Artificial Intelligence

MACHINE LEARNING BASIC ALGORITHMS

Nguyễn Ngọc Thảo – Nguyễn Hải Minh
{nnthao, nhminh}@fit.hcmus.edu.vn

Outline

- Introduction to Machine learning
- ID3 Decision tree
- Naïve Bayesian classification

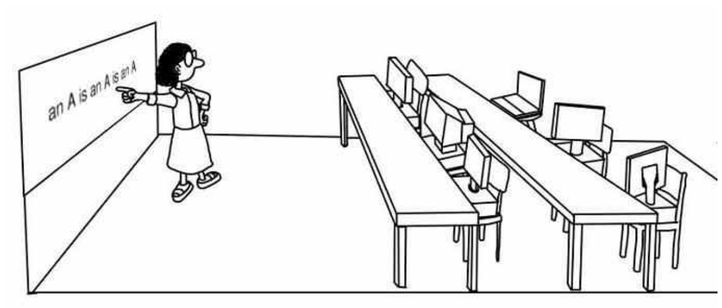
Acknowledgements

- This slide is mainly based on the textbook AIMA (3rd edition)
- Some parts of the slide are adapted from
  - Maria-Florina Balcan, *Introduction to Machine Learning*, 10-401, Spring 2018, Carnegie Mellon University
  - Ryan Urbanowicz, *An Introduction to Machine Learning*, PA CURE Machine Learning Workshop: December 17, School of Medicine, University of Pennsylvania

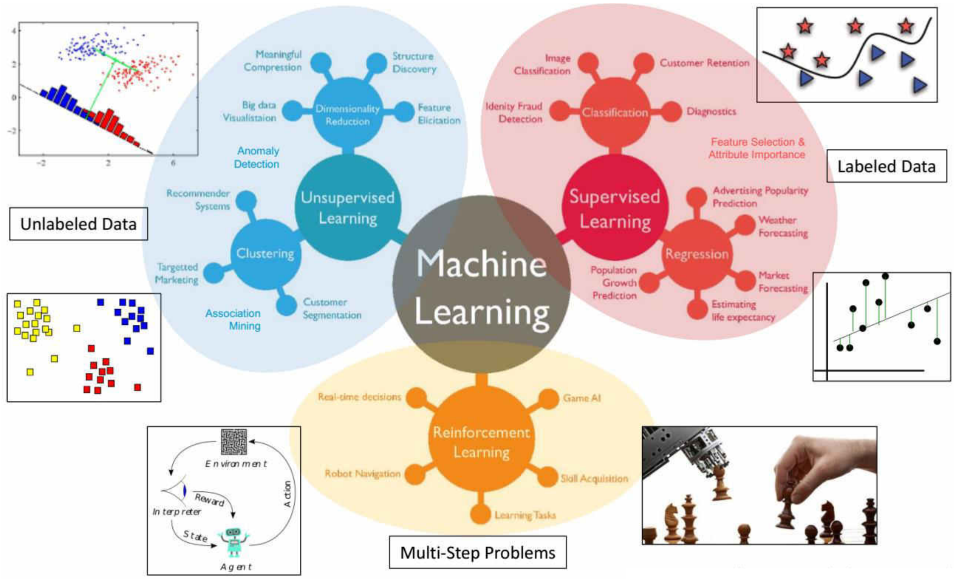
Machine Learning

What is machine learning?

- **Machine learning** involves adaptive mechanisms that enable computers to learn from experience, learn by example and learn by analogy.

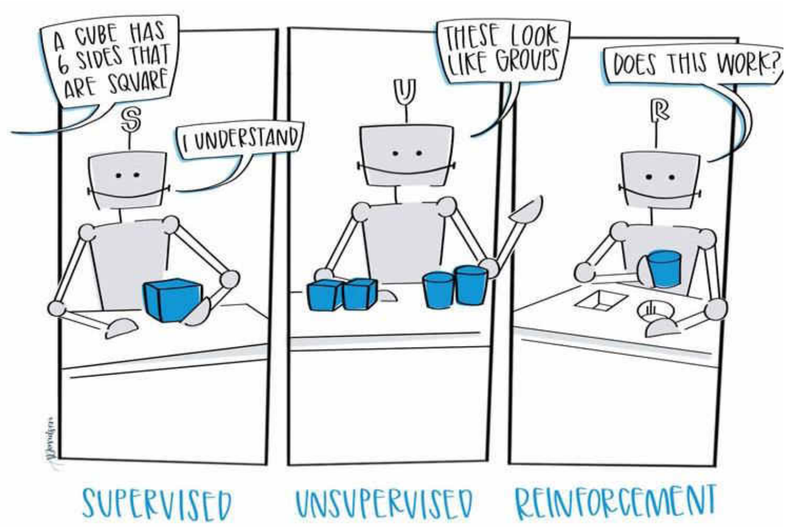


Types of machine learning



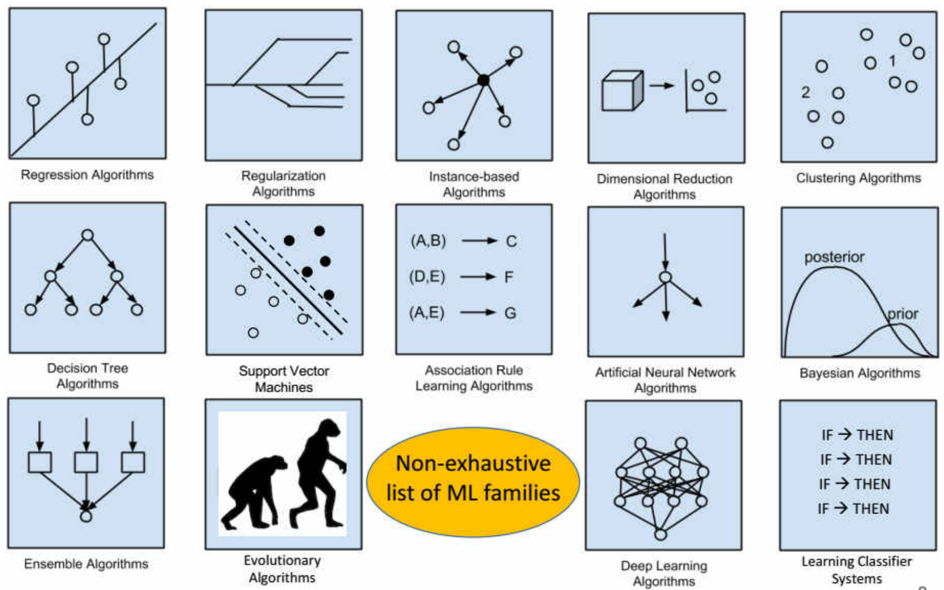
Source: https://ldi.upenn.edu/sites/default/files/Introduction-to-Machine-Learning.pdf

Types of machine learning



Source: https://www.ceralytics.com/3-types-of-machine-learning/

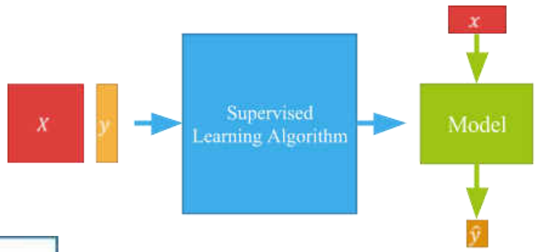
Machine learning algorithms

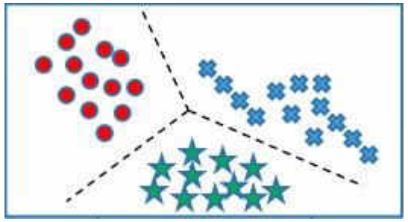


Source: https://ldi.upenn.edu/sites/default/files/Introduction-to-Machine-Learning.pdf

Supervised learning

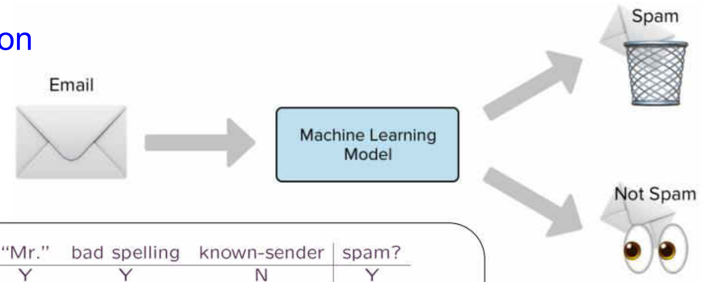
- Learn a function that maps an input to an output based on examples, which are pairs of input-output values.





