Decision Trees

Berrin Yanıkoğlu

- In these slides;
  - you will strengthen your understanding of the introductory concepts of machine learning
  - learn about the Decision tree approach which is one of the fundamental approaches in machine learning
    - Decision trees are also the basis of a new method called “random forest” which we will see towards the end of the course.

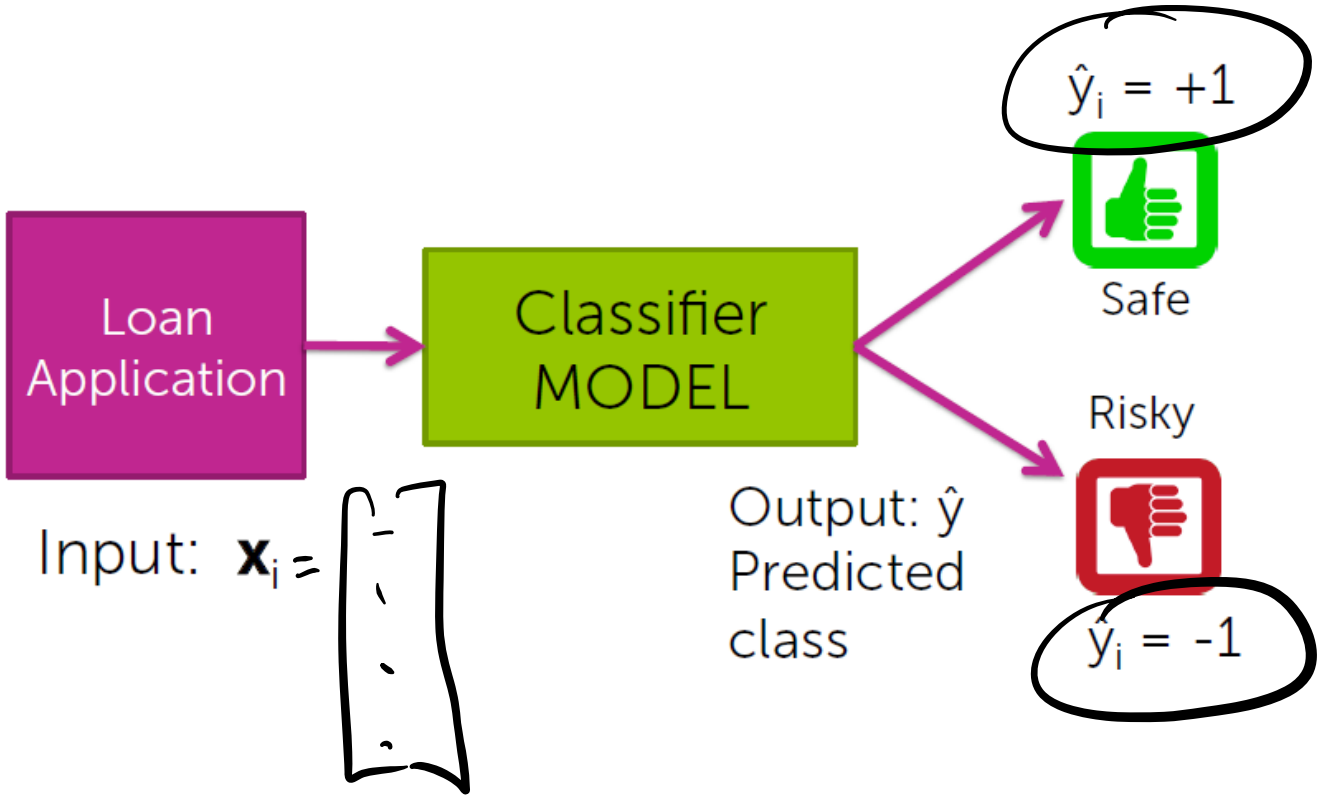
Decision Trees

- One of the most widely used and practical methods for inductive inference
- Can be used for classification or regression problems
- Simple approach, but **foundation** of some state-of-art ML methods

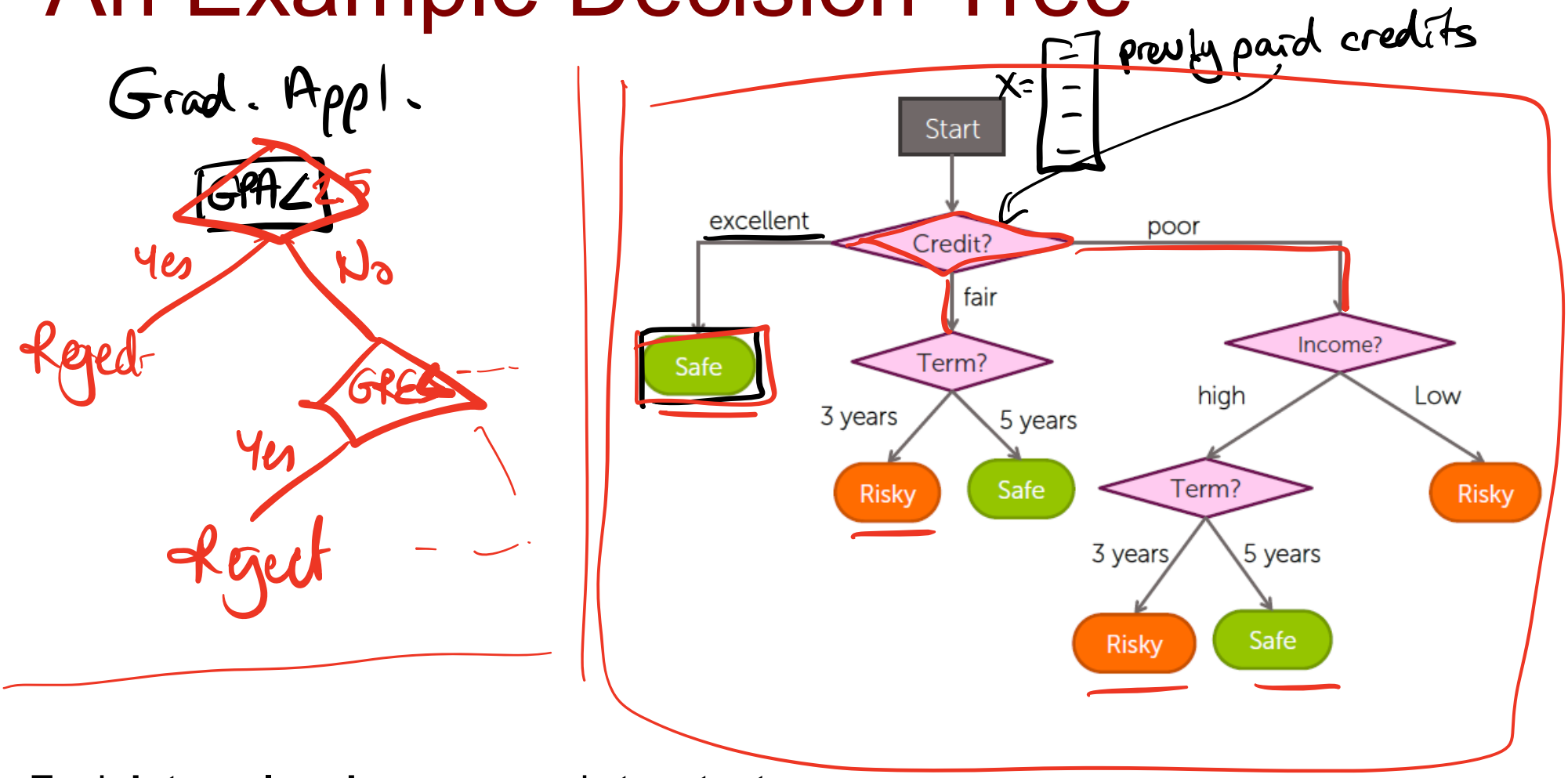
Thanks to Oznur Tastan

Credit Applicant Example

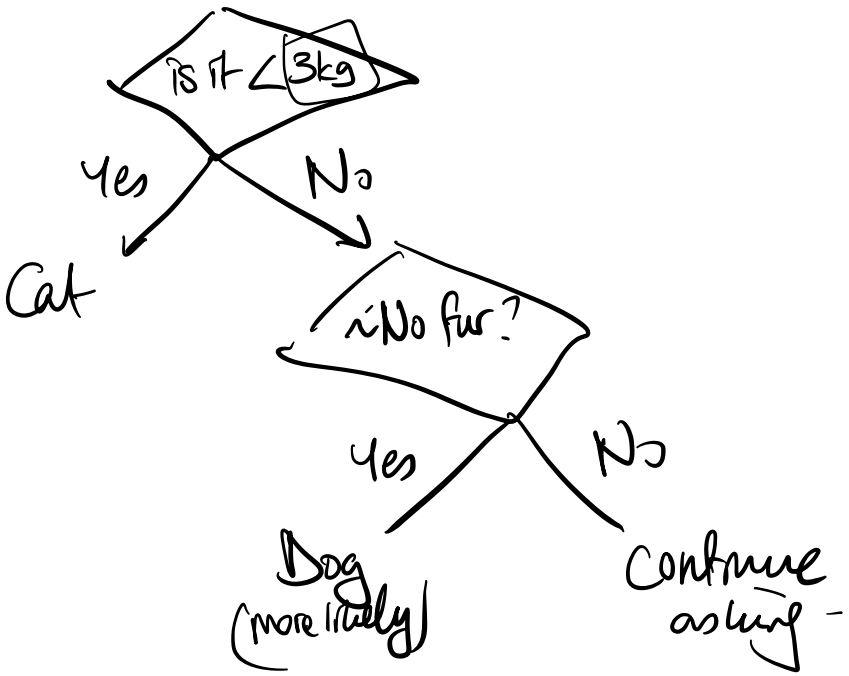
Credit ★★★★	Did I pay previous loans on time?
Income ★★★	What's my income?
Term ★★★★★	How soon do I need to pay the loan?
Personal Info ★★★	Age, reason for the loan, marital status,...



