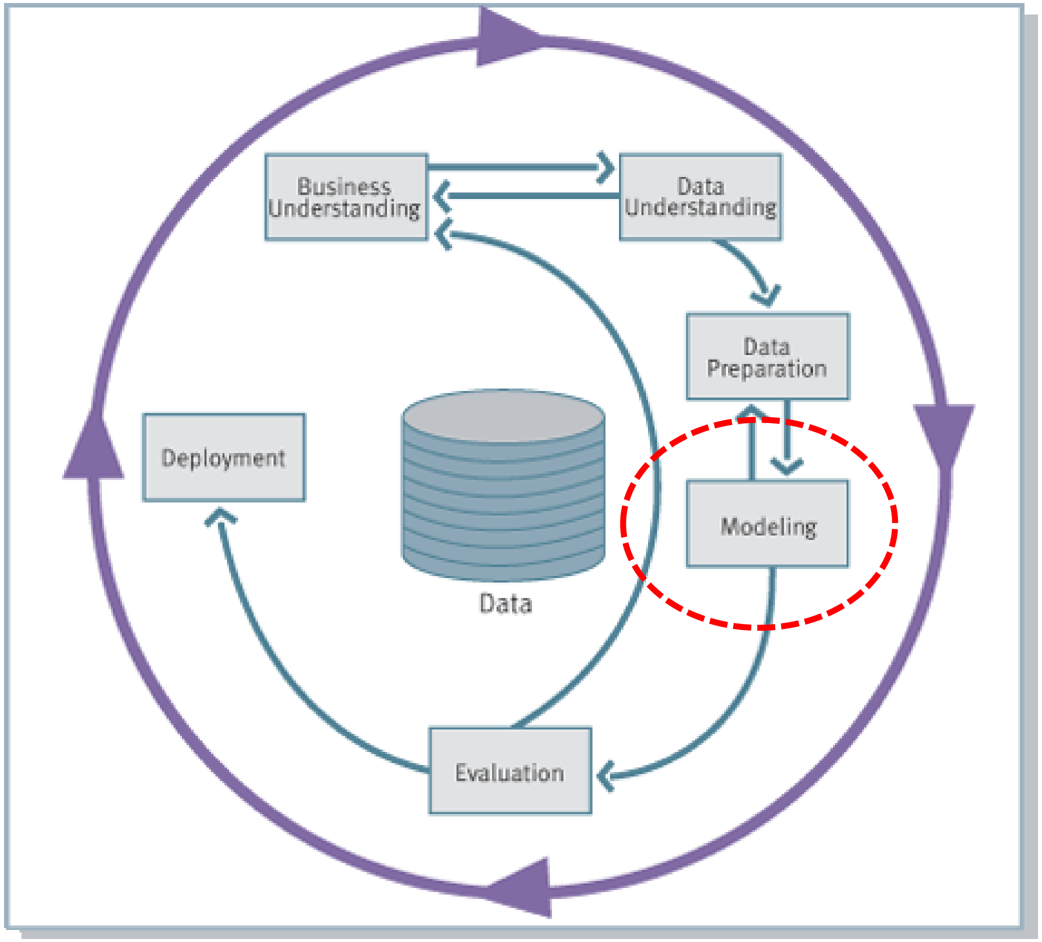
THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Business Analytics (MM 5425)

L3. Decision Tree

Dr. Yue (Katherine) FENG

Recap: A Process View to Run BA



Supervised vs. Unsupervised Learning

> **Supervised learning:** captures relationships between a set of features and a pre-defined, known **target outcome**.

> **Unsupervised learning:** finds relationships in the data without reference to independent/dependent variables.

Key difference: is there a specific, objective *target* that we are trying to predict?

Philosophy of Predictive Modeling

Data universe

Name	Balance	Age	Default
Mike	123,000	50	No
Mary	51,100	40	Yes
Bill	68,000	55	No
Jim	74,000	46	Yes
Dave	23,000	44	No
Anne	100,000	50	Yes
…	…	…	…
…	…	…	…
Henry	61,100	35	???
Amy	68,000	52	???
Allen	22,000	21	???
Tom	123,000	60	???
Jane	100,000	47	???

Data you have (a subset/sample from the universe of the data)

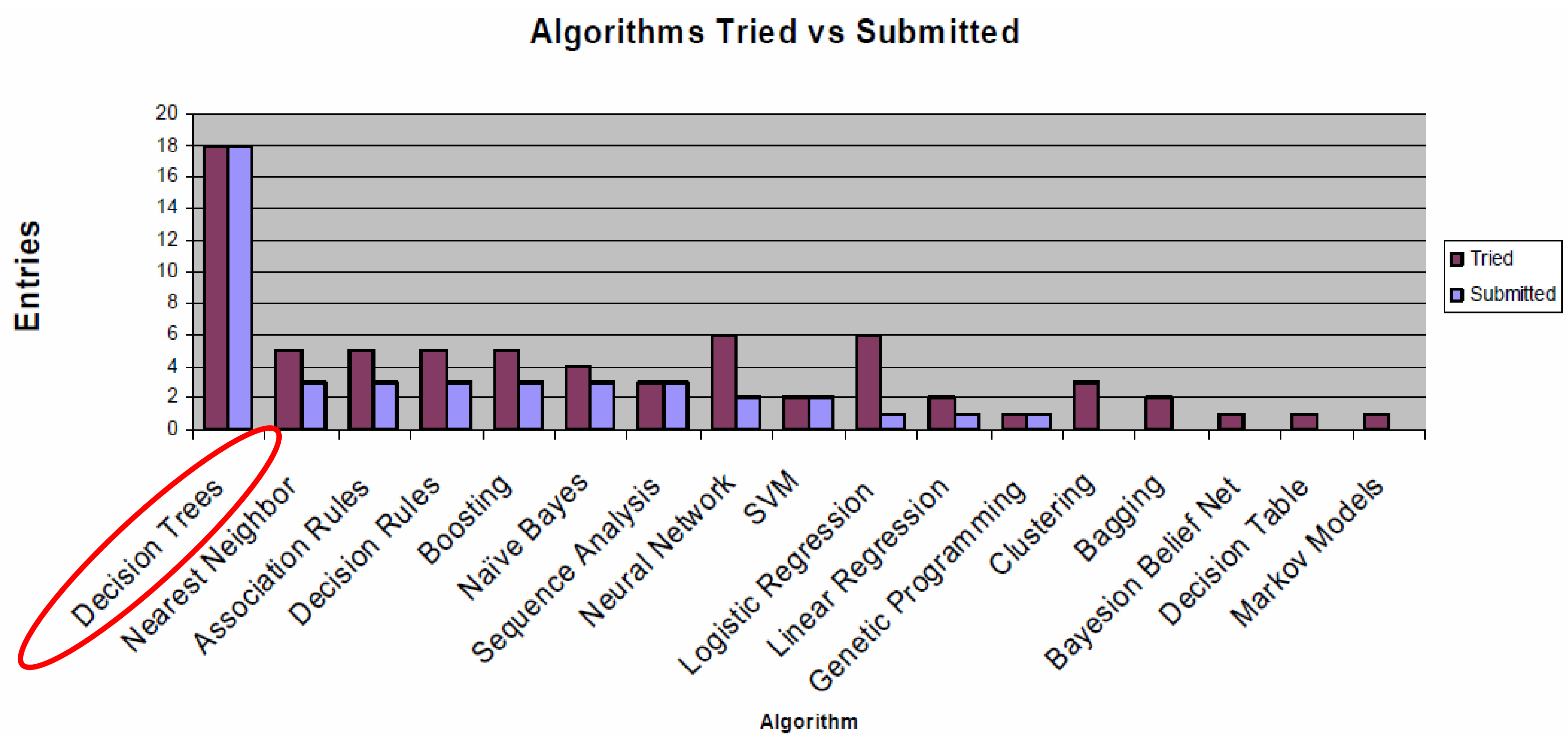
Build model

Model

Apply to new data

Use the subset of data you have to find the pattern of the universe.

Commonly Used Algorithms



Agenda

I. What is a Decision Tree (Classification Tree)?

II. An Example of Classification Tree Induction

III. How to Construct a Classification Tree?

IV. How to Estimate Class Probability?

~~V. Regression Trees (Optional)~~

VI. Overfitting

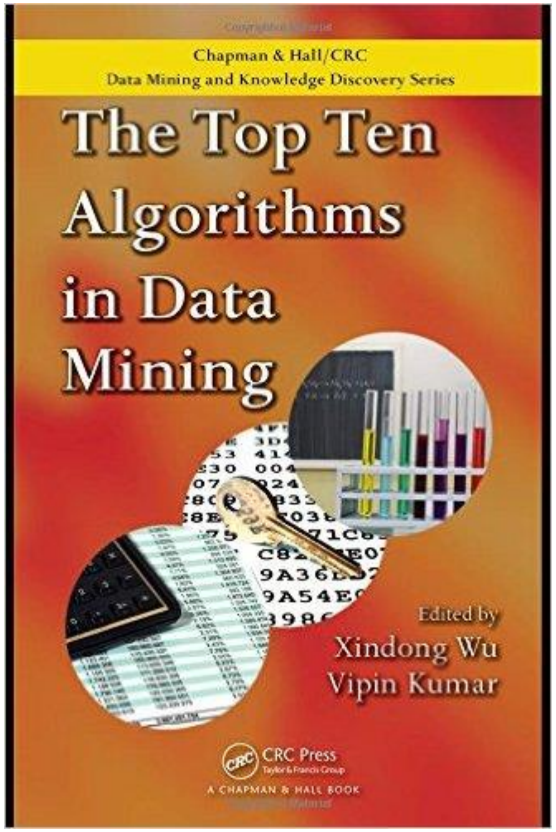
VII. Summary of Decision Tree Learning

What is a Decision Tree?

Why Decision Trees?

> Decision trees, or ***classification trees***, are *one of* the most popular data mining tools.
  - Easy to understand
  - Easy to implement
  - Easy to use
  - Computationally cheap

> They have advantages for model comprehensibility, which is important for model evaluation and communication to non-BA-savvy stakeholders.



Classification Tree

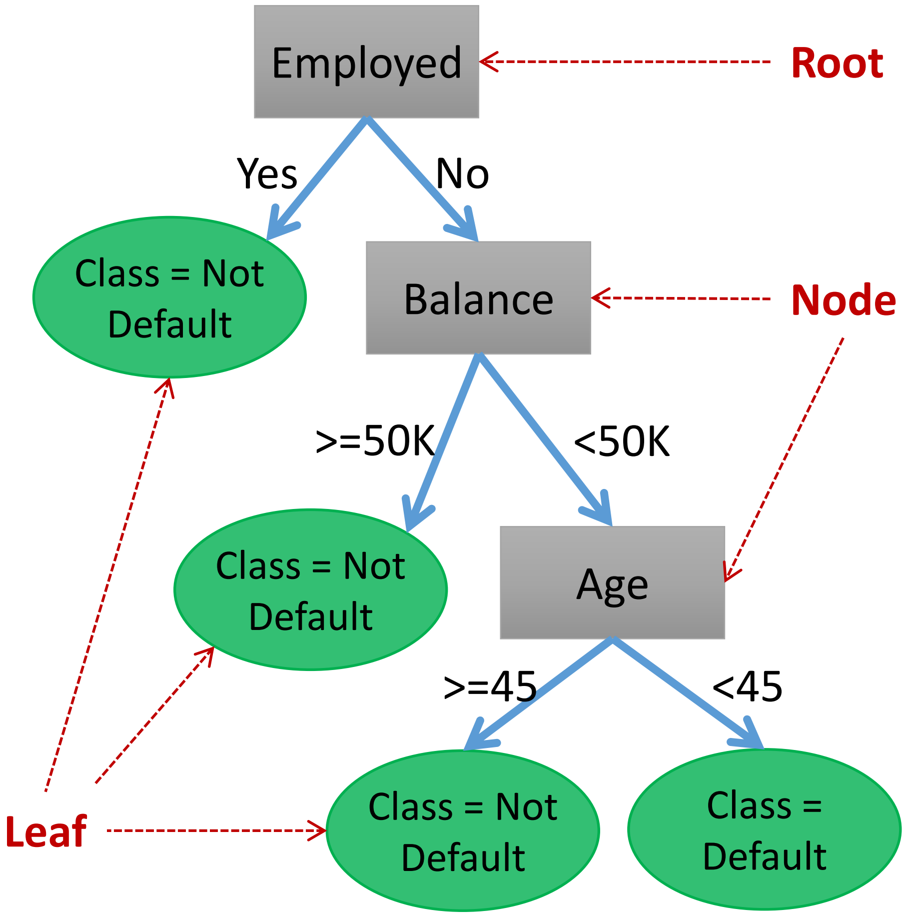
Employed	Balance	Age	Default
Yes	123,000	50	No
No	41,100	40	Yes
No	48,000	55	No
Yes	34,000	46	No
Yes	50,000	46	No
No	100,000	25	No