Supervised learning: Examples

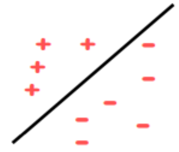
- Spam detection



	"money"	"pills"	"Mr."	bad spelling	known-sender	spam?	
	Y	N	Y	Y	N	Y	
	N	N	N	Y	Y	N	
	N	Y	N	N	N	Y	
example	Y	N	N	N	Y	N	label
	N	N	Y	N	Y	N	
	Y	N	N	Y	N	Y	
	N	N	Y	N	N	N	

Reasonable RULES

- Predict SPAM if unknown AND (money OR pills)
- Predict SPAM if 2money + 3pills – 5 known > 0

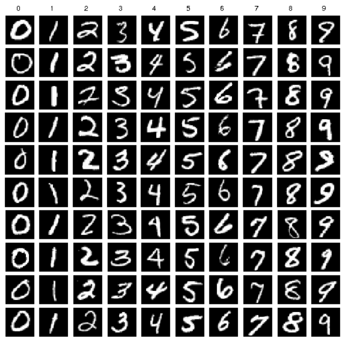


Linearly separable

Supervised learning: Examples

- Object detection

Indoor scene recognition

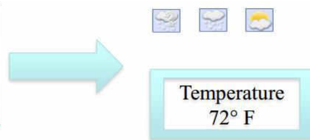


Handwritten digit recognition

Scene text recognition

Supervised learning: More examples

- Weather prediction: Predict the weather type or the temperature at any given location…

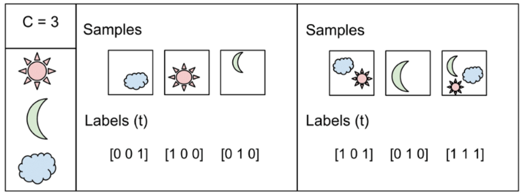


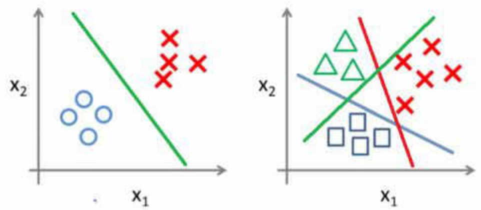
- Medicine: diagnose a disease (or response to chemo drug X, or whether a patient is re-admitted soon?)
    - Input: from symptoms, lab measurements, test results, DNA tests, …
    - Output: one of set of possible diseases, or "none of the above"
    - E.g., audiology, thyroid cancer, diabetes, etc.

- Computational economics:
    - Predict if a user will click on an ad so as to decide which ad to show
    - Predict if a stock will rise or fall (with specific amounts)

Classification vs. Regression

- Train a model to predict a categorical dependent variable
- Case studies: predicting disease, classifying images, predicting customer churn, buy or won't buy, etc.

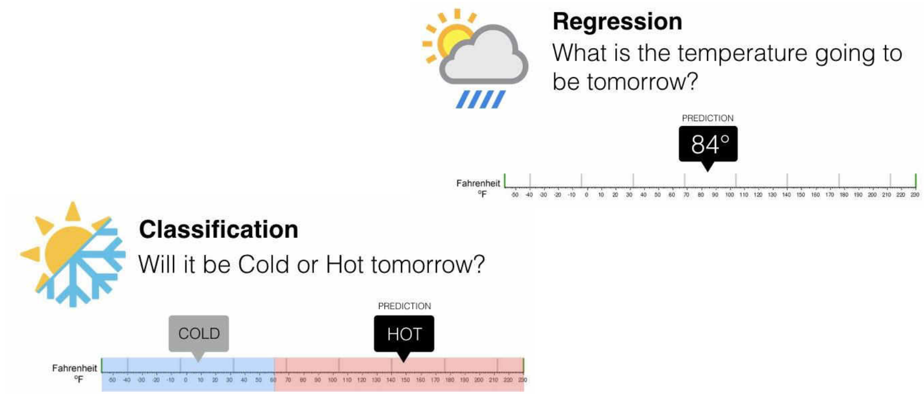




Binary classification
vs.
Multiclass classification
vs.
Multilabel classification

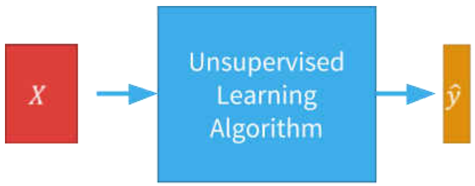
Classification vs. Regression

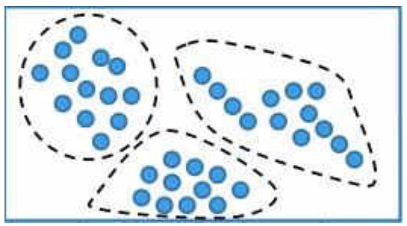
- Train a model to predict a continuous dependent variable
- Case studies: predicting height of children, predicting sales, forecasting stock prices, etc.



Unsupervised learning

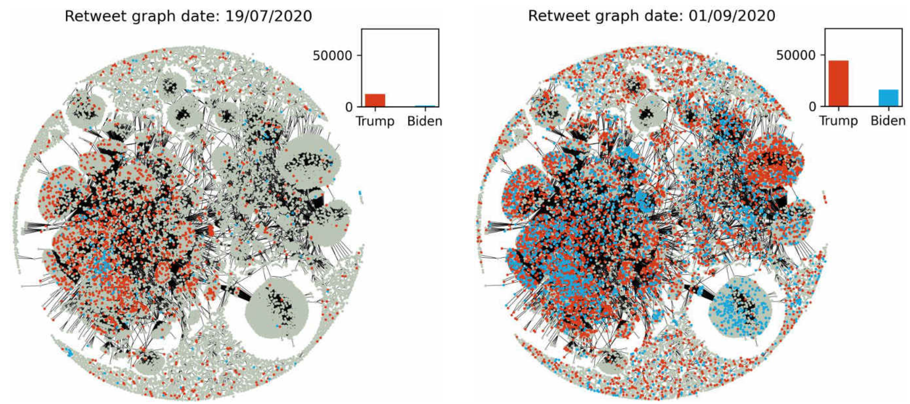
- Infer a function to describe hidden structure from "unlabeled" data
  - A classification (or categorization) is not included in the observations.





