

MIT INSTITUTE FOR DATA,
SYSTEMS, AND SOCIETY



IDSS

Applied Data Science Program

DECISION TREES

Munther A. Dahleh

Introduction



Prof. Munther (Munzer) Dahleh
<https://dahleh.lids.mit.edu>

Outline

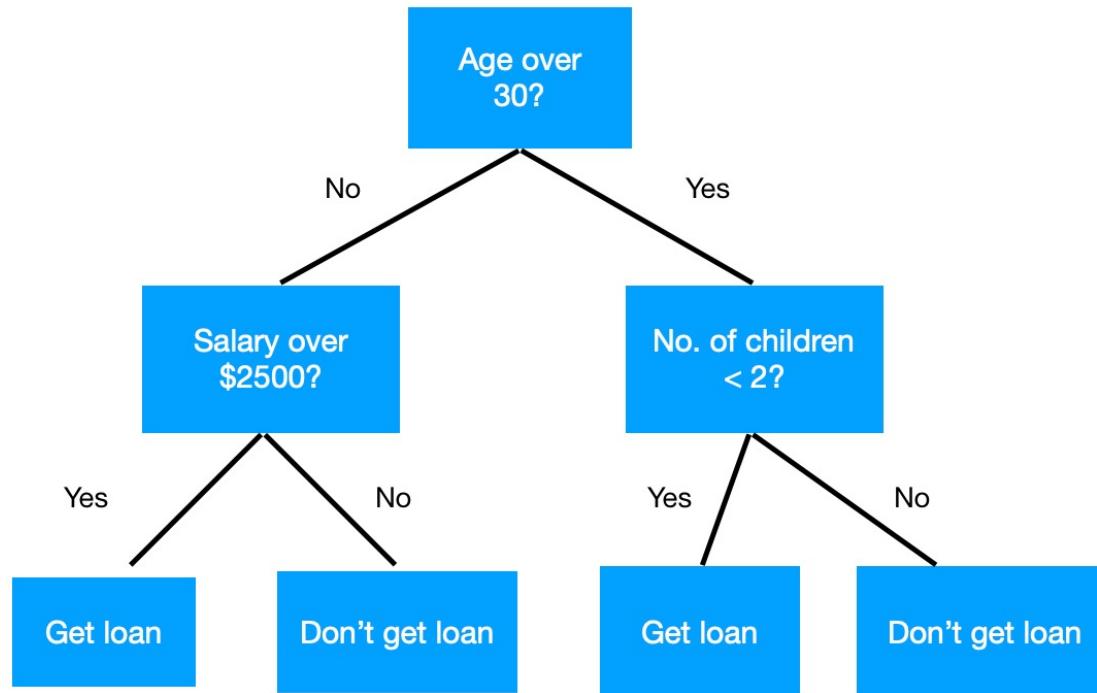
- Learning may require more than linear regression
- Data may not be static (IID)
- Next three lectures:
 - Decision Trees
 - Random Forest
 - Time Series

Decision Trees Outline

- Part I: Definitions, interpretations
- Part II: Learning a Decision Tree
- Part III: Entropy-based Greedy algorithm

Part I: Introduction

Example: Who Gets a Loan?



A Decision Tree is a Flow Chart

- Each internal node represents a "test" on an attribute
- Each branch represents the outcome of the test
- Each leaf node represents a class label (decision taken after computing all attributes)
- The paths from root to leaf represent classification rules

Advantages

- Human-Algorithm Interaction
 - Simple to understand and **interpret**.
 - Mirrors human decision making more closely
 - Uses an open-box model (as opposed to a black box)—Contrast to NNT
- Versatile
 - Able to handle both numerical and categorical data
 - Powerful: Can model arbitrary functions
 - Requires little data preparation
 - Performs well with large datasets.
 - Robust against boosting

Advantages

- Built-in feature selection
 - Naturally de-emphasizes irrelevant features
 - Develops a hierarchy in terms of relevance
- Testable: Possible to validate a model using statistical tests

Limitations

- Trees can be non-robust:
 - A small change in the training data can result in a large change in the tree and consequently the final predictions.
- The problem of learning an optimal decision tree is known to be NP-Complete
 - Practical decision-tree learning algorithms are based on heuristics (greedy algorithm)
 - Such algorithms cannot guarantee to return the globally optimal decision tree.
- Overfitting: Decision-tree learners can create over-complex trees that do not generalize well from the training data.
 - Motivates **pruning and Random Forests**

Implementation

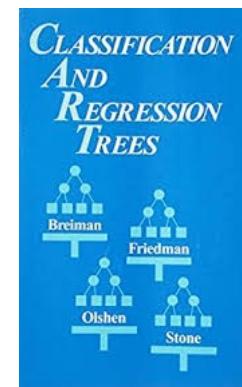
- Many data mining software packages provide implementations of decision tree algorithms.
 - Salford Systems
 - IBM SPSS Modeler
 - RapidMiner
 - SAS Enterprise Miner
 - Matlab
 - R
 - MS SQL Server
 - Scikit learn (a free and open-source machine learning library for the Python programming language)

History and Reference

- Quinlan, J. R. 1986. Induction of Decision Trees. *Mach. Learn.* 1, 1 (Mar. 1986), 81-106

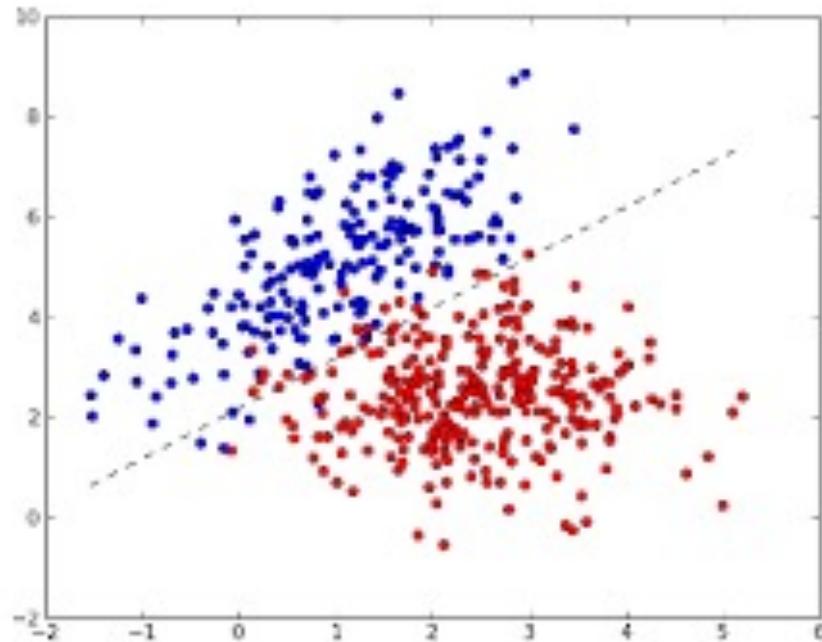


- Breiman, Friedman, Olshen, and Stone: Classification and Regression Trees.

