Lecture 7: Decision Trees

Instructor: Saravanan Thirumuruganathan

Outline

1. Geometric Perspective of Classification
2. Decision Trees

Geometric Perspective of Classification

Perspective of Classification

- Algorithmic
- Geometric
- Probabilistic
- ...

Geometric Perspective of Classification

- Gives some intuition for model selection
- Understand the distribution of data
- Understand the expressiveness and limitations of various classifiers

Feature Space[1]

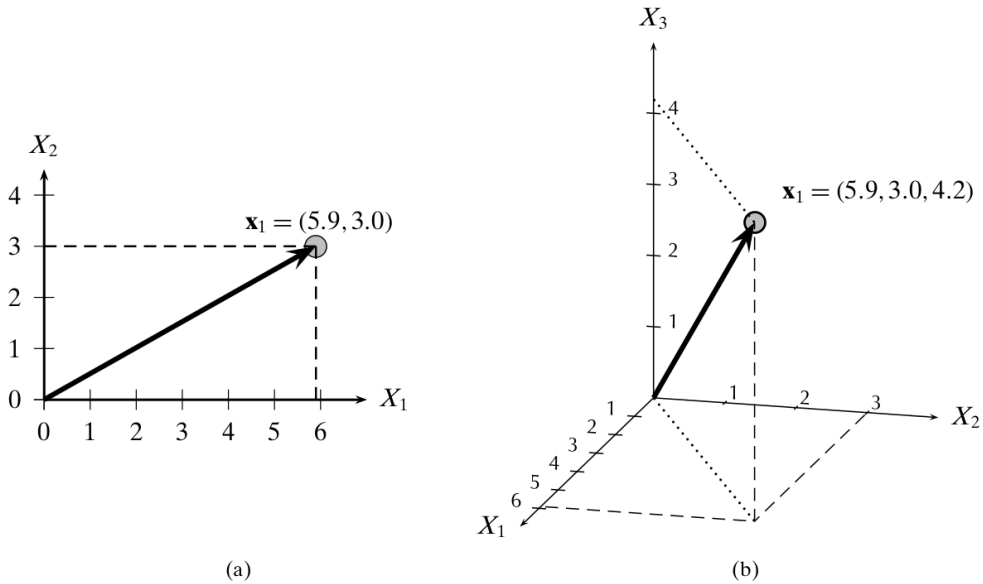
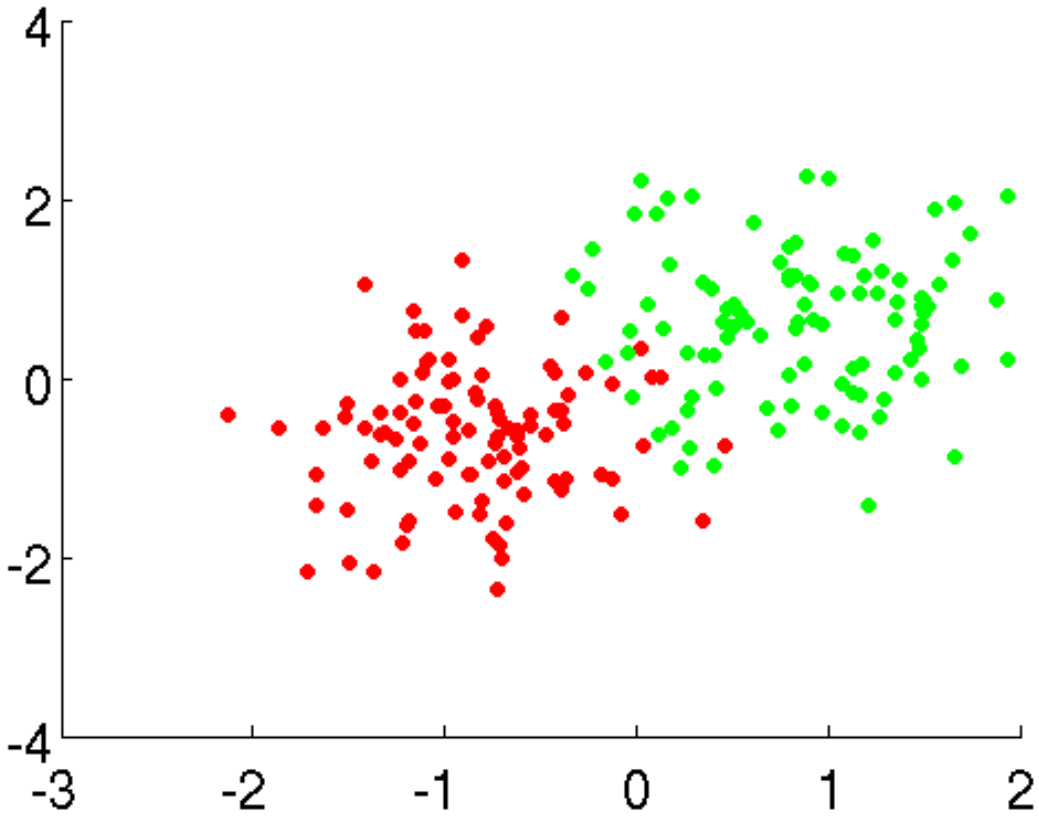


Figure 1.1. Row $\mathbf{x}_1$ as a point and vector in (a) $\mathbb{R}^2$ and (b) $\mathbb{R}^3$.

- **Feature Vector:** d-dimensional vector of features describing the object
- **Feature Space:** The vector space associated with feature vectors

[1] DMA Book

Feature Space in Classification



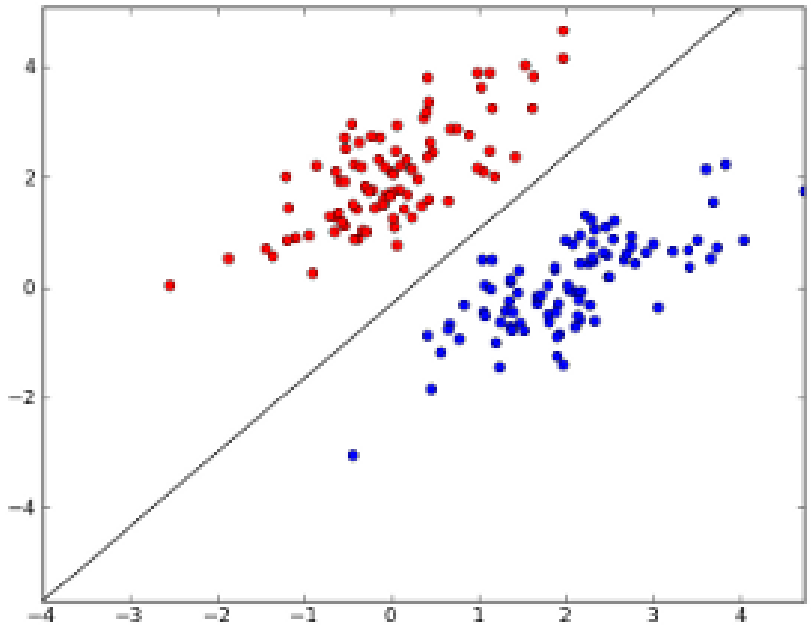
Geometric Perspective of Classification

- **Decision Region:** A partition of feature space such that all feature vectors in it are assigned to same class.
- **Decision Boundary:** Boundaries between neighboring decision regions

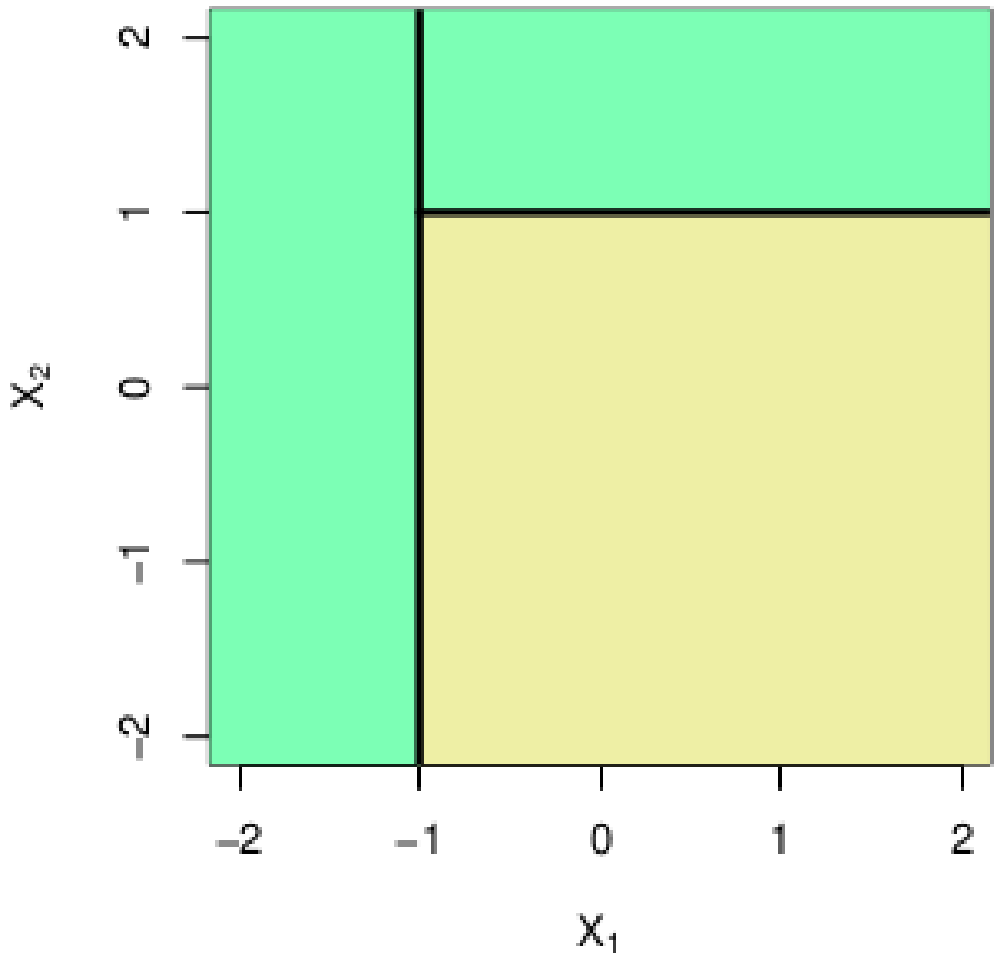
Geometric Perspective of Classification

- Objective of a classifier is to *approximate* the "real" decision boundary as much as possible
- Most classification algorithm has specific expressiveness and limitations
- If they align, then classifier does a good approximation

Linear Decision Boundary

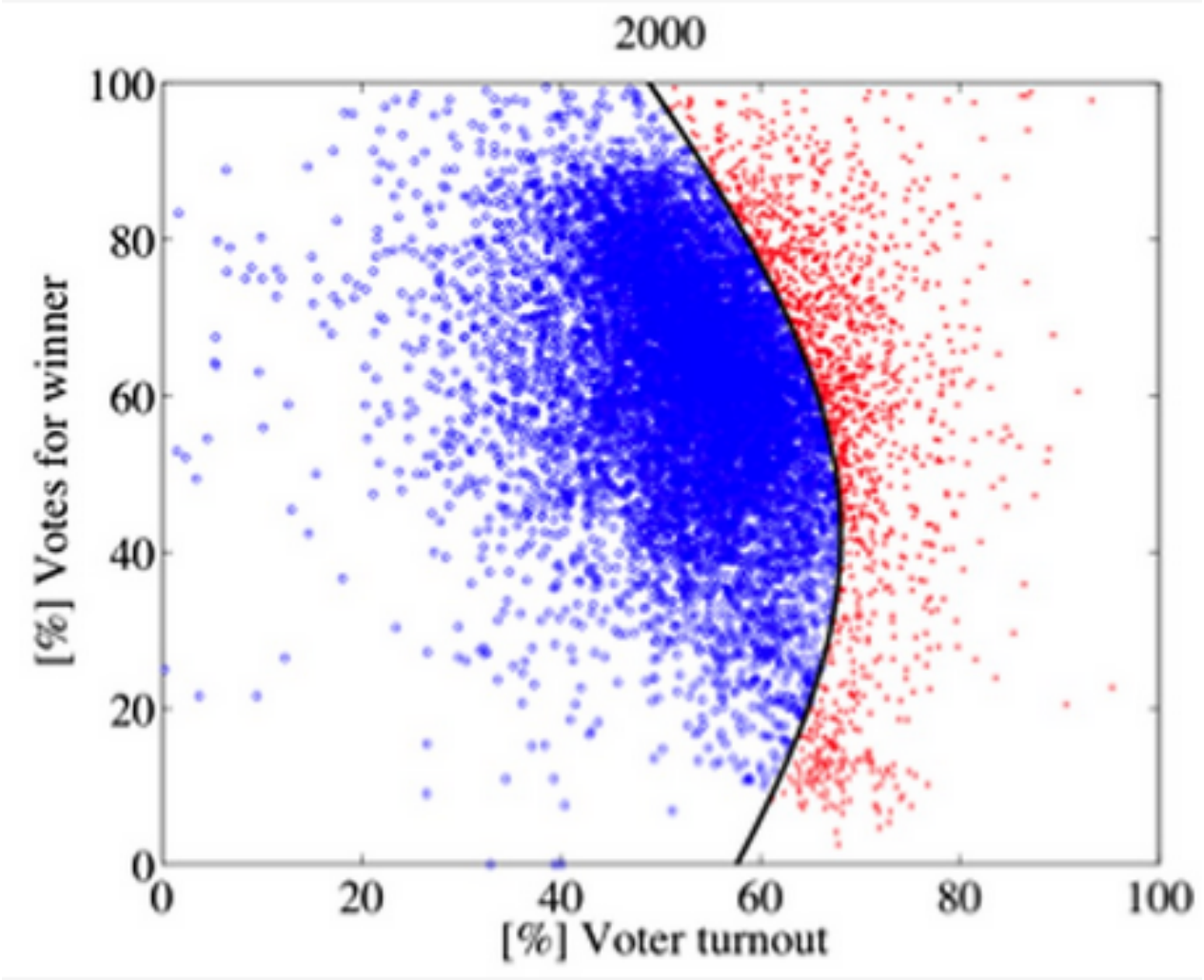


Piecewise Linear Decision Boundary[2]



