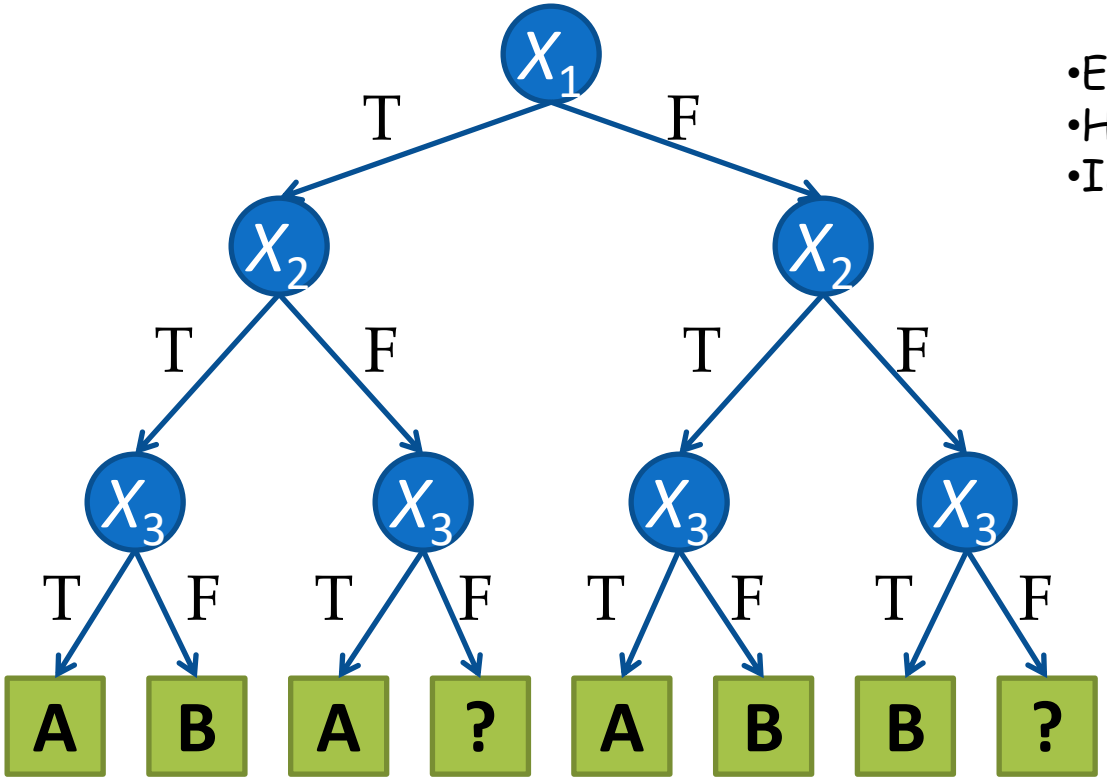
How would you learn a DT?

- That is, how would you choose the nodes and leaves?
  - Which node(s) would you split on?
  - When would you stop splitting?
- Here is a naïve DT learning algorithm
  - The node at the i^{th} level is the i^{th} feature
  - The leaf is the last feature
  - Let's call this algorithm NDT
- Questions
  - What's the empirical error (error on training data)?
  - Can you use it for prediction?
  - Is it interpretable?
  - Does this provide any compression?
  - Can you do any better?