Induces a classification tree from data examples

IF Employed=No and Balance < 50K and Age < 45
Then Default = 'yes'
Else Default = 'no'

Classification tree

Classification Tree (Upside-down) Representation & Terminology



Classification Tree Representation

> Nodes:

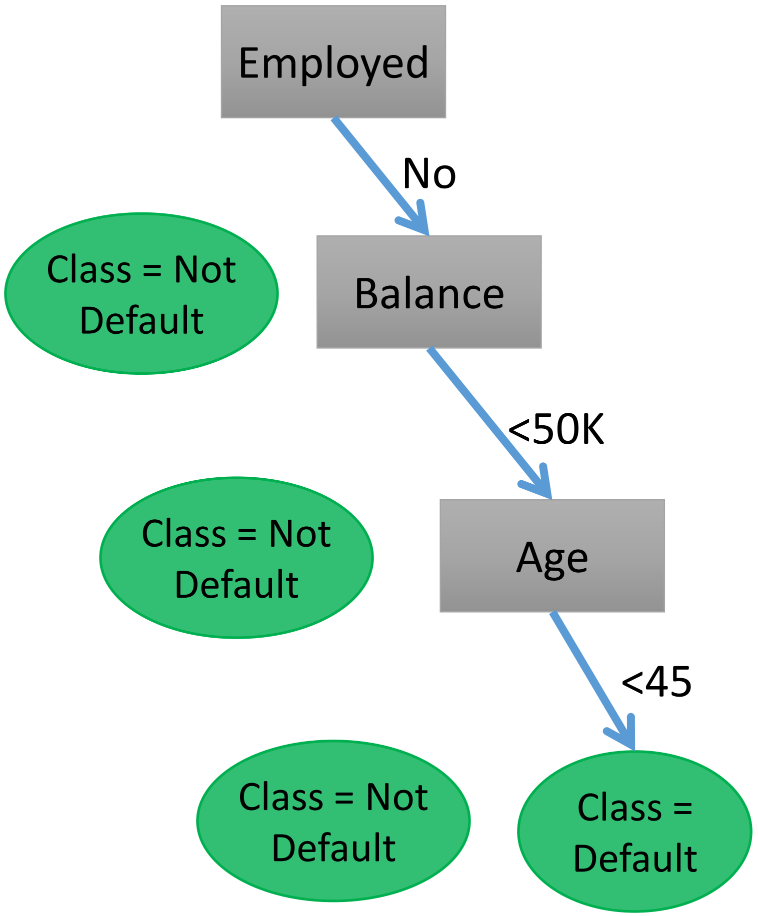
- Each node represents a test on one or more attributes.
- Tests on categorical attribute: number of splits (branches) is number of possible values or one value vs. rest.
- Tests on numeric attributes: need to be discretized (i.e., splitting into intervals by threshold)

> Leaves:

- A class assignment (e.g., default /not default)
- Provide a distribution over all possible classes (e.g., default with probability 0.25, not default with prob. 0.75)

How a Classification Tree is Used for Classification?

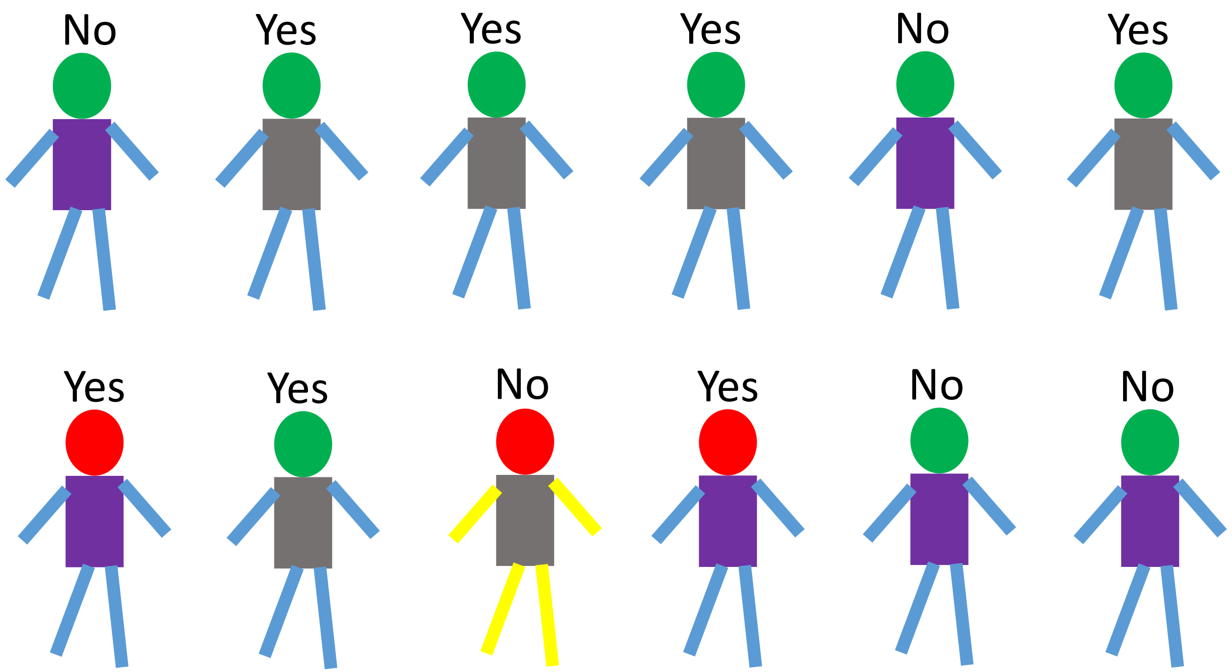
- To determine the class of a new example: e.g., **Mark, age 40, retired, balance 38K.**
- The example is routed down the tree according to values of attributes tested successively.
- At each node, a test is applied to one attribute.
- When a leaf is reached, the example is assigned to a class; or alternatively to a distribution over the possible classes.



An Example of Classification Tree Induction

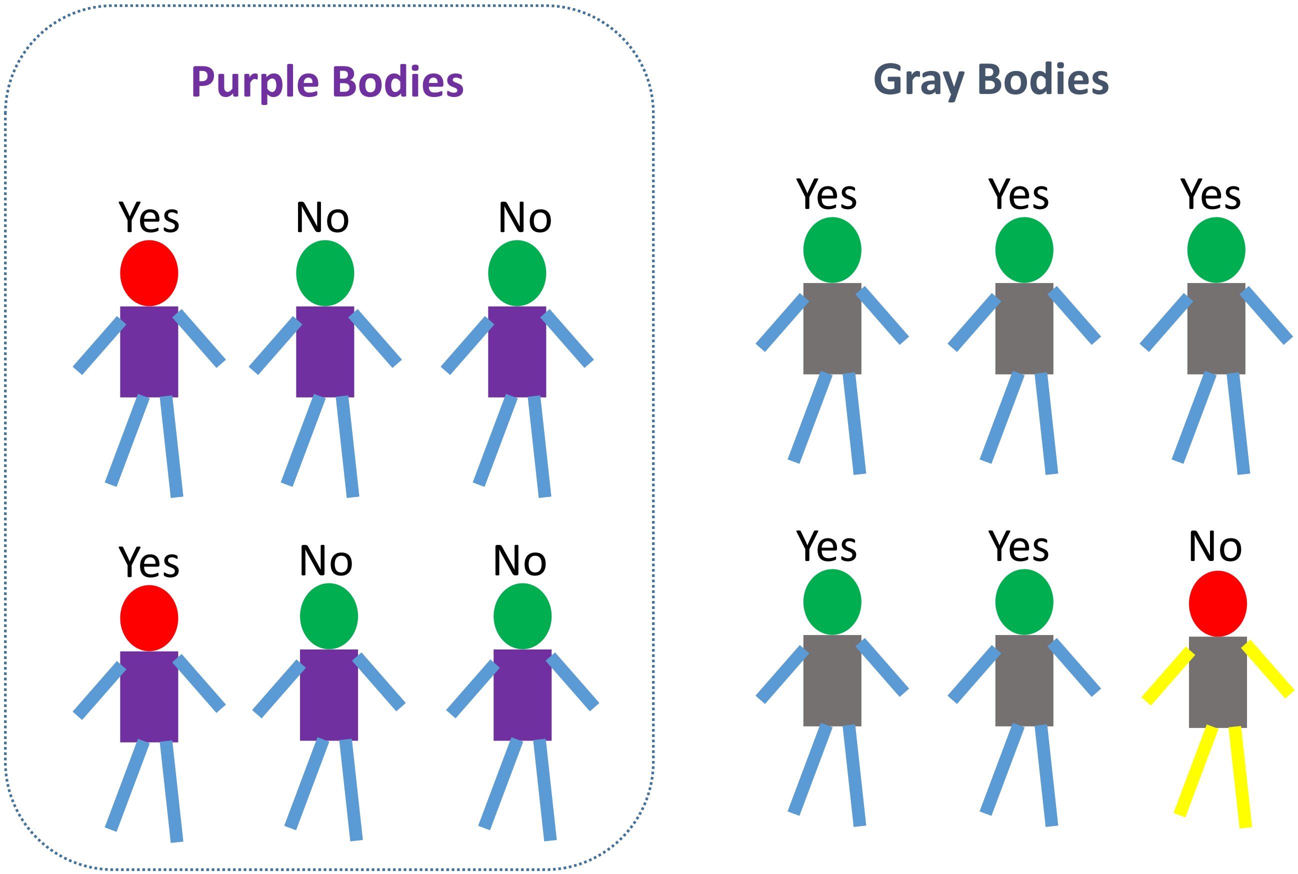
Classification Tree Induction

Objective: Based on customers' attributes, partition the customers into **subgroups that are less impure** with respect to the class (i.e., such that in each group most instances belong to the same class)