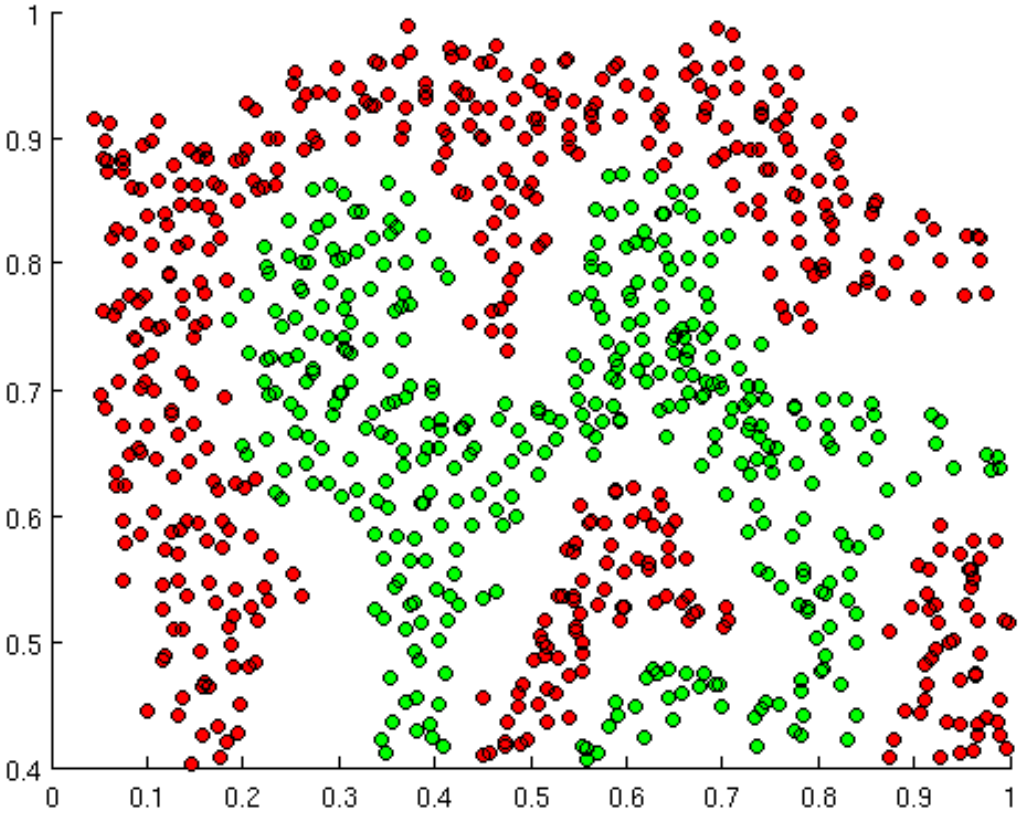
[2]ISLR Book

Quadratic Decision Boundary[3]



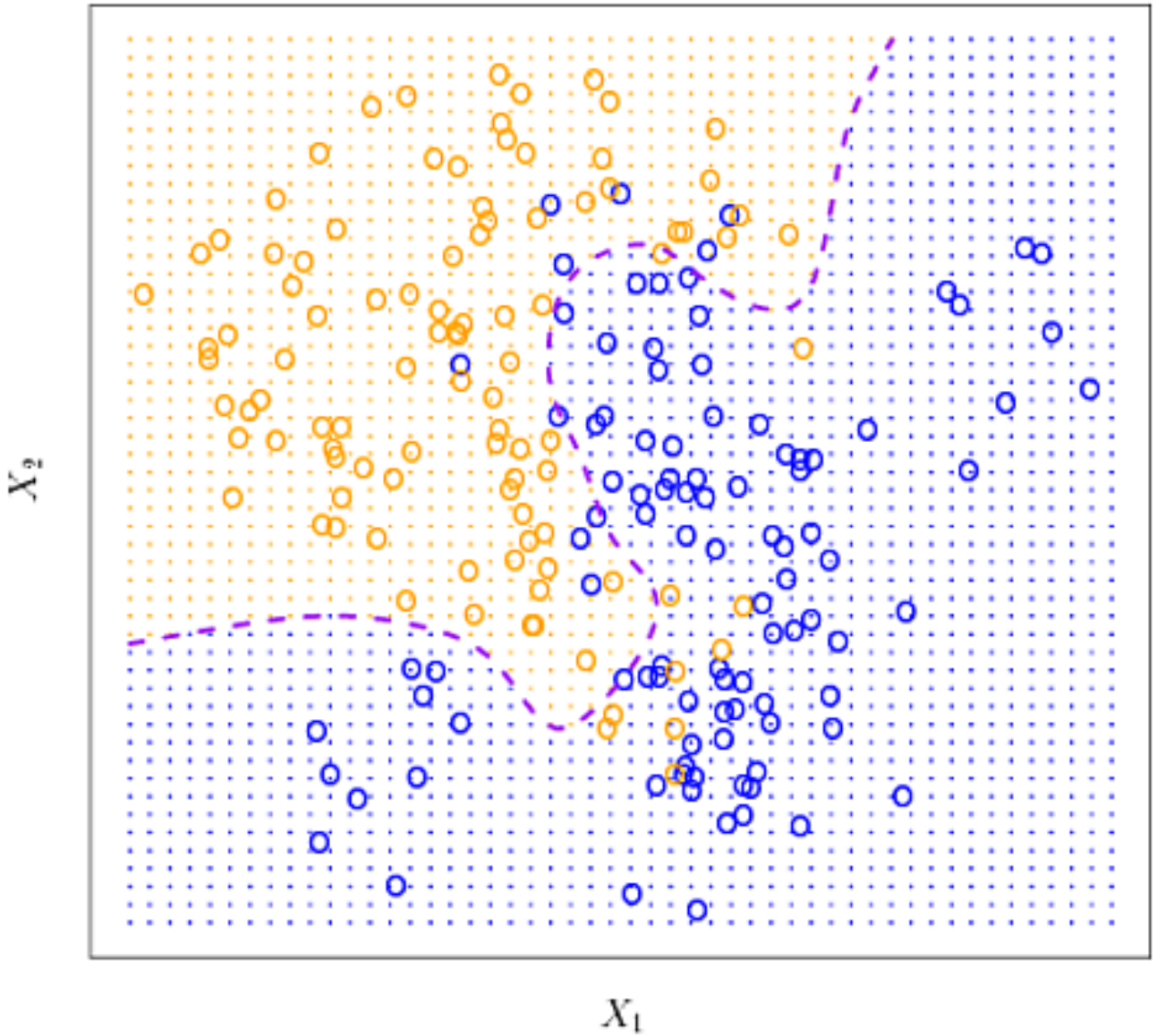
[3] Figshare.com

Non-linear Decision Boundary[4]



[4]ISLR Book

Complex Decision Boundary[5]

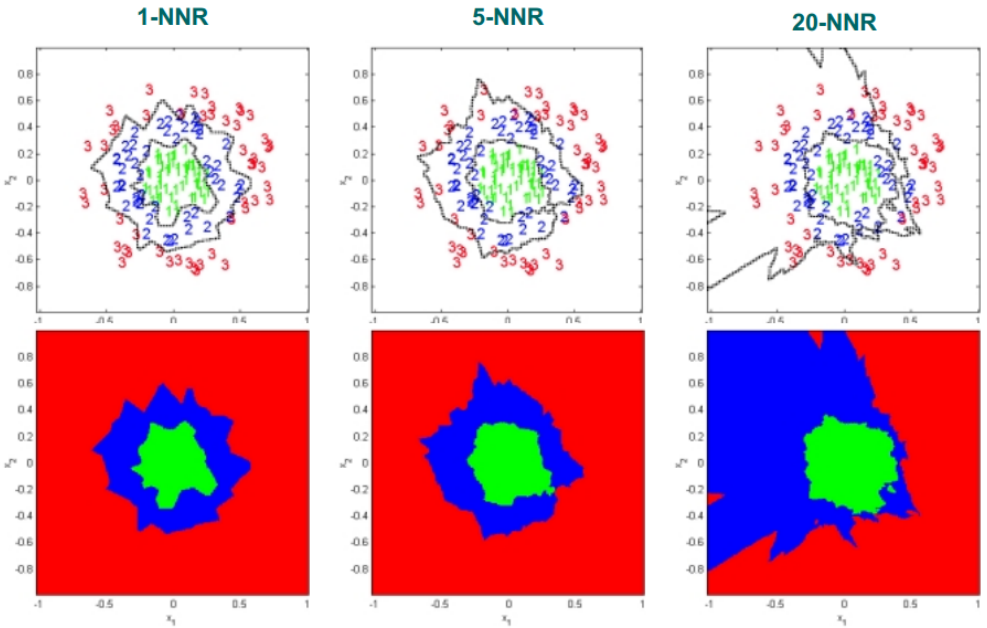


[5]

Classifier Selection Tips

- If decision boundary is linear, most *linear* classifiers will do well
- If decision boundary is non-linear, we sometimes have to use kernels
- If decision boundary is piece-wise, decision trees can do well
- If decision boundary is too complex, k-NN might be a good choice

k-NN Decision Boundary[6]



- Asymptotically Consistent: With infinite training data and large enough k, k-NN approaches the best possible classifier (Bayes Optimal)
- With infinite training data and large enough k, k-NN could approximate most possible decision boundaries