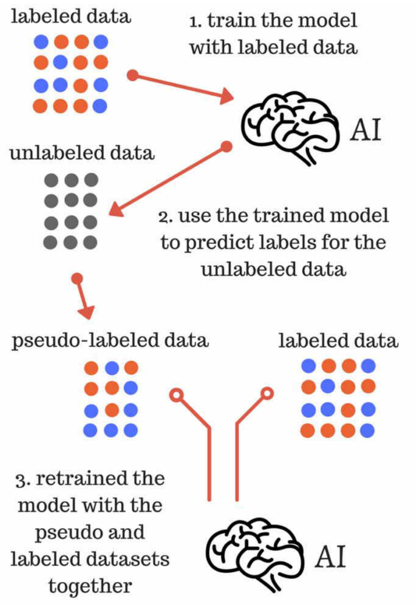
Unsupervised learning: Examples

- Social network analysis: cluster users of social networks by interest (community detection)

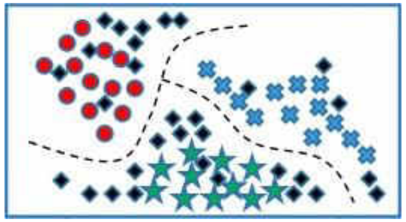


Ref: Shevtsov, Alexander, et al. "Analysis of Twitter and YouTube during US elections 2020." *arXiv e-prints* (2020): arXiv-2010.

Semi-supervised learning

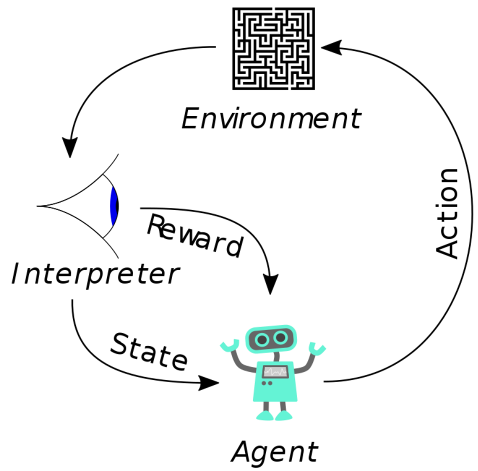


- The model is initially trained with a small amount of labeled data and a large amount of unlabeled data.

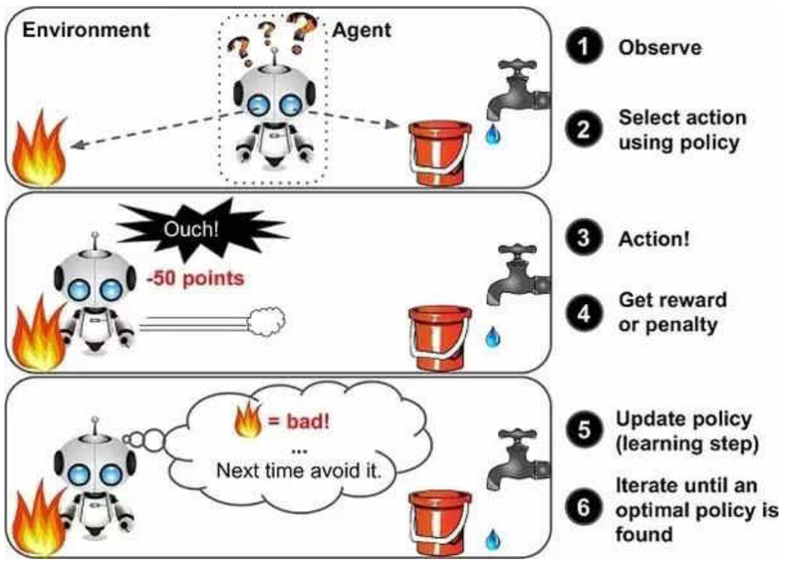


Reinforcement learning

- The agent learns from the environment by interacting with it and receives rewards for performing actions.



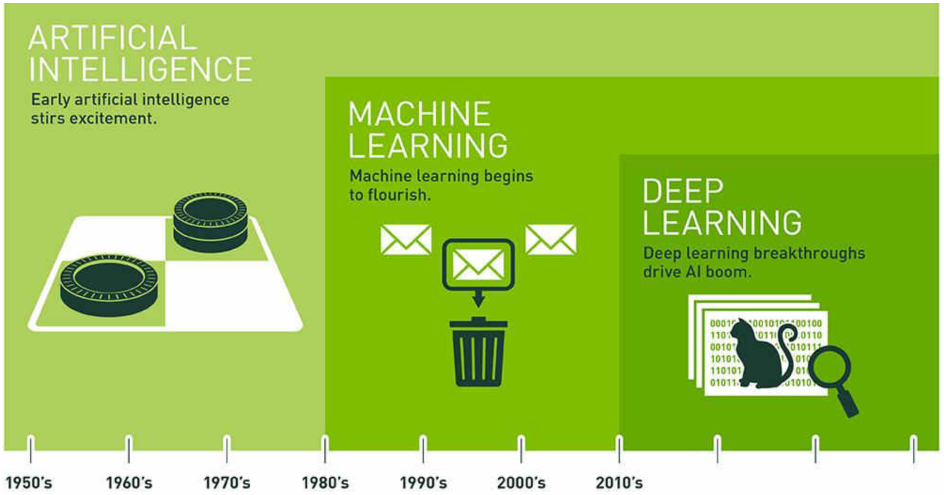
Reinforcement learning: Example



Reinforcement learning: Examples

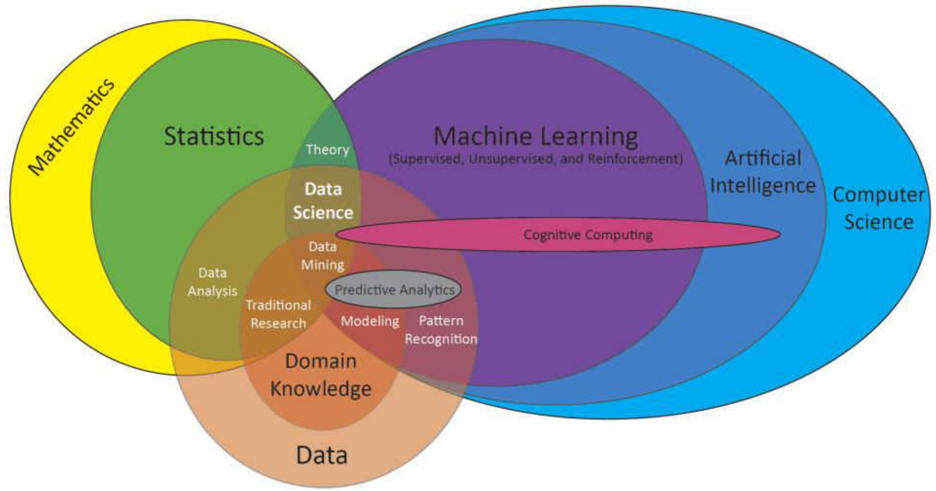
https://openai.com/blog/emergent-tool-use/ https://arxiv.org/pdf/1909.07528.pdf

Machine learning and related concepts

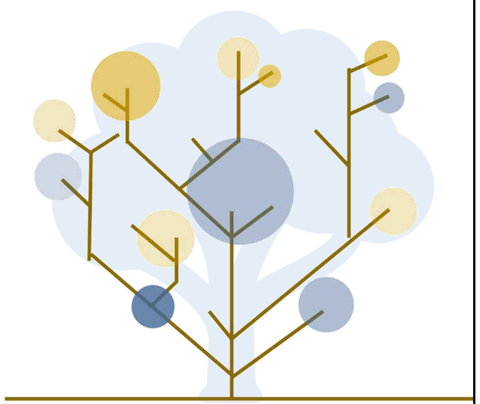


Source: https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

Machine learning and related concepts



ID3
Decision Tree



Learning agents – Why learning?