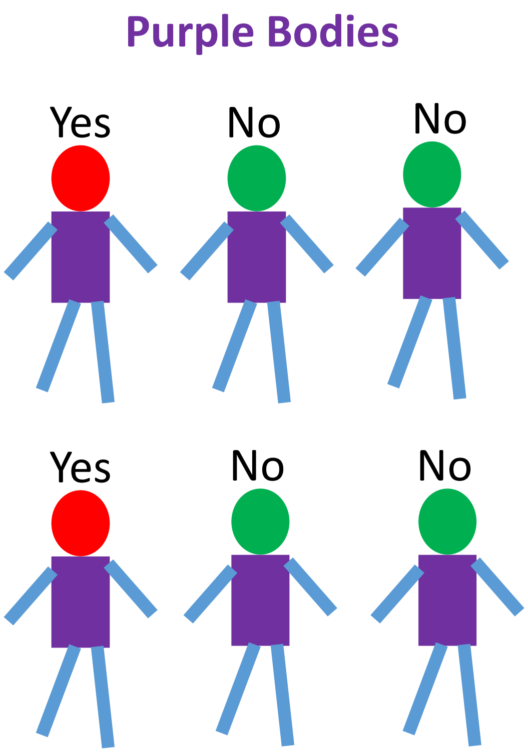
Classification Tree Induction

Partitioning into "***purer***" groups

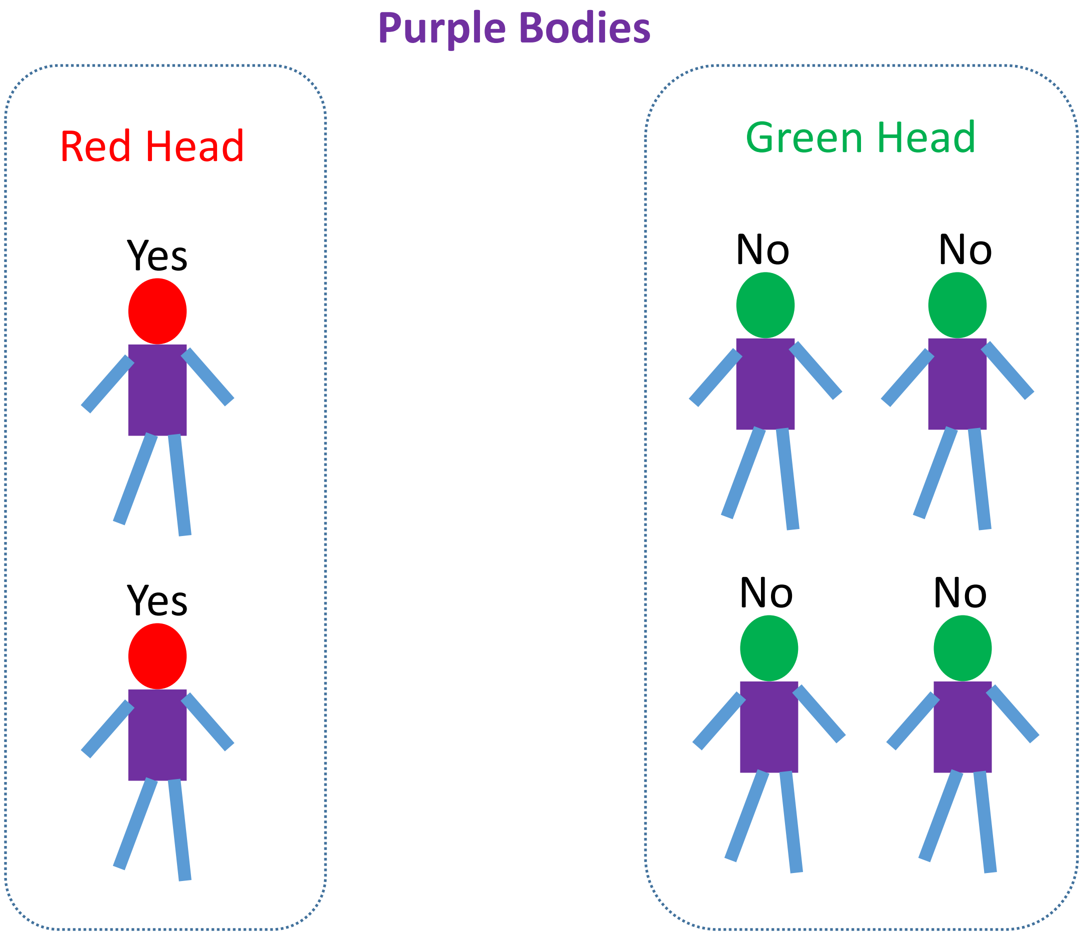


Classification Tree Induction

Partitioning into “purer” groups ***recursively***

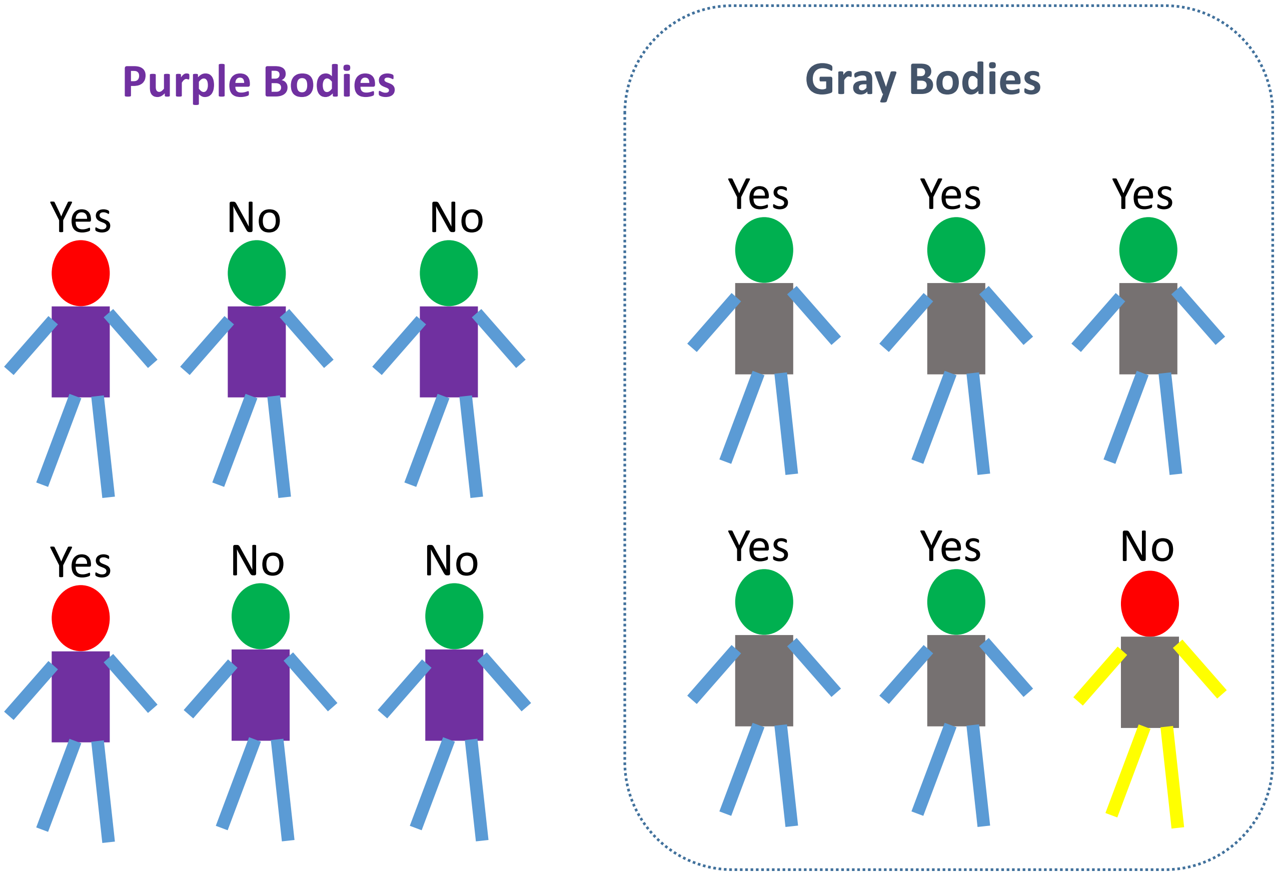


Classification Tree Induction



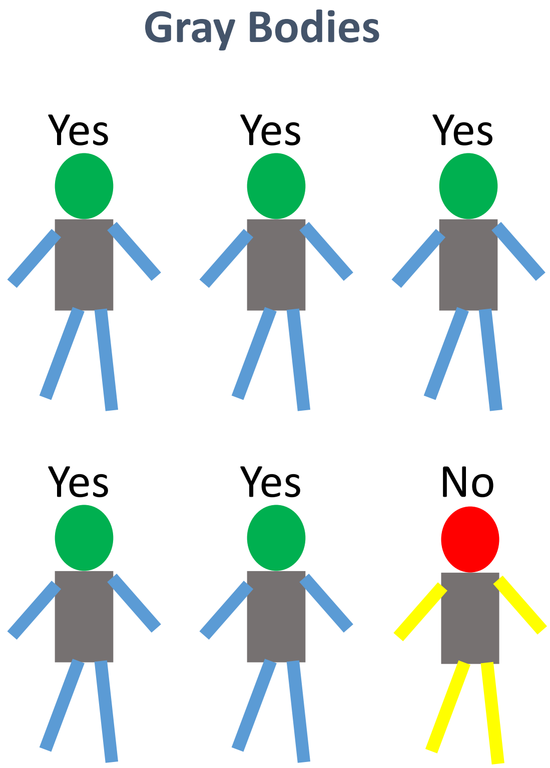
Classification Tree Induction

Partitioning into "***purer***" groups

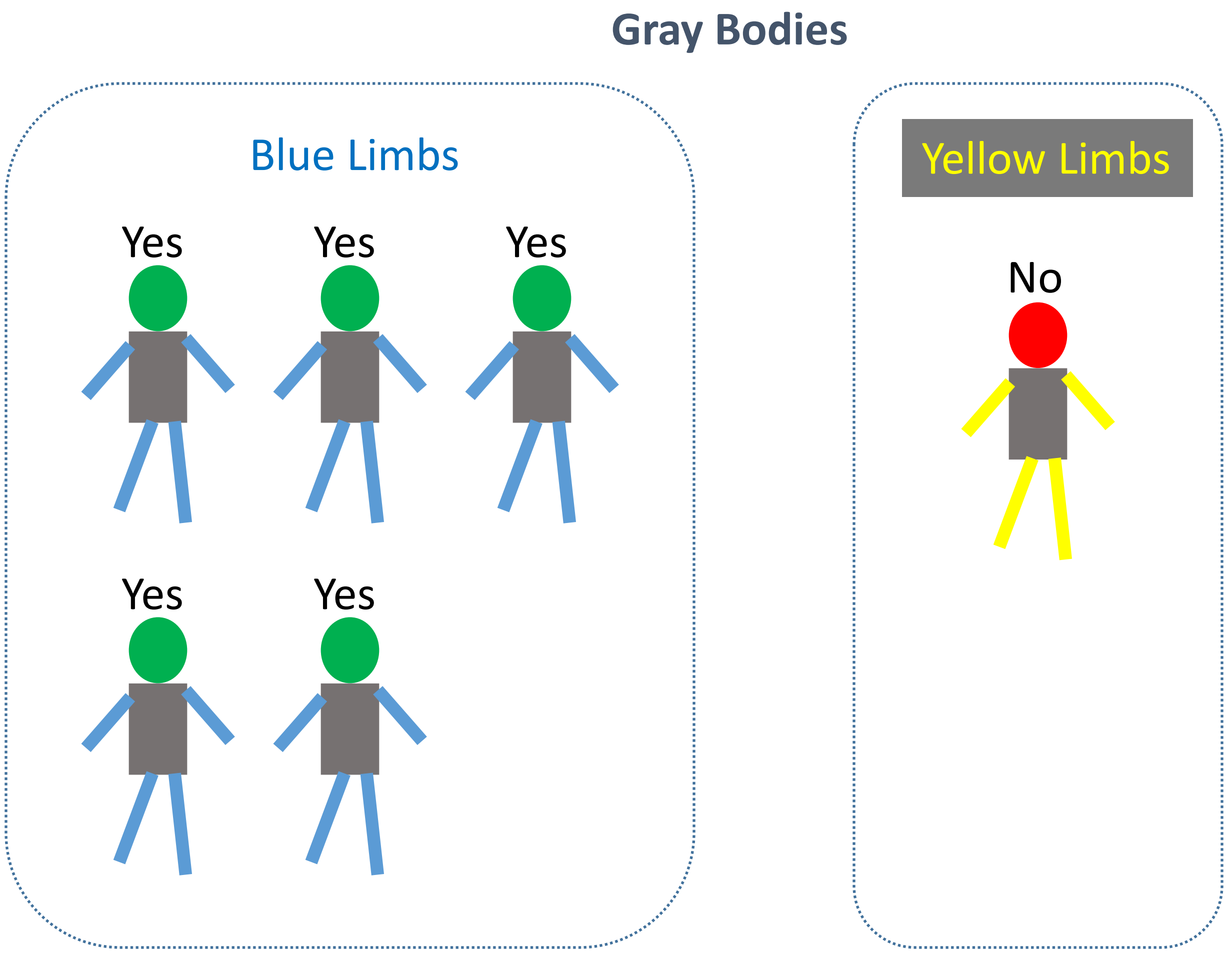


Classification Tree Induction

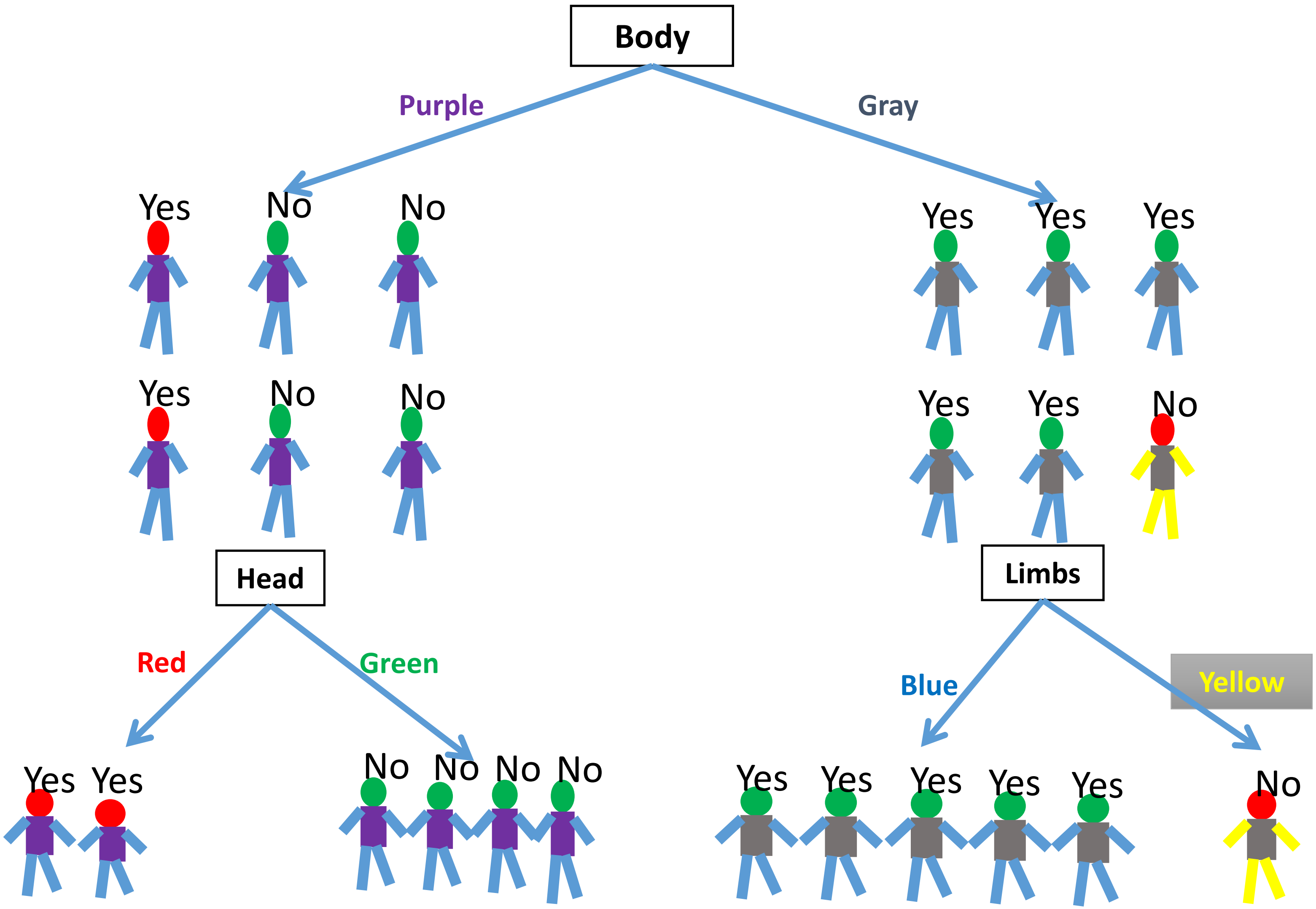
Partitioning into "purer" groups ***recursively***



Classification Tree Induction



Classification Tree Induction



Summary of Classification Tree Induction - Divide and Conquer

> A tree is constructed by ***recursively partitioning*** the instances.

> With each partition, the instances are split into subgroups that are "***increasingly pure***."

- ***Q1: How to automatically choose which attribute to be used to split the data?***
- ***Q2: how would you assign estimates of class probability based on tree induction? (e.g., probability of default)***