Let's Apply NDT: Data1

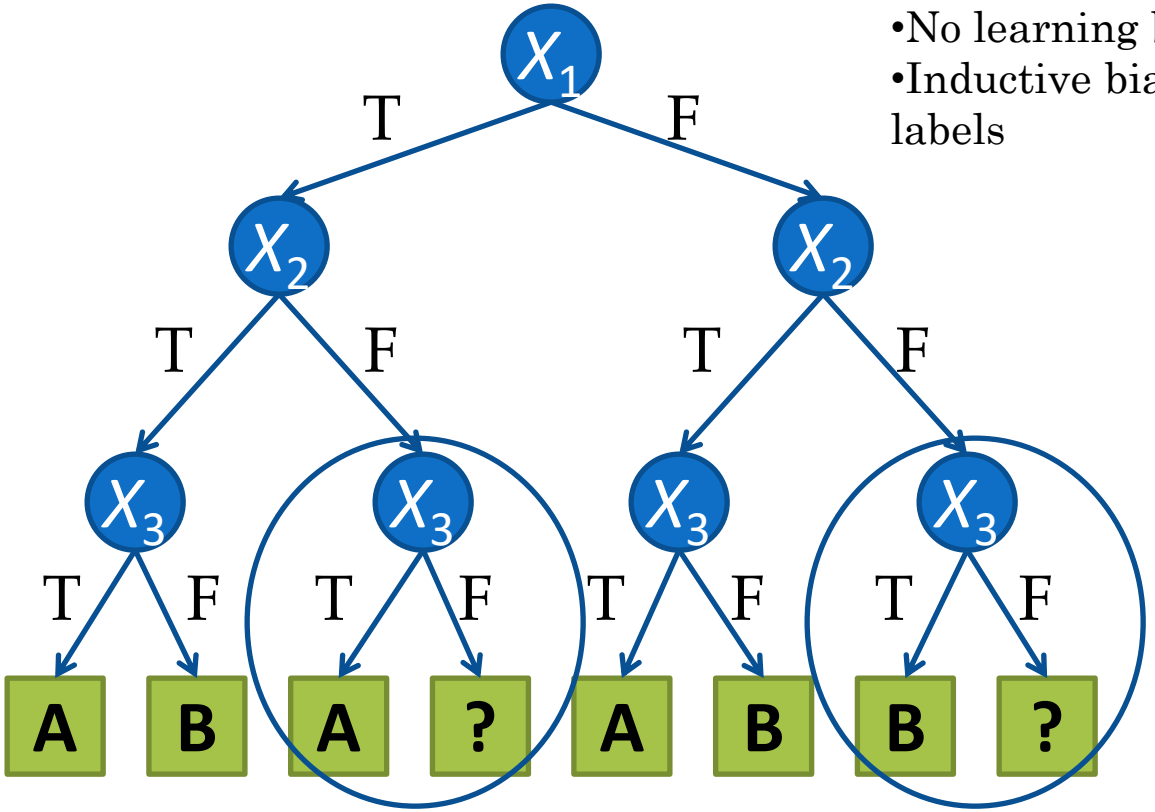
X_1	X_2	X_3	Y
T	T	T	A
T	T	F	B
T	F	T	A
F	T	T	A
F	T	F	B
F	F	T	B

Let's Apply NDT: Data1 – Tree1



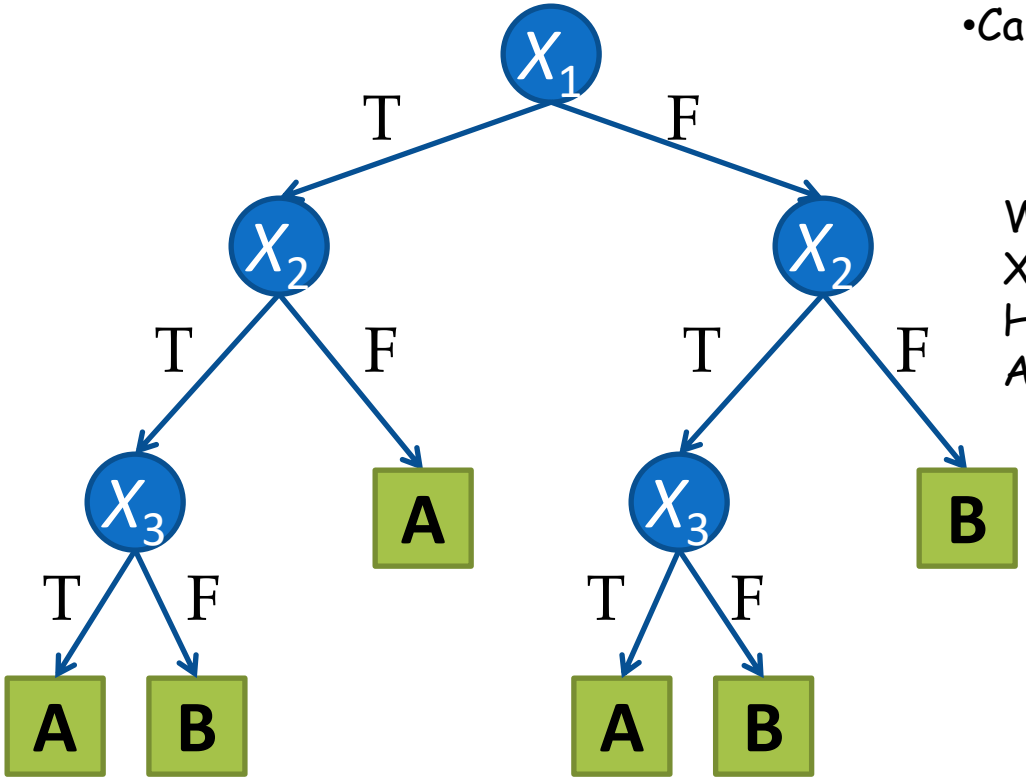
- Empirical error?
- How do you predict F, F, F?
- Introduce inductive bias. How?