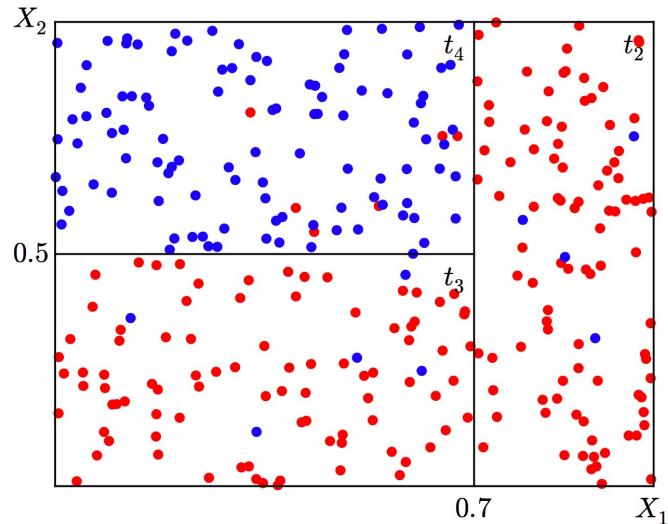


Classification with Regression

- Linear classifies ($a' x > 0$)
- Miss-classification (just point at picture)
- One level depth

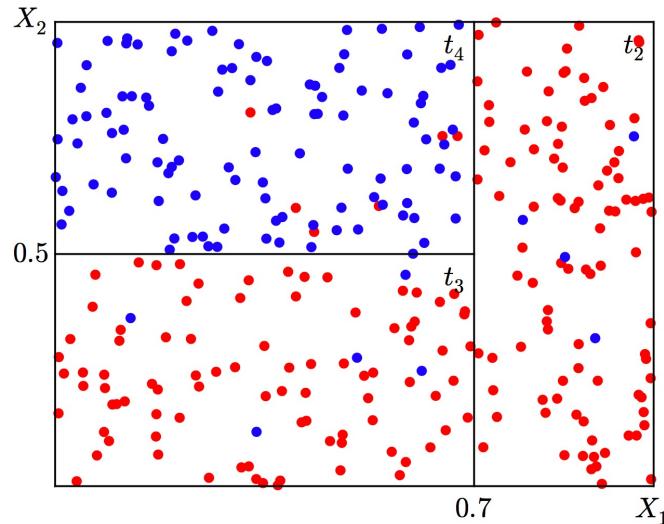


More complicated classification

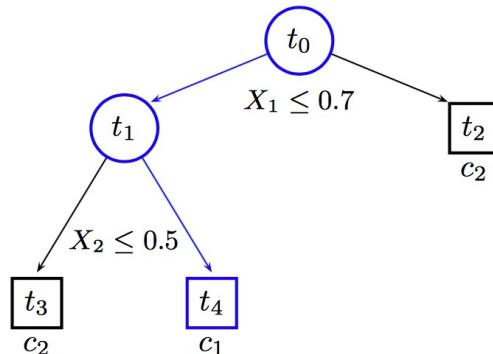


*Need a 2-level tree for classification
Miss classification*

More complicated classification



*Need a 2-level tree for classification
Miss classification*



Decision tree

Numerical vs. Categorical Data

- So far, you focused on continuous variables
- Decision trees are simpler when variables are categorical:
 - Variables take on discrete values
 - Refer to this as *finite outcomes*
 - Binary: equivalent to { 0, 1 }
- Will revisit continuous variables at the end

Illustrative Example: Waiting at a restaurant

Example	Input Attributes										Goal <i>WillWait</i>
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	
x_1	Yes	No	No	Yes	Some	\$\$\$	No	Yes	French	0–10	$y_1 = \text{Yes}$
x_2	Yes	No	No	Yes	Full	\$	No	No	Thai	30–60	$y_2 = \text{No}$
x_3	No	Yes	No	No	Some	\$	No	No	Burger	0–10	$y_3 = \text{Yes}$
x_4	Yes	No	Yes	Yes	Full	\$	Yes	No	Thai	10–30	$y_4 = \text{Yes}$
x_5	Yes	No	Yes	No	Full	\$\$\$	No	Yes	French	>60	$y_5 = \text{No}$
x_6	No	Yes	No	Yes	Some	\$\$	Yes	Yes	Italian	0–10	$y_6 = \text{Yes}$
x_7	No	Yes	No	No	None	\$	Yes	No	Burger	0–10	$y_7 = \text{No}$
x_8	No	No	No	Yes	Some	\$\$	Yes	Yes	Thai	0–10	$y_8 = \text{Yes}$
x_9	No	Yes	Yes	No	Full	\$	Yes	No	Burger	>60	$y_9 = \text{No}$
x_{10}	Yes	Yes	Yes	Yes	Full	\$\$\$	No	Yes	Italian	10–30	$y_{10} = \text{No}$
x_{11}	No	No	No	No	None	\$	No	No	Thai	0–10	$y_{11} = \text{No}$
x_{12}	Yes	Yes	Yes	Yes	Full	\$	No	No	Burger	30–60	$y_{12} = \text{Yes}$

1.	Alternate: whether there is a suitable alternative restaurant nearby.
2.	Bar: whether the restaurant has a comfortable bar area to wait in.
3.	Fri/Sat: true on Fridays and Saturdays.
4.	Hungry: whether we are hungry.
5.	Patrons: how many people are in the restaurant (values are None, Some, and Full).
6.	Price: the restaurant's price range (\$, \$\$, \$\$\$).
7.	Raining: whether it is raining outside.
8.	Reservation: whether we made a reservation.
9.	Type: the kind of restaurant (French, Italian, Thai or Burger).
10.	WaitEstimate: the wait estimated by the host (0-10 minutes, 10-30, 30-60, >60).

Attributes:



Institute of
Technology

[from: Russell & Norvig]