How to Construct a Classification Tree?

Basic Principles

> Objectives

- For each splitting node, choose the attribute that best partitions the population into **less impure groups**.
- All else being equal, **fewer nodes** are better.

> Impurity measure: **Entropy**

> Splitting criterion: **Information Gain** (based on entropy)

- Most commonly used
- How informative is the attribute in distinguishing among instances

Impurity Measure - Entropy

Entropy = $\sum_{i} -p_i \log_2 p_i$

where p_i is the proportion of class *i* in the data

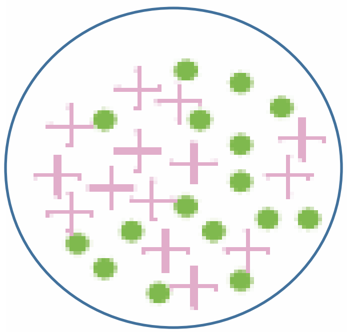
- Measures how disorganized a dataset is
- Comes from ***Information Theory*** (Shannon, 1948)
- Ranges from **0 (minimum disorder)** to **1 (maximal disorder)**

Exercise - Entropy

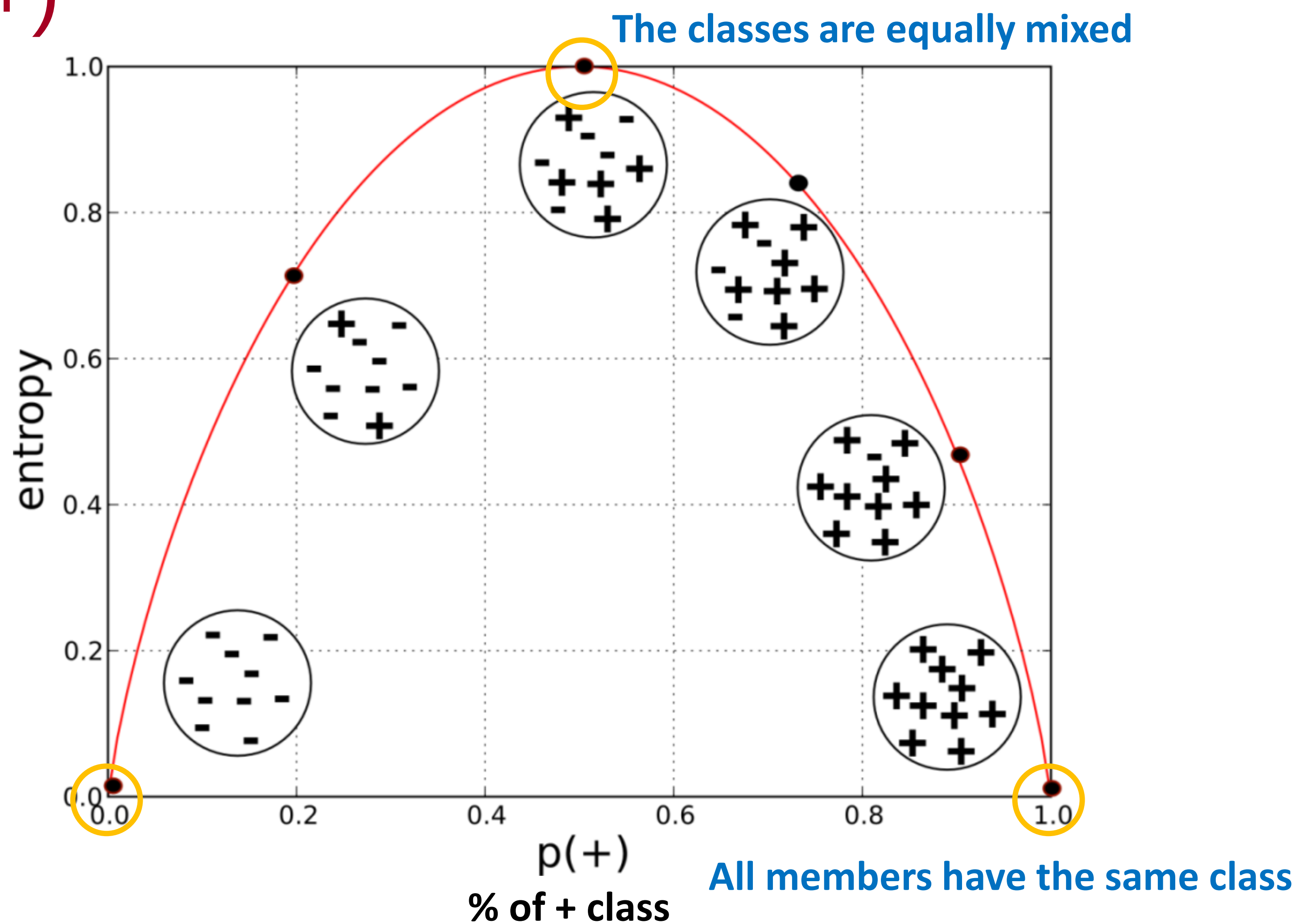
> Our initial population is composed of 14 cases of class "**Not Default**" and 16 cases of class "**Default**"

> Entropy (***entire population*** of examples) =

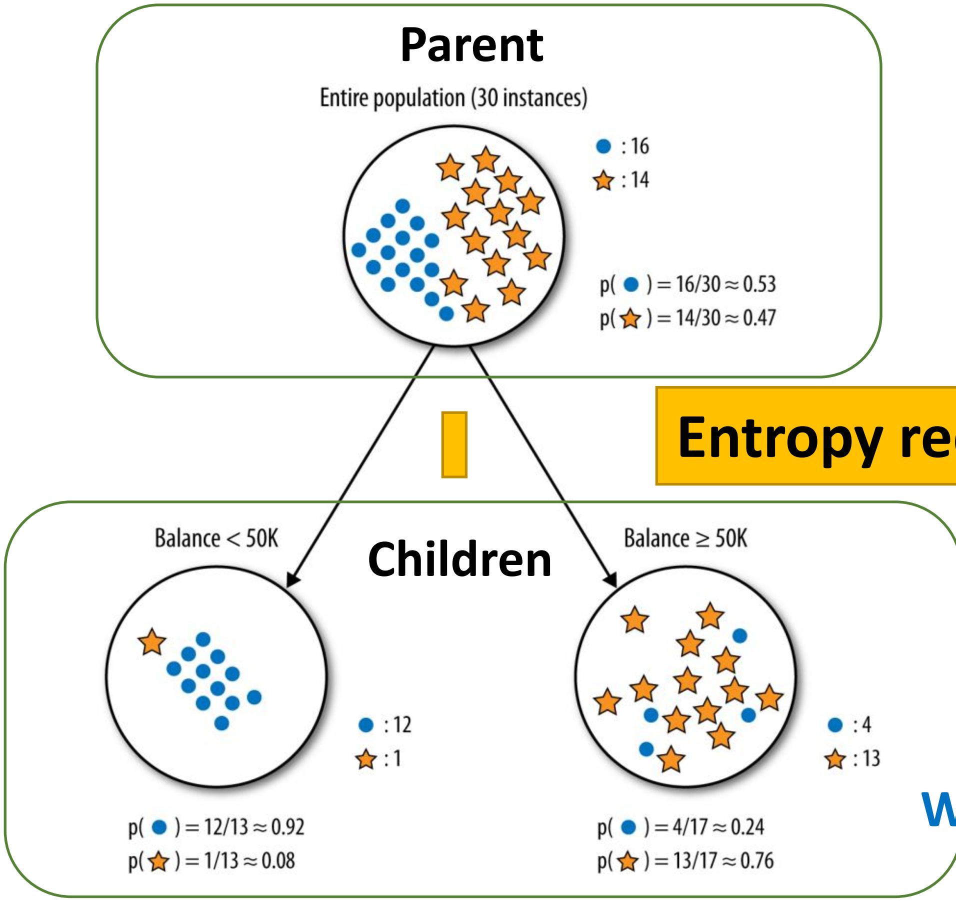
$$-\left(\frac{14}{30}\cdot\log_2\frac{14}{30}\right)-\left(\frac{16}{30}\cdot\log_2\frac{16}{30}\right)=0.996$$