- Unknown environments
  - A robot designed to navigate mazes must learn the layout of each new maze it encounters.
- Environment changes over time
  - An agent designed to predict tomorrow's stock market prices must learn to adapt when conditions change from boom to bust.
- No idea how to program a solution
  - The task to recognizing the faces of family members

Learning element

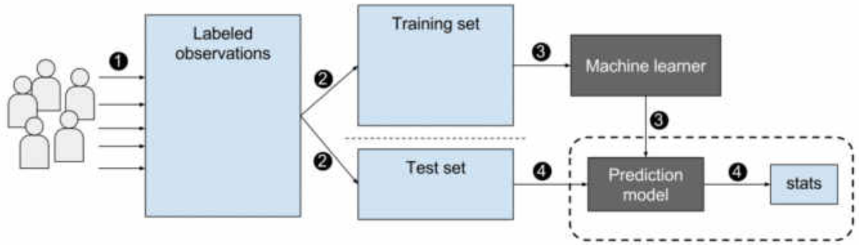
- Design of a learning element is affected by
  - Which *components* is to be improved
  - What *prior knowledge* the agent already has
  - What *representation* is used for the components
  - What ***feedback*** is available to learn these components
- Type of feedback
  - Supervised learning: correct answers for each example
  - Unsupervised learning: correct answers not given
  - Reinforcement learning: occasional rewards

Supervised learning

- Simplest form: learn a function from examples
- Given a **training set** of N example input-output pairs

$$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$

  - where each y_j was generated by an unknown function $y = f(x)$
- Find a hypothesis $\boldsymbol{h}$ such that $\boldsymbol{h} \approx \boldsymbol{f}$
- To measure the accuracy of a hypothesis, give it a **test set** of examples that are different with those in the training set.



Supervised learning

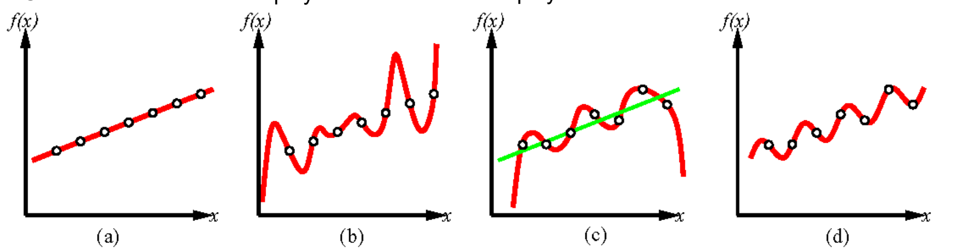
- Construct h so that it agrees with f.
- The hypothesis h is **consistent** if it agrees with f on all observations.
- **Ockham's razor:** Select the simplest consistent hypothesis.

Consistent linear fit

Consistent 7th order polynomial fit

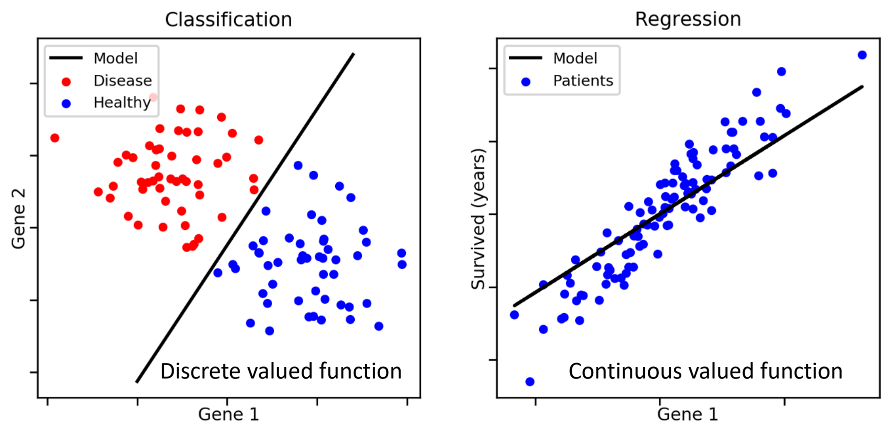
Inconsistent linear fit. Consistent 6th order polynomial fit.

Consistent sinusoidal fit



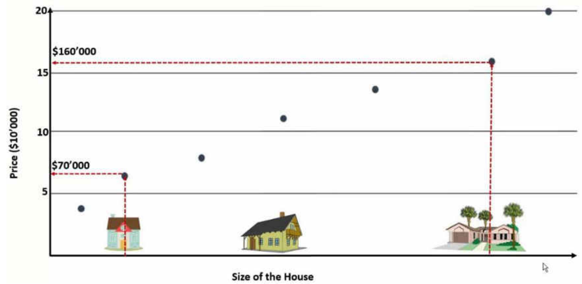
Supervised learning problems

- $\boldsymbol{h(x)} = \textit{the predicted output value for the input } \boldsymbol{x}$



Regression vs. Classification

- Estimating the price of a house



- Will you pass or fail the exam?
  - 2 classes: Fail/Pass

- Is this an apple, an orange or a tomato?
  - 3 classes: Apple / Orange / Tomato

The wait@restaurant problem

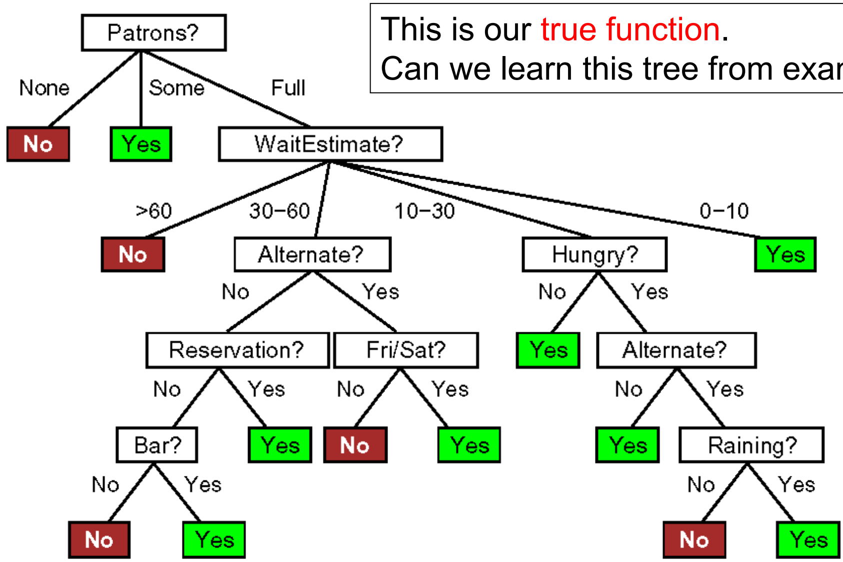
Predicting whether a certain person will wait to have a seat in a restaurant.

The wait@restaurant problem

- The decision is based on the following attributes
  1. **Alternate:** is there an alternative restaurant nearby?
  2. **Bar:** is there a comfortable bar area to wait in?
  3. **Fri/Sat:** is today Friday or Saturday?
  4. **Hungry:** are we hungry?
  5. **Patrons:** number of people in the restaurant (None, Some, Full)
  6. **Price:** price range ($, $$, $$$)
  7. **Raining:** is it raining outside?
  8. **Reservation:** have we made a reservation?
  9. **Type:** kind of restaurant (French, Italian, Thai, Burger)
  10. **WaitEstimate:** estimated waiting time (0-10, 10-30, 30-60, >60)

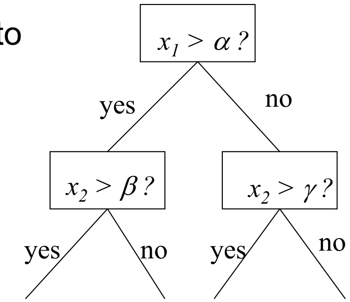
The wait@restaurant decision tree

This is our true function.
Can we learn this tree from examples?