MIT INSTITUTE FOR DATA,
SYSTEMS, AND SOCIETY

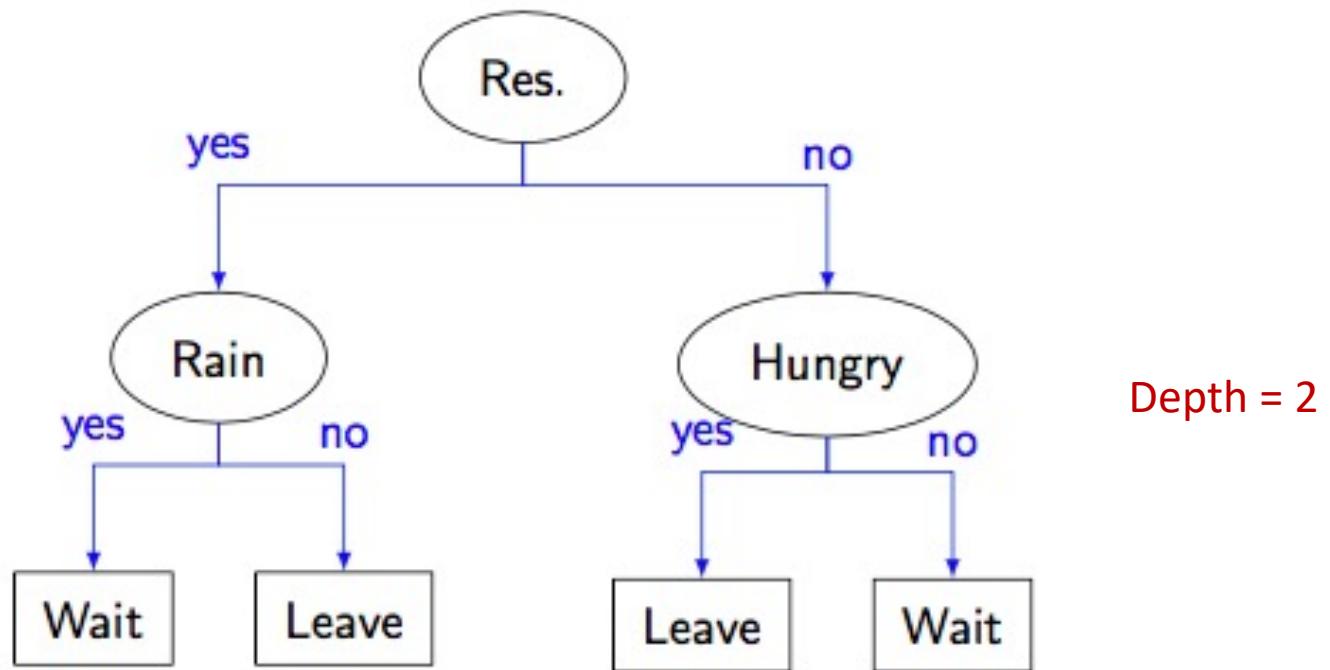
Notation

Feature Space- A vector of categorical data: \mathcal{X} .

Outcome Class (categorical): \mathcal{Y}

A Decision Rule: $f : \mathcal{X} \rightarrow \mathcal{Y}$

One possible Tree



Misclassification = ?

Misclassification Error

Data Set: $\{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}, i = 1, \dots, N\}$

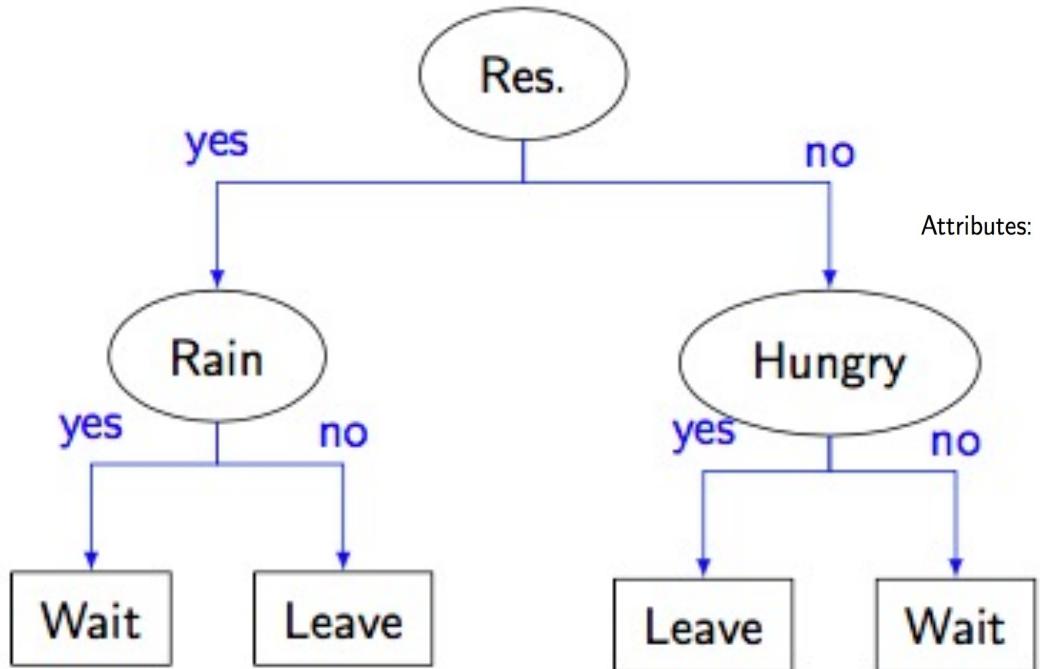
Empirical Error of a decision Rule f :

$$R(f) = \frac{1}{N} \sum_i^N \mathbf{I}(f(x_i) \neq y_i)$$

$$\mathbf{I}(x) = 1 \text{ if } x \neq 0, \text{ otherwise it is 0}$$

Misclassification Error

Example	Input Attributes											Goal WillWait
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est		
x ₁	Yes	No	No	Yes	Some	\$\$\$	No	Yes	French	0-10	y ₁ = Yes	
x ₂	Yes	No	No	Yes	Full	\$	No	No	Thai	30-60	y ₂ = No	
x ₃	No	Yes	No	No	Some	\$	No	No	Burger	0-10	y ₃ = Yes	
x ₄	Yes	No	Yes	Yes	Full	\$	Yes	No	Thai	10-30	y ₄ = Yes	
x ₅	Yes	No	Yes	No	Full	\$\$\$	No	Yes	French	>60	y ₅ = No	
x ₆	No	Yes	No	Yes	Some	\$\$	Yes	Yes	Italian	0-10	y ₆ = Yes	
x ₇	No	Yes	No	No	None	\$	Yes	No	Burger	0-10	y ₇ = No	
x ₈	No	No	No	Yes	Some	\$\$	Yes	Yes	Thai	0-10	y ₈ = Yes	
x ₉	No	Yes	Yes	No	Full	\$	Yes	No	Burger	>60	y ₉ = No	
x ₁₀	Yes	Yes	Yes	Yes	Full	\$\$\$	No	Yes	Italian	10-30	y ₁₀ = No	
x ₁₁	No	No	No	No	None	\$	No	No	Thai	0-10	y ₁₁ = No	
x ₁₂	Yes	Yes	Yes	Yes	Full	\$	No	No	Burger	30-60	y ₁₂ = Yes	