[6] ISLR Book

Decision Trees

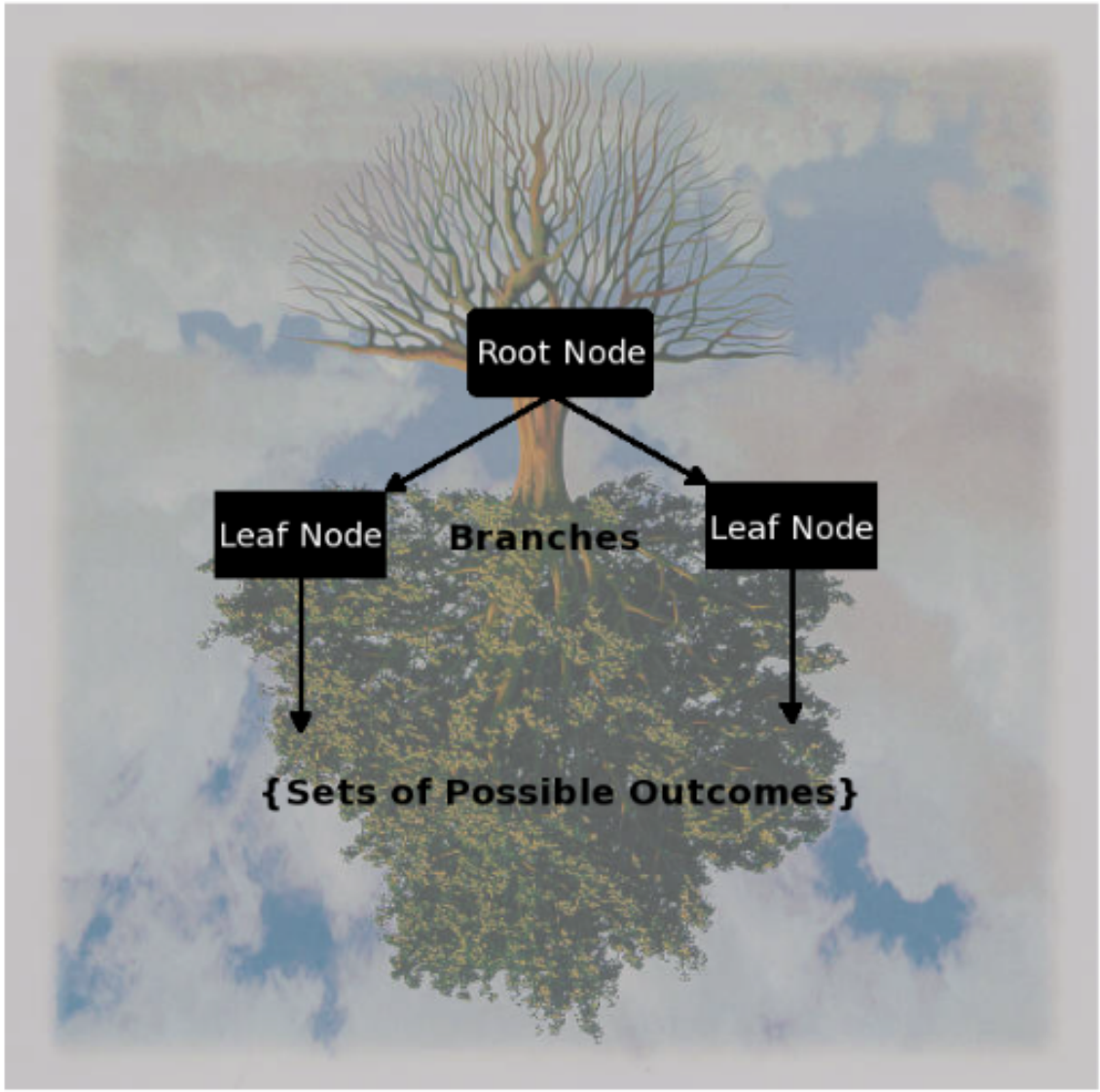
Strategies for Classifiers

- **Parametric Models:** Makes some assumption about data distribution such as density and often use explicit probability models
- **Non-parametric Models:** No prior assumption of data and determine decision boundaries directly.
  - k-NN
  - Decision tree

Tree[7]

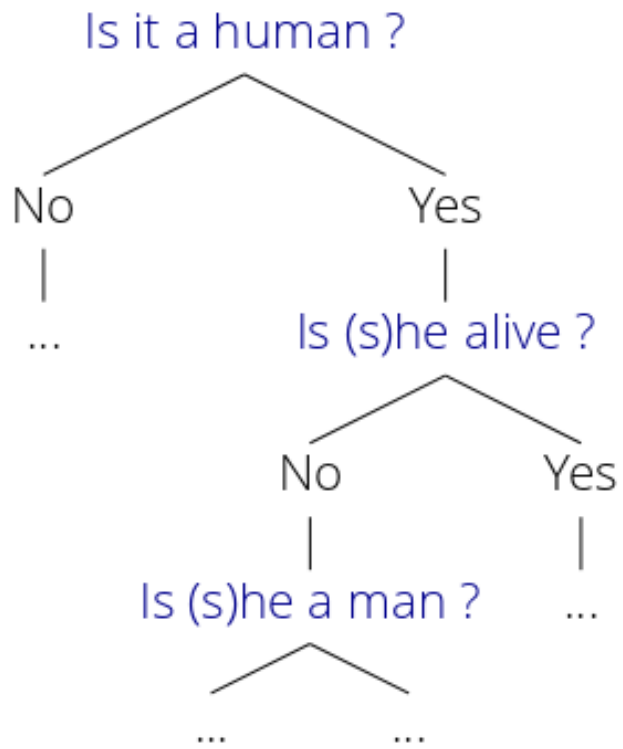
[7] http:

Binary Decision Tree[8]



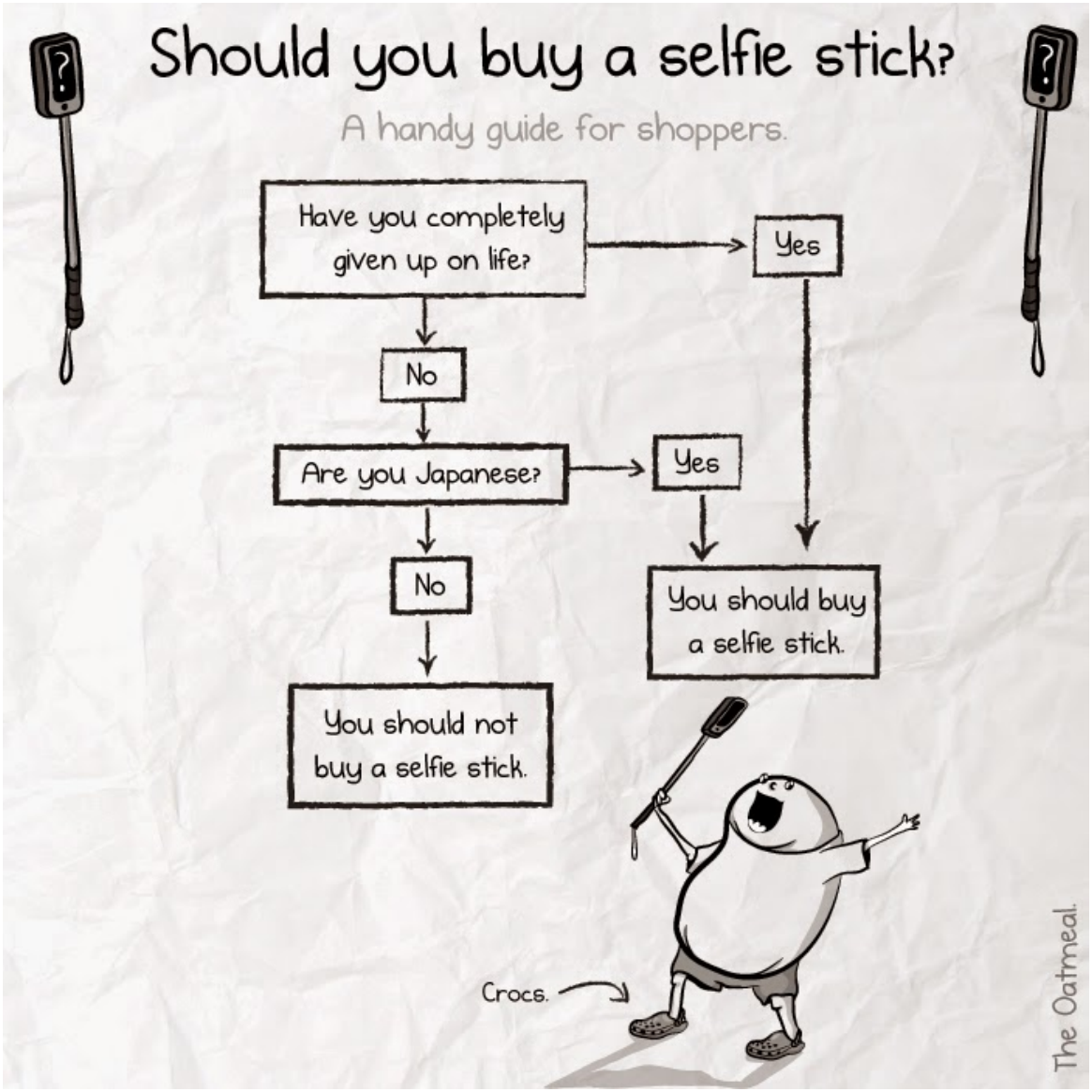
[8] http:

20 Question Intuition[9]



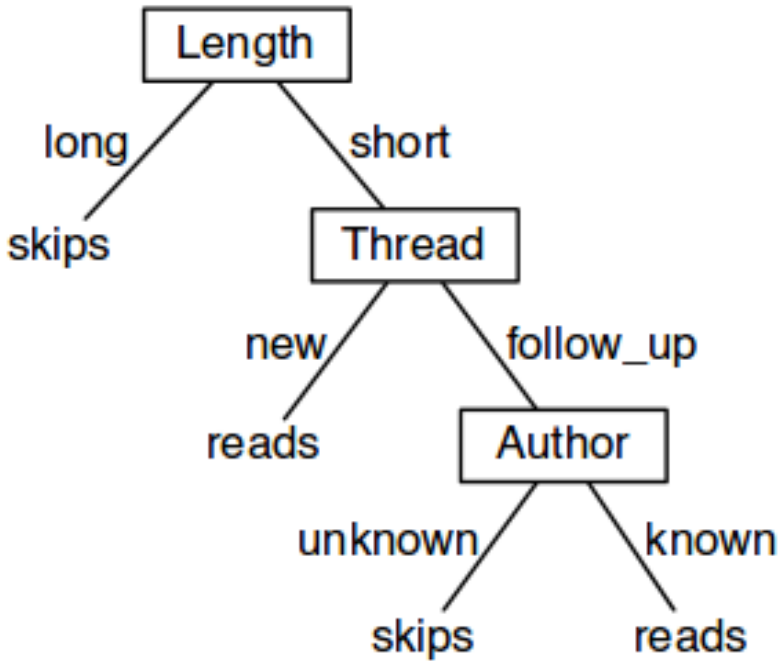
[9] http://www.idiap.ch/~fleuret/files/EE613/EE613-slides-6.pdf

Decision Tree for Selfie Stick[10]



[10] The Oatmeal Comics

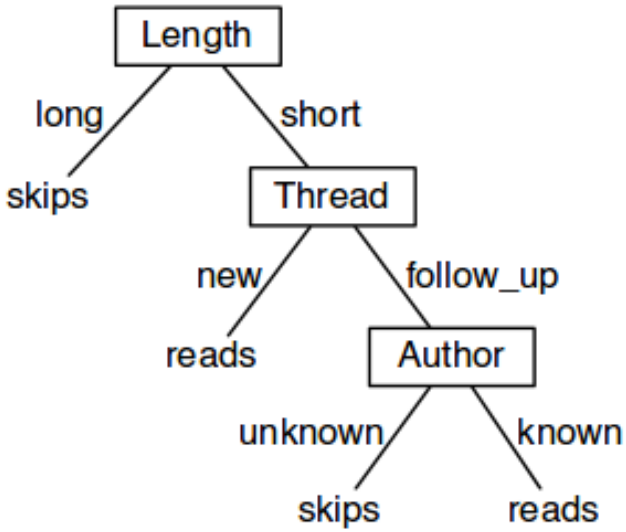
Decision Trees and Rules[11]



[11] http://artint.info/slides/ch07/lect3.pdf

Decision Trees and Rules[12]

- long $\rightarrow$ skips
- short $\wedge$ new $\rightarrow$ reads
- short $\wedge$ follow Up $\wedge$ known $\rightarrow$ reads
- short $\wedge$ follow Up $\wedge$ unknown $\rightarrow$ skips



[12] http://artint.info/slides/ch07/lect3.pdf

Building Decision Trees Intuition[13]

Horsepower	Weight	Mileage
95	low	low
90	low	low
70	low	high
86	low	high
76	high	low
88	high	low