Entropy of a two-class set as a function of p(+)



Information Gain



Information gain:

the **change in entropy** due to any amount of new information being added

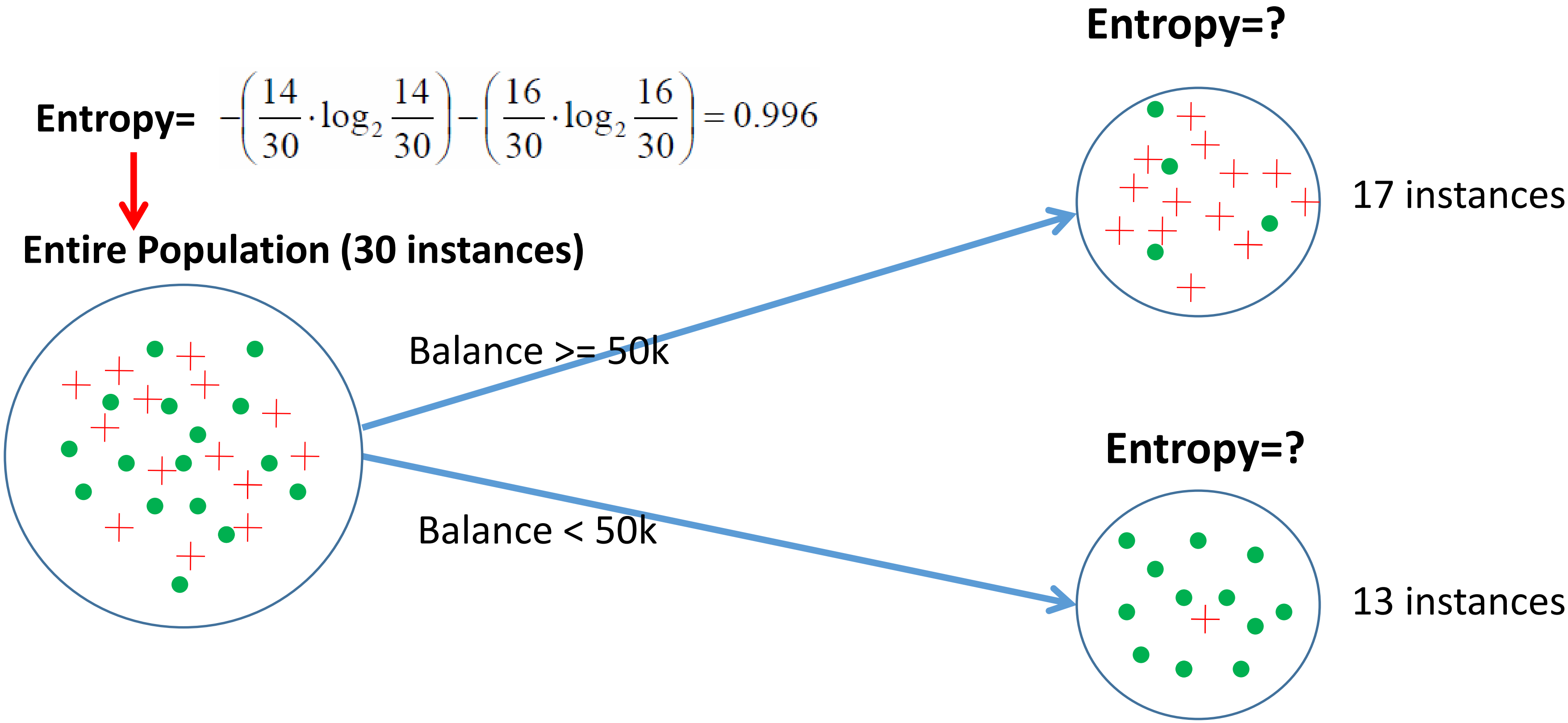
Entropy reduction

Weights of subgroups in children sets:
p(c1); p(c2); …

$$IG(parent, children) = entropy(parent) - [p(c_1) \times entropy(c_1) + p(c_2) \times entropy(c_2) + \cdots]$$

Exercise - Information Gain

Entropy= $-\left(\frac{14}{30}\cdot\log_2\frac{14}{30}\right)-\left(\frac{16}{30}\cdot\log_2\frac{16}{30}\right)=0.996$



(Weighted) Average Entropy of Children = ?

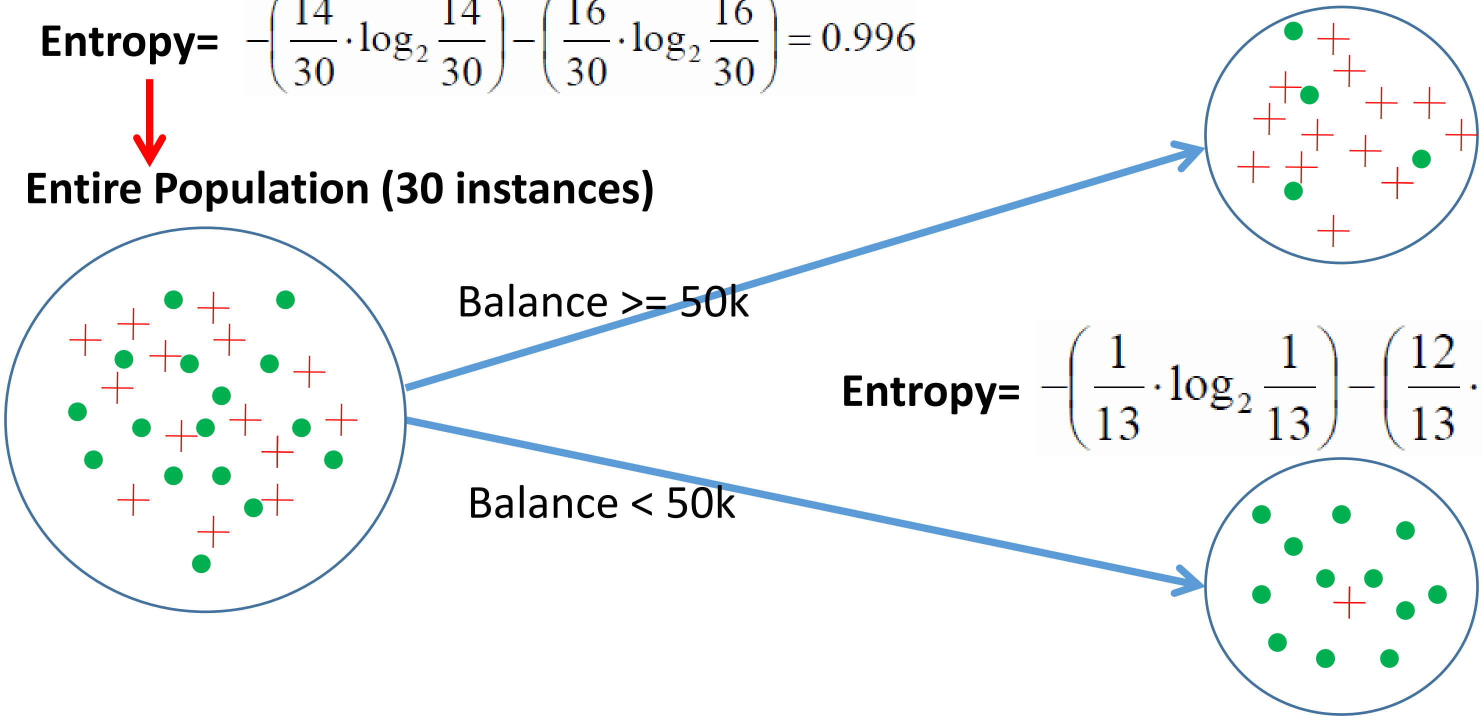
Information Gain ?

Exercise - Information Gain

Entropy= $-\left(\frac{13}{17}\cdot\log_2\frac{13}{17}\right)-\left(\frac{4}{17}\cdot\log_2\frac{4}{17}\right)=0.787$

Entropy= $-\left(\frac{14}{30}\cdot\log_2\frac{14}{30}\right)-\left(\frac{16}{30}\cdot\log_2\frac{16}{30}\right)=0.996$

Entire Population (30 instances)



Entropy= $-\left(\frac{1}{13}\cdot\log_2\frac{1}{13}\right)-\left(\frac{12}{13}\cdot\log_2\frac{12}{13}\right)=0.391$

(Weighted) Average Entropy of Children = $\left(\frac{17}{30}\cdot 0.787\right)+\left(\frac{13}{30}\cdot 0.391\right)=0.615$

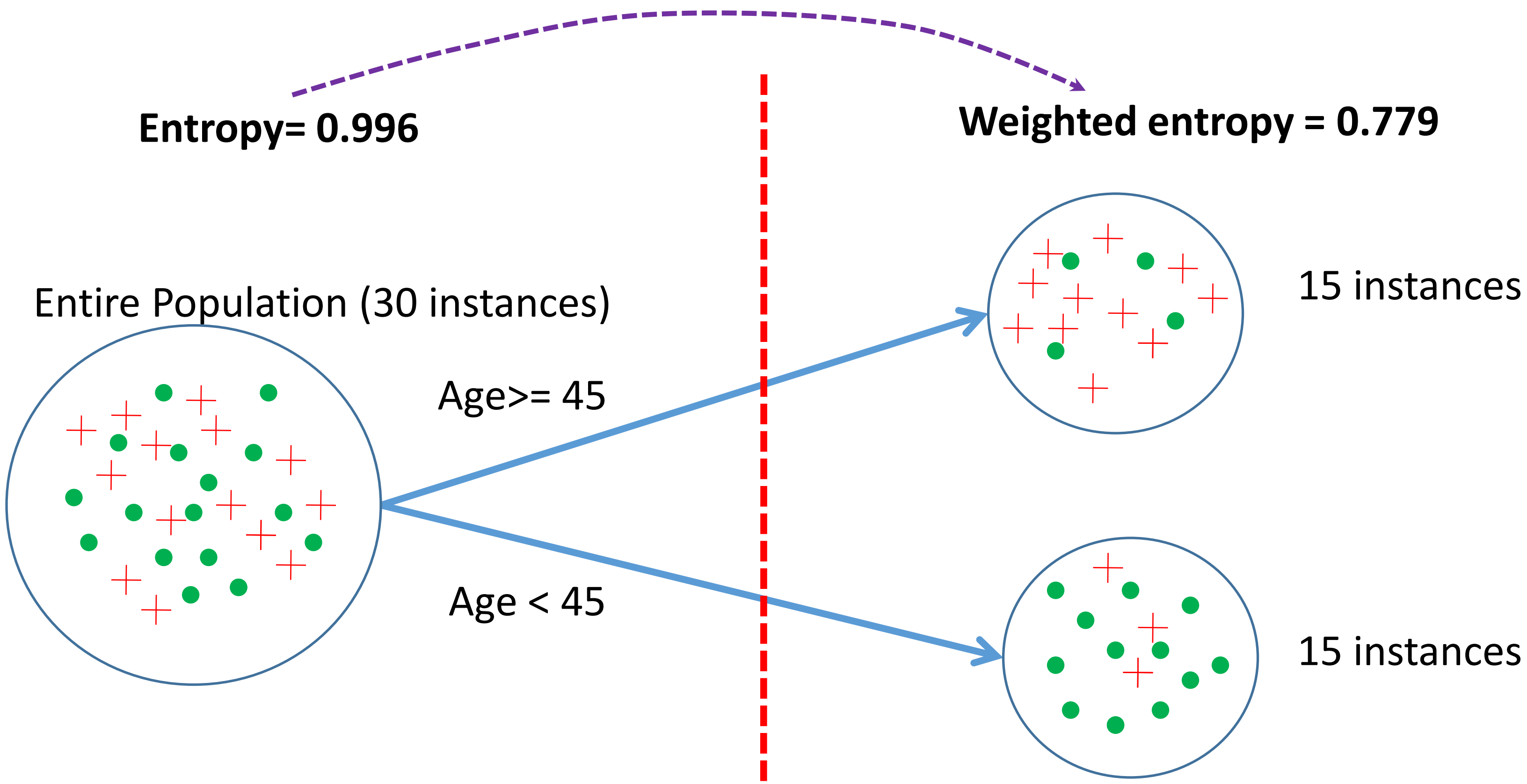
Information Gain = 0.996 - 0.615 = 0.38

Our Original Question

- ***Q1: How to automatically choose which attribute to be used to split the data?***

What if we split over “age” first instead?