DATA1– TREE1



- No learning bias -> no prediction
- Inductive bias = Similar instances -> similar labels

DATA1– TREE2



- Can we use this for prediction?
  - Yes!

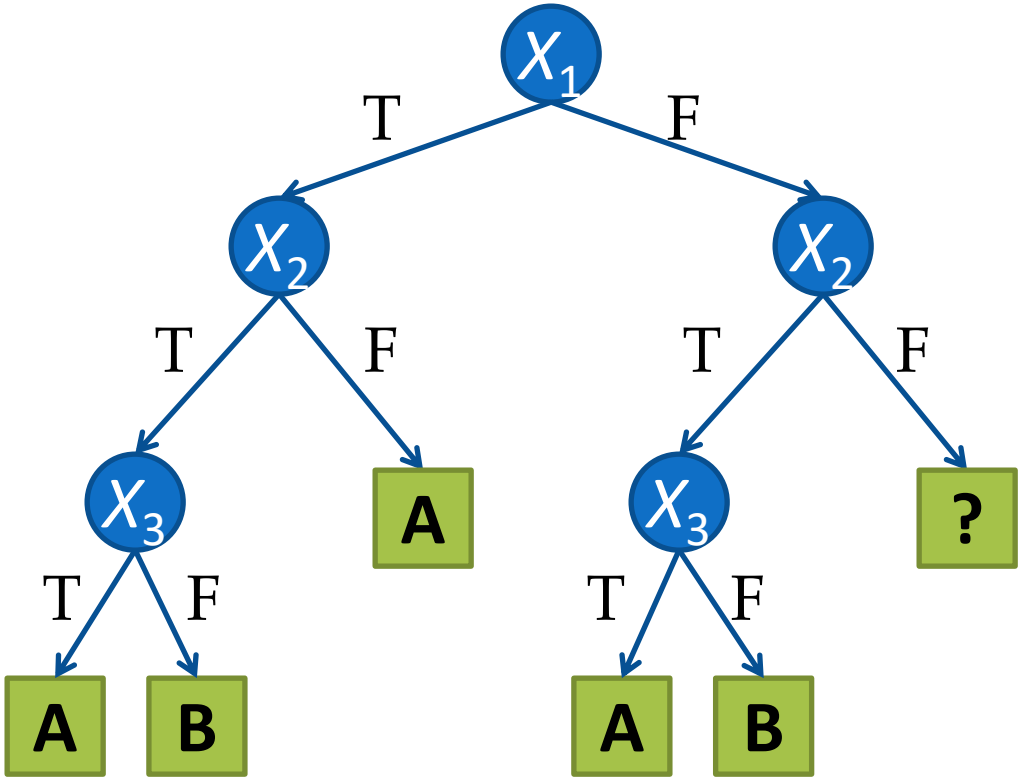
What if we had no case for X1=F, X2=F in the training data? How would you classify those cases? A or B?