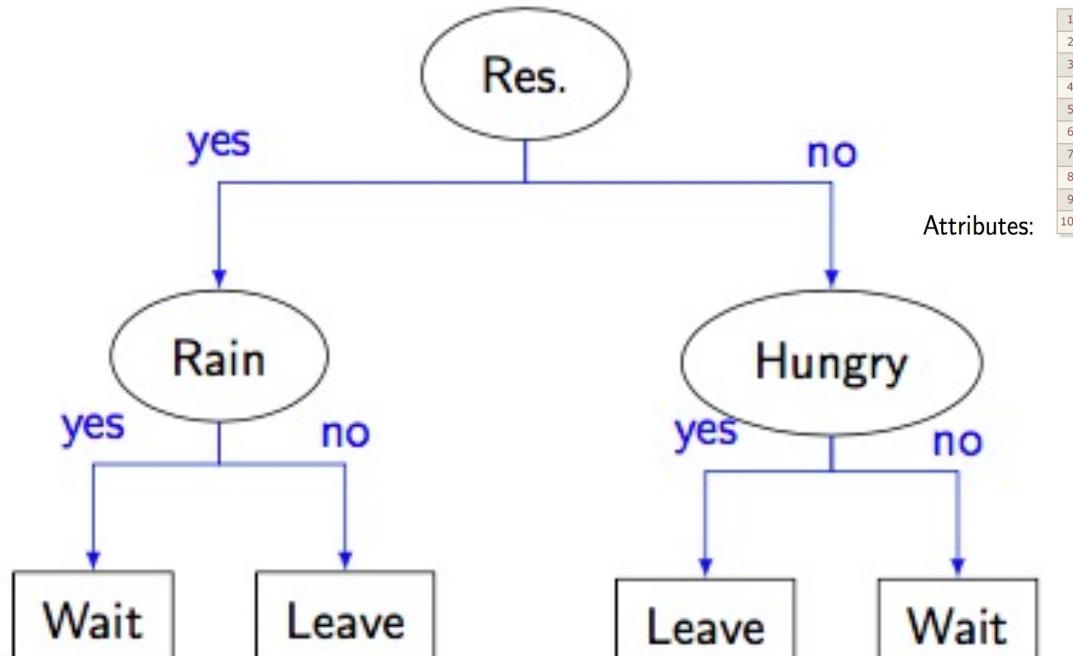


1. Alternate: whether there is a suitable alternative restaurant nearby.
2. Bar: whether the restaurant has a comfortable bar area to wait in.
3. Fri/Sat: true on Fridays and Saturdays.
4. Hungry: whether we are hungry.
5. Patrons: how many people are in the restaurant (values are None, Some, and Full).
6. Price: the restaurant's price range (\$, \$\$, \$\$\$).
7. Raining: whether it is raining outside.
8. Reservation: whether we made a reservation.
9. Type: the kind of restaurant (French, Italian, Thai or Burger).
10. WaitEstimate: the wait estimated by the host (0-10 minutes, 10-30, 30-60, >60).

[from: Russell & Norvig]

Misclassification Error

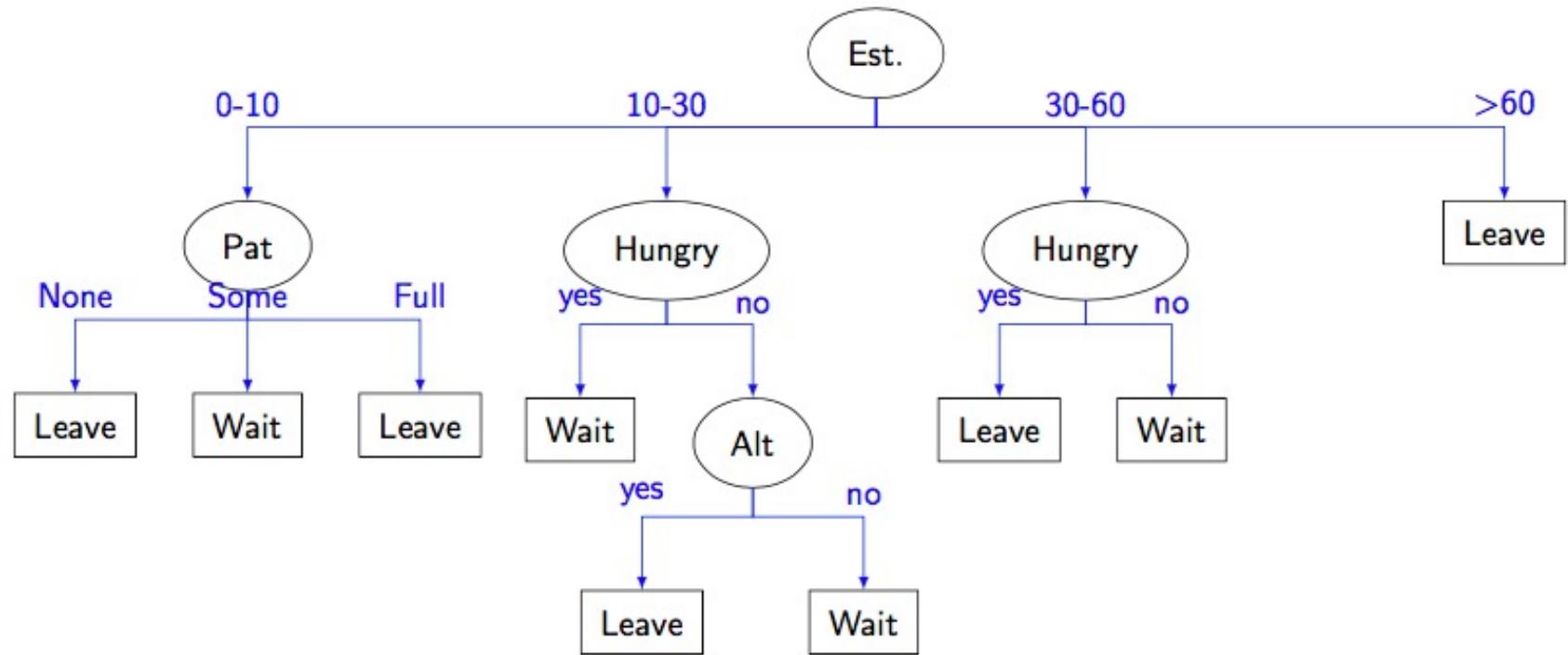
Example	Input Attributes											Goal WillWait
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est		
x ₁	Yes	No	No	Yes	Some	\$\$\$	No	Yes	French	0-10	y ₁ = Yes	
x ₂	Yes	No	No	Yes	Full	\$	No	No	Thai	30-60	y ₂ = No	
x ₃	No	Yes	No	No	Some	\$	No	No	Burger	0-10	y ₃ = Yes	
x ₄	Yes	No	Yes	Yes	Full	\$	Yes	No	Thai	10-30	y ₄ = Yes	
x ₅	Yes	No	Yes	No	Full	\$\$\$	No	Yes	French	>60	y ₅ = No	
x ₆	No	Yes	No	Yes	Some	\$\$	Yes	Yes	Italian	0-10	y ₆ = Yes	
x ₇	No	Yes	No	No	None	\$	Yes	No	Burger	0-10	y ₇ = No	
x ₈	No	No	No	Yes	Some	\$\$	Yes	Yes	Thai	0-10	y ₈ = Yes	
x ₉	No	Yes	Yes	No	Full	\$	Yes	No	Burger	>60	y ₉ = No	
x ₁₀	Yes	Yes	Yes	Yes	Full	\$\$\$	No	Yes	Italian	10-30	y ₁₀ = No	
x ₁₁	No	No	No	No	None	\$	No	No	Thai	0-10	y ₁₁ = No	
x ₁₂	Yes	Yes	Yes	Yes	Full	\$	No	No	Burger	30-60	y ₁₂ = Yes	


 $(x_6 \ x_8)$
 $(x_1 \ x_5 \ x_{10})$
 $(x_2 \ x_4 \ x_{12}) \quad (x_3 \ x_7 \ x_9 \ x_{11})$

- 1. Alternate: whether there is a suitable alternative restaurant nearby.
- 2. Bar: whether the restaurant has a comfortable bar area to wait in.
- 3. Fri/Sat: true on Fridays and Saturdays.
- 4. Hungry: whether we are hungry.
- 5. Patrons: how many people are in the restaurant (values are None, Some, and Full).
- 6. Price: the restaurant's price range (\$, \$\$, \$\$\$).
- 7. Raining: whether it is raining outside.
- 8. Reservation: whether we made a reservation.
- 9. Type: the kind of restaurant (French, Italian, Thai or Burger).
- 10. WaitEstimate: the wait estimated by the host (0-10 minutes, 10-30, 30-60, >60).