Information Gain = 0.996 – 0.779 = 0.215



recall, gain from first splitting on *balance* = 0.38

Answer to Our Original Question

- ***Q1: How to automatically choose which attribute to be used to split the data?***

Answer: at each node, choose the attribute that obtains maximum information gain!

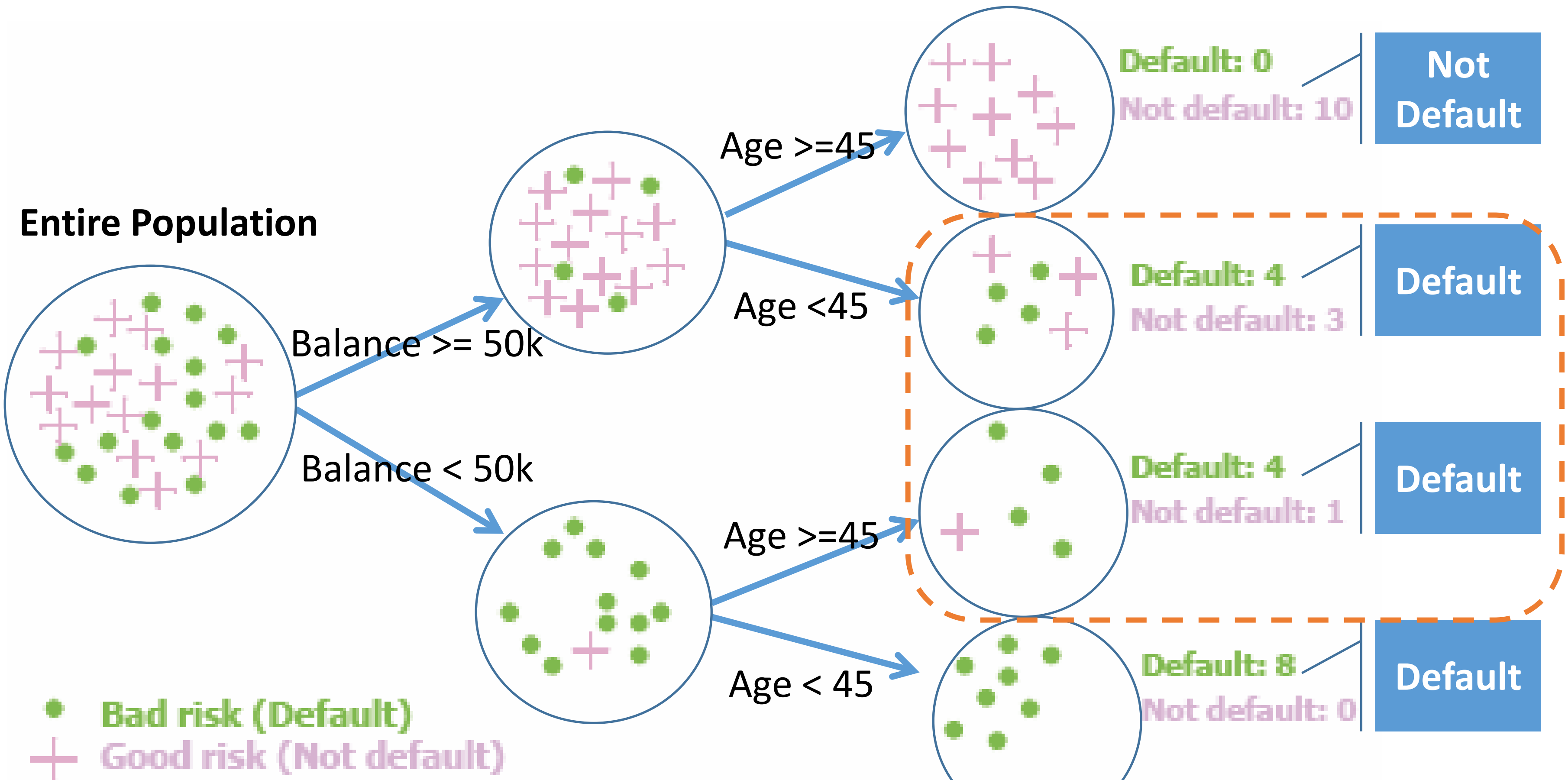
How to Estimate Class Probability?

Class Probability Estimation

> **Frequency-based estimate**

- Basic assumption: each member of a segment corresponding to a tree leaf has the same probability to belong in the corresponding class
- If a leaf contains n positive instances and m negative instances (binary classification), the probability of any new instance being positive is estimated as $\frac{n}{n+m}$

Exercises: Default Problem



16: Default

14: Not default

Exercises

Based on the tree you learned from the past defaulting data,

1) A new person, who is 45 years old and has 20K balance, is applying for a credit card issued by your company.

Please predict if this new person is gonna default? How confident are you about your prediction?

2) Another girl is also applying for the same credit card. But the only information we have about her is she has 70k balance.

Can you predict if she will default? How sure about that?

Discussion

> If one attribute only splits off one single data point into the pure subset. Is this better than another split that does not produce any pure subset, but in some sense reduces the impurity more broadly?

$$IG(parent, children) = entropy(parent) - [p(c_1) \times entropy(c_1) + p(c_2) \times entropy(c_2) + \cdots]$$

Weights of subgroups in children sets

Answer: splitting off a single example may not be as good as splitting the parent set into two large, relatively pure subsets, even if neither is pure!

How to Handle Continuous Variables

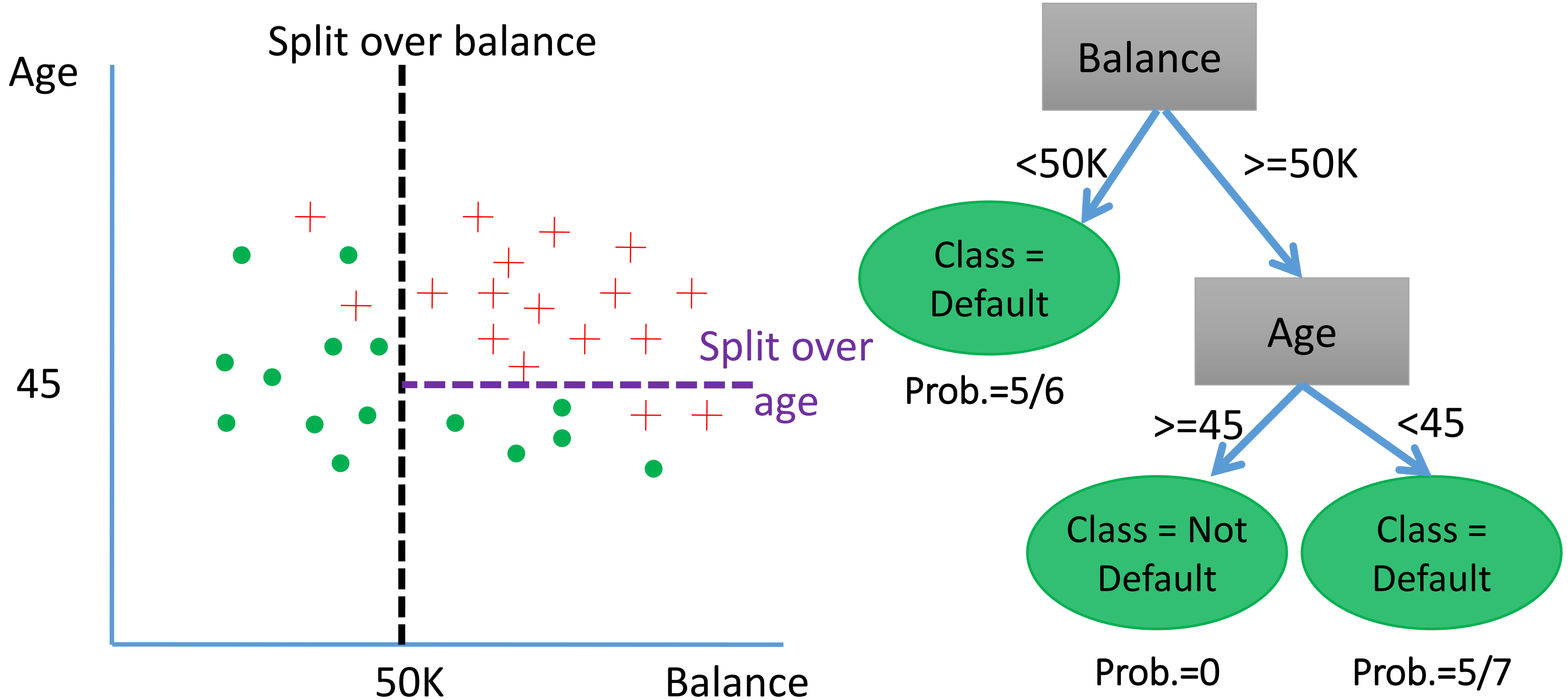
> Handle continuous attribute by splitting into two **intervals** (can be more) at each node.

> How to find the best threshold to divide?

- Answer: try them all !
- Use information gain (or other measure) again
- Sort all the values of an continuous attribute in increasing order {v1, v2, …, vr},
- One possible threshold between two adjacent values vi and vi+1. **Try all possible thresholds and find the one that maximizes the purity.**

Geometric Interpretation

> Classification tree partitions space of examples with axis-parallel decision boundaries