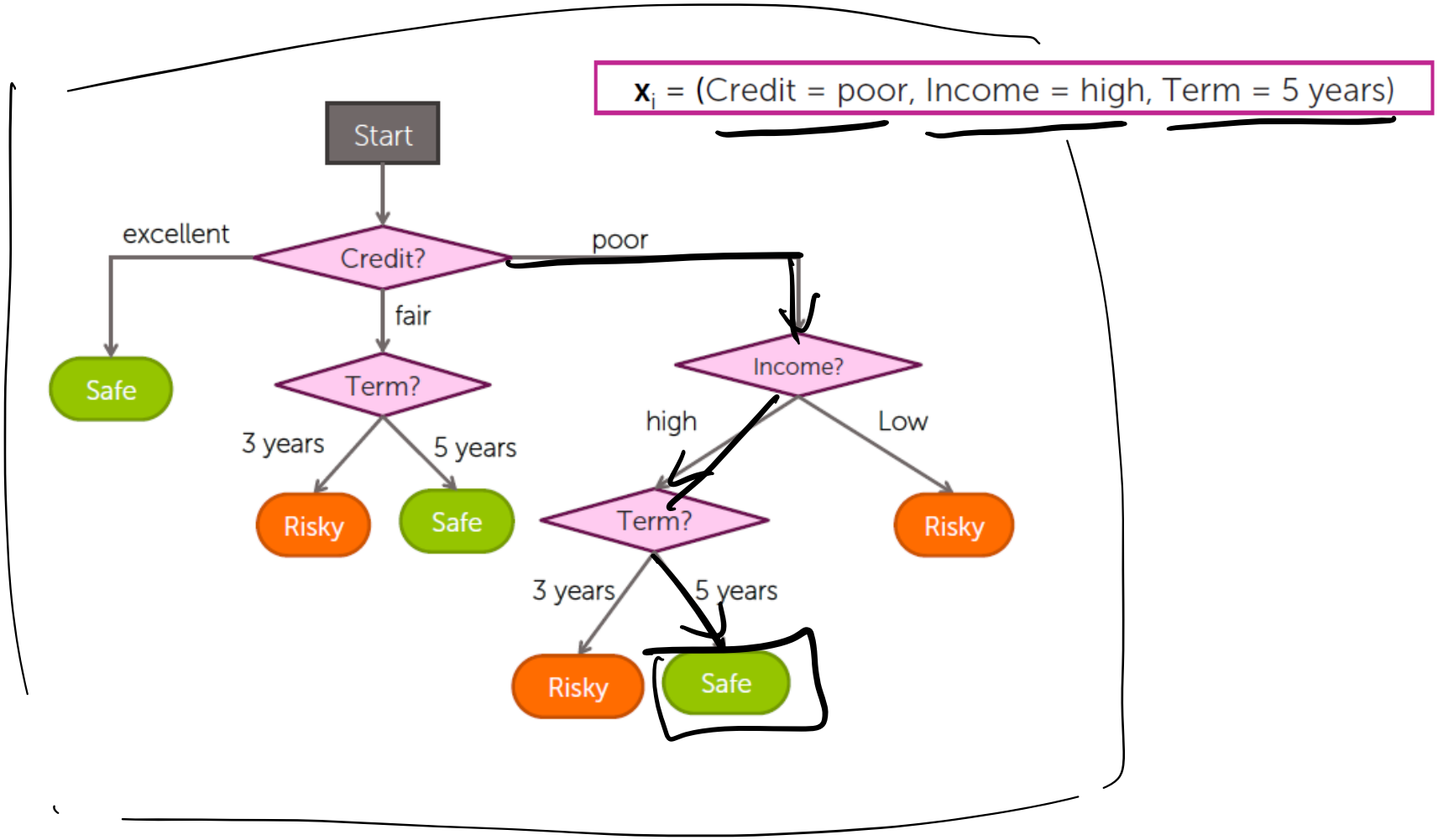
An Example Decision Tree



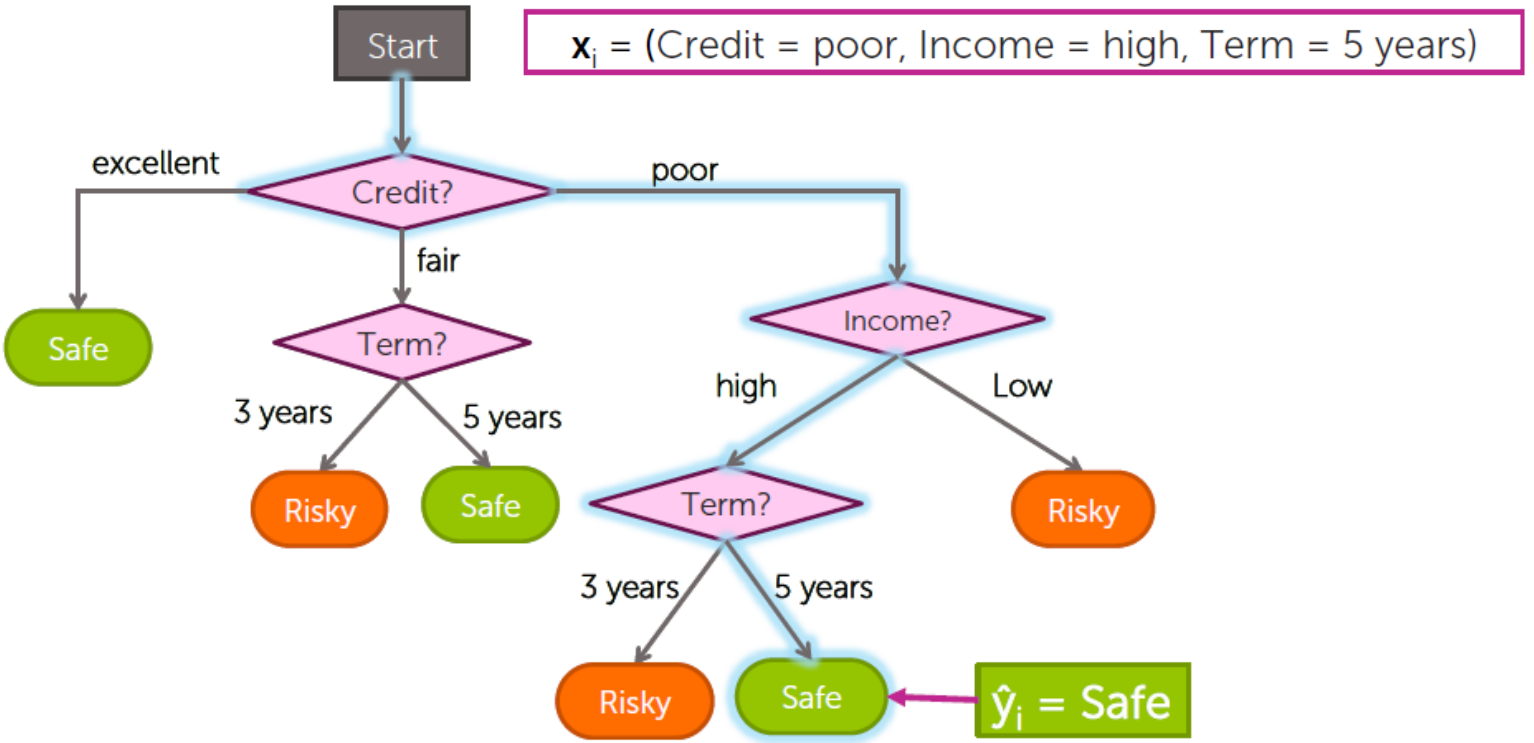
Each **internal node** corresponds to a **test**
Each **branch** corresponds to a **output of the test**
Each **leaf node** represents a classification label


Is it < 3kg
Yes
No
Cat
No fur?
Yes
No
Dog
(more likely)
Continue
asking

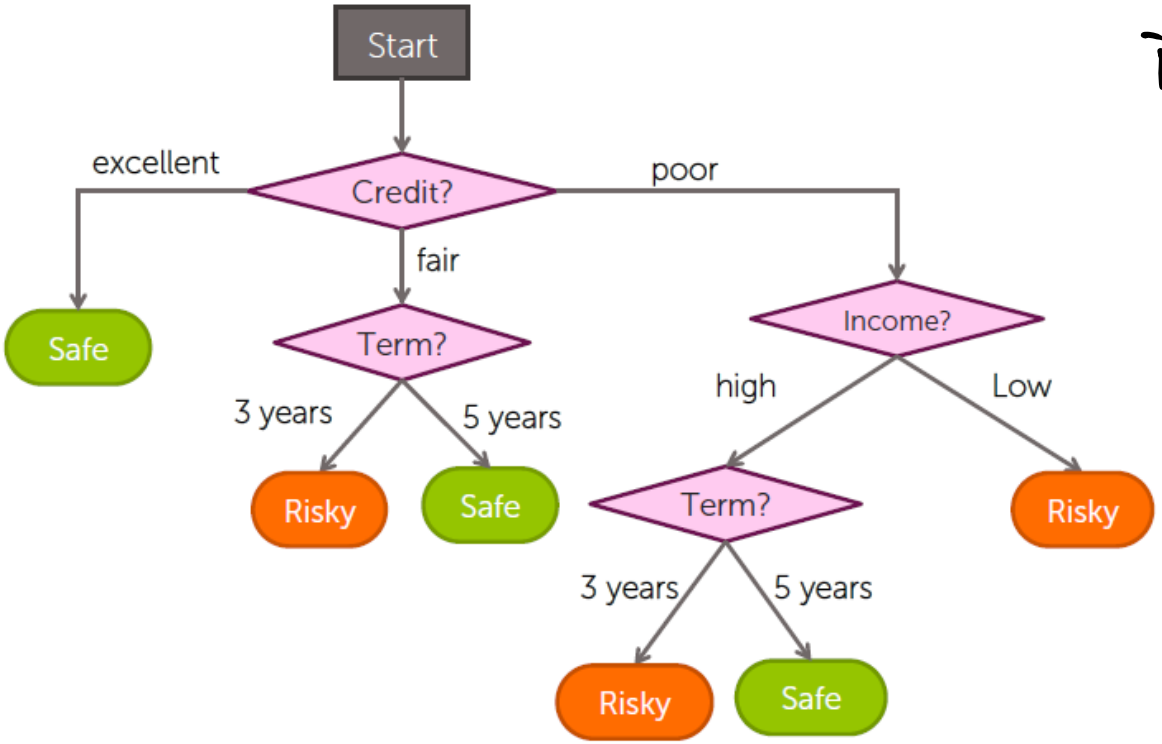
Using a Decision Tree for prediction



Using a Decision Tree for Prediction



- Decision trees learns a set of rules from the training data
  - A learned tree can be represented as a set of **if-then rules**
  - A decision tree is an interpretable & explainable model



if (credit = excellent)
or
(credit = fair
&
term = 5yr)
or
(_____)
→ Safe

Decision Tree Learning (a.k.a DT Induction)

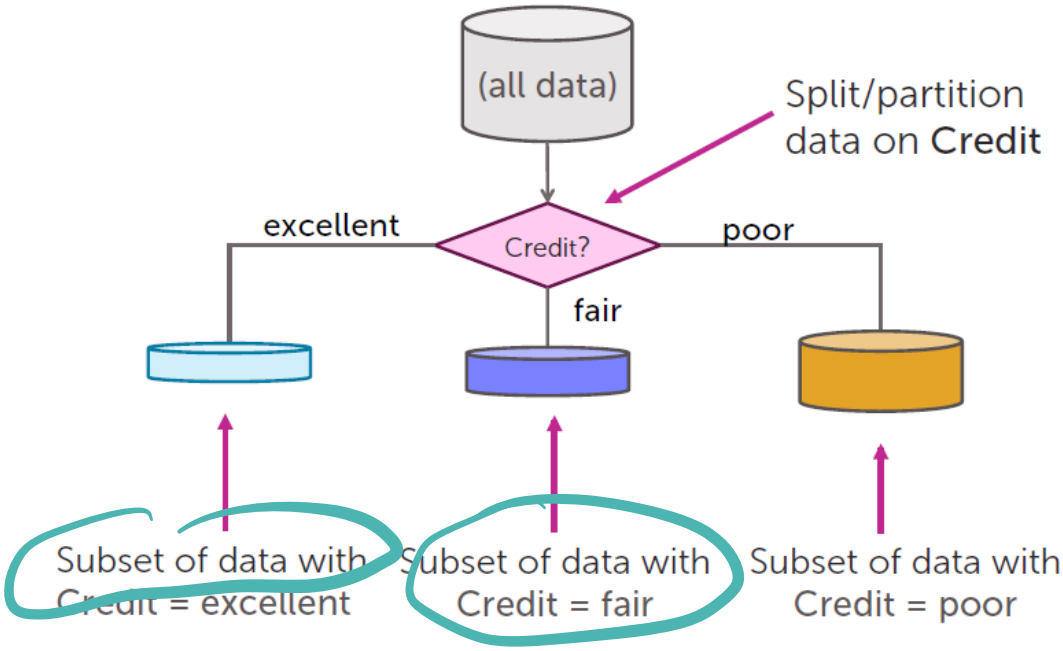
Decision tree learning algorithm

- For a given training set, there are many trees that code it without any error

- Finding the smallest tree is NP-complete (Quinlan 1986), hence we are forced to use some (local) search algorithm to find reasonable solutions

ID3 Decision Tree Learning Algorithm

- Learning is **greedy**; find the best split **recursively** (Breiman et al, 1984; Quinlan, 1986, 1993)
- At each internal node:
  - **Choose an attribute** to question about
  - According to the answers given, the problem is **split into sub-problems**.
  - Continue **recursively**.