[from: Russell & Norvig]

Detailed Tree



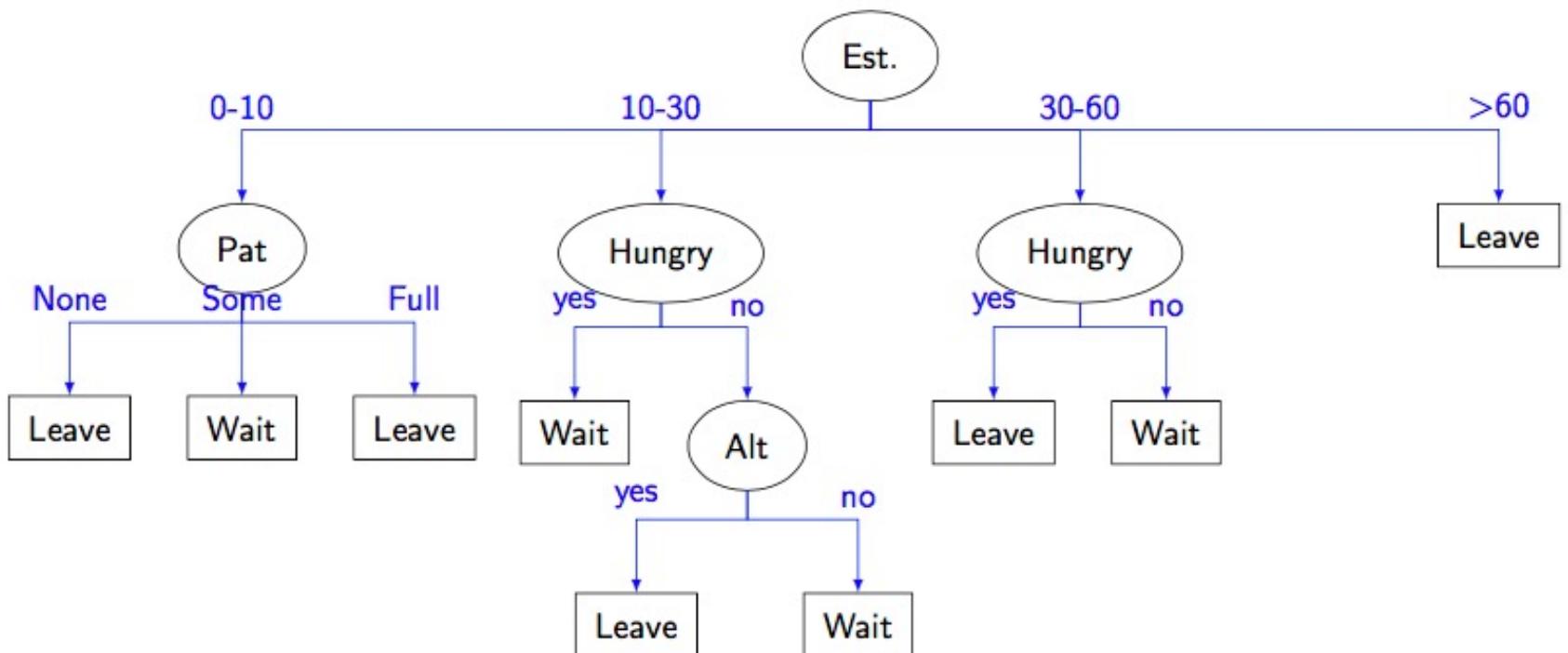
Misclassification: ?

Detailed Tree

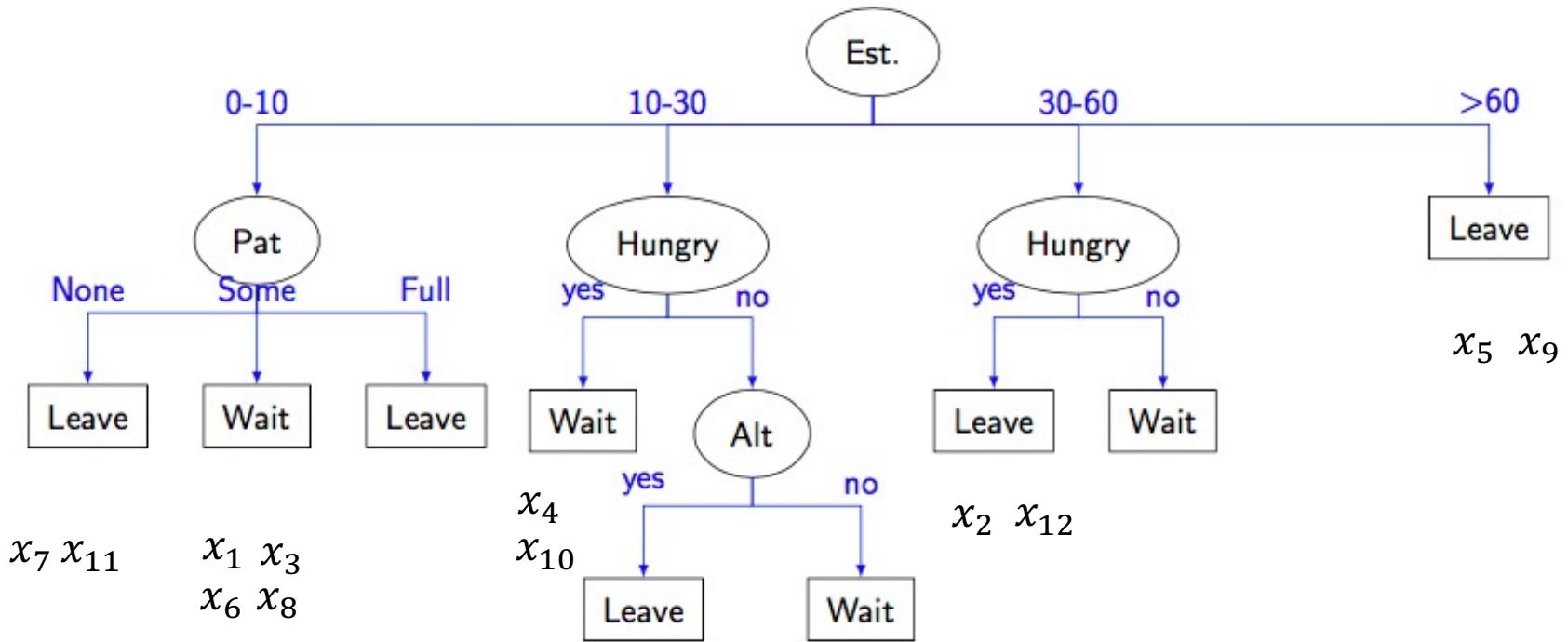
Example	Input Attributes										Goal WillWait
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	
x ₁	Yes	No	No	Yes	Some	\$\$\$	No	Yes	French	0-10	y ₁ = Yes
x ₂	Yes	No	No	Yes	Full	\$	No	No	Thai	30-60	y ₂ = No
x ₃	No	Yes	No	No	Some	\$	No	No	Burger	0-10	y ₃ = Yes
x ₄	Yes	No	Yes	Yes	Full	\$	Yes	No	Thai	10-30	y ₄ = Yes
x ₅	Yes	No	Yes	No	Full	\$\$\$	No	Yes	French	>60	y ₅ = No
x ₆	No	Yes	No	Yes	Some	\$\$	Yes	Yes	Italian	0-10	y ₆ = Yes
x ₇	No	Yes	No	No	None	\$	Yes	No	Burger	0-10	y ₇ = No
x ₈	No	No	No	Yes	Some	\$\$	Yes	Yes	Thai	0-10	y ₈ = Yes
x ₉	No	Yes	Yes	No	Full	\$	Yes	No	Burger	>60	y ₉ = No
x ₁₀	Yes	Yes	Yes	Yes	Full	\$\$\$	No	Yes	Italian	10-30	y ₁₀ = No
x ₁₁	No	No	No	No	None	\$	No	No	Thai	0-10	y ₁₁ = No
x ₁₂	Yes	Yes	Yes	Yes	Full	\$	No	No	Burger	30-60	y ₁₂ = Yes