Table: Car Mileage Prediction from 1971

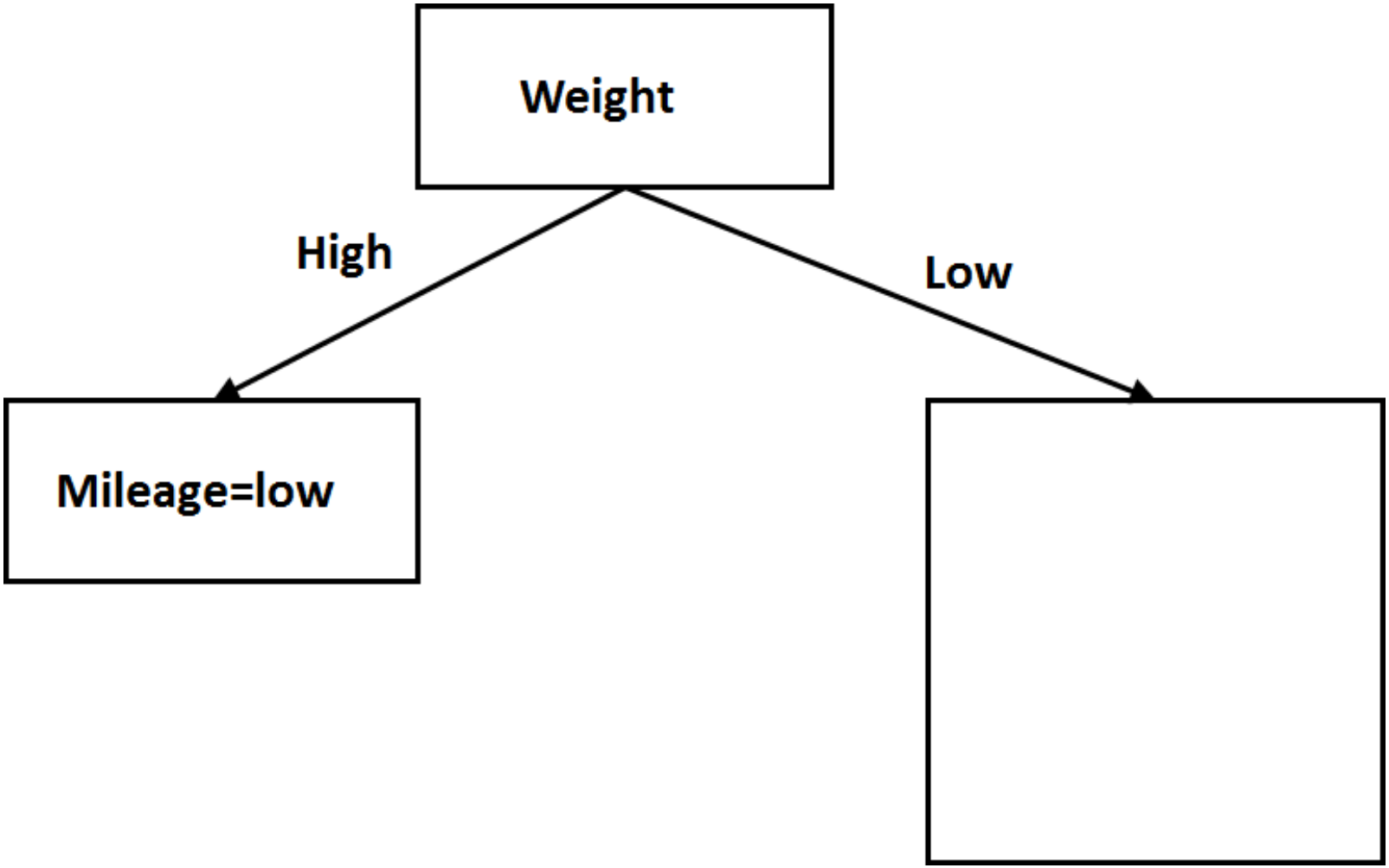
[13] http://spark-summit.org/wp-content/uploads/2014/07/Scalable-Distributed-Decision-Trees-in-Spark-Made-Das-Sparks-Talwalkar.pdf

Building Decision Trees Intuition

Horsepower	Weight	Mileage
95	low	low
90	low	low
70	low	high
86	low	high
76	high	low
88	high	low

Table: Car Mileage Prediction from 1971

Building Decision Trees Intuition



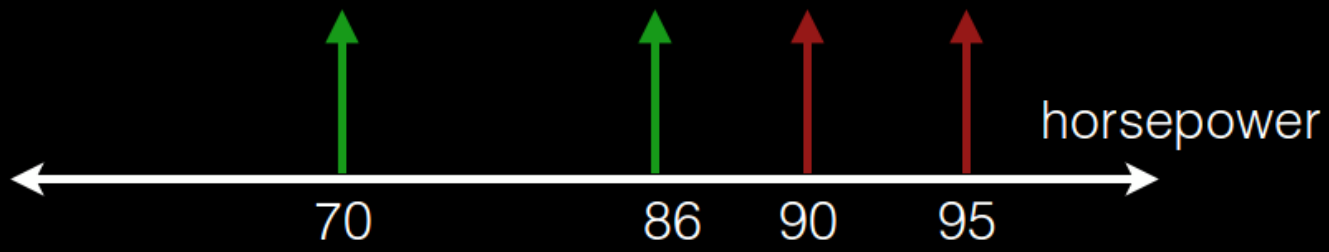
Building Decision Trees Intuition

Horsepower	Weight	Mileage
95	low	low
90	low	low
70	low	high
86	low	high

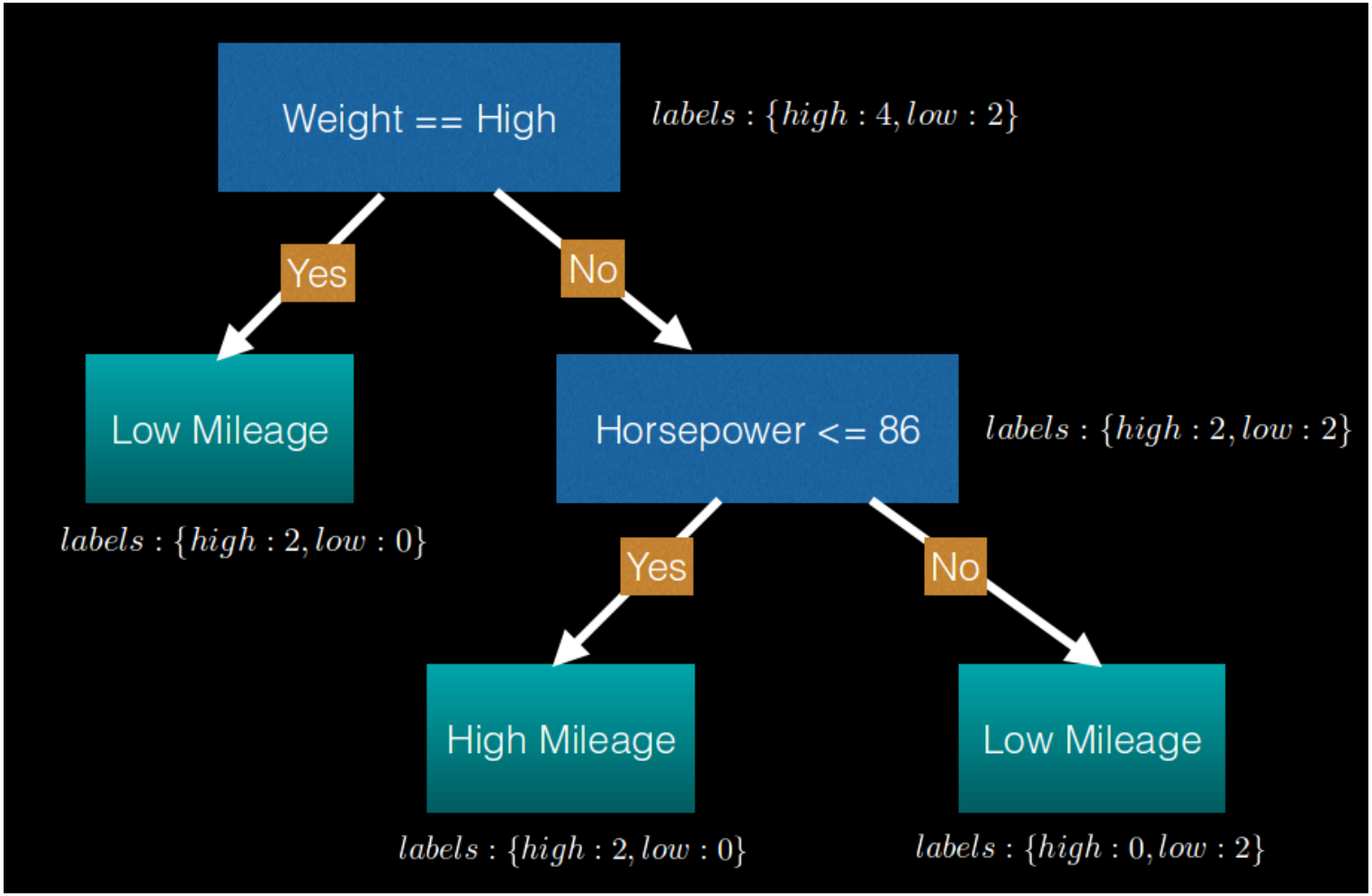
Table: Car Mileage Prediction from 1971

Building Decision Trees Intuition

horsepower	weight	mileage
95	low	low
90	low	low
70	low	high
86	low	high



Building Decision Trees Intuition



Building Decision Trees Intuition

Prediction:

horsepower	weight	mileage prediction
90	high	
80	low	
70	high	

Building Decision Trees Intuition

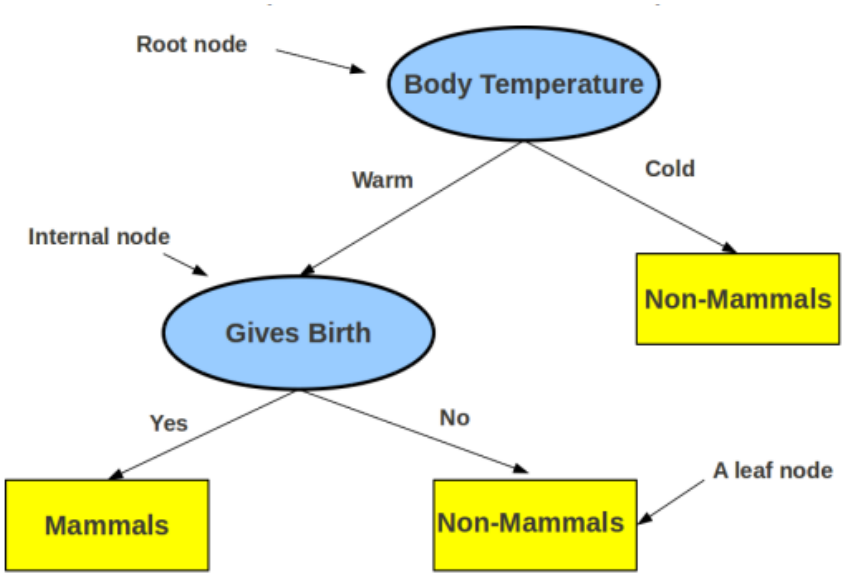
Prediction:

horsepower	weight	mileage prediction	
90	high	low	Correct!
80	low	low	Correct!
70	high	high	Wrong!

Learning Decision Trees

Decision Trees

- Defined by a **hierarchy** of rules (in form of a tree)



- Rules form the internal nodes of the tree (topmost internal node = root)
- Each rule (internal node) tests the value of some property the data
- Leaf nodes make the prediction

Decision Tree Learning

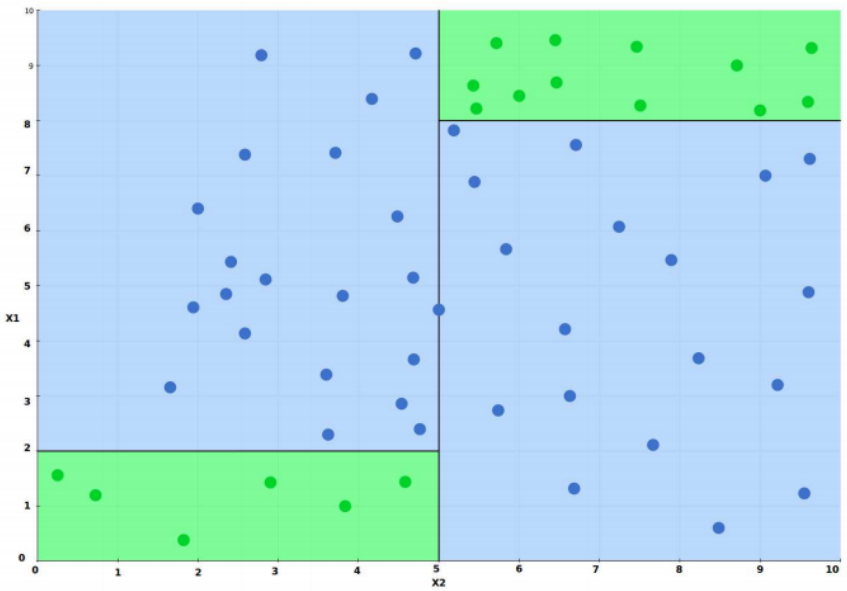
Objective:

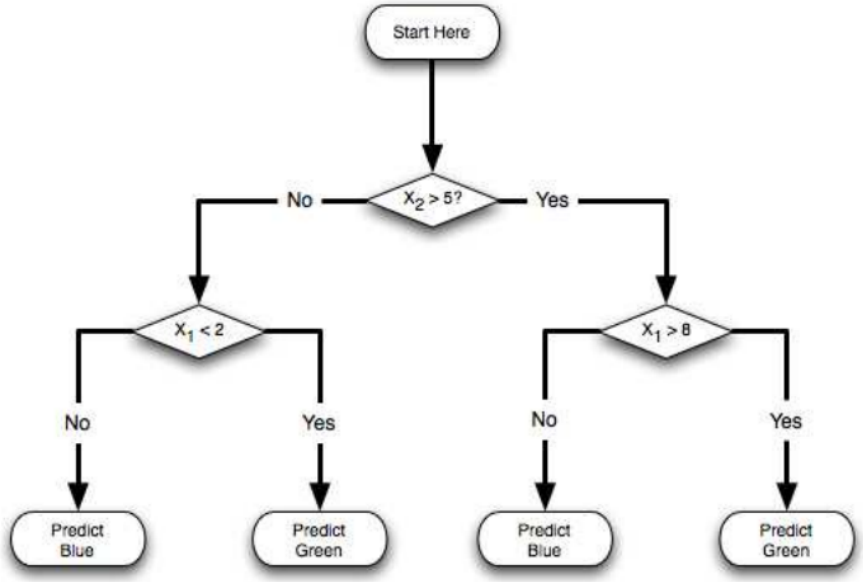
- Use the training data to construct a good decision tree
- Use the constructed Decision tree to predict labels for test inputs

Decision Tree Learning

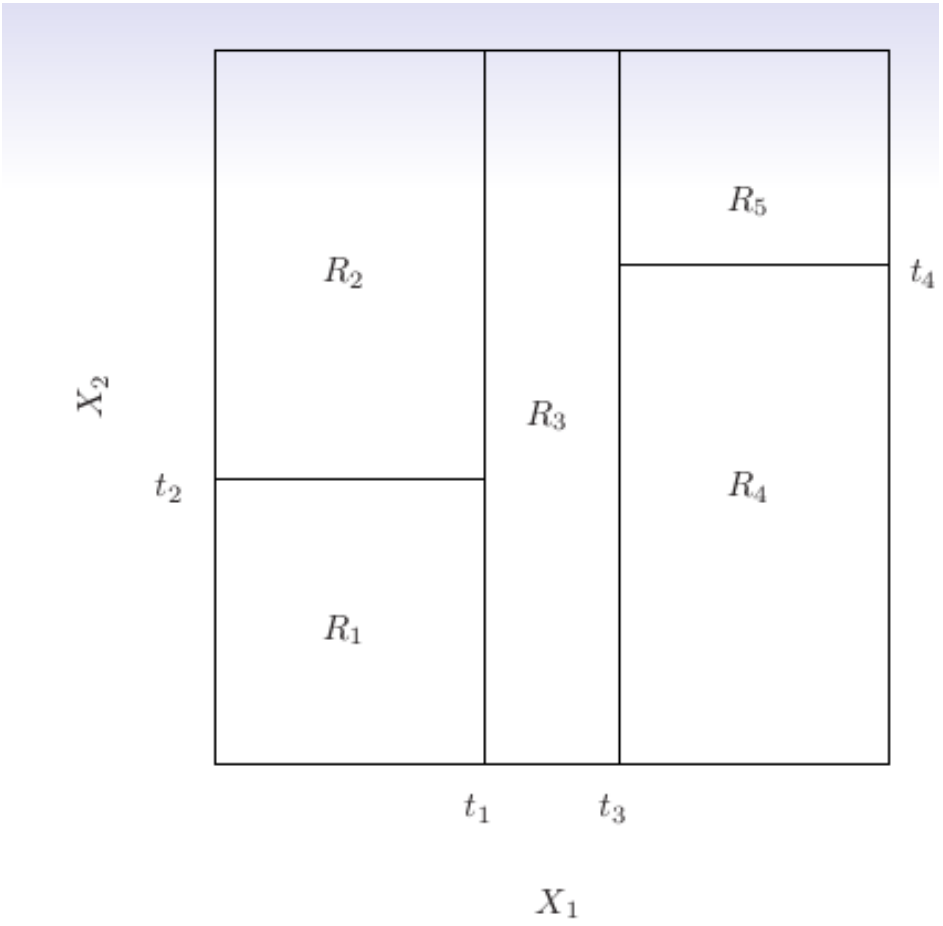
- Identifying the region (blue or green) a point lies in
  - A classification problem (blue vs green)
  - Each input has 2 features: co-ordinates $\{x_1, x_2\}$ in the 2D plane
- Once learned, the decision tree can be used to predict the region (blue/green) of a new test point

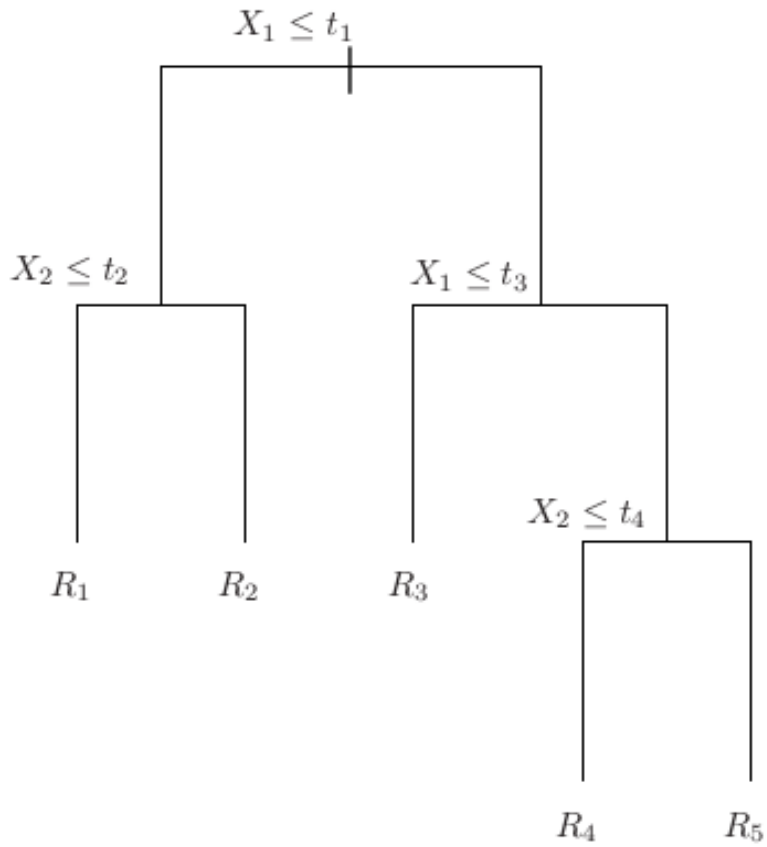
Decision Tree Learning





Expressiveness of Decision Trees

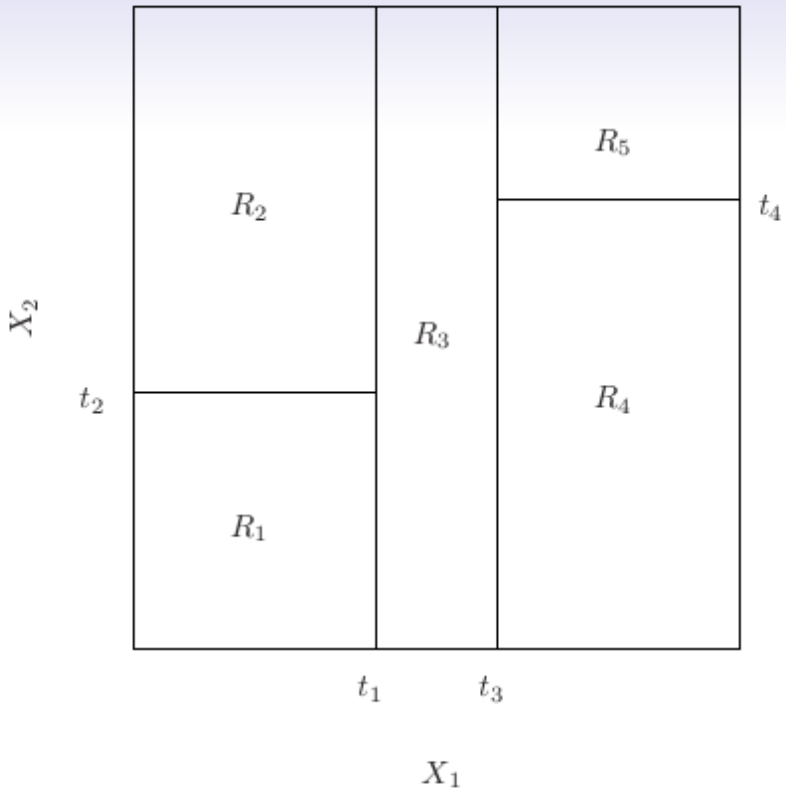


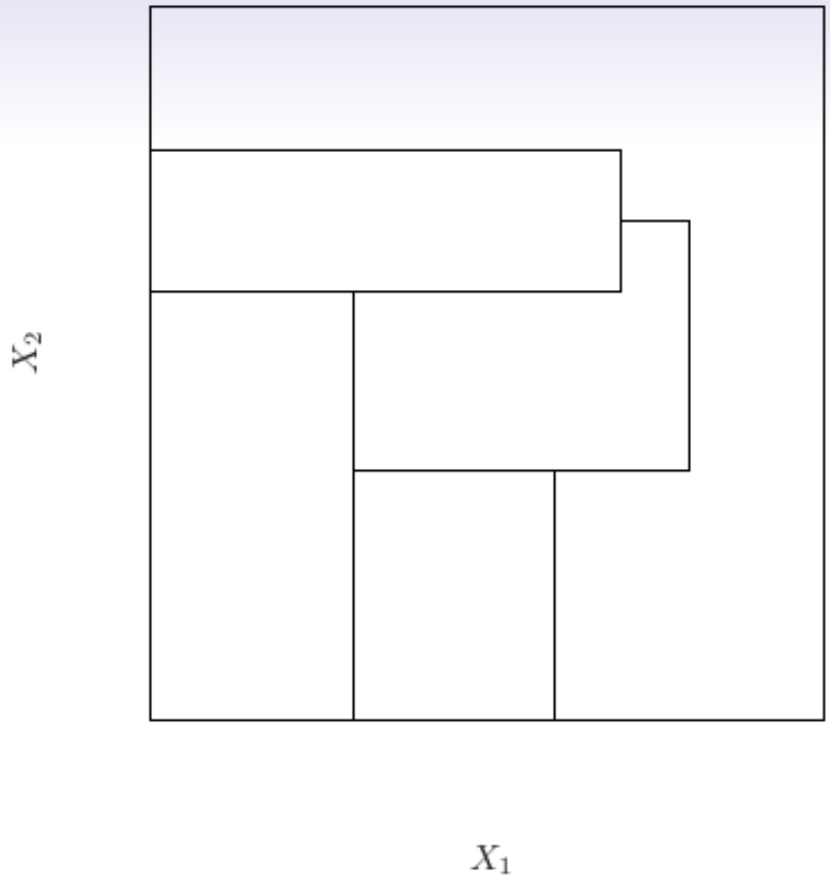


Expressiveness of Decision Trees

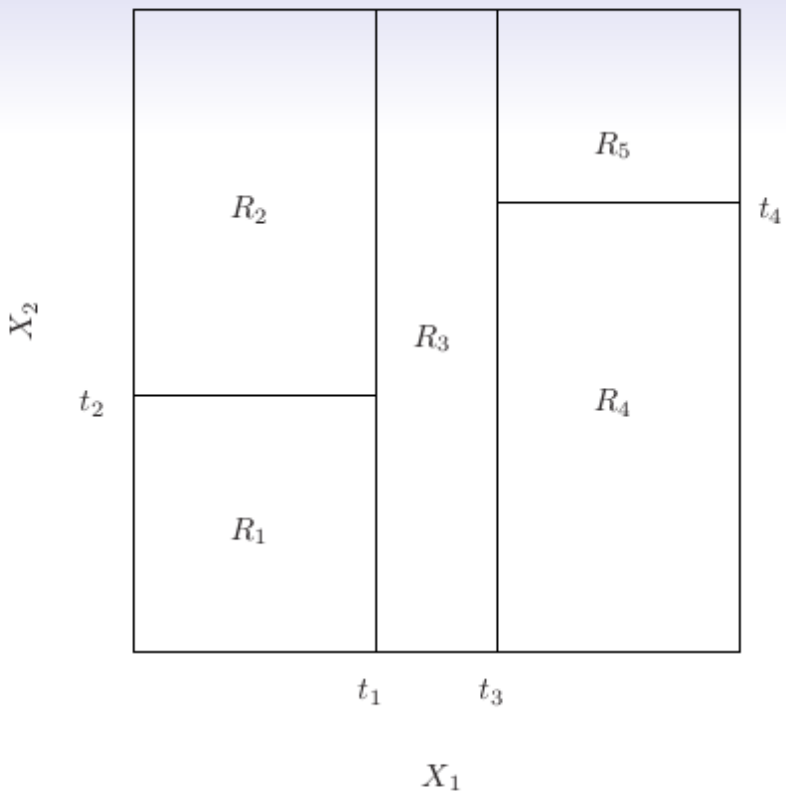
- Decision tree divides feature space into **axis-parallel** rectangles
- Each rectangle is labelled with one of the C classes
- Any partition of feature space by recursive binary splitting can be simulated by Decision Trees