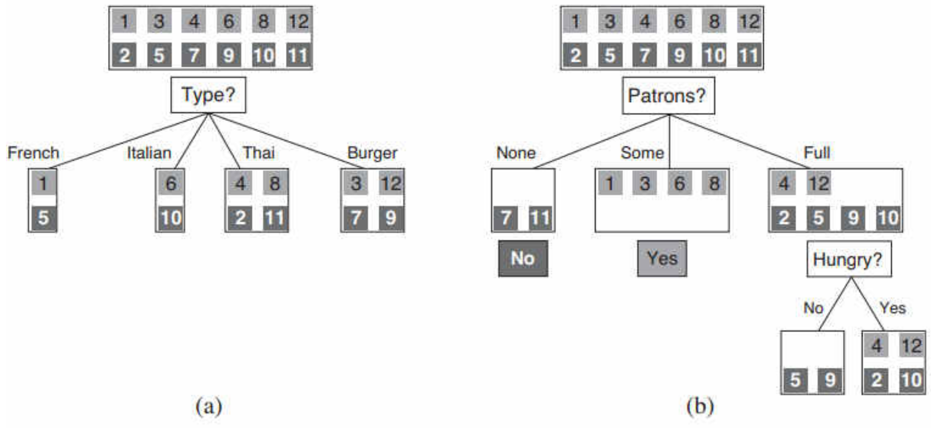
Learning decision trees

- **Divide and conquer:** Split data into smaller and smaller subsets
- Splits are usually on a single variable



- After splitting up, each outcome is a new decision tree learning problem with fewer examples and one less attribute.

Learning decision trees



Splitting the examples by testing on attributes

ID3 Decision tree algorithm

1. The remaining examples are **all** positive (or **all** negative), $\rightarrow$ DONE, it is possible to answer Yes or No.
   - E.g., in Figure (b), None and Some branches
2. There are **some** positive and **some** negative examples $\rightarrow$ choose the **best** attribute to split them
   - E.g., in Figure (b), Hungry is used to split the remaining examples

ID3 Decision tree algorithm

3. **No** examples left at a branch $\rightarrow$ return a default value.
   - No example has been observed for a combination of attribute values
   - The default value is calculated from the plurality classification of all the examples that were used in constructing the node's parent.
   - These are passed along in the variable parent examples
4. **No** attributes left but both positive and negative examples $\rightarrow$ return the plurality classification of remaining ones.
   - Examples of the same description, but different classifications
   - Usually an error or noise in the data, nondeterministic domain, or no observation of an attribute that would distinguish the examples.

ID3 Decision tree: Pseudo-code

function DECISION-TREE-LEARNING($examples, attributes, parent\ examples$)
returns a tree
  if $examples$ is empty — No examples left
    then return PLURALITY-VALUE($parent\ examples$)
  else if all $examples$ have the same classification — remaining examples are all pos/all neg
    then return the classification
  else if $attributes$ is empty — No attributes left but examples are still pos & neg
    then return PLURALITY-VALUE($examples$)
  else
    …

ID3 Decision tree: Pseudo-code

function DECISION-TREE-LEARNING($examples, attributes, parent\ examples$)
returns a tree
  …
  else
    $A \leftarrow argmax_{a \in attributes}$ IMPORTANCE($a, examples$)
    $tree \leftarrow$ a new decision tree with root test A
    for each value v_k of A **do**
      $exs \leftarrow \{e : e \in examples$ **and** $e.A = v_k\}$
      $subtree \leftarrow$ DECISION-TREE-LEARNING($exs, attributes - A, examples$)
      add a branch to $tree$ with label $(A = v_k)$ and subtree $subtree$
    return $tree$

Decision tree: Inductive learning

- **Simplest:** Construct a decision tree with one leaf for every example
  - → memory based learning
  - → *worse generalization.*

- **Advanced:** Split on each variable so that the purity of each split increases (i.e. either only yes or only no)
  - E.g., using Entropy to measure the purity of data

A purity measure with entropy

- **Entropy** is a measure of the uncertainty of a random variable V with values v_k.

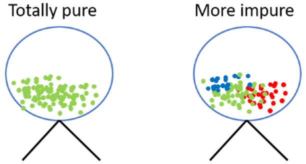
An indicator of how messy your data is

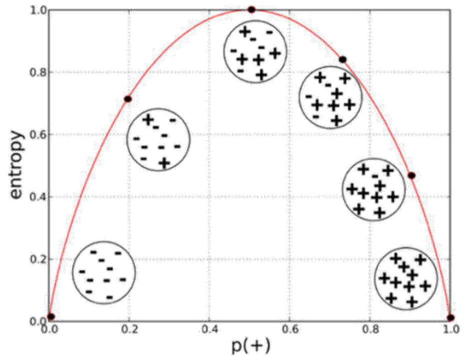
$$H(V) = \sum_k P(v_k) \log_2 \frac{1}{P(v_k)} = -\sum_k P(v_k) \log_2 P(v_k)$$

- v_k is a class in V (e.g., yes/no in binary classification)
- $P(v_k)$ is the proportion of the number of elements in class v_k to the number of elements in V

A purity measure with entropy

- Entropy is maximal when all possibilities are equally likely.
- Entropy is zero in a pure "yes" (or pure "no") node.





Provost, Foster; Fawcett, Tom. Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking

- Decision tree aims to **decrease the entropy** in each node.

The wait@restaurant training data

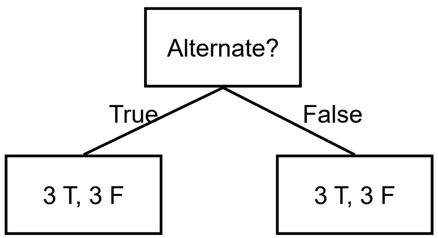
T = True, F = False

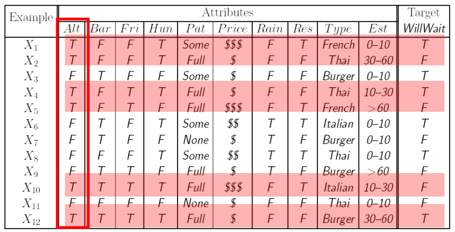
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

$$H(S) = -\left({}^{6}\!/_{12}\right)\log_2\left({}^{6}\!/_{12}\right) - \left({}^{6}\!/_{12}\right)\log_2\left({}^{6}\!/_{12}\right) = 1$$

6 True, 6 False

ID3 Decision tree: An example





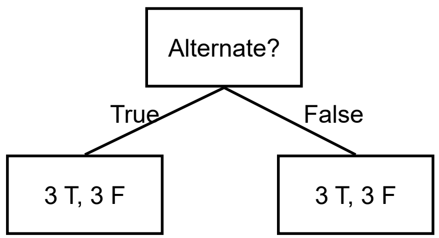
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

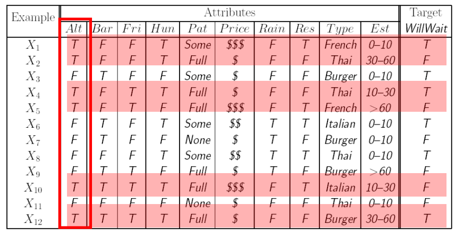
- Calculate Average Entropy of attribute Alternate

$$AE_{Alternate} = P(Alt = T) \times H(Alt = T) + P(Alt = F) \times H(Alt = F)$$

$$AE_{Alternate} = \frac{6}{12}\left[-\left(\frac{3}{6}\log_2\frac{3}{6}\right) - \left(\frac{3}{6}\log_2\frac{3}{6}\right)\right] + \frac{6}{12}\left[-\left(\frac{3}{6}\log_2\frac{3}{6}\right) - \left(\frac{3}{6}\log_2\frac{3}{6}\right)\right] = 1$$