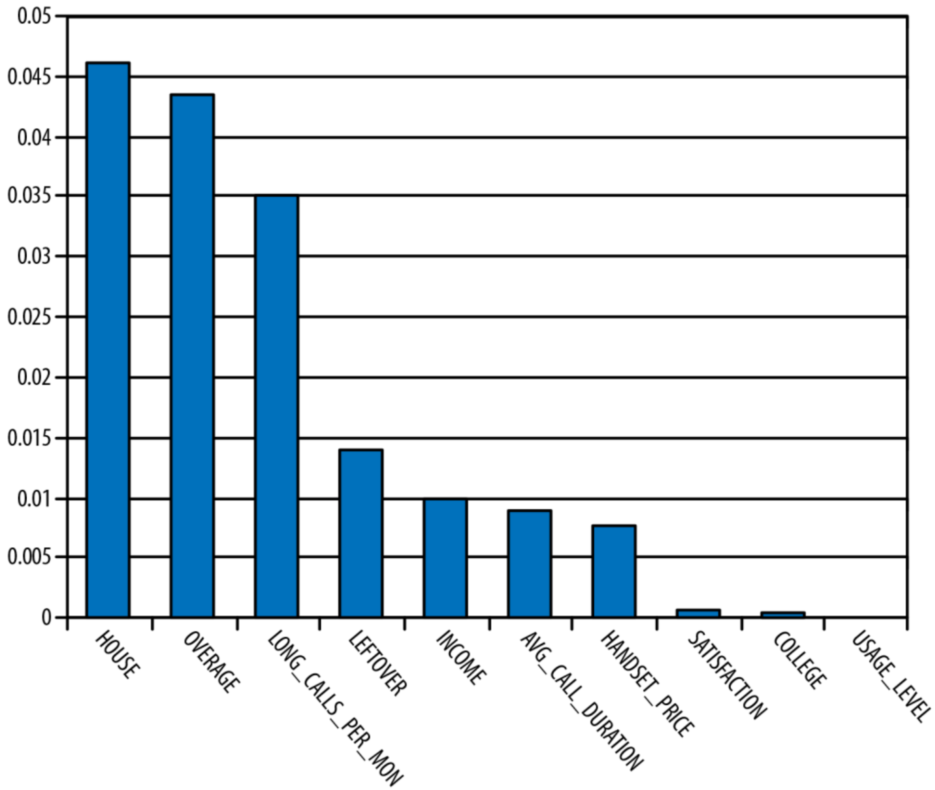


- Bad risk (Default) – 15 cases
+ Good risk (Not default) – 17 cases

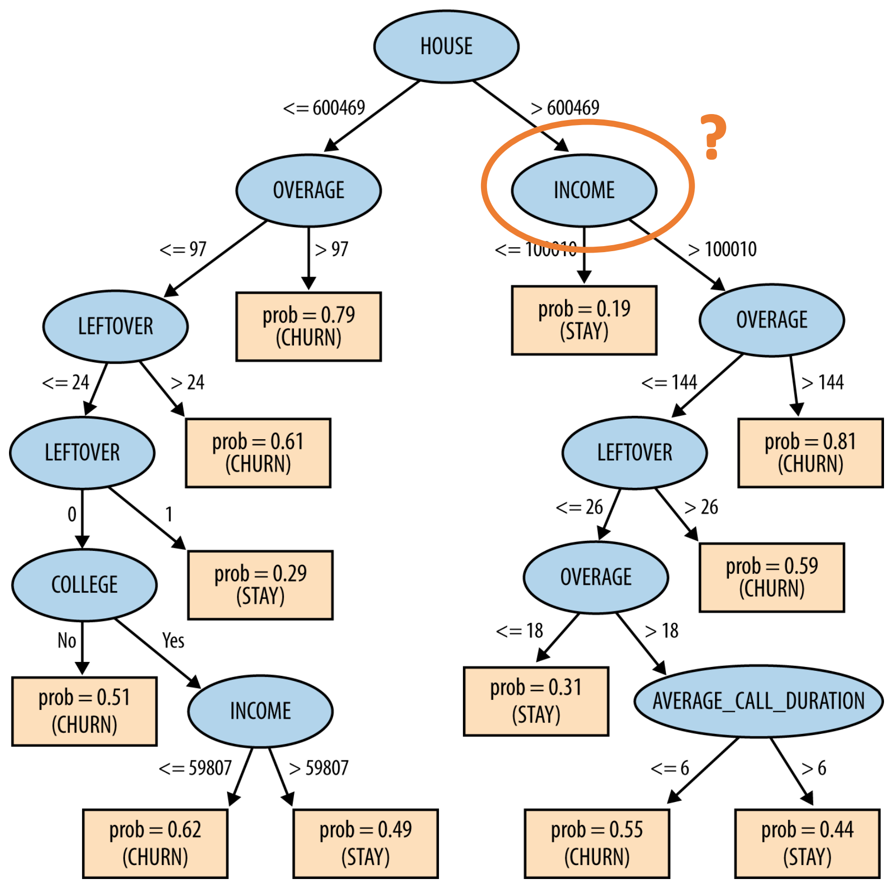
TelCo: Predicting Churn with Tree Induction

Variable	Explanation
COLLEGE	Is the customer college educated?
INCOME	Annual income
OVERAGE	Average overcharges per month
LEFTOVER	Average number of leftover minutes per month
HOUSE	Estimated value of dwelling (from census tract)
HANDSET_PRICE	Cost of phone
LONG_CALLS_PER_MONTH	Average number of long calls (15 mins or over) per month
AVERAGE_CALL_DURATION	Average duration of a call
REPORTED_SATISFACTION	Reported level of satisfaction
REPORTED_USAGE_LEVEL	Self-reported usage level
LEAVE *(Target variable)*	Did the customer stay or leave (churn)?

Which customers should TelCo target with a special offer, prior to contract expiration?



Rank	Info. gain	Attribute name
1	0.0461	HOUSE
2	0.0436	OVERAGE
3	0.0350	LONG_CALLS_PER_MON
4	0.0136	LEFTOVER
5	0.0101	INCOME
6	0.0089	AVG_CALL_DURATION
7	0.0076	HANDSET_PRICE
8	0.0003	SATISFACTION
9	0.000	COLLEGE
10	0.000	USAGE_LEVEL



Regression Trees

Regression Trees

> CART: Classification and Regression Trees

> Regression trees: when the target variable is **numeric**

> **Predicted output:** average of the training examples in the leaf

> **Impurity measure:** sum of the squared errors in the subset

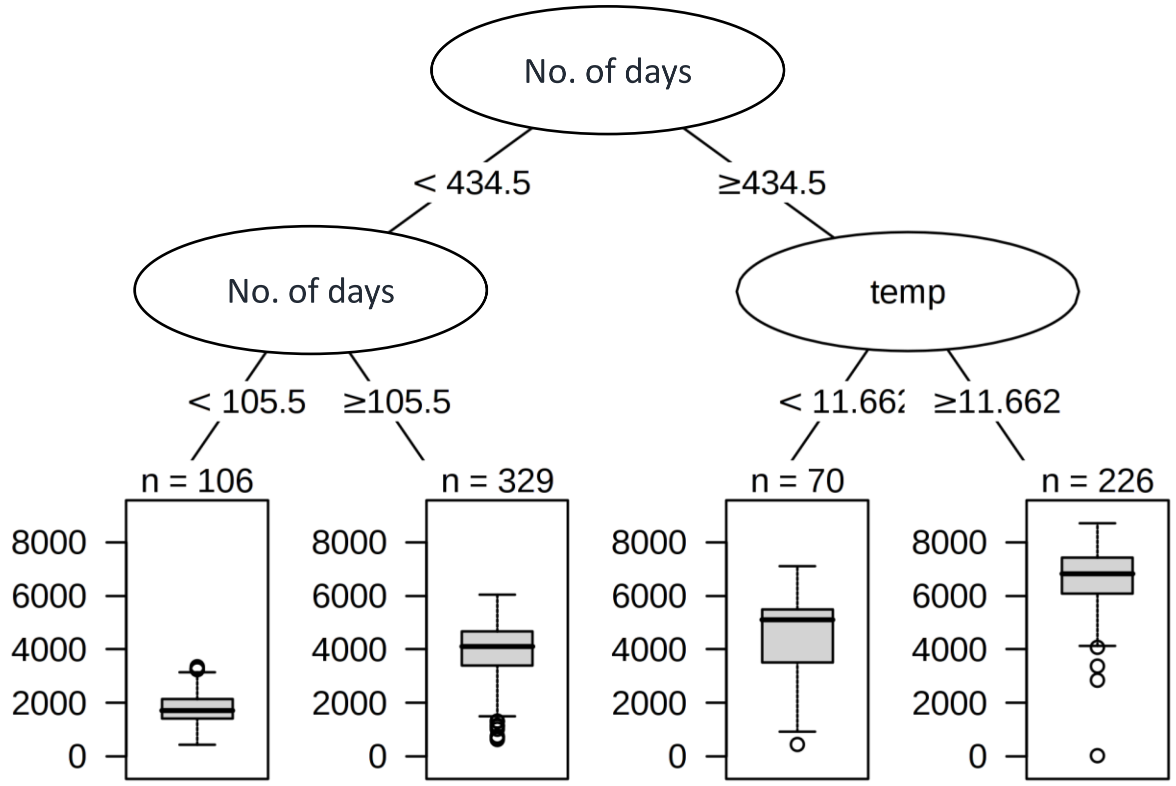
- 0 when all values are the same

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where y_i is the true value of the target and $\hat{y}_i$ is the predicted value.

> Splits are based on how much they reduce the impurity.

An Example of Regression Tree



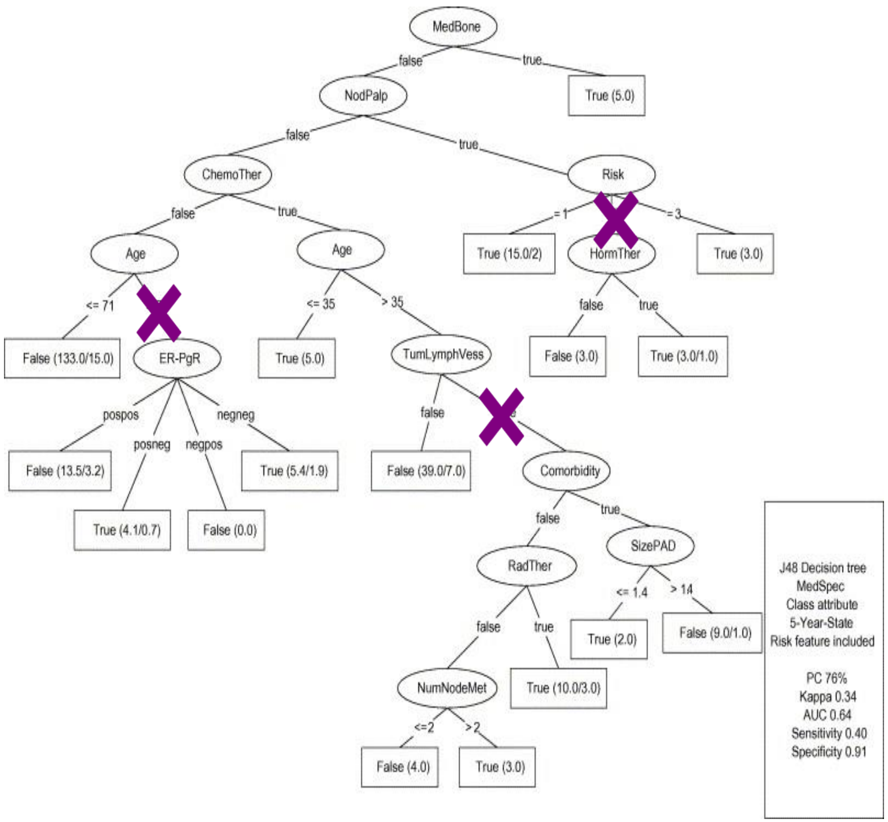
Overfitting

Now We Stop the Partitioning When

> Maximum purity is obtained (i.e., all training examples reaching the node are of the same class)

> Additional splits obtain no information gain

> Question: ***Do we really need to grow the tree fully?***

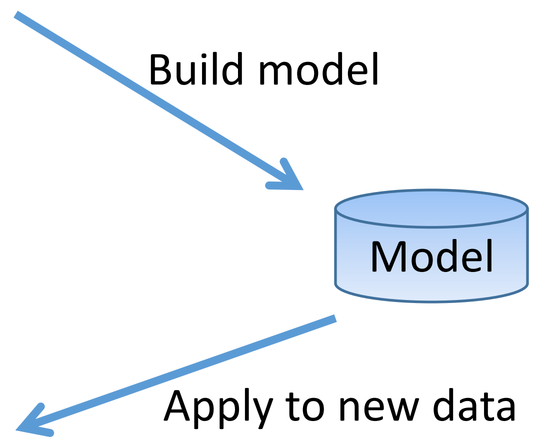


A Problem: Overfitting!

Data universe

Name	Balance	Age	Default
Mike	123,000	50	No
Mary	51,100	40	Yes
Bill	68,000	55	No
Jim	74,000	46	Yes
Dave	23,000	44	No
Anne	100,000	50	Yes
…	…	…	…
…	…	…	…
Henry	61,100	35	???
Amy	68,000	52	???
Allen	22,000	21	???
Tom	123,000	60	???
Jane	100,000	47	???