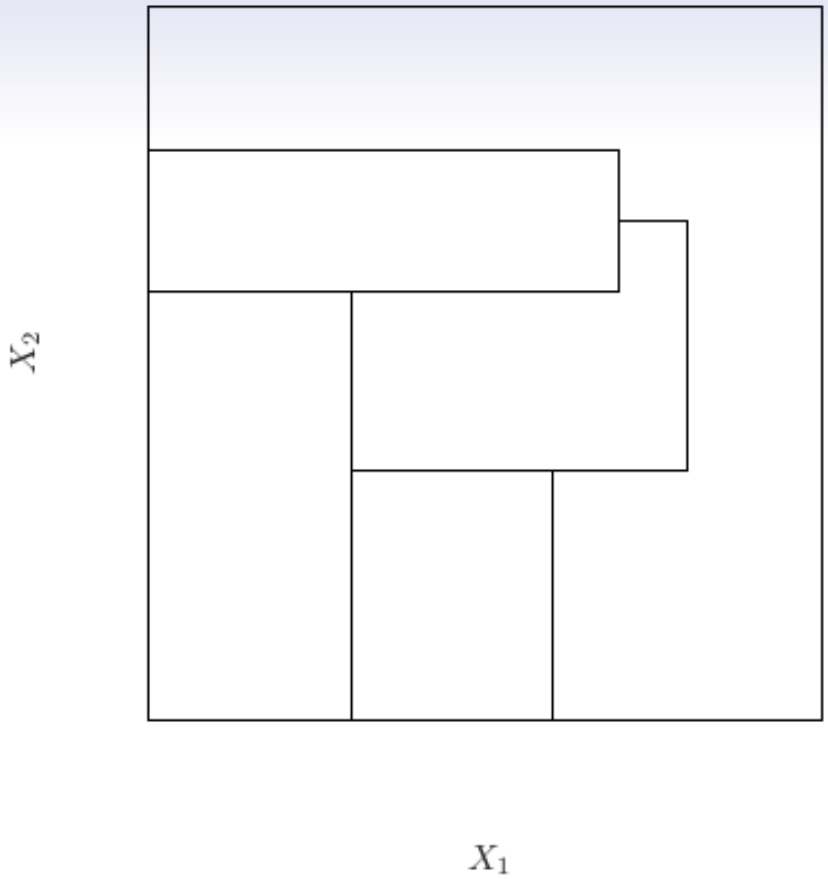
Expressiveness of Decision Trees





Expressiveness of Decision Trees



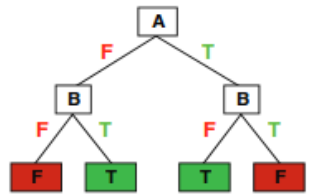


Feature space on left can be simulated by Decision tree but not the one on right.

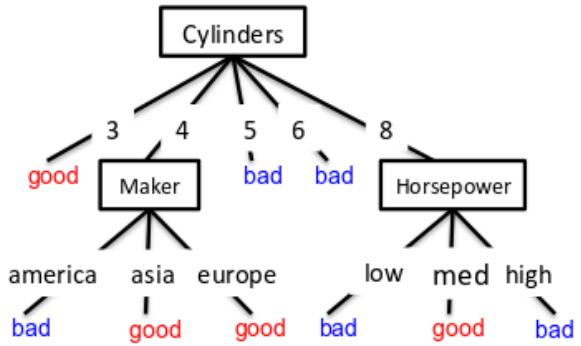
Expressiveness of Decision Tree

- Can express any logical function on input attributes
- Can express **any** boolean function
- For boolean functions, path to leaf gives truth table row
- Could require exponentially many nodes
- $cyl = 3 \vee (cyl = 4 \wedge (maker = asia \vee maker = europe)) \vee \ldots$

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



(Figure from Stuart Russell)



Hypothesis Space

- Exponential search space wrt set of attributes
- If there are d boolean attributes, then the search space has 2^{2^d} trees
  - If $d = 6$, then it is approximately $18,446,744,073,709,551,616$ (or approximately 1.8×10^{18})
  - If there are d boolean attributes, each truth table has 2^d rows
  - Hence there must be 2^{2^d} truth tables that can take all possible variations
  - Alternate argument: the number of trees is same as
    - number of bolean functions with d variables
    - $=$ number of distinct truth tables with 2^d rows $= 2^{2^d}$
- NP-Complete to find optimal decision tree
- Idea: Use greedy approach to find a locally optimal tree

Decision Tree Learning Algorithms

- 1966: Hunt and colleagues from Psychology developed first known algorithm for human concept learning
- 1977: Breiman, Friedman and others from Statistics developed CART
- 1979: Quinlan developed proto-ID3
- 1986: Quinlan published ID3 paper
- 1993: Quinlan's updated algorithm C4.5
- 1980's and 90's: Improvements for handling noise, continuous attributes, missing data, non-axis parallel DTs, better heuristics for pruning, overfitting, combining DTs

Decision Tree Learning Algorithms

Main Loop:

1. Let A be the "best" decision attribute for next node
2. Assign A as decision attribute for node
3. For each value of A, create a new descendent of node
4. Sort training examples to leaf nodes
5. If training examples are perfectly classified, then STOP else iterate over leaf nodes

Recursive Algorithm for Learning Decision Trees

DT(*Examples*, *Labels*, *Features*):

- If all examples are positive, return a single node tree *Root* with label = +
- If all examples are negative, return a single node tree *Root* with label = -
- If all features exhausted, return a single node tree *Root* with majority label
- Otherwise, let F be the feature having the highest information gain
- $Root \leftarrow F$
- For each possible value f of F
  - Add a tree branch below *Root* corresponding to the test $F = f$
  - Let $Examples_f$ be the set of examples with feature F having value f
  - Let $Labels_f$ be the corresponding labels
  - If $Examples_f$ is empty, add a leaf node below this branch with label = most common label in *Examples*
  - Otherwise, add the following subtree below this branch:

    $$\mathrm{DT}(Examples_f, Labels_f, Features - \{F\})$$

  - Note: *Features* - $\{F\}$ removes feature F from the feature set *Features*

Decision Tree Learning

- Greedy Approach: Build tree, top-down by choosing one attribute at a time
- Choices are locally optimal and may or may not be globally optimal
- Major issues
  - Selecting the next attribute
  - Given an attribute, how to specify the split condition
  - Determining termination condition

Termination Condition

Stop expanding a node further when:

Termination Condition

Stop expanding a node further when:

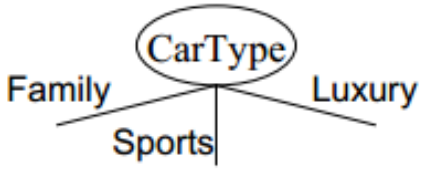
- It consist of examples all having the same label
- Or we run out of features to test!

How to Specify Test Condition?

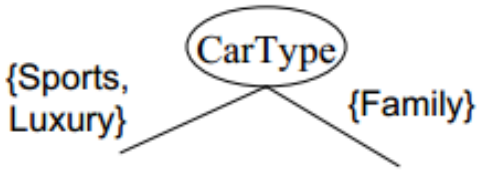
- Depends on attribute types
  - Nominal
  - Ordinal
  - Continuous
- Depends on number of ways to split
  - 2-way split
  - Multi-way split

Splitting based on Nominal Attributes

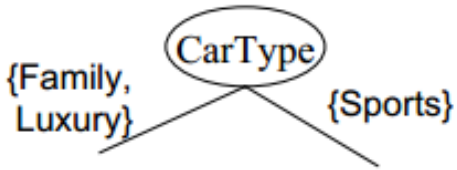
- **Multi-way split:** Use as many partitions as distinct values.



- **Binary split:** Divides values into two subsets. Need to find optimal partitioning.

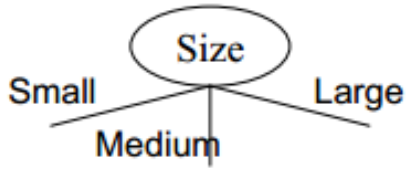


OR

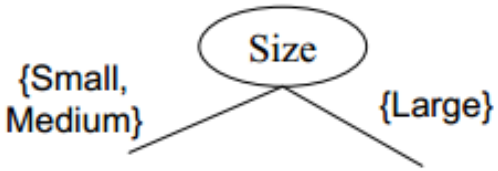


Splitting based on Ordinal Attributes

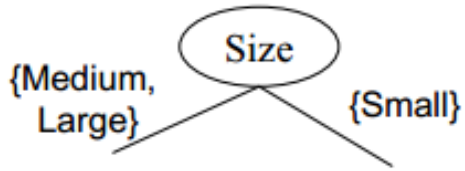
- Multi-way split: Use as many partitions as distinct values.



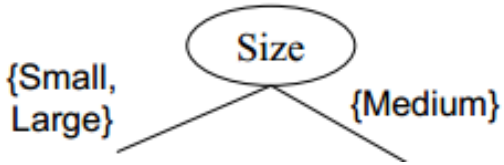
- Binary split: Divides values into two subsets. Need to find optimal partitioning.



OR



- What about this split?



Splitting based on Continuous Attributes

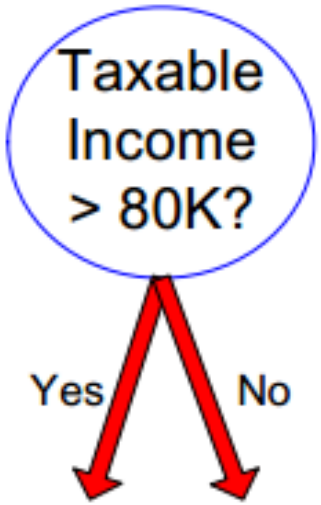
How to split continuous attributes such as Age, Income etc

Splitting based on Continuous Attributes

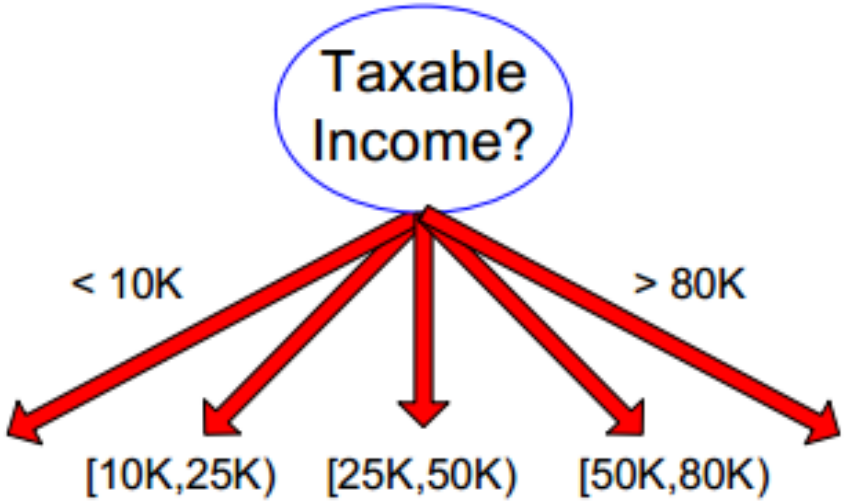
How to split continuous attributes such as Age, Income etc

- **Discretization** to form an ordinal categorical attribute
  - *Static:* discretize once at the beginning
  - *Dynamic:* find ranges by equal interval bucketing, equal frequency bucketing, percentiles, clustering etc
- **Binary Decision:** $(A < v) or (A \geq v)$
  - Consider all possible split and find the best cut
  - Often, computationally intensive