ID3 Decision tree: An example



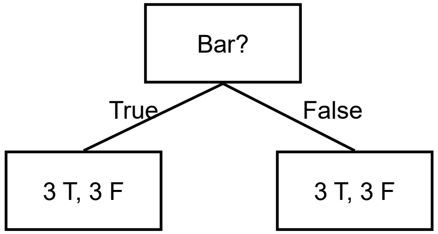


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

- **Information Gain** is the difference in entropy from before to after the set S is split on the selected attribute.

$$IG(Alternate, S) = H(S) - AE_{Alternate} = 1 - 1 = 0$$

ID3 Decision tree: An example

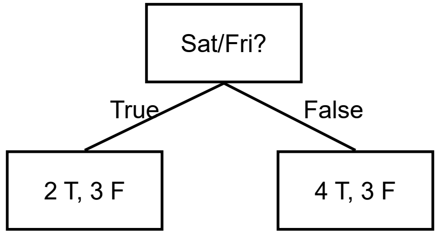


Example	Attributes										Target
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

$$AE_{Bar} = \frac{6}{12}\left[-\left(\frac{3}{6}\log_2\frac{3}{6}\right) - \left(\frac{3}{6}\log_2\frac{3}{6}\right)\right] + \frac{6}{12}\left[-\left(\frac{3}{6}\log_2\frac{3}{6}\right) - \left(\frac{3}{6}\log_2\frac{3}{6}\right)\right] = 1$$

$$IG(Bar, S) = H(S) - AE_{Bar} = 1 - 1 = 0$$

ID3 Decision tree: An example

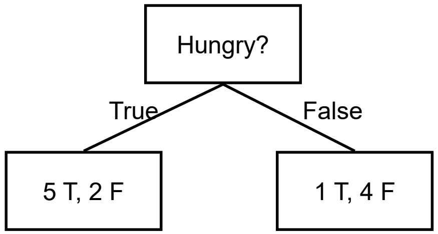


Example	Attributes										Target
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

$$AE_{Sat/Fri?} = \frac{5}{12}\left[-\left(\frac{2}{5}\log_2\frac{2}{5}\right) - \left(\frac{3}{5}\log_2\frac{3}{5}\right)\right] + \frac{7}{12}\left[-\left(\frac{4}{7}\log_2\frac{4}{7}\right) - \left(\frac{3}{7}\log_2\frac{3}{7}\right)\right]$$
$$= 0.979$$

$$IG(Sat/Fri?, S) = H(S) - AE_{Sat/Fri?} = 1 - 0.979 = 0.021$$

ID3 Decision tree: An example

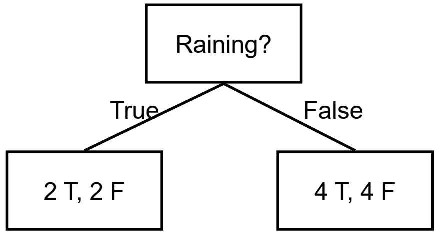


Example	Attributes										Target
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

$$AE_{Hungry} = \frac{7}{12}\left[-\left(\frac{5}{7}\log_2\frac{5}{7}\right) - \left(\frac{2}{7}\log_2\frac{2}{7}\right)\right] + \frac{5}{12}\left[-\left(\frac{1}{5}\log_2\frac{1}{5}\right) - \left(\frac{4}{5}\log_2\frac{4}{5}\right)\right]$$
$$= 0.804$$

$$IG(Hungry, S) = H(S) - AE_{Hungry} = 1 - 0.804 = 0.196$$

ID3 Decision tree: An example

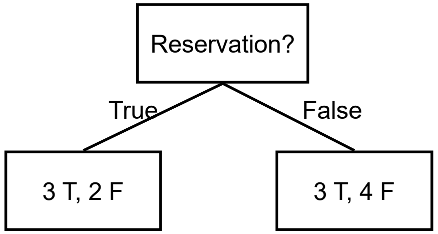


Example	Attributes										Target
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

$$AE_{Raining} = \frac{4}{12}\left[-\left(\frac{2}{4}\log_2\frac{2}{4}\right) - \left(\frac{2}{4}\log_2\frac{2}{4}\right)\right] + \frac{8}{12}\left[-\left(\frac{4}{8}\log_2\frac{4}{8}\right) - \left(\frac{4}{8}\log_2\frac{4}{8}\right)\right] = 1$$

$$IG(Raining, S) = H(S) - AE_{Hungry} = 1 - 1 = 0$$

ID3 Decision tree: An example

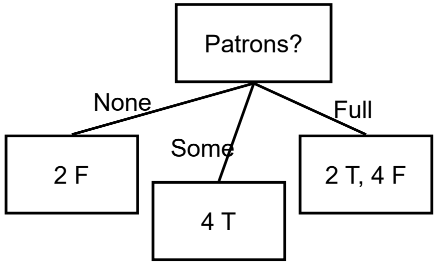


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

$$AE_{Reservation} = \frac{5}{12}\left[-\left(\frac{3}{5}\log_2\frac{3}{5}\right)-\left(\frac{2}{5}\log_2\frac{2}{5}\right)\right]+\frac{7}{12}\left[-\left(\frac{3}{7}\log_2\frac{3}{7}\right)-\left(\frac{4}{7}\log_2\frac{4}{7}\right)\right]$$
$$= 0.979$$

$$IG(Reservation, S) = H(S) - AE_{Reservation} = 1 - 0.979 = 0.021$$

ID3 Decision tree: An example

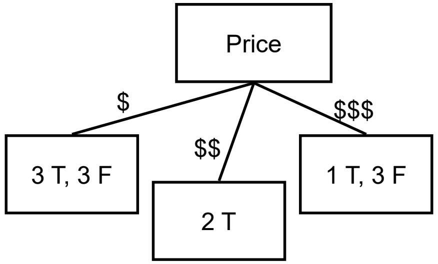


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

AE_{Patron}
$$= \frac{2}{12}\left[-\left(\frac{0}{2}\log_2\frac{0}{2}\right)-\left(\frac{2}{2}\log_2\frac{2}{2}\right)\right]+\frac{4}{12}\left[-\left(\frac{4}{4}\log_2\frac{4}{4}\right)-\left(\frac{0}{4}\log_2\frac{0}{4}\right)\right]$$
$$+\frac{6}{12}\left[-\left(\frac{2}{6}\log_2\frac{2}{6}\right)-\left(\frac{4}{6}\log_2\frac{4}{6}\right)\right] = 0.541$$

$$IG(Patron, S) = H(S) - AE_{Patron} = 1 - 0.541 = 0.459$$

ID3 Decision tree: An example

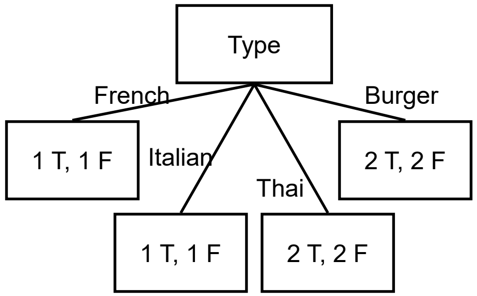


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

AE_{Price}
$$= \frac{6}{12}\left[-\left(\frac{3}{6}\log_2\frac{3}{6}\right)-\left(\frac{3}{6}\log_2\frac{3}{6}\right)\right]+\frac{2}{12}\left[-\left(\frac{2}{2}\log_2\frac{2}{2}\right)-\left(\frac{0}{2}\log_2\frac{0}{2}\right)\right]$$
$$+\frac{4}{12}\left[-\left(\frac{1}{4}\log_2\frac{1}{4}\right)-\left(\frac{3}{4}\log_2\frac{3}{4}\right)\right] = 0.770$$