Let's Apply it: Data2

X_1	X_2	X_3	Y
T	T	T	A
T	T	F	B
T	F	T	A
F	T	T	A
F	T	F	B

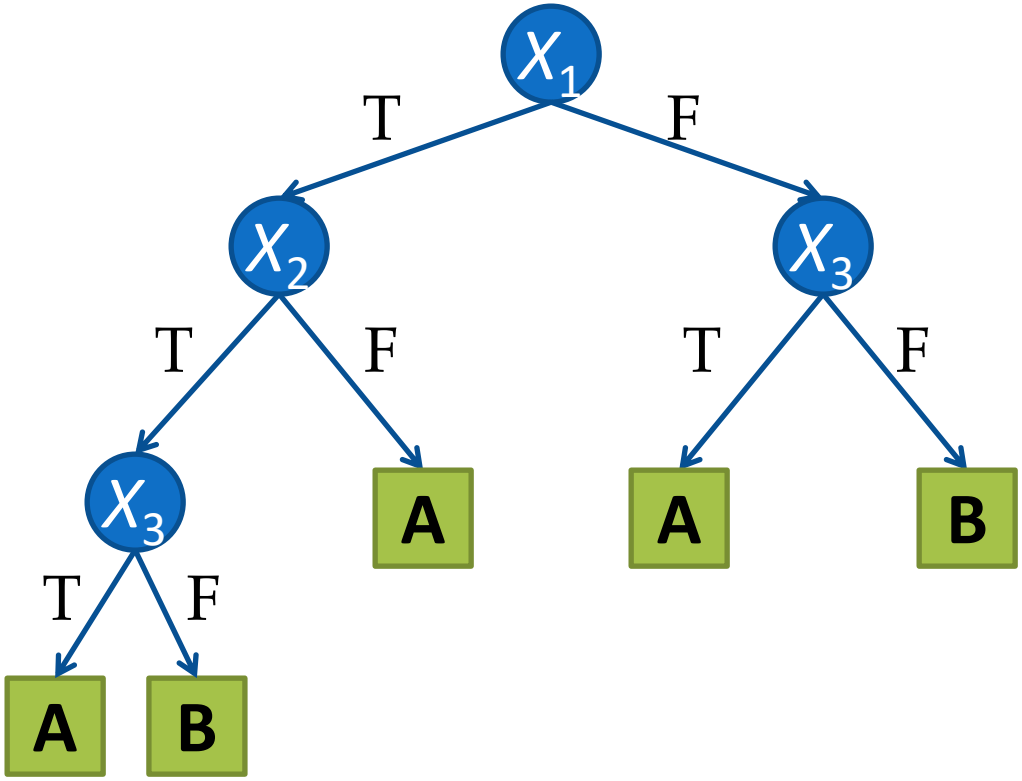
Same as Data1, except the last row is removed.

DATA2 – TREE1



What if we reorder X_2 and X_3 for the $X_1 = F$ branch?

DATA2 – TREE2



What kind of tree are we even looking for?

The Best Tree on Validation Set?

- An approach: consider all trees that are consistent with the training data and choose the one that has the best performance on the validation set
- What's wrong with this approach?

An Inductive Bias

- Given two trees that are consistent with the training data, prefer the "smaller" one
  - Why?
- The simplicity principle; simpler solutions that work are preferred over the more complex ones
  - Occam's razor; law of parsimony; "Entities should not be multiplied beyond necessity"
  - "We are to admit no more causes of natural things than such as are both true and sufficient to explain their appearances. Therefore, to the same natural effects we must, so far as possible, assign the same causes."
    - Isaac Newton
- This does not mean simplicity is a rule; it's rather a preference; an inductive bias in this case

Bad News

- Finding the smallest tree that is consistent with the training data is NP-complete
- What do we do?
  - Be greedy!
  - Start with the "best" feature at the top and then the next "best" and then the next "best"
  - Is this guaranteed to be optimal?
    - Of course not, but it's faster than exponential ☺
- Then, how do we measure how "good" a given feature is?

Purity

- Feature X_i is "better" than feature X_j if splitting the data using X_i leads to a "purer" split
- For example, given a dataset with 5 Yes and 5 No cases
  - X_1 splits the data into <2Y, 3N> and <3Y, 2N>
  - X_2 splits the data into <1Y, 4N> and <4Y, 1N>
  - X_3 splits the data into <0Y, 5N> and <5Y, 0N>
  - Then, $X_3 \succ X_2 \succ X_1$
- We understand this visually; how do we formulize it?

Purity

- A node is pure if it contains instances that belong to the same class
- Some *impurity* measures
  - Let p represent the proportion of instances that belong to one class