Splitting based on Continuous Attributes

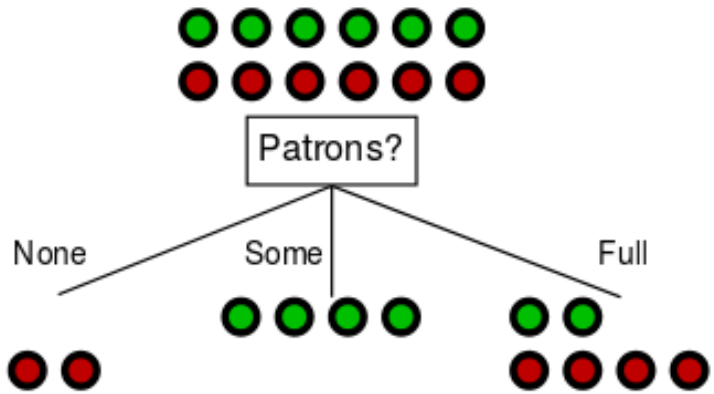


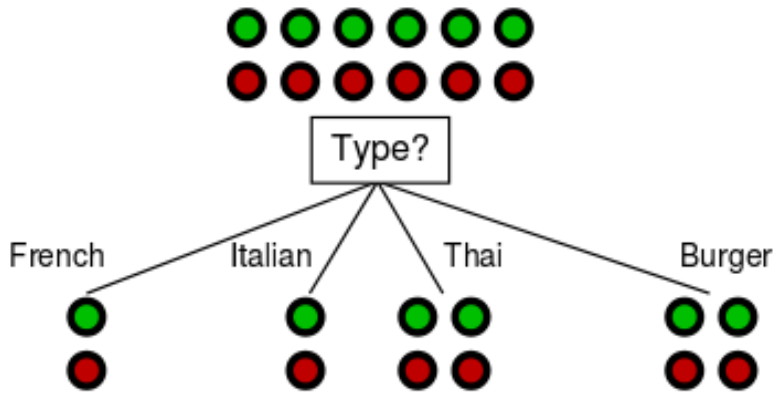
(i) Binary split



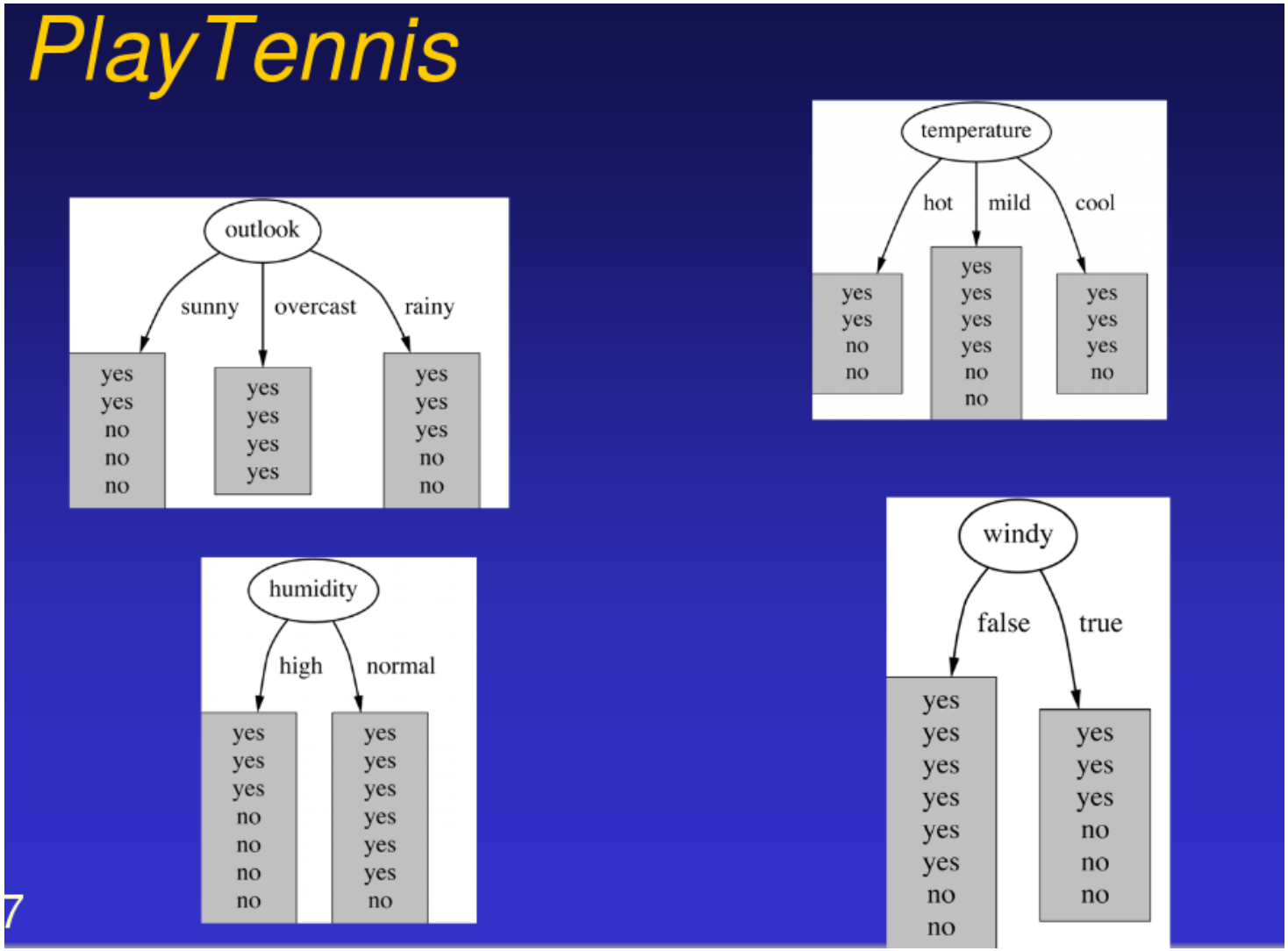
(ii) Multi-way split

Choosing the next Attribute - I



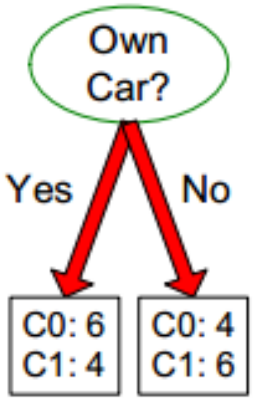


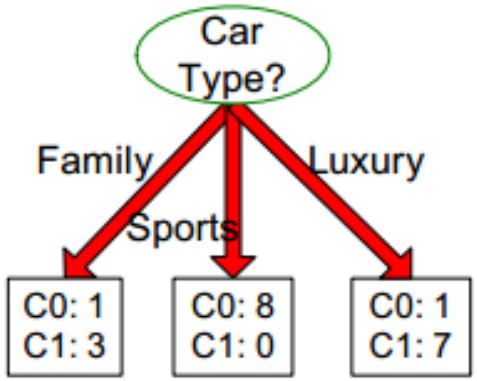
Choosing the next Attribute - II[14]

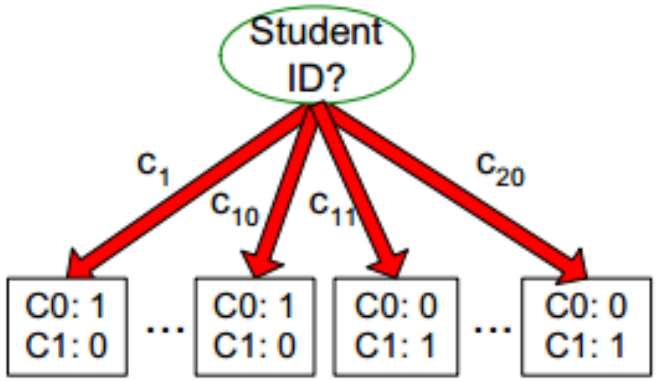


[14] http://www.cedar.buffalo.edu/~srihari/CSE574/Chap16/Chap16.1-InformationGain.pdf

Choosing the next Attribute - III







Choosing an Attribute

- Good Attribute

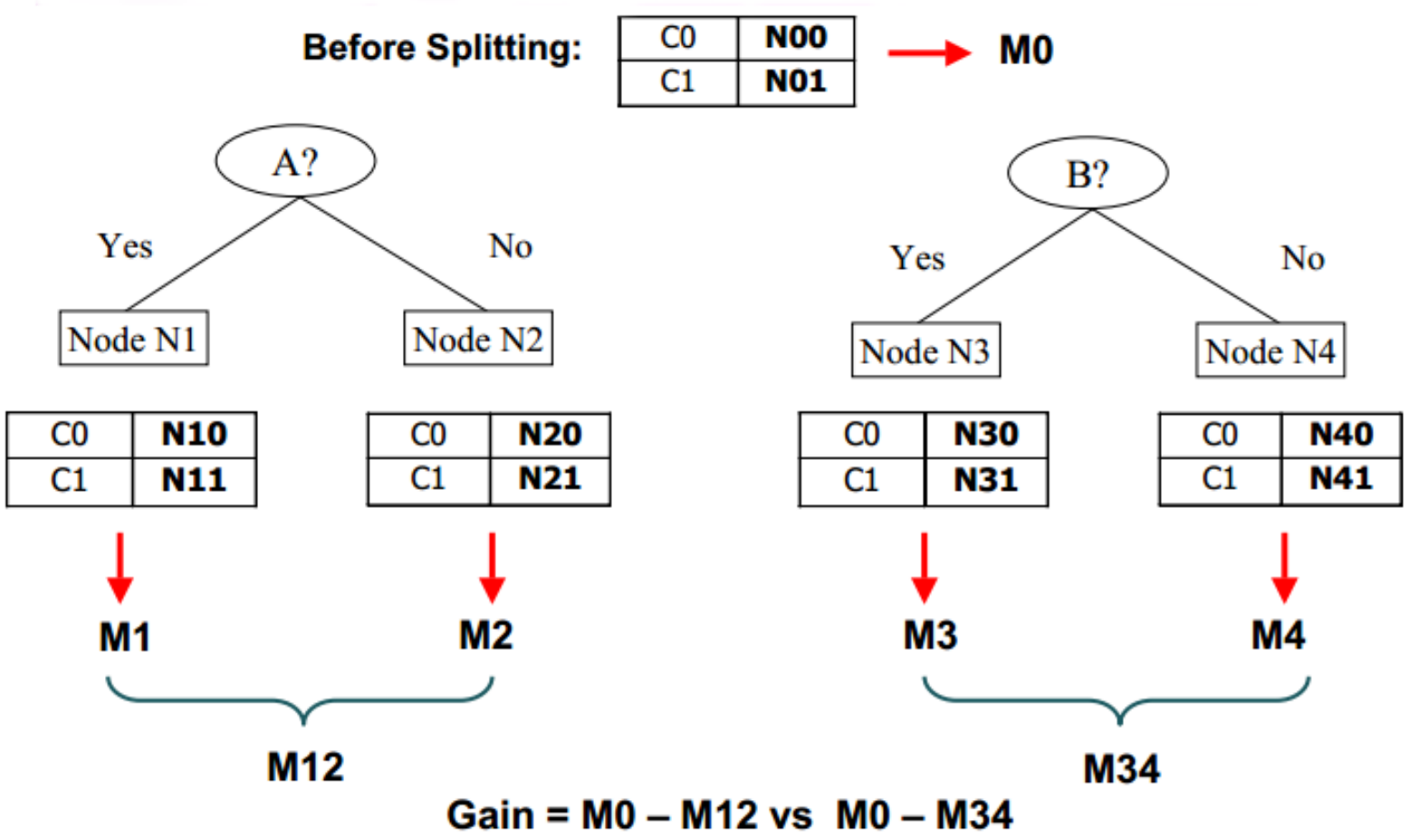
Choosing an Attribute

- Good Attribute
  - for one value we get all instances as positive
  - for other value we get all instances as negative
- Bad Attribute

Choosing an Attribute

- Good Attribute
  - for one value we get all instances as positive
  - for other value we get all instances as negative
- Bad Attribute
  - it provides no discrimination
  - attribute is immaterial to the decision
  - for each value we have same number of positive and negative instances

How to Find the Best Split?



Measures of Node Impurity

- Gini Index
- Entropy
- Misclassification Error

Gini Index

- An important measure of statistical dispersion
- Used in Economics to measure income inequality in countries
- Proposed by Corrado Gini

Gini Index

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Gini = 1 – P(C1)² – P(C2)² = 1 – 0 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Gini = 1 – (1/6)² – (5/6)² = 0.278

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Gini = 1 – (2/6)² – (4/6)² = 0.444

Gini Index

- Gini Index for a given node t :

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

  - Maximum (1 - $1/n_c$) when records are equally distributed among all classes, implying least interesting information
  - Minimum (0.0) when all records belong to one class, implying most interesting information

C1	0
C2	6
Gini=0.000	

C1	1
C2	5
Gini=0.278	

C1	2
C2	4
Gini=0.444	

C1	3
C2	3
Gini=0.500	

Splitting Based on Gini

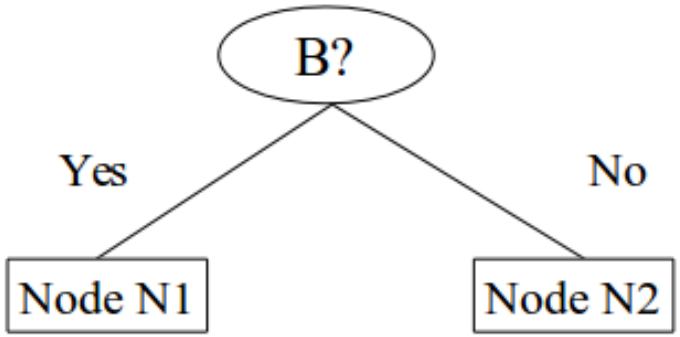
- Used in CART, SLIQ, SPRINT
- When a node p is split into k partitions (children), the quality of split is computed as,

$$Gini_{split} = \sum_{i=1}^{k} \frac{n_i}{n} Gini(i)$$

  - n_i = number of records at child i
  - n = number of records at node p

Gini Index for Binary Attributes

- Splits into two partitions
- Effect of Weighing partitions:
  - Larger and Purer Partitions are sought for.