: cat
: dog
large
Wet
Very light
[1-2kg]
CAT
Normal
[2,4]
Fur
Large
[4,+[
Dog

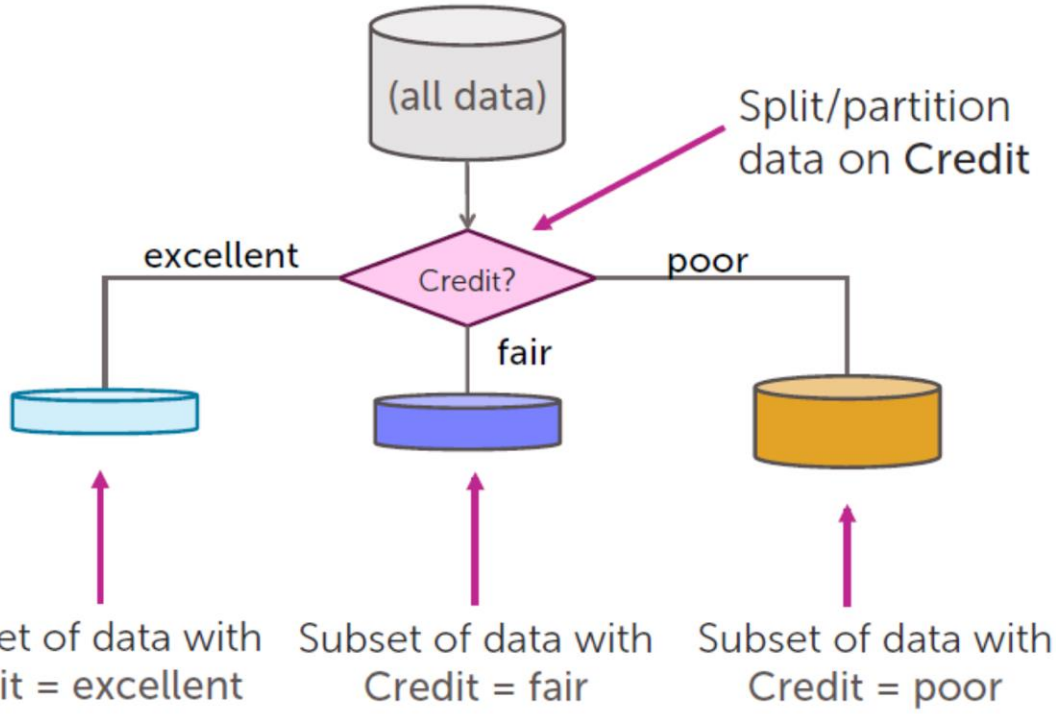
N=7 samples

Top-Down Induction of Decision Trees

Main loop:

1. $A \leftarrow$ the "best" decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A, create new descendant of *node*
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

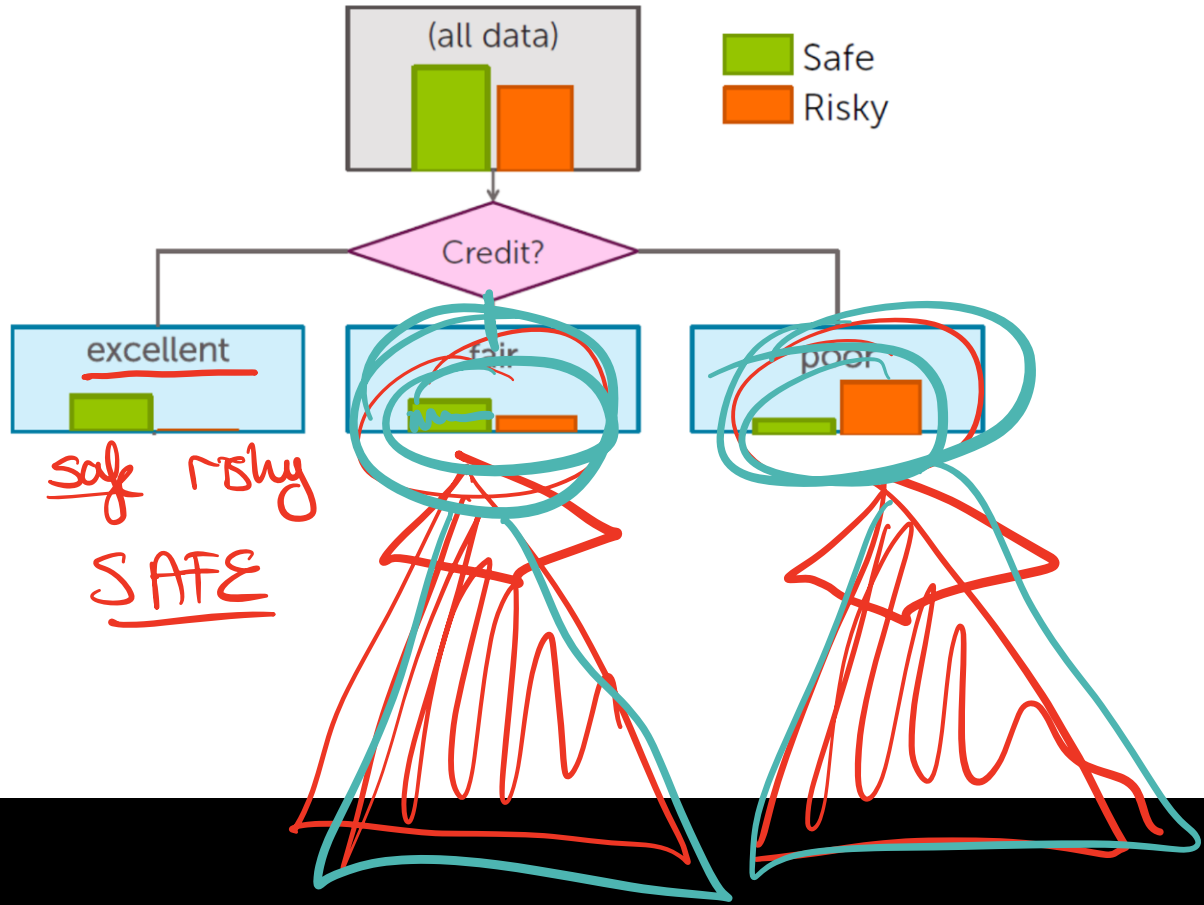
1. $A \leftarrow$ the "best" decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A, create new descendant of *node*



2. Assign A as decision attribute for *node*

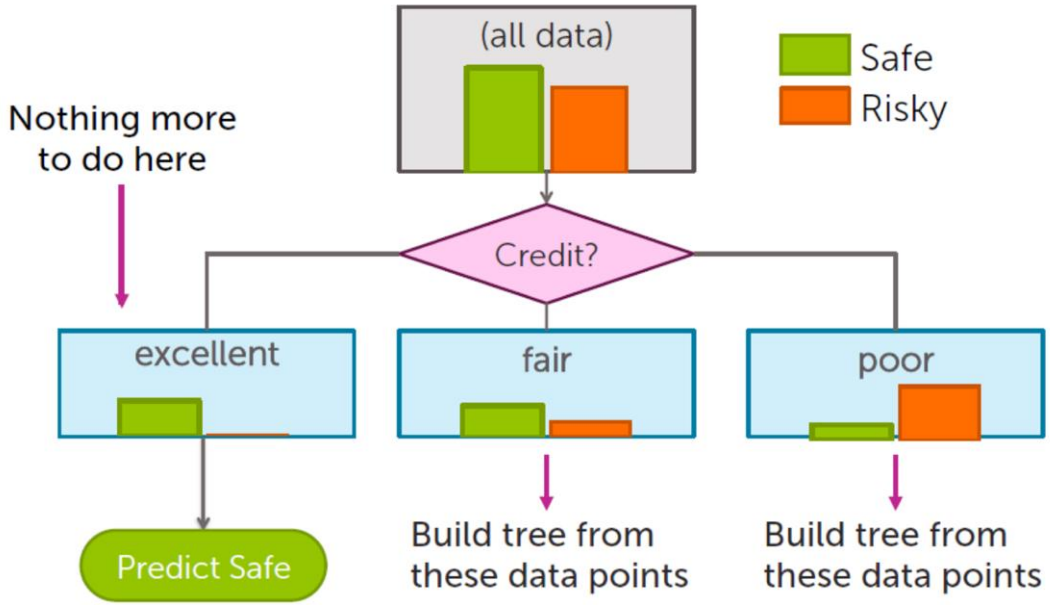
3. For each value of A, create new descendant of *node*

4. Sort training examples to leaf nodes



4. Sort training examples to leaf nodes

5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes



ID3 Learning Algorithm

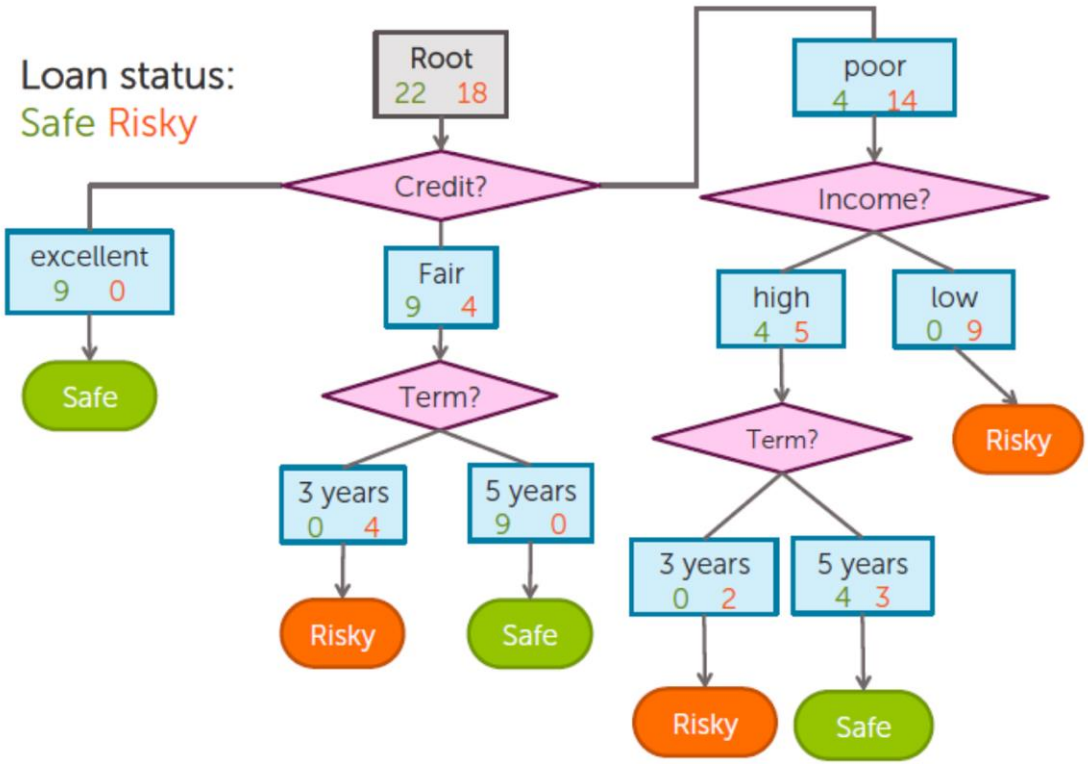
Until stopped:

- Select one of the unused attributes to partition the remaining examples at each non-terminal node using only the training samples associated with that node

Stopping criteria:

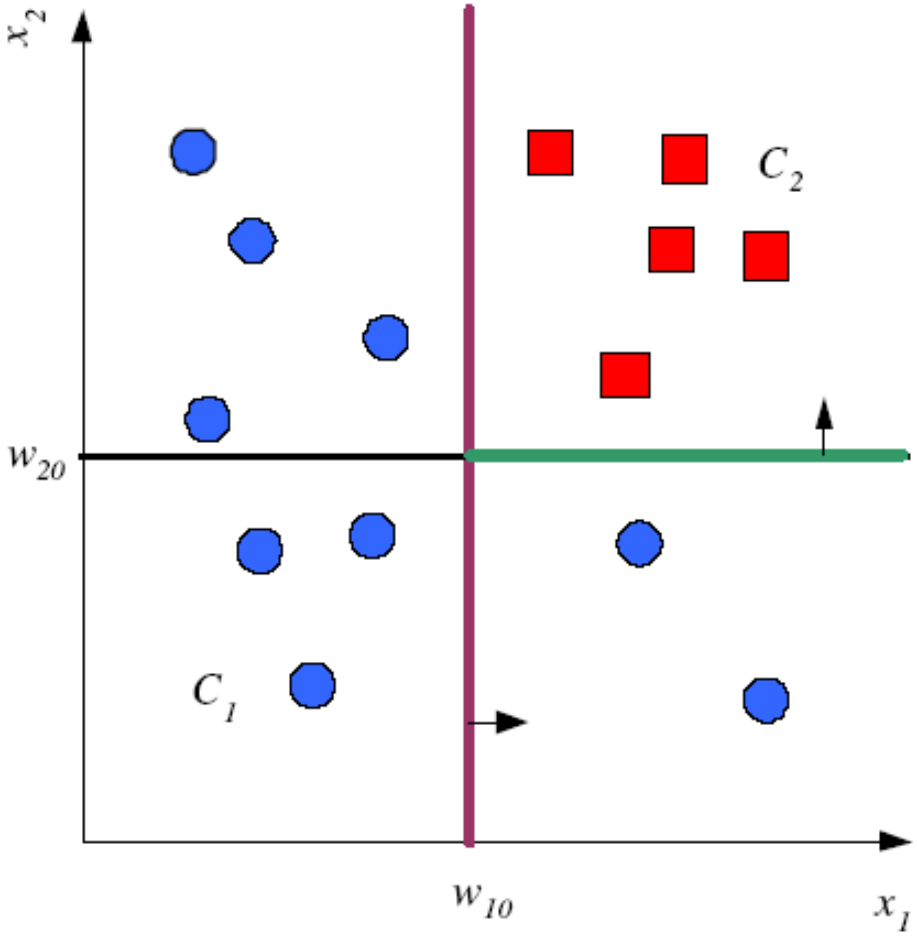
- each leaf-node contains examples of one type
- algorithm ran out of attributes
- …

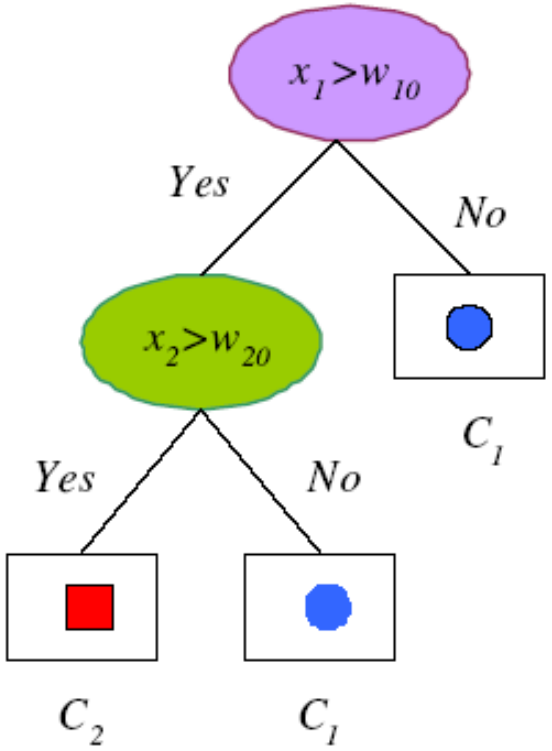
- If training examples are not perfectly classified, but there is no more attribute to split into, then STOP
  - Assign the majority class



Decision Boundaries w. **Univariate** Trees

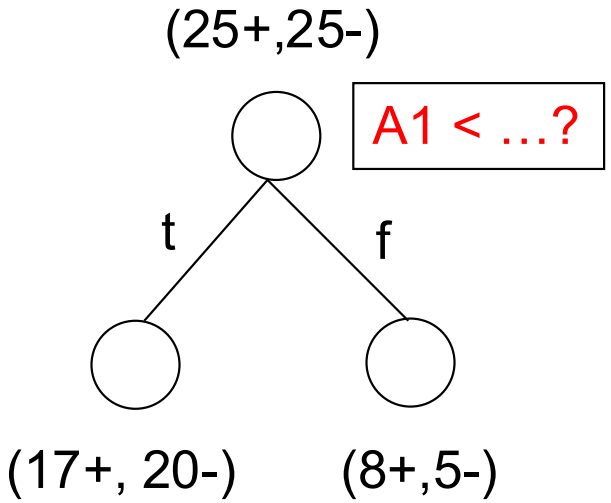
Skip for now

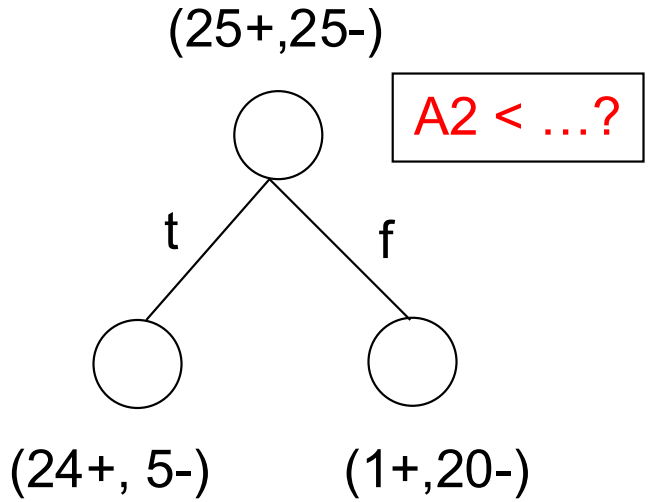




Entropy

What is a better attribute?





- Assume you have 25 + and 25 – samples to learn the decision tree from.
- You consider features A1 and A2 for split.
- With A1, you have 17+ and 20- samples following the t branch.
- With A2, you have 24+ and 5- samples following the t branch.
- ...

Entropy

- **Measure of uncertainty**
- Entropy of a random variable X is the expected number of bits to resolve the uncertainty about the value that X may take:

$$H[X] = -\sum_{X=x} p(x) \log_2 p(x)$$

- Important quantity in
  - coding theory
  - statistical physics
  - machine learning
  - ...

Entropy of a **Binary** Random Variable

- High school form example with gender field

- Example: Consider a binary random variable X s.t. Pr{X = 0} = 0.1

  Entropy(X) =

$$H[X] = -\sum_{X=x} p(x) \log_2 p(x)$$

Selecting the Next Attribute

- We will use the **remaining entropy** as our measure to prefer one attribute over another.

- In summary, we will consider
  - the entropy over the distribution of samples falling under each leaf node and
  - we will take a weighted average of that entropy – weighted by the proportion of samples falling under that leaf.

- We will then choose the attribute that brings us the biggest information gain, or equivalently, results in a tree with the lower weighted entropy.

Information Gain

$$Gain(S, A) = \text{expected reduction in entropy due to sorting on } A$$

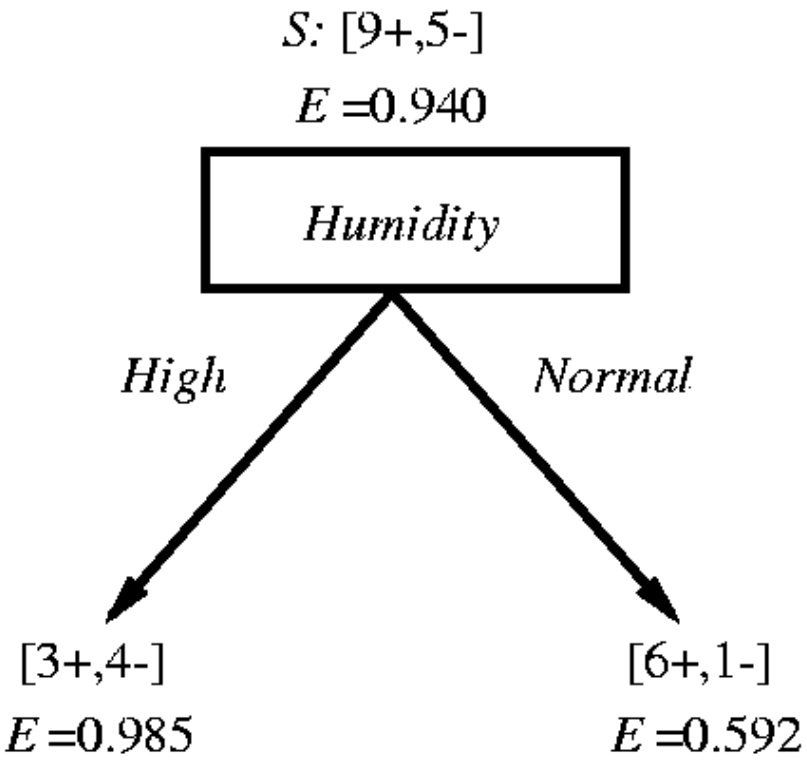
$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

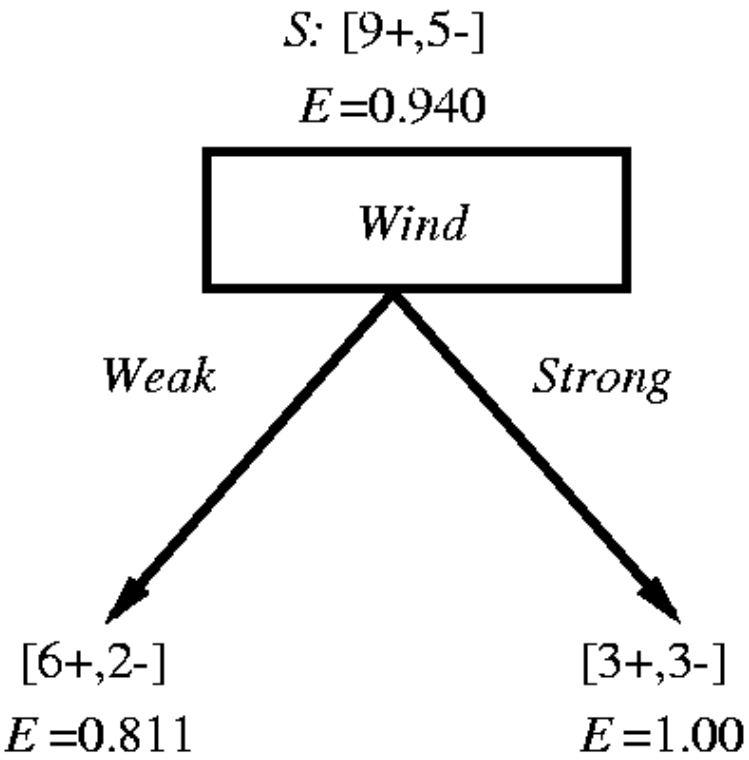
Please note: Gain is [what you started with – Remaining entropy].
So we can simply choose the tree with the **smallest remaining entropy**!