- Attributes:
1. Alternate: whether there is a suitable alternative restaurant nearby.
 2. Bar: whether the restaurant has a comfortable bar area to wait in.
 3. Fri/Sat: true on Fridays and Saturdays.
 4. Hungry: whether we are hungry.
 5. Patrons: how many people are in the restaurant (values are None, Some, and Full).
 6. Price: the restaurant's price range (\$, \$\$, \$\$\$).
 7. Raining: whether it is raining outside.
 8. Reservation: whether we made a reservation.
 9. Type: the kind of restaurant (French, Italian, Thai or Burger).
 10. WaitEstimate: the wait estimated by the host (0-10 minutes, 10-30, 30-60, >60).

[from: Russell & Norvig]



How does Pruning work here?



Misclassification: 2/12

Questions and Break

Part II: Learning a DT

Probabilistic Description

Feature Space- A vector of categorical data: \mathcal{X} .

Outcome Class (categorical): \mathcal{Y}

A Decision Rule: $f : \mathcal{X} \rightarrow \mathcal{Y}$

A Probabilistic Model: $X \subset \mathcal{X}$ and $Y \subset \mathcal{Y}$ are Random Variables

Error of a decision Rule $f : R^*(f) = \mathbf{P} \{(X, Y) : f(X) \neq Y\}$

Empirical Estimates

Data Set: $\{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}, i = 1, \dots, N\}$

Empirical Error of a decision Rule f :

$$R(f) = \frac{1}{N} \sum_i^N \mathbf{I}(f(x_i) \neq y_i)$$

$\mathbf{I}(x) = 1 \text{ if } x \neq 0, \text{ otherwise it is } 0$

Trees as Decision Rules

- Define a sub-class as:

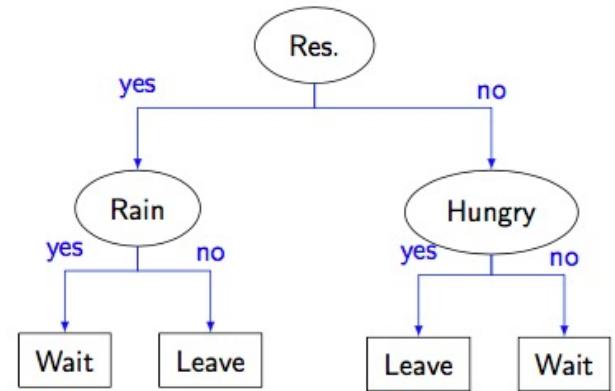
$$C = \{(x, y) \mid x(k) = v_k, k \text{ subset of all feature indices}\}$$

- The assignment mapping is given by

$$f(C) = y, \quad y = \max_y \mathbf{P}(y|C)$$

- Assignment from Data

Majority of $\{y_i, (x_i, y_i) \in C\}$



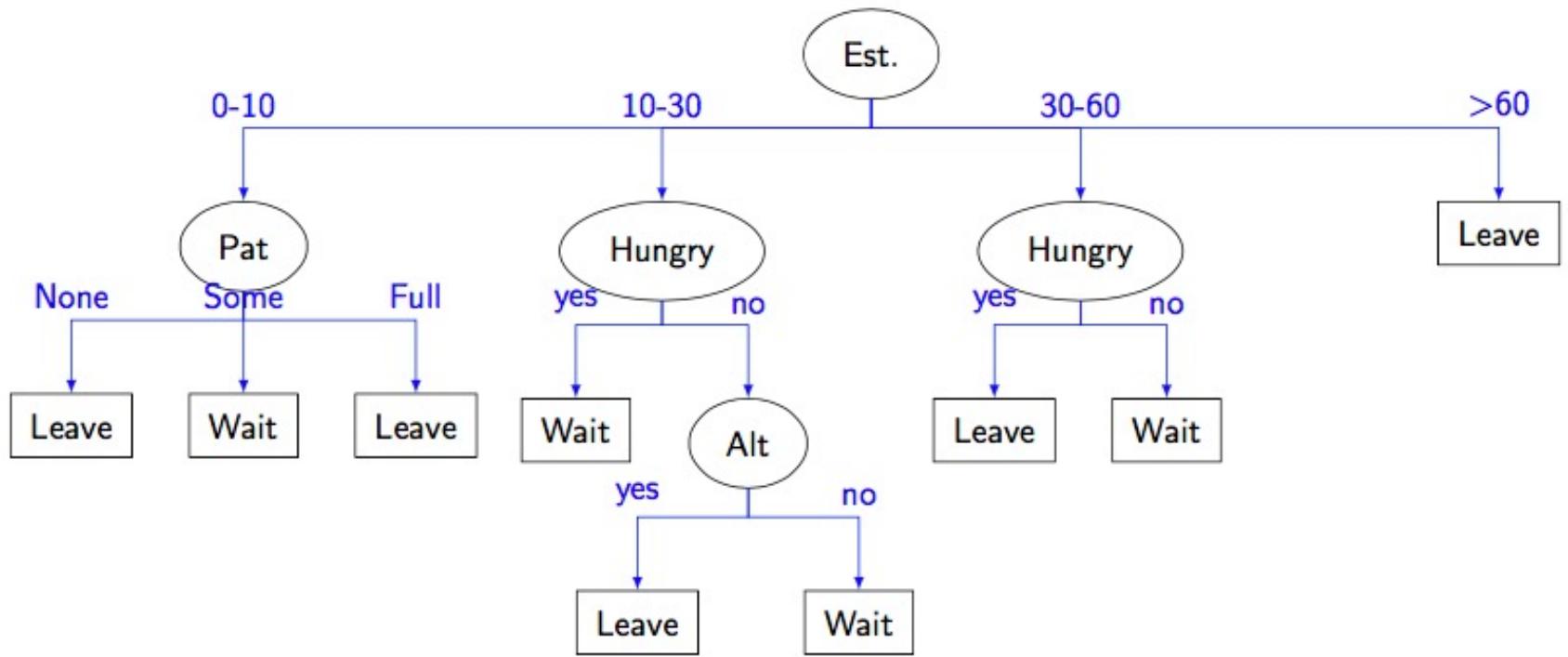
Learning a DT from Data

- Split the data
 - Training Data
 - Cross Validation data
- Train the model using the “Training Data”
- Evaluate the performance on the “Cross Validation Data”
- 80-20 ratio generally works!