$$\text{Entropy} = -p \log_2 p - (1-p) \log_2 (1-p)$$

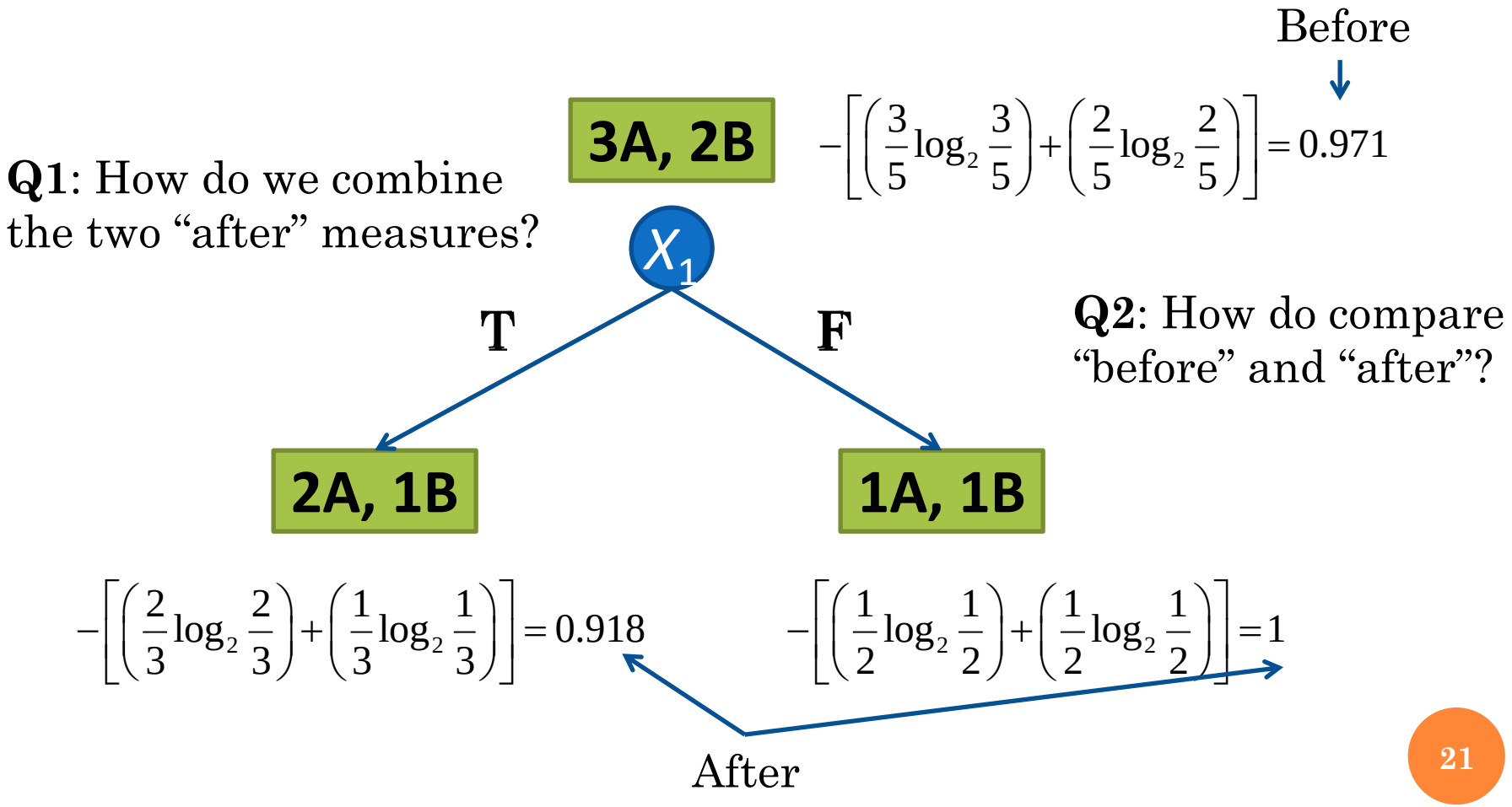
$$\text{Gini Index} = p(1-p)$$

Locally Optimal Feature

- A feature, X_i, is locally optimal if the *impurity* is *smallest after* we split using X_i
- Example from Data2
  - Before split: 3A, 2B
  - Entropy =

$$-\left[\left(\frac{3}{5}\log_2\frac{3}{5}\right)+\left(\frac{2}{5}\log_2\frac{2}{5}\right)\right]=0.971$$