$$IG(Price, S) = H(S) - AE_{Price} = 1 - 0.770 = 0.23$$

ID3 Decision tree: An example

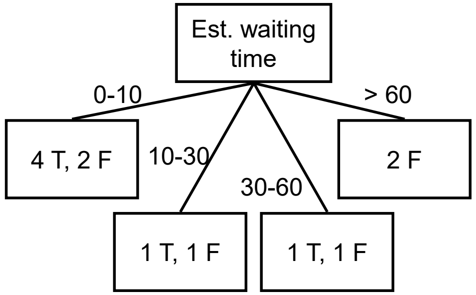


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

AE_{Type}
$$= \frac{2}{12}\left[-\left(\frac{1}{2}\log_2\frac{1}{2}\right)-\left(\frac{1}{2}\log_2\frac{1}{2}\right)\right]+\frac{2}{12}\left[-\left(\frac{1}{2}\log_2\frac{1}{2}\right)-\left(\frac{1}{2}\log_2\frac{1}{2}\right)\right]$$
$$+\frac{4}{12}\left[-\left(\frac{2}{4}\log_2\frac{2}{4}\right)-\left(\frac{2}{4}\log_2\frac{2}{4}\right)\right]+\frac{4}{12}\left[-\left(\frac{2}{4}\log_2\frac{2}{4}\right)-\left(\frac{2}{4}\log_2\frac{2}{4}\right)\right] = 1$$

$$IG(Type, S) = H(S) - AE_{Type} = 1 - 1 = 0$$

ID3 Decision tree: An example



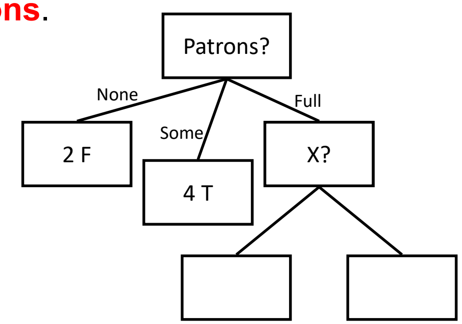
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

$$
\begin{aligned}
&AE_{Est.waiting\ time}\\
&= \frac{6}{12}\left[-\left(\frac{4}{6}\log_2\frac{4}{6}\right)-\left(\frac{2}{6}\log_2\frac{2}{6}\right)\right]+\frac{2}{12}\left[-\left(\frac{1}{2}\log_2\frac{1}{2}\right)-\left(\frac{1}{2}\log_2\frac{1}{2}\right)\right]\\
&+\frac{2}{12}\left[-\left(\frac{1}{2}\log_2\frac{1}{2}\right)-\left(\frac{1}{2}\log_2\frac{1}{2}\right)\right]+\frac{2}{12}\left[-\left(\frac{0}{2}\log_2\frac{0}{2}\right)-\left(\frac{2}{2}\log_2\frac{2}{2}\right)\right]=0.792
\end{aligned}
$$

$$IG(Est.waiting\ time, S) = H(S) - AE_{Est.waiting\ time} = 1 - 0.792$$
$$= 0.208$$

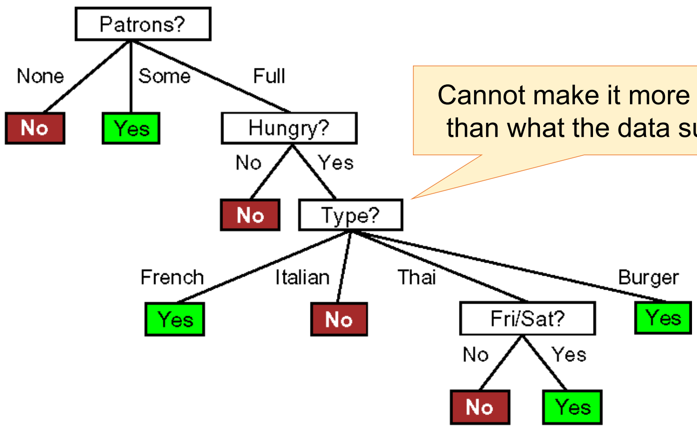
ID3 Decision tree: An example

- Largest Information Gain (0.459) / Smallest Entropy (0.541) achieved by splitting on **Patrons**.



- Continue making new splits, always purifying nodes

ID3 Decision tree algorithm

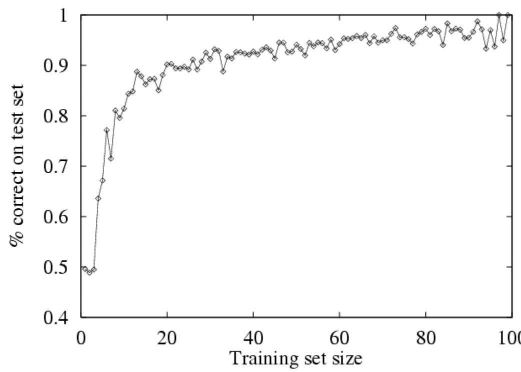


Cannot make it more complex than what the data supports.

Induced tree (from examples)

Performance measurement

- How do we know that $h \approx f$?
  1. Use theorems of computational or statistical learning theory
  2. Try h on a new test set of examples
     - Use the same distribution over example space as training set



Learning curve = % correct on test set as a function of training set size

Quiz 01: ID3 decision tree

- The data represent files on a computer system. Possible values of the class variable are "infected", which implies the file has a virus infection, or "clean" if it doesn't.
- Derive decision tree for virus identification.

No.	Writable	Updated	Size	Class
1	Yes	No	Small	Infected
2	Yes	Yes	Large	Infected
3	No	Yes	Med	Infected
4	No	No	Med	Clean
5	Yes	No	Large	Clean
6	No	No	Large	Clean

Naïve Bayesian classification

Bayesian classification

- A statistical classifier performs probabilistic prediction, i.e., predicts class membership probabilities
- Foundation: Based on **Bayes' Theorem**

$$\underbrace{P(c \mid x)}_{\text{Posterior Probability}} = \frac{\overbrace{P(x \mid c)}^{\text{Likelihood}}\,\overbrace{P(c)}^{\text{Class Prior Probability}}}{\underbrace{P(x)}_{\text{Predictor Prior Probability}}}$$

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Bayesian classification

- Performance
  - A simple Bayesian classifier (e.g., naïve Bayesian), has comparable performance with decision tree and selected neural networks.
- Incremental
  - Each training example can incrementally increase/decrease the probability that a hypothesis is correct
  - That is, prior knowledge can be combined with observed data.
- Standard
  - Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

The buying computer dataset

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Bayes' Theorem

- Total Probability Theorem: $P(B) = \sum_{i=1}^{M} P(B|A_i)P(A_i)$
- Let $\mathbf{X}$ be a data sample ("evidence") with unknown class label and H be a hypothesis that $\mathbf{X}$ belongs to class C
- **Bayes' Theorem:** $\boldsymbol{P(H \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid H)P(H)}{P(\mathbf{X})}}$
- Classification is to determine $P(H \mid \mathbf{X})$, the probability that the hypothesis H holds given the observed data sample $\mathbf{X}$.