Training Examples

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

Which attribute is the best classifier?





Gain (S, Humidity)

= .940 - (7/14).985 - (7/14).592
= .151

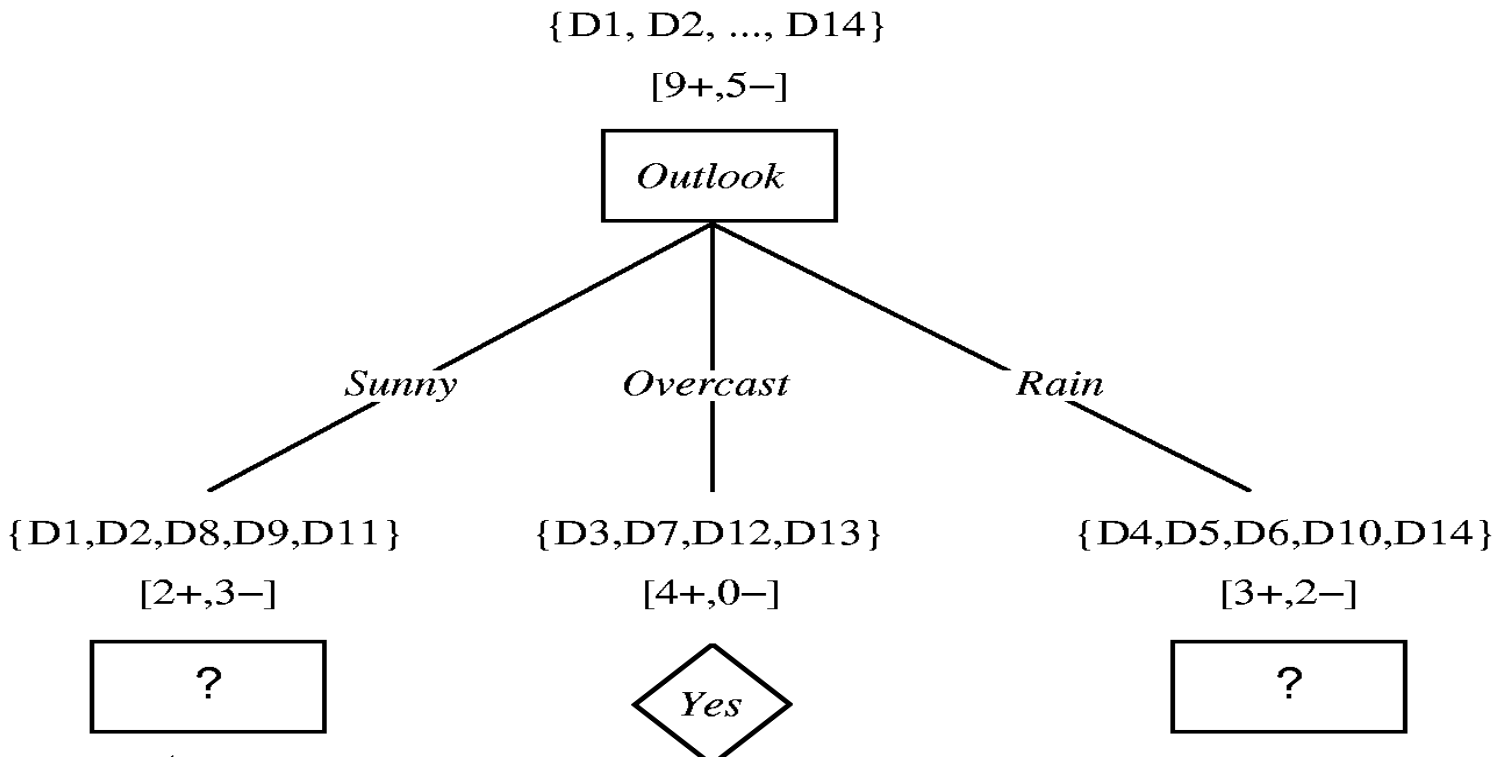
Gain (S, Wind)

= .940 - (8/14).811 - (6/14)1.0
= .048

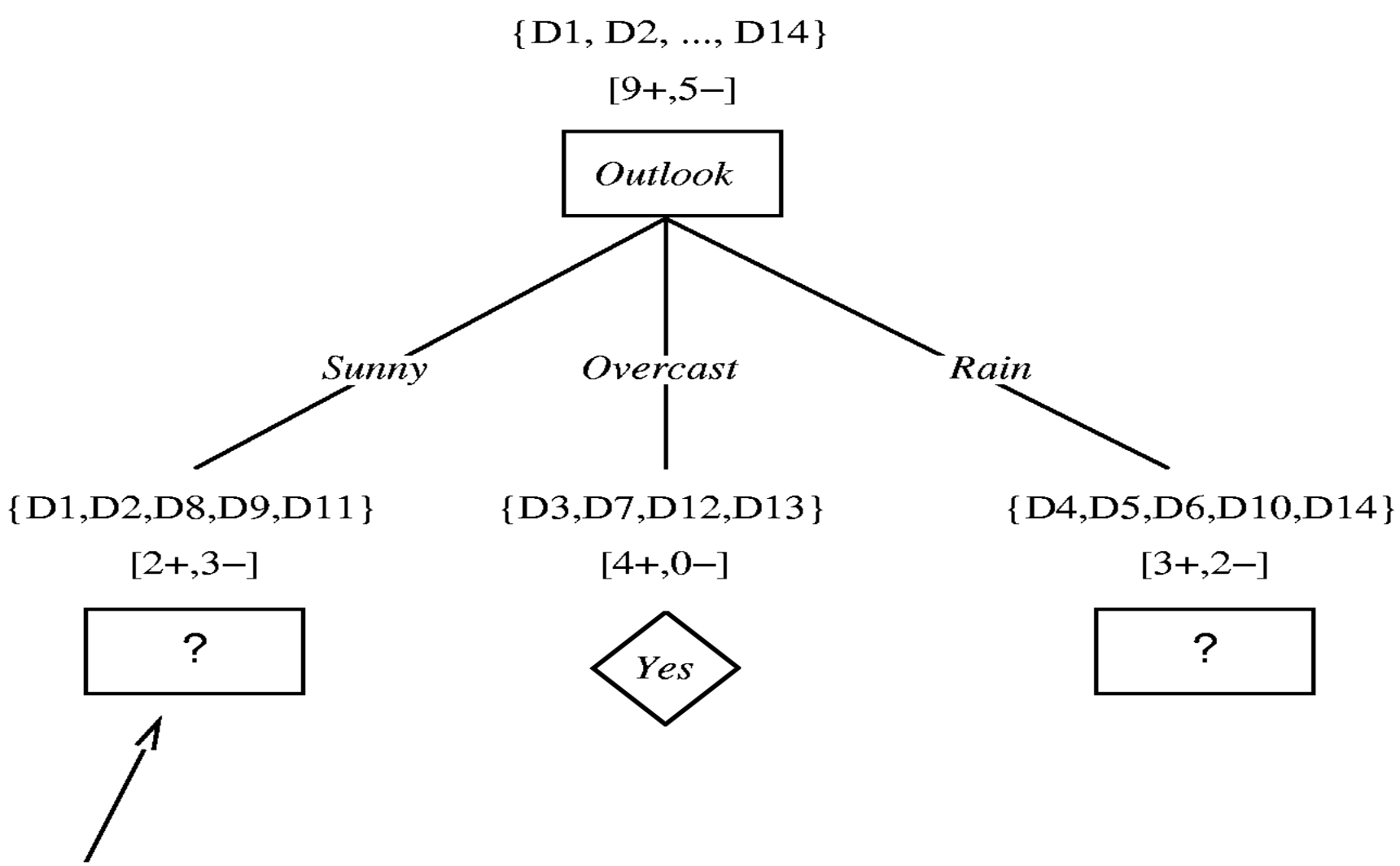
We would select the Humidity attribute to split the root node as it has a higher Information Gain.

Selecting the Next Attribute

- Computing the information gain for each attribute, we selected the *Outlook* attribute as the first test, resulting in the following partially learned tree:



- We can repeat the same process recursively, until Stopping conditions are satisfied.



Which attribute should be tested here?

S_{sunny} = {D1,D2,D8,D9,D11}

Gain (S_{sunny}, *Humidity*) = .970 − (3/5) 0.0 − (2/5) 0.0 = .970

Gain (S_{sunny}, *Temperature*) = .970 − (2/5) 0.0 − (2/5) 1.0 − (1/5) 0.0 = .570

Gain (S_{sunny}, *Wind*) = .970 − (2/5) 1.0 − (3/5) .918 = .019

Other measures of impurity

- Entropy is not the **only** measure of impurity. If a function satisfies certain criteria, it can be used as a measure of impurity.
  - **Misclassification Rate:**

$$\text{Classification error}(t) = 1 - \max_i p(i \mid t)$$

  - **Gini index**
  - **…**

Generalization in DTs

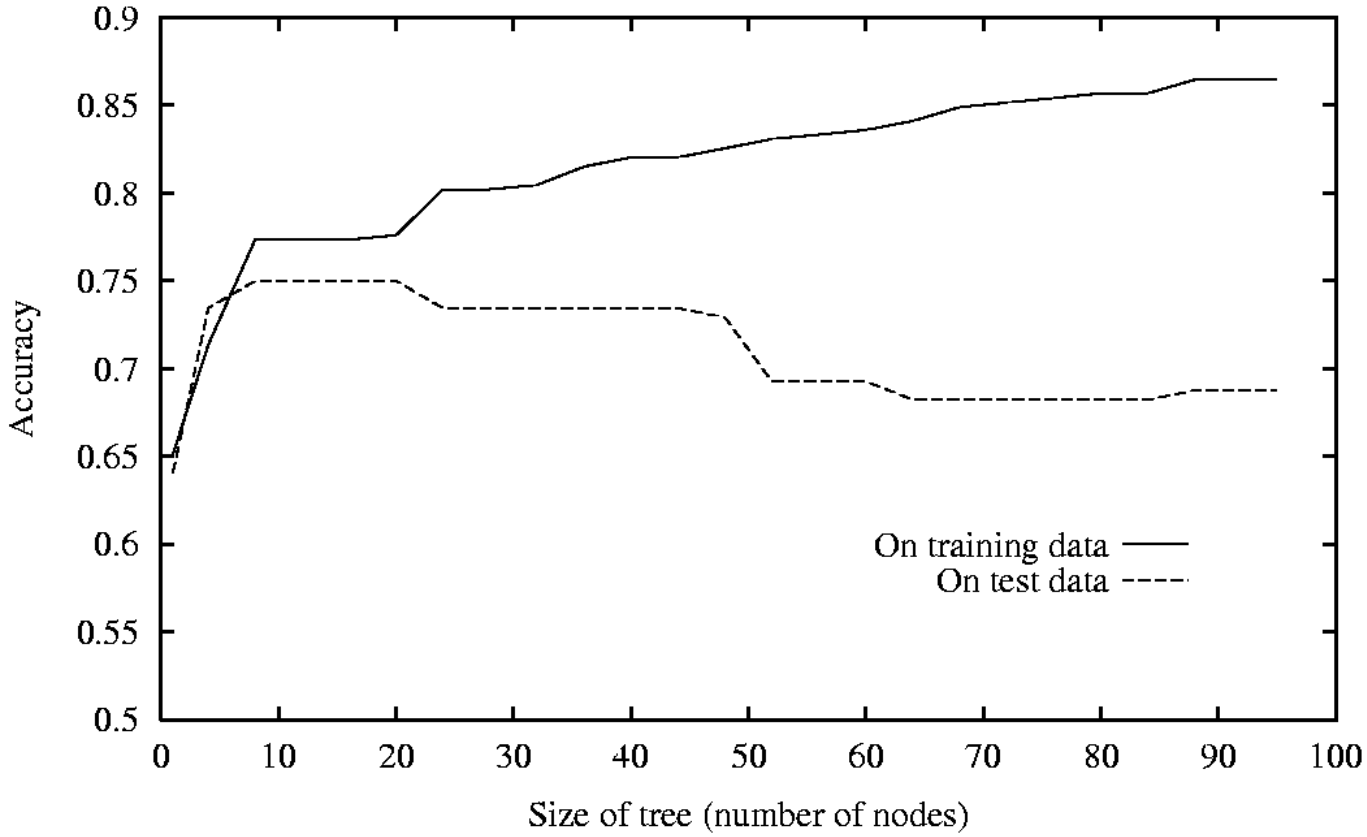
When to stop growing the tree?