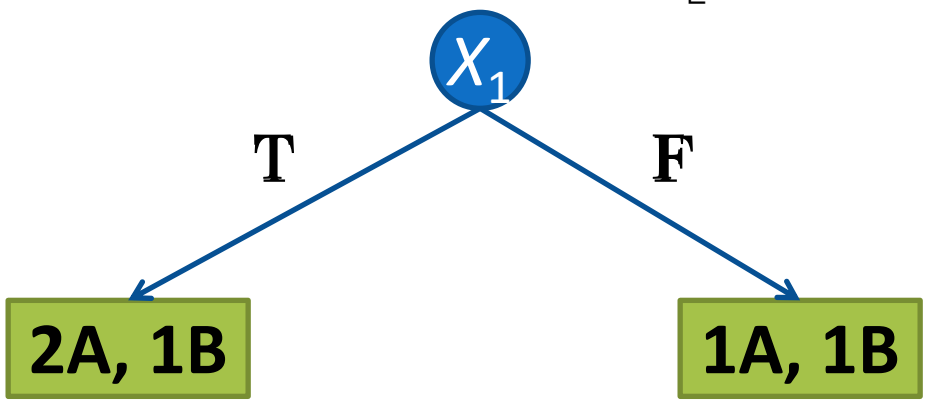
Split on X_1


Before
3A, 2B
$-\left[\left(\frac{3}{5}\log_2\frac{3}{5}\right)+\left(\frac{2}{5}\log_2\frac{2}{5}\right)\right]=0.971$
Q1: How do we combine the two "after" measures?
X_1
T
F
Q2: How do compare "before" and "after"?
2A, 1B
1A, 1B
$-\left[\left(\frac{2}{3}\log_2\frac{2}{3}\right)+\left(\frac{1}{3}\log_2\frac{1}{3}\right)\right]=0.918$
$-\left[\left(\frac{1}{2}\log_2\frac{1}{2}\right)+\left(\frac{1}{2}\log_2\frac{1}{2}\right)\right]=1$
After

Split on X_1

Before

$$-\left[\left(\frac{3}{5}\log_2\frac{3}{5}\right)+\left(\frac{2}{5}\log_2\frac{2}{5}\right)\right]=0.971$$



$$-\left[\left(\frac{2}{3}\log_2\frac{2}{3}\right)+\left(\frac{1}{3}\log_2\frac{1}{3}\right)\right]=0.918$$

$$-\left[\left(\frac{1}{2}\log_2\frac{1}{2}\right)+\left(\frac{1}{2}\log_2\frac{1}{2}\right)\right]=1$$

After

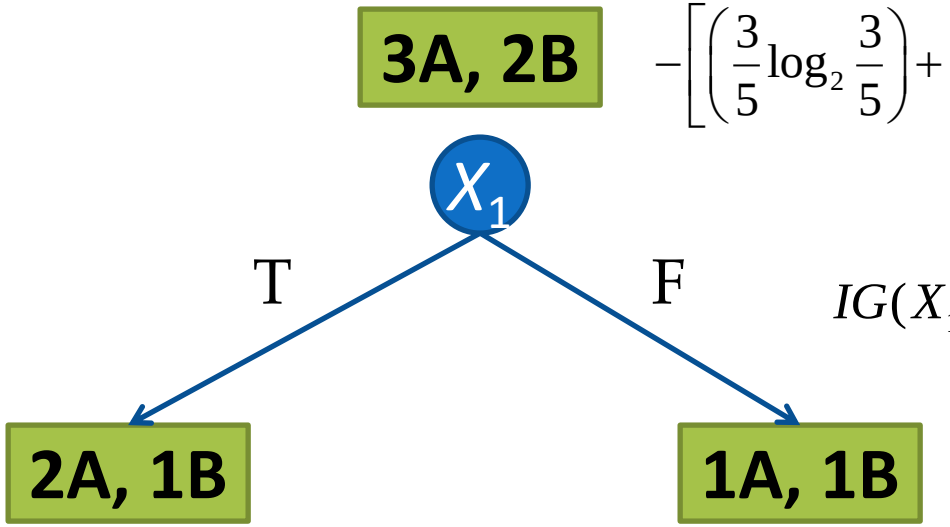
$$\begin{aligned}\text{After} &= \text{weighted average}\\ &= prob(X_1=T)\times Entropy(LeftTree)+prob(X_1=F)\times Entropy(RightTree)\\ &= \frac{3}{5}\times 0.918+\frac{2}{5}\times 1\\ &= 0.951\end{aligned}$$

Information Gain

$$InformationGain(X_i) = \text{Entropy before} - \text{Expected entropy after}$$

Split on X_1

Before



$$-\left[\left(\frac{3}{5}\log_2\frac{3}{5}\right)+\left(\frac{2}{5}\log_2\frac{2}{5}\right)\right]=0.971$$

$$IG(X_1)=0.971-0.951=0.02$$

$$-\left[\left(\frac{2}{3}\log_2\frac{2}{3}\right)+\left(\frac{1}{3}\log_2\frac{1}{3}\right)\right]=0.918$$

$$-\left[\left(\frac{1}{2}\log_2\frac{1}{2}\right)+\left(\frac{1}{2}\log_2\frac{1}{2}\right)\right]=1$$