Learning a DT from Data

- How do we learn:
 - Pick a feature
 - Split the data based on that feature
 - Define new classes
 - Repeat
- The model does not have to split all the features at each level
- The model does not have to use all the features
- Combinatorial explosion
- Order matters!

Back to Waiting for a Table



- How do we pick the feature at each level
- Greedy Algorithm
- Define the cost!

Optimizing Splits

- Motivate Entropy as an '**Impurity**' measure
- Greedy algorithm for splits minimize impurity?

Introduce Entropy as a measure of uncertainty

- Review Entropy; conditional entropy
- Impurity measure
- Define greedy algorithm
- Show how it works

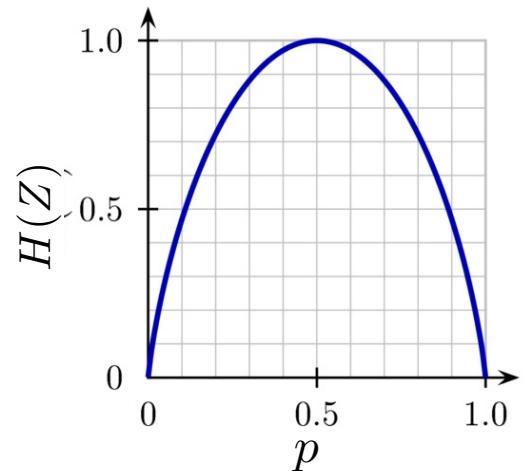
Entropy

$Z \in \mathcal{Z}$ is a random variable with probability mass function $P(z)$

$$\textbf{Entropy: } H(Z) = -\sum_{z \in \mathcal{Z}} P(z) \log P(z)$$

Example: Coin flip with $P(\text{head}) = p$

$$\textbf{Entropy: } -p \log p - (1-p) \log(1-p)$$



Measures uncertainty in a random variable

Maximum at $p=0.5$

Example

Example: if $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, then

		$Y = 0$	$Y = 1$
		$X = 0$	$1/8$
$X = 1$	$Y = 0$	$3/8$	$1/8$
	$Y = 1$		

Entropy: $H(X, Y) = - \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} P(x, y) \log P(x, y)$

$$= -1/8 \log 1/8 - 3/8 \log 3/8 - 1/8 \log 1/8 - 3/8 \log 3/8$$

Conditional Entropy:

$$H(Y|X = 0) = - \sum_{y \in \mathcal{Y}} P(y|x = 0) \log P(y|x = 0) = -1/4 \log 1/4 - 3/4 \log 3/4$$

$$H(Y|X = 1) = - \sum_{y \in \mathcal{Y}} P(y|x = 1) \log P(y|x = 1) = -3/4 \log 3/4 - 1/4 \log 1/4$$

$$H(Y|X) = - \sum_{x \in \mathcal{X}} P(x) H(Y|X = x) = 1/2 H(Y|X = 0) + 1/2 H(Y|X = 1)$$

Observe: $H(Y|X) \leq H(Y)$

Questions and Break

Part III: Information Gain

Information Gain

Information Gain: $IG(Y|X) = H(Y) - H(Y|X)$

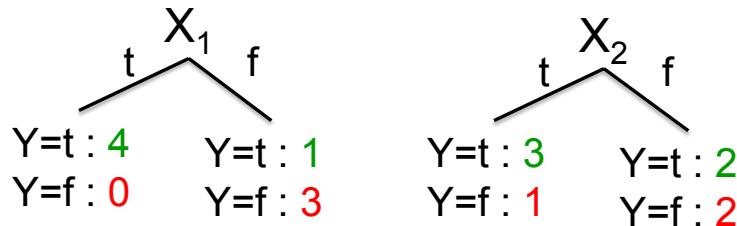
If $X \perp Y$ then $IG = 0$ X is not informative

If $IG(Y|X) = H(Y)$ then Y is most informative

Simple Illustration

Splitting: choosing a good attribute

Would we prefer to split on X_1 or X_2 ?

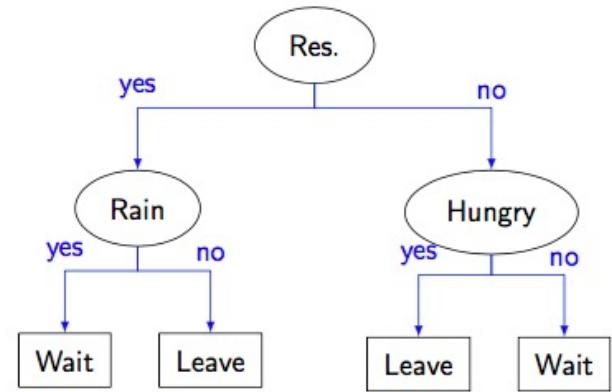


Idea: use counts at leaves to define probability distributions, so we can measure uncertainty!

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F

Greedy Algorithm

- Start with the complete data set



- Pick a feature (formally: $X(m)$)
- Describe the data based on this feature

$$\{(x_i(m), y_i), i = 1, \dots, N\}$$

- Split the outcome data based on the classes

$$S_1 = \{(y_i \mid x_i(m) = 0\}$$

$$S_2 = \{(y_i \mid x_i(m) = 1\}$$

Empirical Computation of Entropy

- Empirically compute the conditional entropy

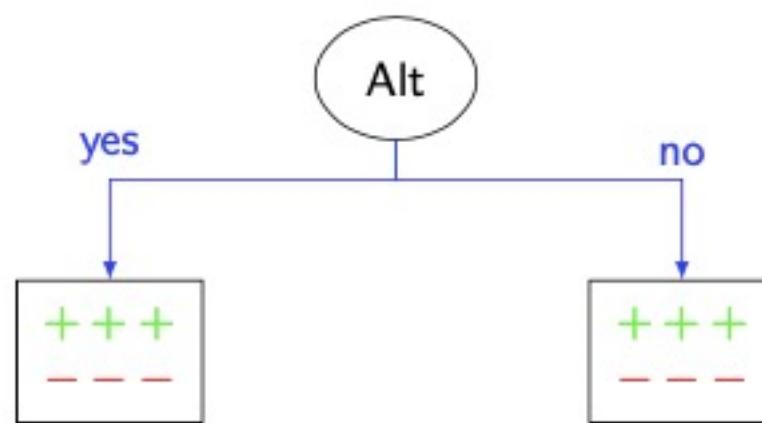
$$H(Y|X(m))$$

- Estimate:

$$\mathbf{P}(S_1)H(S_1) + \mathbf{P}(S_2)H(S_2)$$

- Pick m to minimize this (maximize information gain)