- Typical stopping conditions:
  - **Stop if all instances belong to the same class**
  - Stop if all the attribute values are the same

    (=no more attribute to split, will happen with binary or categorical attributes)

- But **if you make a decision tree deep enough,** it can often do a perfect job of predicting class labels on training set.
  - Is this a good thing?

- A typical behaviour in machine learning is that as the complexity (here the size of the tree) of the model increases, it has more chance to better represent the training data.
- But then, often the generalization performance suffer.



Addresing Overfitting

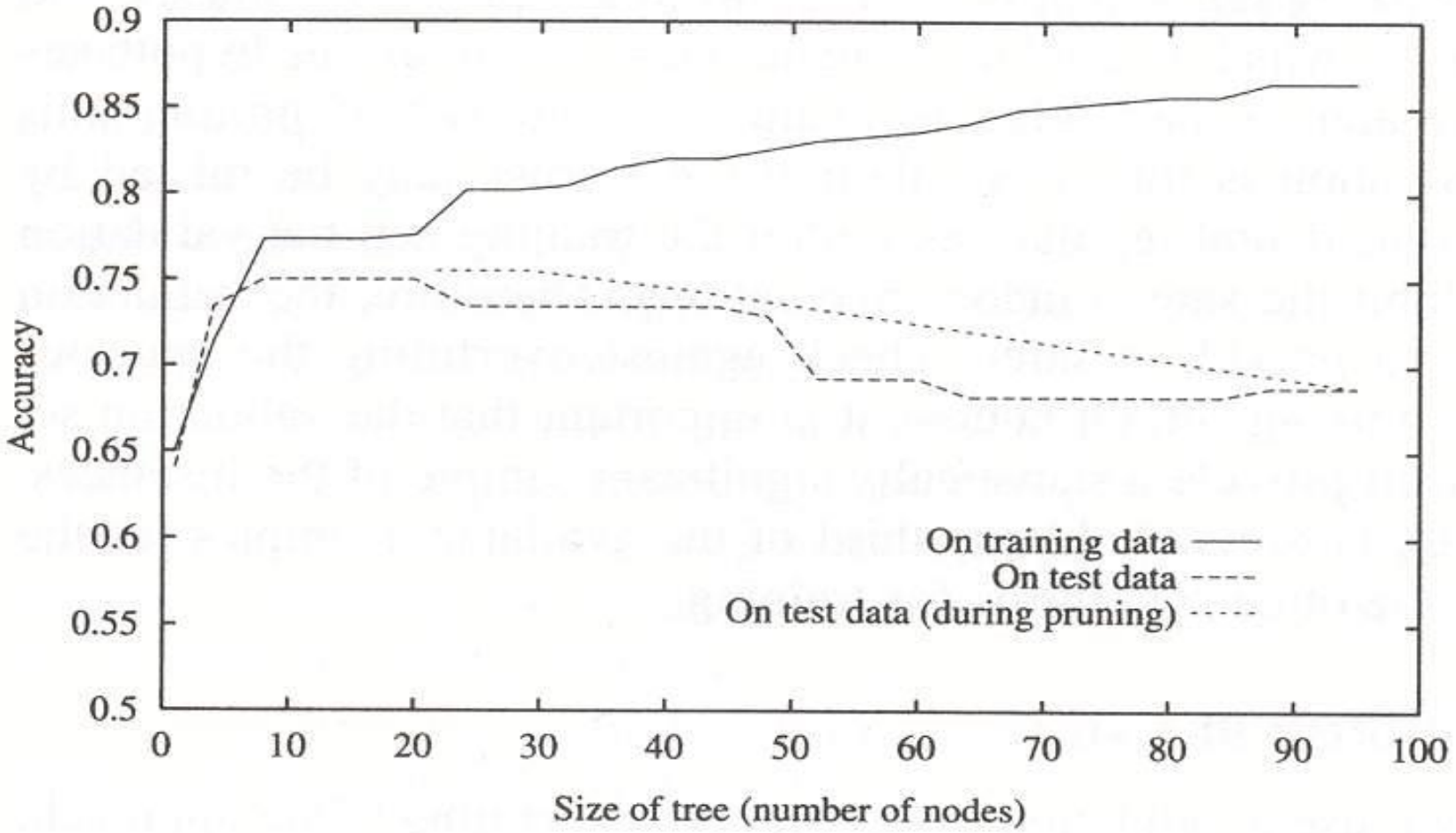
- Making decision trees simpler
- Two approaches:
  - Early stopping
    - stop growing the tree before it becomes too complex
  - Pruning
    - first grow the tree and then simplify it
- Both early stopping and pruning uses validation data

Early Stopping

- Set a maximum depth for the tree
  - Stop growing the tree when depth reaches the maximum depth
- Set a minimum number of instances in a node
  - If number of examples in a node becomes lower then this minimum number, then stop growing
  - When the number of instances are low in a node, decision in that leaf would be prone to noisy data and overfitting

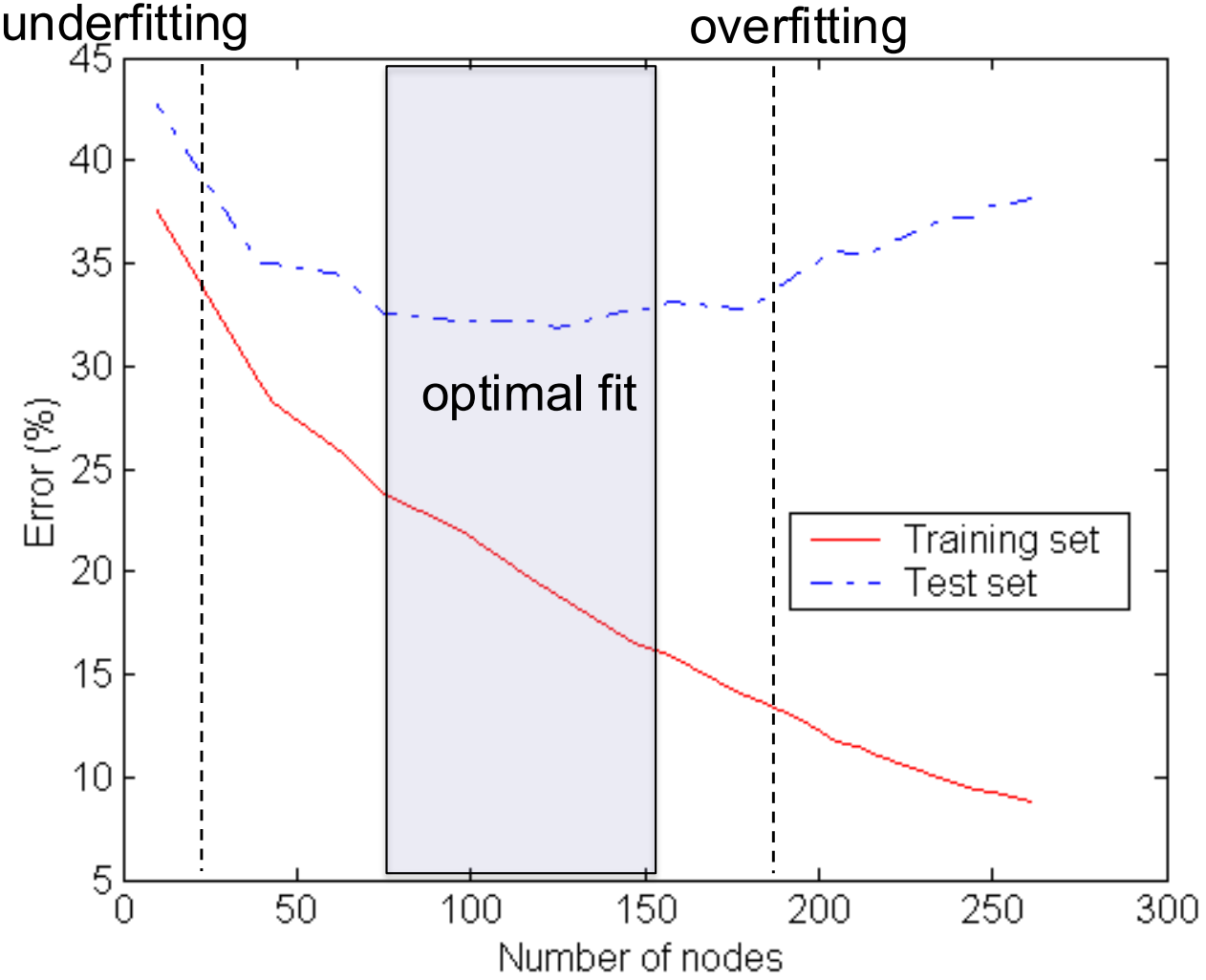
Reduced-Error Pruning (Quinlan 1987)

- Split data into *training* and *validation* set
- Starting with the leaves, do until further pruning is harmful:
  - 1. Evaluate impact of pruning each possible node (plus those below it) on the *validation* set
    - In pruning a node, we return the majority decision before the considered split
  - 2. Greedily remove the one that most improves *validation* set accuracy
- Produces smallest version of the (most accurate) tree
- What if data is limited?
  - We would not want to separate a validation set.
  - Use cross-validation or statistical approaches to evaluate impact



Any model selection should be done over the validation set, so you shd take the “test set” here to mean validation set.