Bayes' Theorem

- $\boldsymbol{P(H)}$ (prior probability): the initial probability
  - E.g., $\mathbf{X}$ will buy computer, regardless of age, income, …
- $P(\mathbf{X})$: the probability that sample data is observed
  - E.g., $\mathbf{X}$ is 31..40 and has a medium income, regardless of the buying
- $P(\mathbf{X} \mid H)$ (likelihood): the probability of observing the sample $\mathbf{X}$, given that the hypothesis holds
  - E.g., given that $\mathbf{X}$ will buy computer, the probability that $\mathbf{X}$ is 31..40 and has a medium income
- $\boldsymbol{P(H \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid H)P(H)}{P(\mathbf{X})}}$(posterior probability)
  - E.g., given that $\mathbf{X}$ is 31..40 and has a medium income, the probability that $\mathbf{X}$ will buy computer

Bayes' Theorem

- Informally, $P(H \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid H)P(H)}{P(\mathbf{X})}$ can be viewed as

  $posteriori \;=\; likelihood * prior \,/\, evidence$
- $\mathbf{X}$ belongs to C_i iff the probability $P(C_i \mid \mathbf{X})$ is the highest among all the $P(C_k \mid \mathbf{X})$ for all the k classes
- **Practical difficulty**
  - Require initial knowledge of many probabilities
  - Significant computational cost involved

Classification with Bayes' Theorem

- Let D be a training set of tuples and associated class labels
- Each tuple is represented by a n-attribute $\mathbf{X} = (x_1, x_2, \dots, x_n)$
- Suppose there are m classes $C_1, C_2, \dots, C_m$
- Classification is to derive the maximum posteriori $P(C_i | \mathbf{X})$ from **Bayes' theorem**

$$P(C_i \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid C_i) P(C_i)}{P(\mathbf{X})}$$

- $P(X)$ is constant for all classes, only $P(\mathbf{X} \mid C_i) P(C_i)$ needs to be maximized.

Naïve Bayesian classification

- Class-conditional independence: There are no dependence relationships **among the attributes**
- The **naïve Bayesian classification** formula is written as

$$P(\mathbf{X} \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) = P(x_1 \mid C_i) \times P(x_2 \mid C_i) \times \cdots \times P(x_n \mid C_i)$$

  - A_k is categorical: $P(x_k \mid C_i)$ is the number of tuples in C_i having value x_k for A_k divided by $|C_{i,D}|$ (# of tuples of C_i in D)
  - A_k is continuous: $P(x_k \mid C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$ with the Gaussian distribution $g(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
- Count class distributions only → computation cost reduced

Naïve Bayesian classification: An example

P(buys_computer = "yes")	9/14
P(buys_computer = "no")	5/14

	buys_computer = "yes"	buys_computer = "no"
age = "<=30"	2/9	3/5
age = "31…40"	4/9	0/5
age = ">40"	3/9	2/5
income = "low"	3/9	1/5
income = "medium"	4/9	2/5
income = "high"	2/9	2/5
student = "yes"	6/9	1/5
student = "no"	3/9	4/5
credit_rating = "fair"	6/9	2/5
credit_rating = "excellent"	3/9	3/5

Naïve Bayesian classification: An example

age	income	student	credit_rating	buys_computer
<=30	medium	yes	fair	?

- $P(\mathbf{X} | C_i)$
  - $P(\mathbf{X} \mid buys\_computer = \text{"yes"}) = 2/9 * 4/9 * 6/9 * 6/9 = 0.044$
  - $P(\mathbf{X} \mid buys\_computer = \text{"no"}) = 3/5 * 2/5 * 1/5 * 2/5 = 0.019$
- $P(\mathbf{X} | C_i) * P(C_i)$
  - $P(\mathbf{X} \mid buys\_computer = \text{"yes"}) * P(buys\_computer = \text{"yes"}) = 0.028$
  - $P(\mathbf{X} | buys\_computer = \text{"no"}) * P(buys\_computer = \text{"no"}) = 0.007$
- $P(C_i \mid \mathbf{X})$
  - $P(buys\_computer = \text{"yes"} \mid \mathbf{X}) = 0.8$
  - $P(buys\_computer = \text{"no"} \mid \mathbf{X}) = 0.2$

Therefore, $\mathbf{X}$ belongs to class ("buys_computer = yes")

Avoiding the zero-probability issue

- The naïve Bayesian prediction requires each conditional probability be **non-zero**.

$$P(\mathbf{X} \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$

- Otherwise, the predicted probability will be zero
- For example,

age	income	student	credit_rating	buys_computer
31…40	medium	yes	fair	?

  - $P(\mathbf{X} \mid buys\_computer = \text{"}no\text{"}) = 0 * 2/5 * 1/5 * 2/5 = 0$
  - Therefore, the conclusion is always **yes** regardless the value of $P(\mathbf{X} \mid buys\_computer = \text{"}yes\text{"})$

Avoiding the zero-probability issue

- **Laplacian correction** (or Laplacian estimator)

$$P(C_i) = \frac{|C_i| + 1}{|D| + m} \qquad P(x_k \mid C_i) = \frac{|x_k \cup C_i| + 1}{|C_i| + r}$$

  - where m is the number of classes, $|x_k \cup C_i|$ denotes the number of tuples contains both $A_k = x_k$ and C_i, and r is the number of values of attribute A_k
- The "corrected" probability estimates are close to their "uncorrected" counterparts

Naïve Bayesian classification: An example

P(buys_computer = "yes")	10/16
P(buys_computer = "no")	6/16

	buys_computer = "yes"	buys_computer = "no"
age = "<=30"	3/12	4/8
age = "31…40"	5/12	1/8
age = ">40"	4/12	3/8
income = "low"	4/12	2/8
income = "medium"	5/12	3/8
income = "high"	3/12	3/8
student = "yes"	7/11	2/7
student = "no"	4/11	5/7
credit_rating = "fair"	7/11	3/7
credit_rating = "excellent"	4/11	4/7

Naïve Bayesian classification: An example

age	income	student	credit_rating	buys_computer
31..40	medium	yes	fair	?

- $P(\mathbf{X} \mid C_i)$
  - $P(\mathbf{X} \mid buys\_computer = \text{"}yes\text{"}) = 5/12 * 5/12 * 7/11 * 7/11 = 0.070$
  - $P(\mathbf{X} \mid buys\_computer = \text{"}no\text{"}) = 1/8 * 3/8 * 2/7 * 3/7 = 0.006$
- $P(\mathbf{X} \mid C_i) * P(C_i)$
  - $P(\mathbf{X} \mid buys\_computer = \text{"}yes\text{"}) * P(buys\_computer = \text{"}yes\text{"}) = 0.044$
  - $P(\mathbf{X} \mid buys\_computer = \text{"}no\text{"}) * P(buys\_computer = \text{"}no\text{"}) = 0.002$
- $P(C_i \mid \mathbf{X})$
  - $P(buys\_computer = \text{"}yes\text{"} \mid \mathbf{X}) = 0.953$
  - $P(buys\_computer = \text{"}no\text{"} \mid \mathbf{X}) = 0.047$

Therefore, $\mathbf{X}$ belongs to class ("buys_computer = yes")

Handling missing values

- If the values of some attributes are missing, these attributes are omitted from the product of probabilities
- As a result, the estimation is less accurate
- For example,

age	income	student	credit_rating	buys_computer
?	medium	yes	fair	?

Naïve Bayesian classification: Evaluation

- Advantages
  - Easy to implement
  - Good results obtained in most of the cases
- Disadvantages
  - Class conditional independence → loss of accuracy
  - Practically, dependencies exist among variables, which cannot be modeled by Naïve Bayes
    - E.g., in medical records, patients' profile (age, family history, etc.), symptoms (fever, cough etc.), disease (lung cancer, diabetes, etc.)
- *How to deal with these dependencies?*
  - Bayesian Belief Networks

Quiz 02: Naïve Bayesian classification

- The data represent files on a computer system. Possible values of the class variable are "infected", which implies the file has a virus infection, or "clean" if it doesn't.
- Derive naïve Bayesian probabilities for virus identification in either cases, with or without Laplacian correction.

No.	Writable	Updated	Size	Class
1	Yes	No	Small	Infected
2	Yes	Yes	Large	Infected
3	No	Yes	Med	Infected
4	No	No	Med	Clean
5	Yes	No	Large	Clean
6	No	No	Large	Clean

THE END