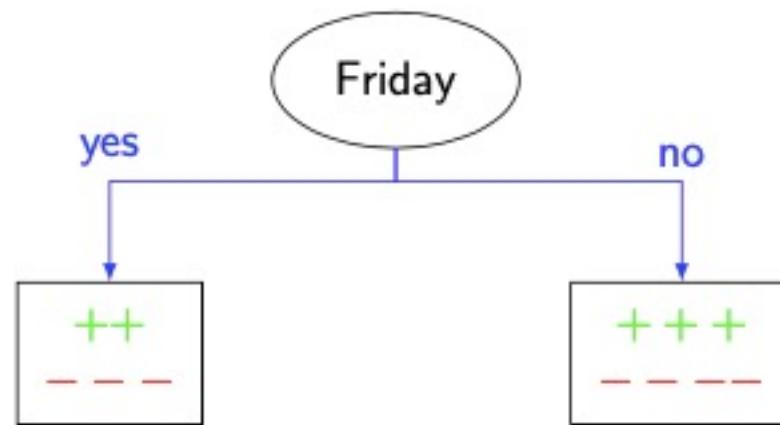
Splitting on Alt



Loss: 1.

$$IG = 0$$

Back to Example: Splitting on Friday

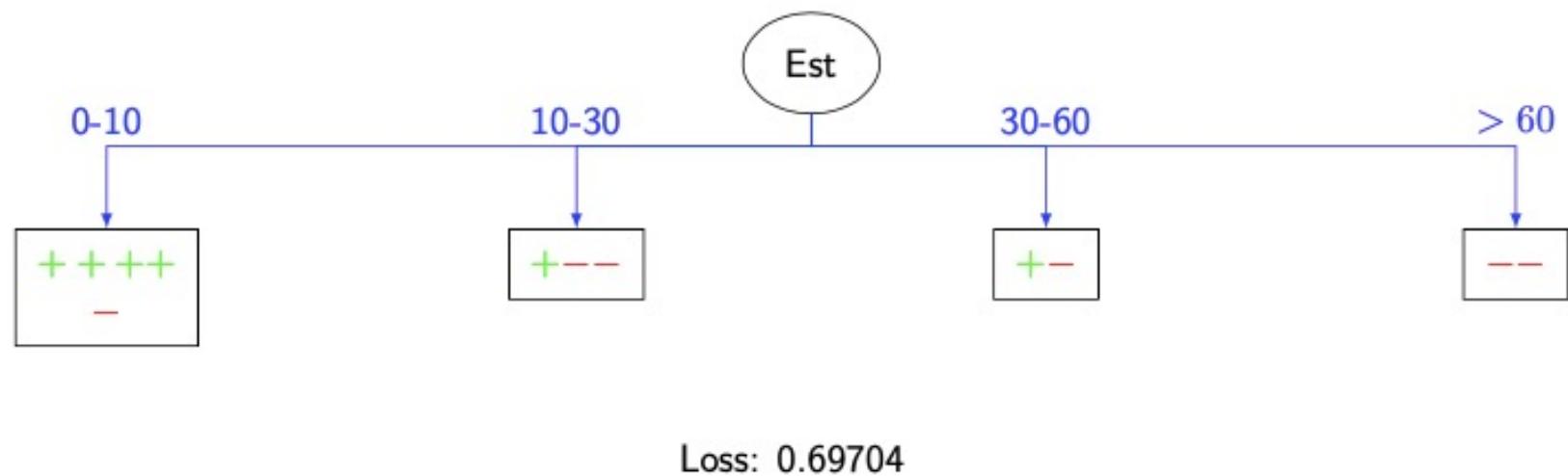


Loss: 0.97928

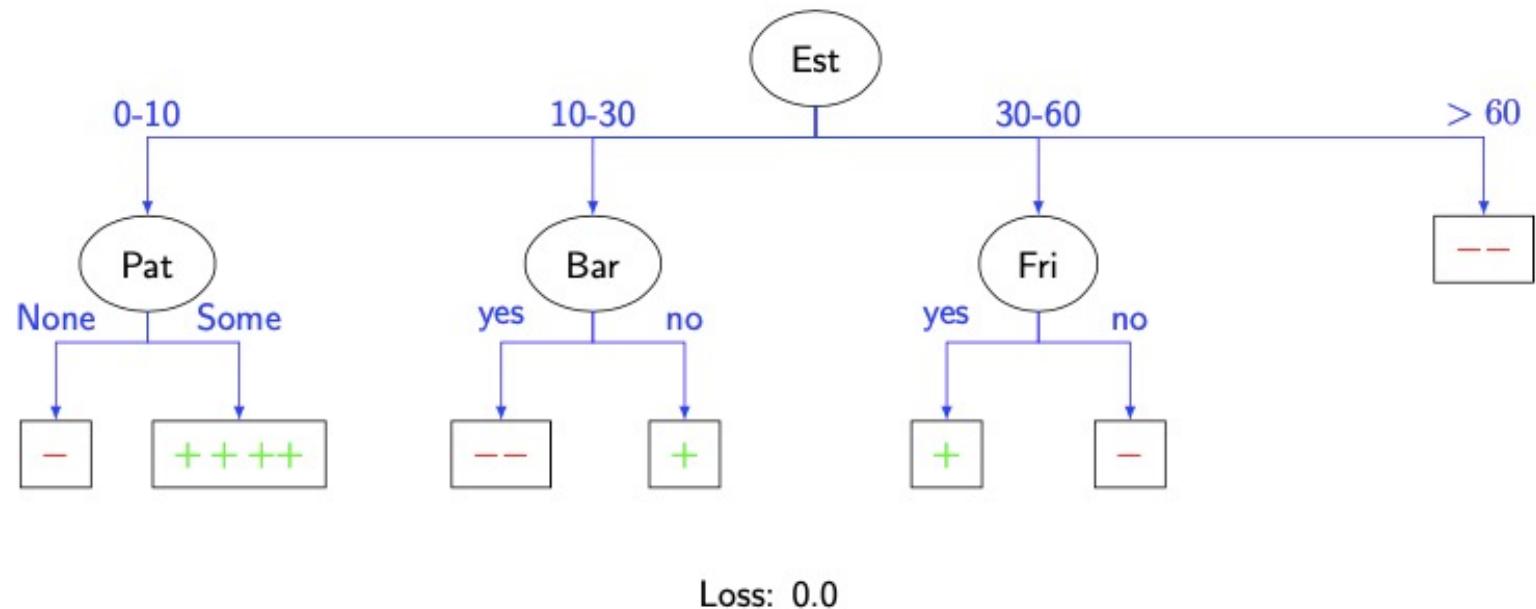
Computation

- $H(S_1) = -\frac{2}{5} \log \frac{2}{5} - \frac{3}{5} \log \frac{3}{5} = .97$
- $H(S_2) = -\frac{3}{7} \log \frac{3}{7} - \frac{4}{7} \log \frac{4}{7} = .99$
- Entropy of Split: .
$$.97 * \frac{5}{12} + .99 * \frac{7}{12} = .979$$
- $IG = H(Y) - .979 = 1 - .979 = .021$
- *Not a great split.*

Back to Example: Splitting on Est



Back to Example: Multiple Splits



Gini Index

Replace Entropy with $\sum_x p(x) (1 - p(x)) = 1 - \sum_x p(x)^2$

Gini Index: Favors larger partitions

Summary

- Interpreting a decision tree
- Learning a Decision Tree
 - ✓ Mis-classification error
 - ✓ Probabilistic interpretation
 - ✓ Formal definition of a Decision Tree
- Impurity minimization
 - ✓ Entropy minimization
 - ✓ Information Gain

Questions



Thank You