Underfitting and Overfitting



Underfitting: when model is too simple, both training and test errors are large

Overfitting: when model is too complex it models the details of the training set and fails on the test set

Regression with Decision Trees

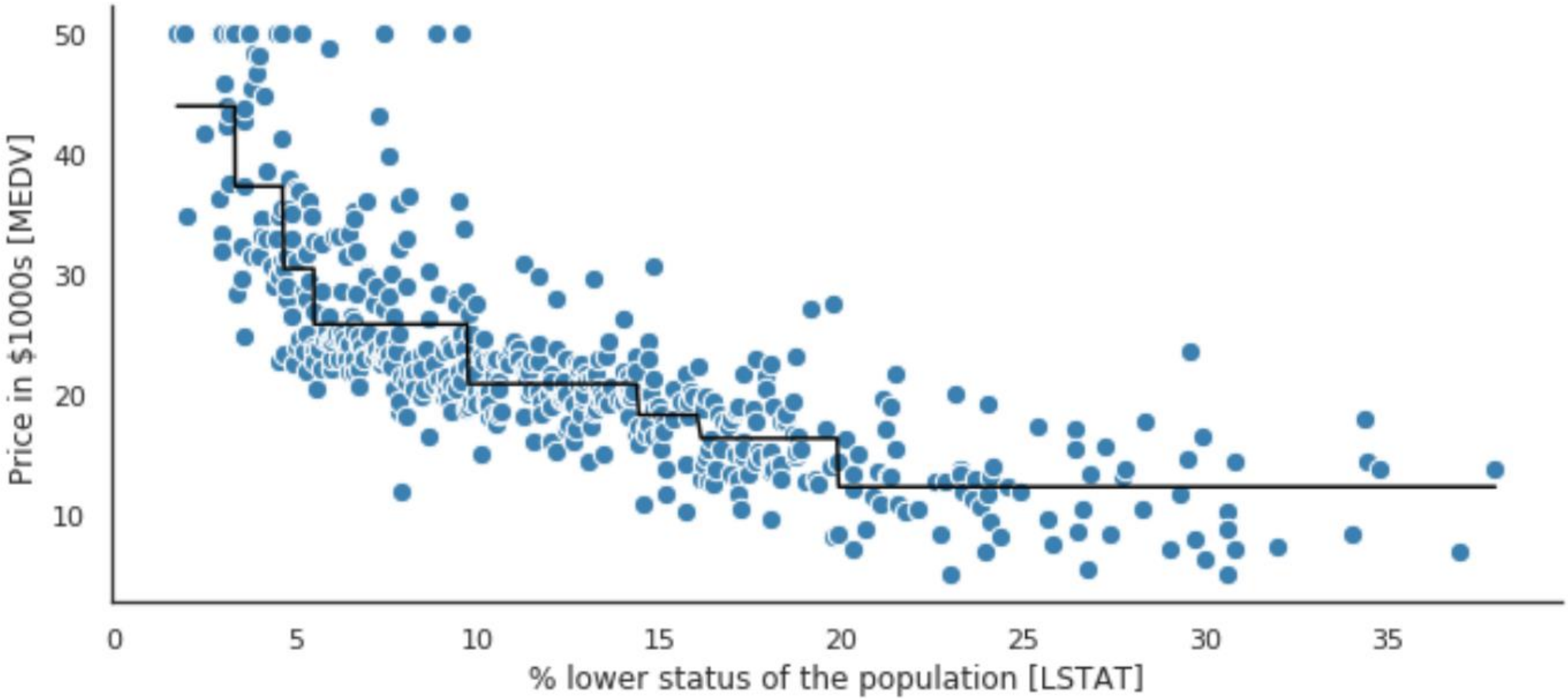
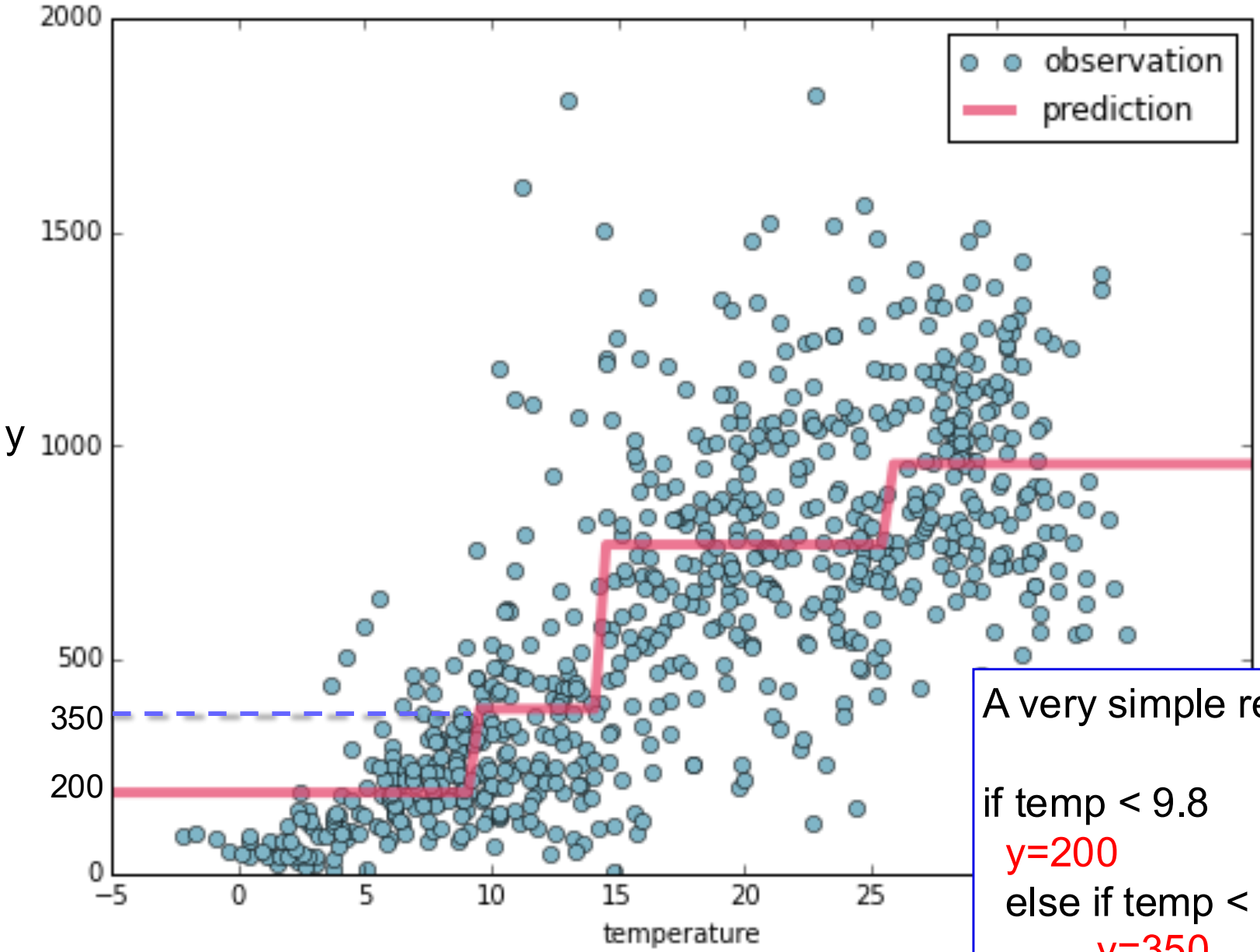


Image credit: https://towardsdatascience.com/https-medium-com-lorrli-classification-and-regression-analysis-with-decision-trees-c43cdbc58054