Data you have (a subset/sample from the universe of the data)



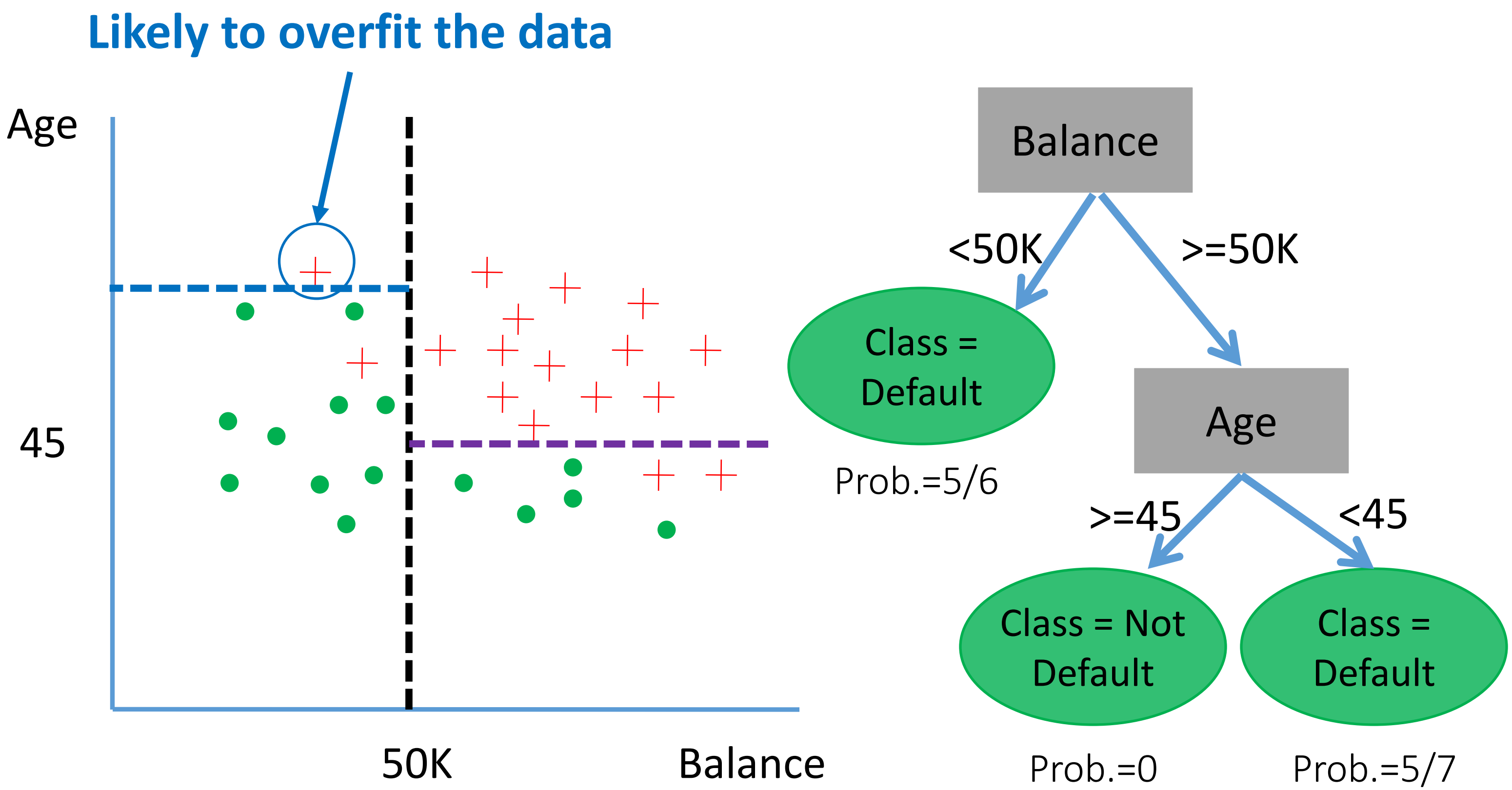
Overfitting: the pattern learned is too specific to be generalized to the universe.

Overfitting the Training Data

> Symptoms:
- Good accuracy on training data but poor on test data

> Reason:
- Trees are too deep and have too many branches, some may reflect anomalies due to noise or outliers

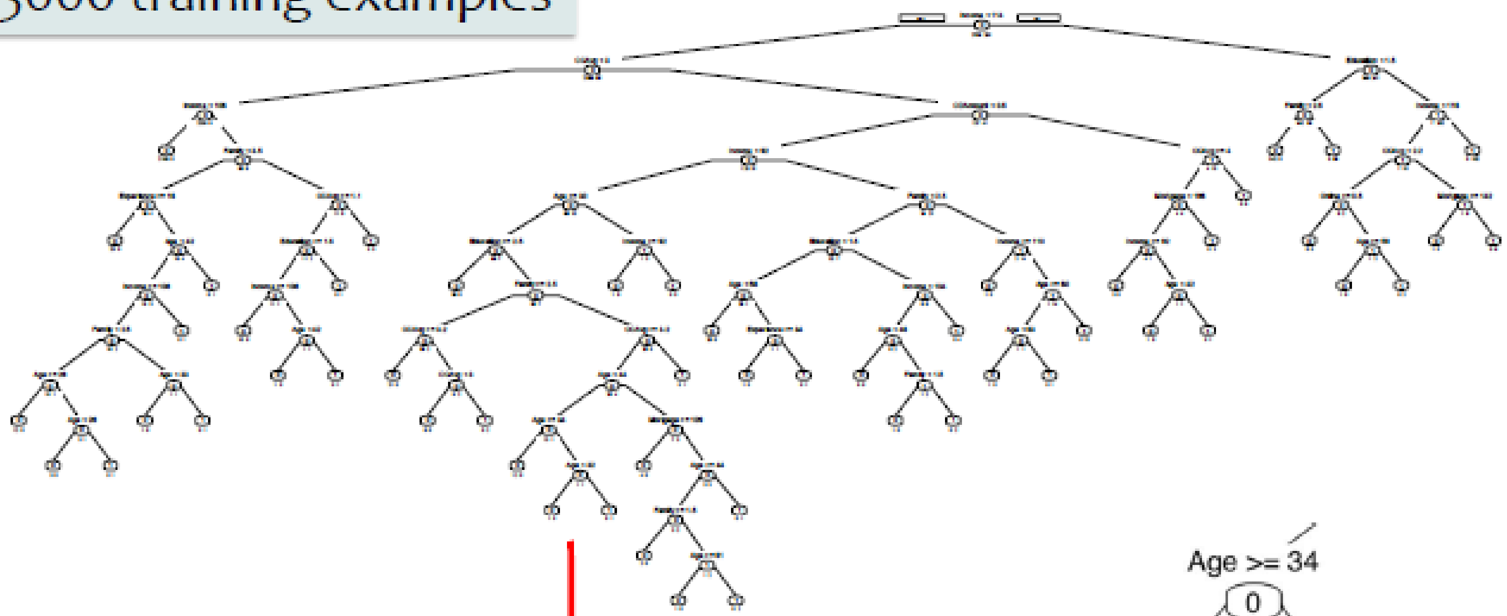
An Example

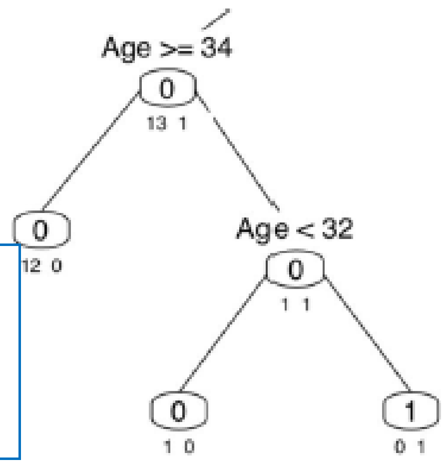


- Bad risk (Default) – 15 cases
- Good risk (Not default) – 17 cases

A Fully-Grown Tree

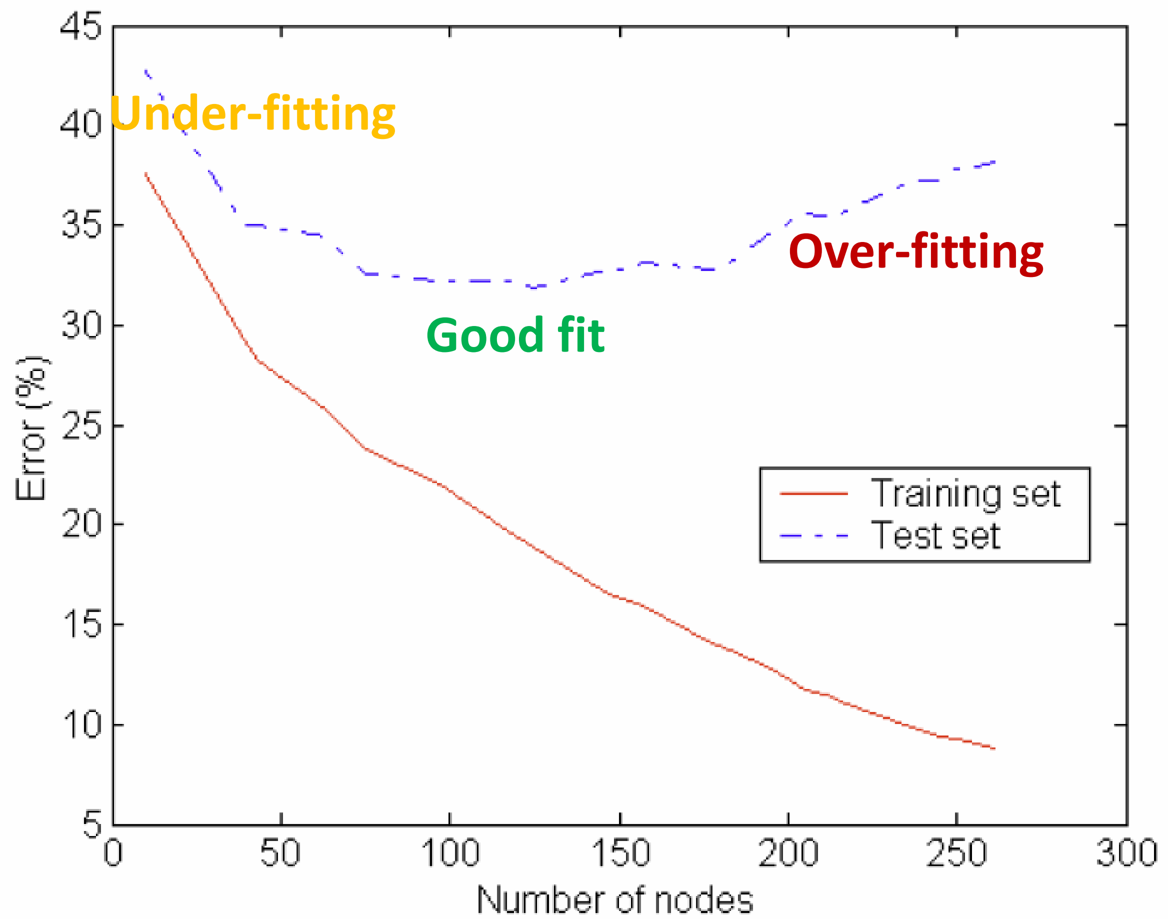
3000 training examples





A fully-grown tree has training error of 0.
However, this tree does not learn any pattern, it simply memorize the training data!

Problem of Overfitting



Error rate = 1 - Accuracy

Over-fitting

Model "***memorizes***" the properties of the particular training set rather than learning the underlying concept or phenomenon.

How to Avoid Overfitting?

> Stop growing via
- Limiting the **maximum depth** of the tree
  - Depth of a tree: the number of edges from the deepest node to the tree's root node
- Limiting the **minimum number of instances** in a node required for splitting
- Limiting the **minimum impurity decrease** required

> Not data-driven, but can incorporate domain knowledge

> In python, fine-tuning tree parameters by **max_depth; min_samples_split; min_impurity_decrease**

Summary

Summary for Decision Tree Construction

> A tree is constructed by ***recursively partitioning*** the examples.

> With each partition the examples are split into subgroups that are "***increasingly pure***."

> How to choose the attribute to split data? ***Maximize information gain!***

Pros and Cons of Decision Tree

Pros:

> Generate transparent rules so that simple to understand and interpret (not applicable for the ensemble versions of trees)

> Require little data preparation and variable selection is automatic

> The relationships between attributes and target are nonlinear.

Pros and Cons of Decision Tree

Cons

> Tree structure is not stable, sensitive to small changes in the data

> Can overfit, thus requires pruning steps and a large dataset to construct a good model

> Splits are done on one attribute at a time, not able to cover the combinations of attributes (e.g., interactions in regression models)

Thank You !