	Parent
C1	6
C2	6
Gini = 0.500	

Gini(N1)
= 1 – $(5/7)^2$ – $(2/7)^2$
= 0.408

Gini(N2)
= 1 – $(1/5)^2$ – $(4/5)^2$
= 0.32

	N1	N2
C1	5	1
C2	2	4
Gini=0.371		

Gini(Children)
= 7/12 * 0.408 +
5/12 * 0.32
= 0.371

Gini Index for Categorical Attributes

- For each distinct value, gather counts for each class in the dataset
- Use the count matrix to make decisions

Multi-way split

	CarType		
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	0.393		

Two-way split
(find best partition of values)

	CarType	
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

	CarType	
	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini	0.419	

Entropy and Information Gain

- You are watching a set of independent random samples of a random variable X
- Suppose the probabilities are equal: $P(X = A) = P(X = B) = P(X = C) = P(X = D) = \frac{1}{4}$
- Suppose you see a text like $BAAC$
- You want to transmit this information in a **binary** communication channel
- How many bits will you need to transmit this information?

Entropy and Information Gain

- You are watching a set of independent random samples of a random variable X
- Suppose the probabilities are equal: $P(X = A) = P(X = B) = P(X = C) = P(X = D) = \frac{1}{4}$
- Suppose you see a text like $BAAC$
- You want to transmit this information in a **binary** communication channel
- How many bits will you need to transmit this information?
- Simple idea: Represent each character via 2 bits: $A = 00, B = 01, C = 10, D = 11$
- So, $BAAC$ becomes 01000010
- Communication Complexity: 2 on average bits per symbol

Entropy and Information Gain

- Suppose you knew probabilities are unequal:
  $P(X = A) = \frac{1}{2}, P(X = B) = \frac{1}{4}, P(X = C) = P(X = D) = \frac{1}{8}$
- It is now possible to send information 1.75 bits on average per symbol

Entropy and Information Gain

- Suppose you knew probabilities are unequal:
  $P(X = A) = \frac{1}{2}, P(X = B) = \frac{1}{4}, P(X = C) = P(X = D) = \frac{1}{8}$
- It is now possible to send information 1.75 bits on average per symbol
- Choose a frequency based code!
- $A = 0, B = 10, C = 110, D = 111$

Entropy and Information Gain

- Suppose you knew probabilities are unequal: $P(X = A) = \frac{1}{2}, P(X = B) = \frac{1}{4}, P(X = C) = P(X = D) = \frac{1}{8}$
- It is now possible to send information 1.75 bits on average per symbol
- Choose a frequency based code!
- $A = 0, B = 10, C = 110, D = 111$
- $BAAC$ becomes 1000110
- Requires only 7 bits for transmitting $BAAC$

Entropy

- Entropy at a given node t:

$$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

  - Measures homogeneity of a node.
    - Maximum (log n_c) when records are equally distributed among all classes implying least information
    - Minimum (0.0) when all records belong to one class, implying most information
  - Entropy based computations are similar to the GINI index computations

Entropy

$$Entropy(t) = -\sum_j p(j \mid t) \log_2 p(j \mid t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

Entropy

- Information Gain:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

Parent Node, p is split into k partitions;

n_i is number of records in partition i

  - Measures Reduction in Entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN)
  - Used in ID3 and C4.5
  - Disadvantage: Tends to prefer splits that result in large number of partitions, each being small but pure.

Entropy

- Gain Ratio:

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO} \qquad SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

Parent Node, p is split into k partitions

n_i is the number of records in partition i

- Adjusts Information Gain by the entropy of the partitioning (SplitINFO). Higher entropy partitioning (large number of small partitions) is penalized!
- Used in C4.5
- Designed to overcome the disadvantage of Information Gain

Splitting based on Classification Error

- Classification error at node t is

$$Error(t) = 1 - max_i P(i|t)$$

- Measures misclassification error made by a node.
  - Minimum (0.0) when all records belong to one class, implying most interesting information
  - Maximum $(1 - \frac{1}{n_c})$ when records are equally distributed among all classes, implying least interesting information

Classification Error

$$Error(t) = 1 - \max_i P(i \mid t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Error = 1 – max (0, 1) = 1 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

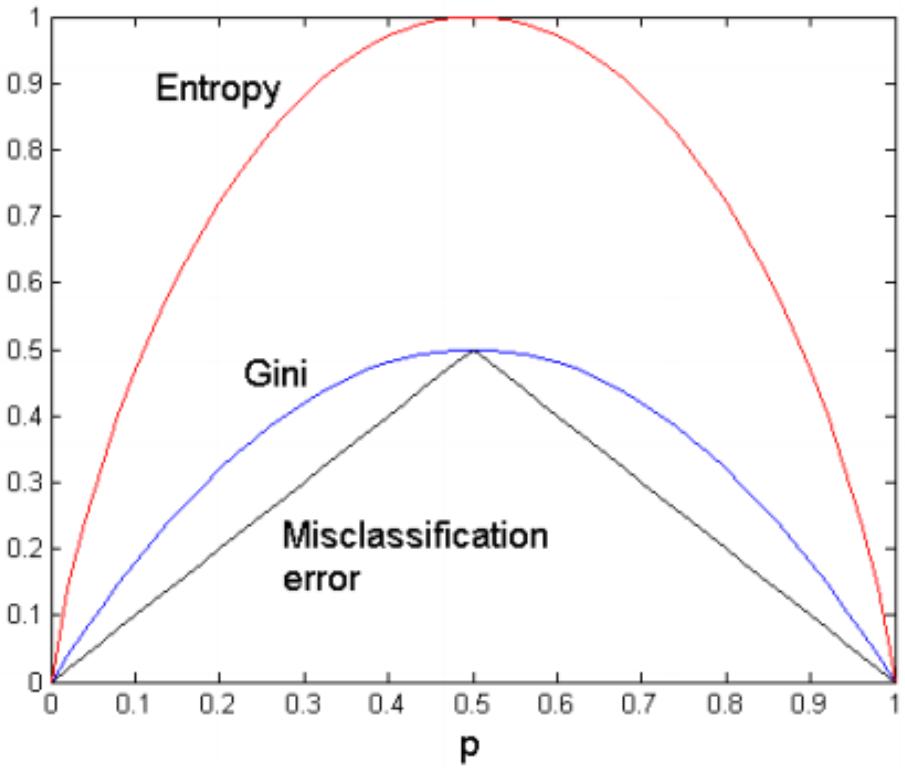
C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Error = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

Comparison among Splitting Criteria

For a 2-class problem:



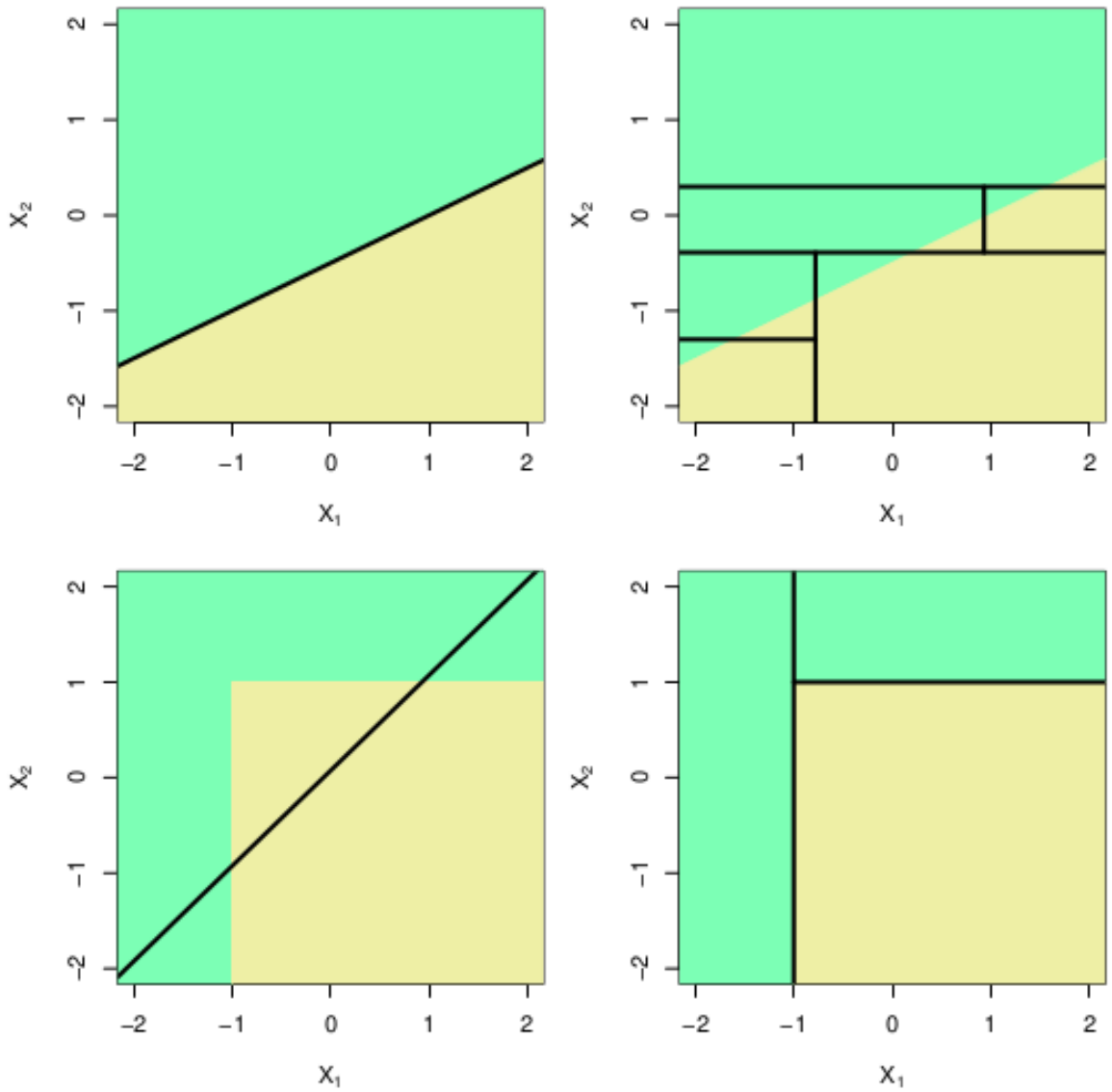
Splitting Criteria

- Gini Index (CART, SLIQ, SPRINT):
  - select attribute that minimize impurity of a split
- Information Gain (ID3, C4.5)
  - select attribute with largest information gain
- Normalized Gain ratio (C4.5)
  - normalize different domains of attributes
- Distance normalized measures (Lopez de Mantaras)
  - define a distance metric between partitions of the data
  - chose the one closest to the perfect partition
- χ^2 contingency table statistics (CHAID)
  - measures correlation between each attribute and the class label
  - select attribute with maximal correlation

Overfitting in Decision Trees

- Decision trees will always overfit in the absence of label noise
- Simple strategies for fixing:
  - Fixed depth
  - Fixed number of leaves
  - Grow the tree till the gain is above some threshold
  - Post pruning

Trees vs Linear Models



Advantages and Disadvantages

- Very easy to explain to people.
- Some people believe that decision trees more closely mirror human decision-making
- Trees can be displayed graphically, and are easily interpreted even by a non-expert (especially if they are small)
- Trees can easily handle qualitative predictors without the need to create dummy variables.
- Inexpensive to construct
- Extremely fast at classifying new data
- Unfortunately, trees generally do not have the same level of predictive accuracy as other classifiers

Summary

Major Concepts:

- Geometric interpretation of Classification
- Decision trees

Slide Material References

- Slides from ISLR book
- Slides by Piyush Rai
- Slides for Chapter 4 from "Introduction to Data Mining" book by Tan, Steinbach, Kumar
- Slides from Andrew Moore, CMU
- See also the footnotes