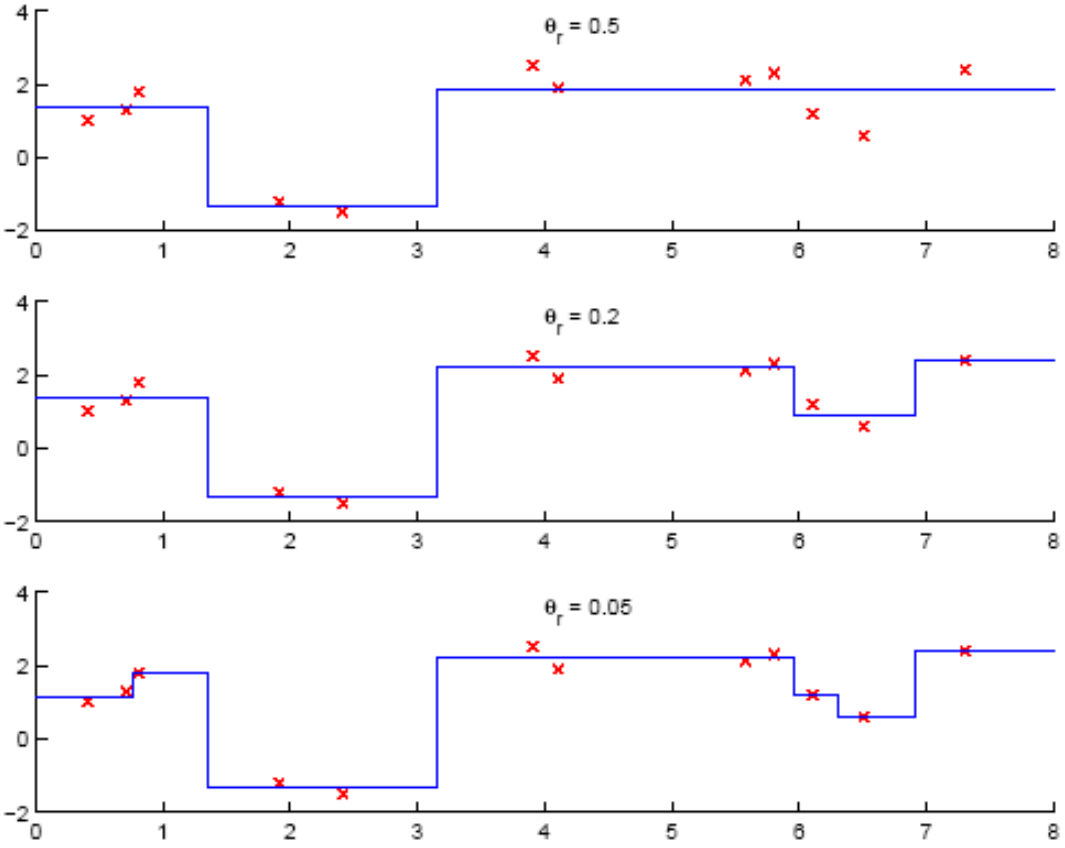
Decision Tree Regression

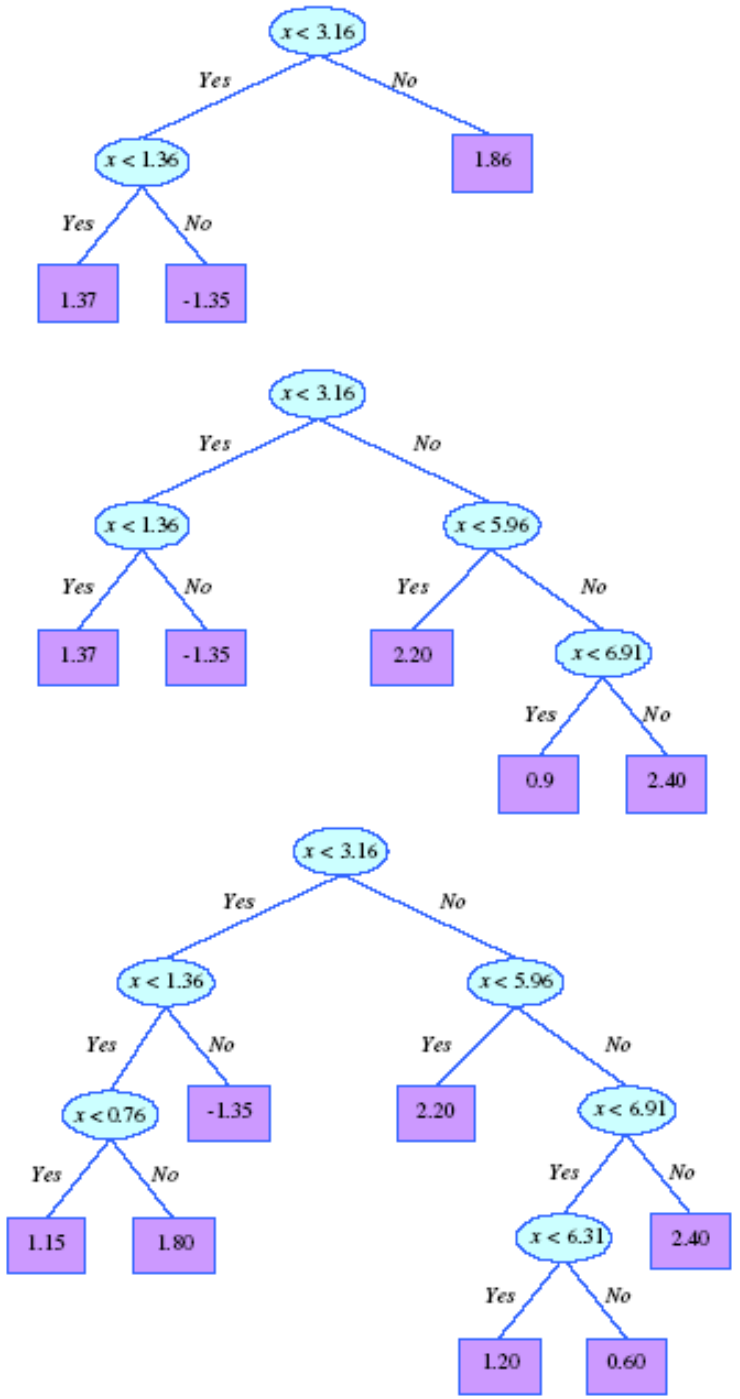


A very simple regression: ☺

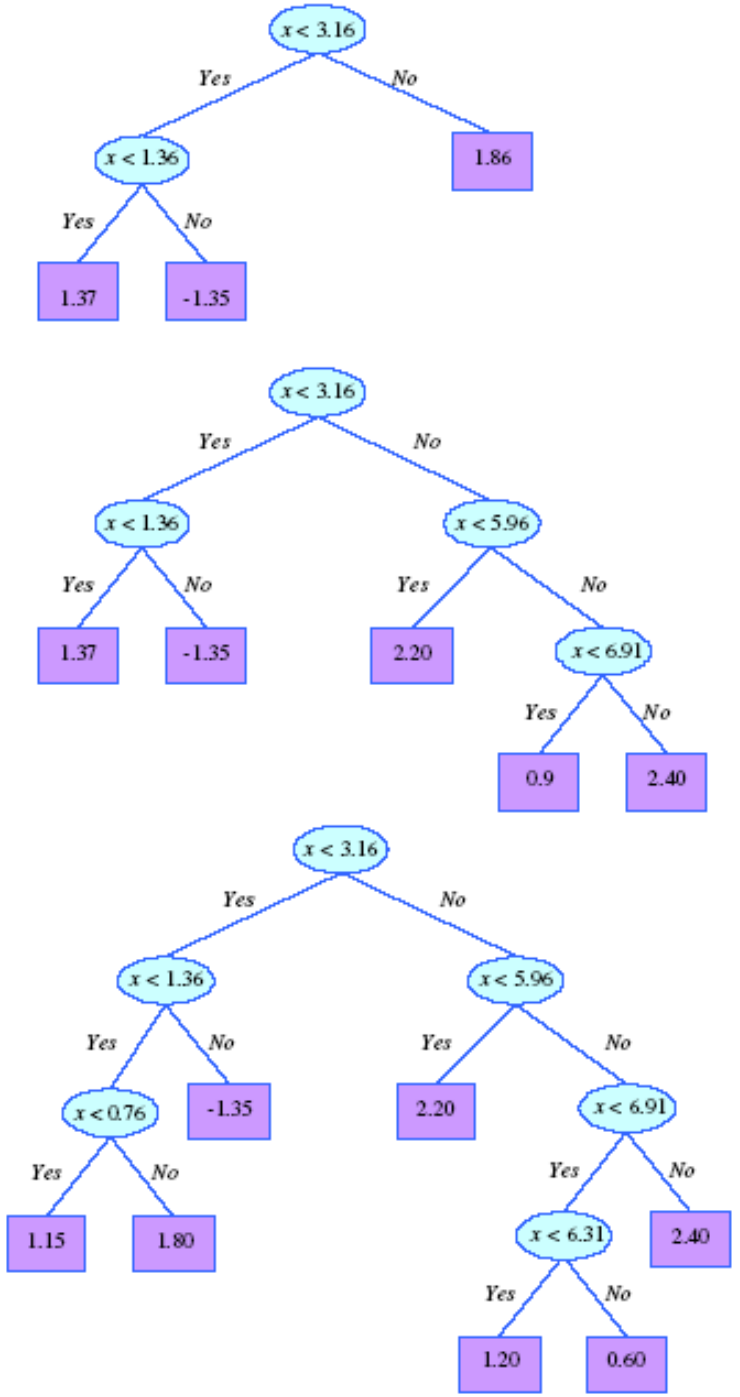
if temp < 9.8
  y=200
  else if temp < 15
    y=350
  else if ...

Model Selection in Trees:





- We do not know which tree is better, one simple way to select the model (tree) is to use validation set performance.



Other Issues

Handling attributes with differing costs

- Sometimes, some attribute values are more expensive or difficult to prepare.
  - medical diagnosis, BloodTest has cost $150
- In practice, it may be desired to postpone acquisition of such attribute values until they become necessary.
- To this purpose, one may modify the attribute selection measure to penalize expensive attributes.
  - Tan and Schlimmer (1990) $\frac{Gain^2(S,A)}{Cost(A)}$
  - Nunez (1988) $\frac{2^{Gain(S,A)} - 1}{(Cost(A)+1)^w}, w \in [0,1]$

Handling training examples with missing attribute values

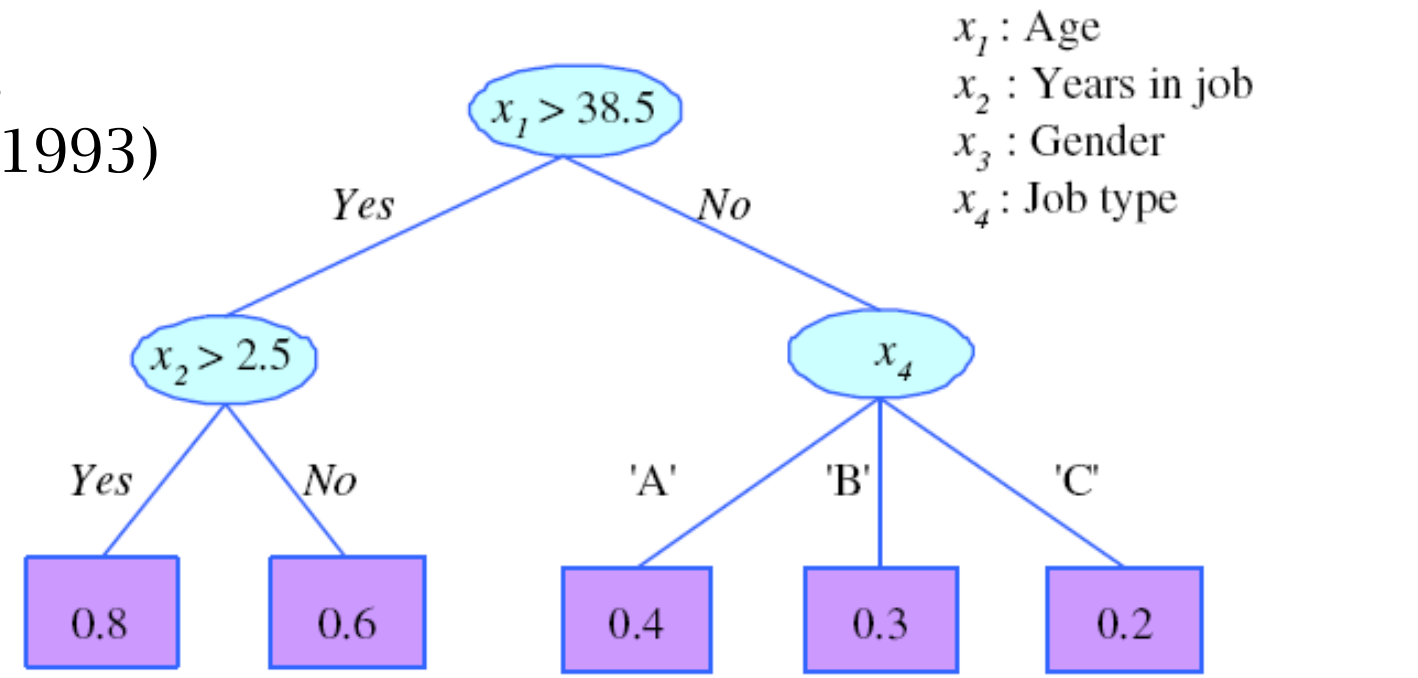
- What if an example x is missing the value an attribute A?

- Simple solution:
  - Use the most common value among examples at node n.
  - Or use the most common value among examples at node n that have classification c(x)

- More complex, probabilistic approach
  - Assign a probability to each of the possible values of A based on the observed frequencies of the various values of A
  - Then, propagate examples down the tree with these probabilities.
  - The same probabilities can be used in classification of new instances (used in C4.5)

Rule Extraction from Trees

C4.5Rules
(Quinlan, 1993)



R1: IF (age>38.5) AND (years-in-job>2.5) THEN y =0.8
R2: IF (age>38.5) AND (years-in-job≤2.5) THEN y =0.6
R3: IF (age≤38.5) AND (job-type='A') THEN y =0.4
R4: IF (age≤38.5) AND (job-type='B') THEN y =0.3
R5: IF (age≤38.5) AND (job-type='C') THEN y =0.2

Summary

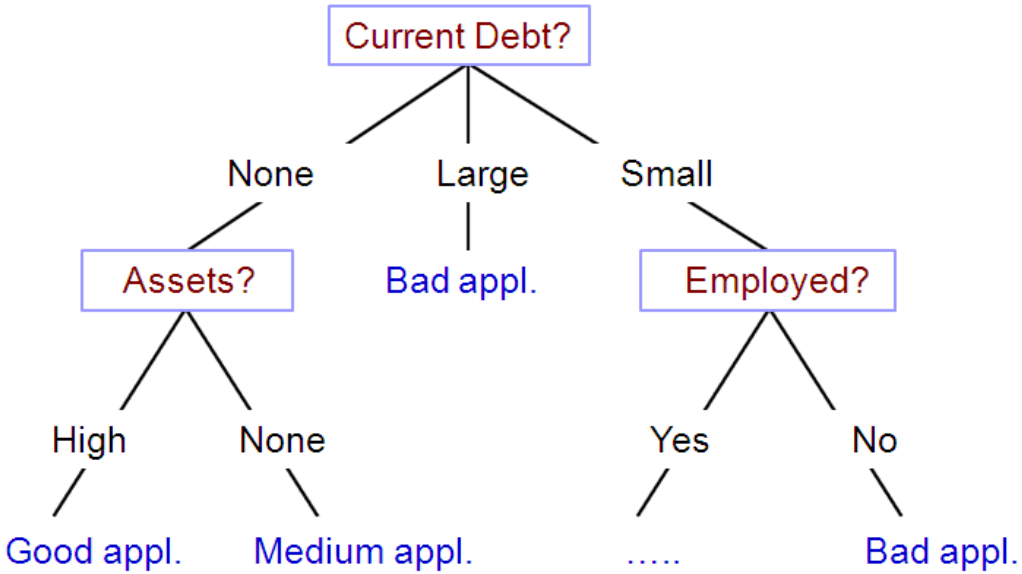
Decision Trees

- Decision trees learns a set of rules training data
  - Learned trees can be represented as sets of **if-then rules** to improve human readability
  - They are very interpretable

Decision Tree

A decision tree can represent a disjunction of conjunctions of constraints on the attribute values of instances.

- Each path corresponds to a conjunction
- The tree itself corresponds to a disjunction



If (Current Debt= Large) OR (Current Debt= Small AND Employed? = No)
then **Bad Applicant**

Strengths and Advantages of Decision Trees

- Interpretability: human experts may verify and/or discover patterns
- Rule extraction from trees
  - A decision tree can be used for feature extraction (e.g. seeing which features are useful)
- It is a compact and fast classification method
- Not affected by scale differences or correlations among attributes

Strengths and Advantages of Decision Trees

- In its simplest form, not the most advanced machine learning model
  - Random forests (we will see later)
  - Gradient Boosted Decision Trees (XGBoost)

- Decision trees are prone (inclined) to overfitting

- Not very robust (not a stable learning algorithm)
  - A small change in the training data can result in a large change in the tree.