After = weighted average

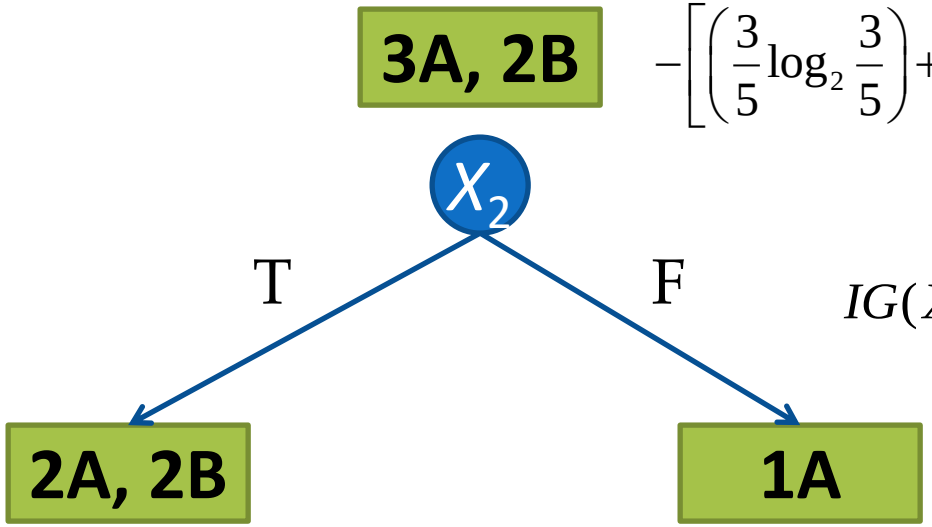
$$= prob(X_1=T)\times Entropy(LeftTree)+prob(X_1=F)\times Entropy(RightTree)$$

$$=\frac{3}{5}\times 0.918+\frac{2}{5}\times 1$$

$$=0.951$$

Split on X_2

Before



$$-\left[\left(\frac{3}{5}\log_2\frac{3}{5}\right)+\left(\frac{2}{5}\log_2\frac{2}{5}\right)\right]=0.971$$

$$IG(X_2)=0.971-0.8=0.171$$

$$-\left[\left(\frac{2}{4}\log_2\frac{2}{4}\right)+\left(\frac{2}{4}\log_2\frac{2}{4}\right)\right]=1$$

$$-\left[\left(\frac{1}{1}\log_2\frac{1}{1}\right)+\left(\frac{0}{1}\log_2\frac{0}{1}\right)\right]=0$$

After = weighted average

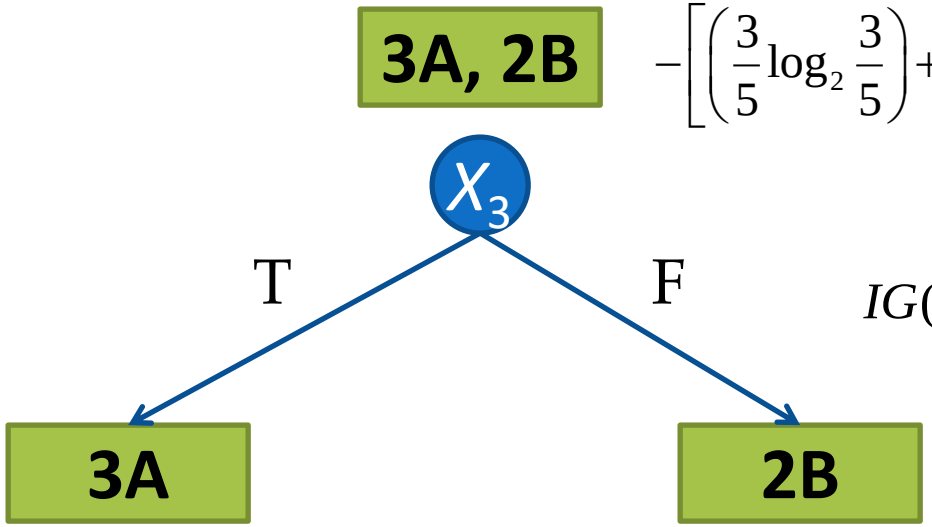
$$= prob(X_2=T)\times Entropy(LeftTree)+prob(X_2=F)\times Entropy(RightTree)$$

$$=\frac{4}{5}\times 1+\frac{1}{5}\times 0$$

$$=0.8$$

Split on X_3

Before



$$-\left[\left(\frac{3}{5}\log_2\frac{3}{5}\right)+\left(\frac{2}{5}\log_2\frac{2}{5}\right)\right]=0.971$$

$$IG(X_3)=0.971-0=0.971$$

$$-\left[\left(\frac{3}{3}\log_2\frac{3}{3}\right)+\left(\frac{0}{3}\log_2\frac{0}{3}\right)\right]=0$$

$$-\left[\left(\frac{0}{2}\log_2\frac{0}{2}\right)+\left(\frac{2}{2}\log_2\frac{2}{2}\right)\right]=0$$

After = weighted average

$$= prob(X_3=T)\times Entropy(LeftTree)+prob(X_3=F)\times Entropy(RightTree)$$

$$=\frac{3}{5}\times 0+\frac{2}{5}\times 0$$

$$=0$$

Information Gain on Data2

$$IG(X_1) = 0.971 - 0.951 = 0.02$$

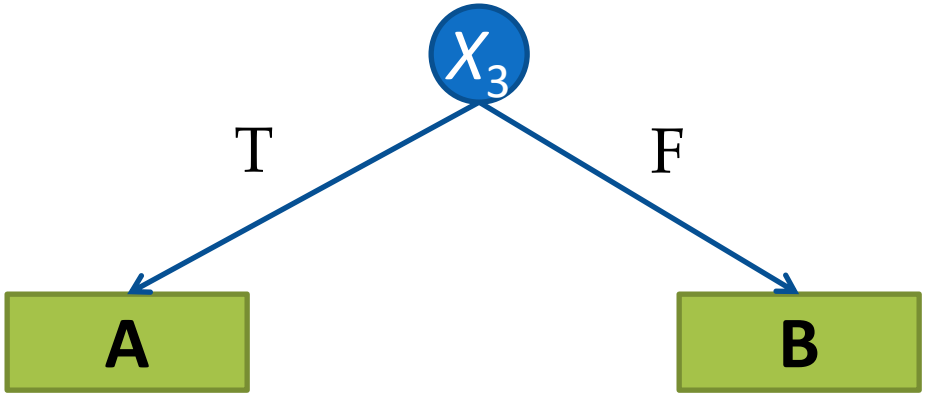
$$IG(X_2) = 0.971 - 0.8 = 0.171$$

$$IG(X_3) = 0.971 - 0 = 0.971$$

A DT Algorithm (ID3)

- Start with the empty tree
- At each iteration
  - Pick the locally optimal feature and split on it
  - Stop when all leaves are pure (or no more features are left to split)

Let's Apply it: Data2-Tree3



- Empirical error?
- Prediction power?
- Size?

Example

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Features With Many Categorical Values

- Let
  - A node have m data points
  - Feature F_i have n possible values
- On average, each split by each value of F_i will have m/n data points
- When n is large, m/n data points are expected to more likely to be pure
- For e.g., take the extreme case where $m \leq n$
- Hence, having a large n by itself is an advantage
- Solution: instead of Information Gain, use Gain Ratio
  - Gain Ratio = IG / (Entropy of the Split Proportions)

Continuous Features

- Let F_i be a continuous feature, such as temperature, age, etc.
- Assume that you would like to create a binary split, asking $F_i >? \delta$
- How many possible δ values do you need to consider?
- How do you find the best δ efficiently?

Hypotheses represented by trees?

- Conjunctions?
- Disjunctions?
- Negations?
- What is the inductive bias of the ID3 algorithm?

Overfitting

- Given a hypothesis h
  - $error_{train}(h)$: Error of h on the train set
  - $error_{distribution}(h)$: Error of h on the entire distribution of the data
- $h \in H$ **overfits** the train set if there is an $h' \in H$ such that:
  - $error_{train}(h) < error_{train}(h')$ and
  - $error_{distribution}(h) > error_{distribution}(h')$

When to stop growing the tree?

- Technically
  - Stop when the leaf is pure
  - Otherwise, stop when no more attributes are left to test
- If there are errors in the training data
  - The tree can end up being larger than it needs to be
- Remember that we want a small tree; larger trees tend to overfit the training data
- Two solutions:
  - Stop early based on a criteria
  - Post-prune the tree

Early Stopping (pre-Pruning)

- Stop growing a branch based on some fixed criteria
- Example criteria:
  - Stop when the number of instances in a leaf gets below a threshold
  - Stop when the information gain of the remaining attributes gets below a threshold
  - Stop when the entropy at a leaf is below a threshold
  - Stop when the depth of the tree reaches a threshold
  - (and so on)

Post-Pruning using validation data

- Keep a separate data for validation
- First, grow the full tree using the training data
- Then, prune a node (and those below it) as long as pruning improves performance on the validation data

Scikit-Learn – Decision Trees

- http://scikit-learn.org/stable/modules/tree.html
- http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html