INTELLIGENZA ARTIFICIALE

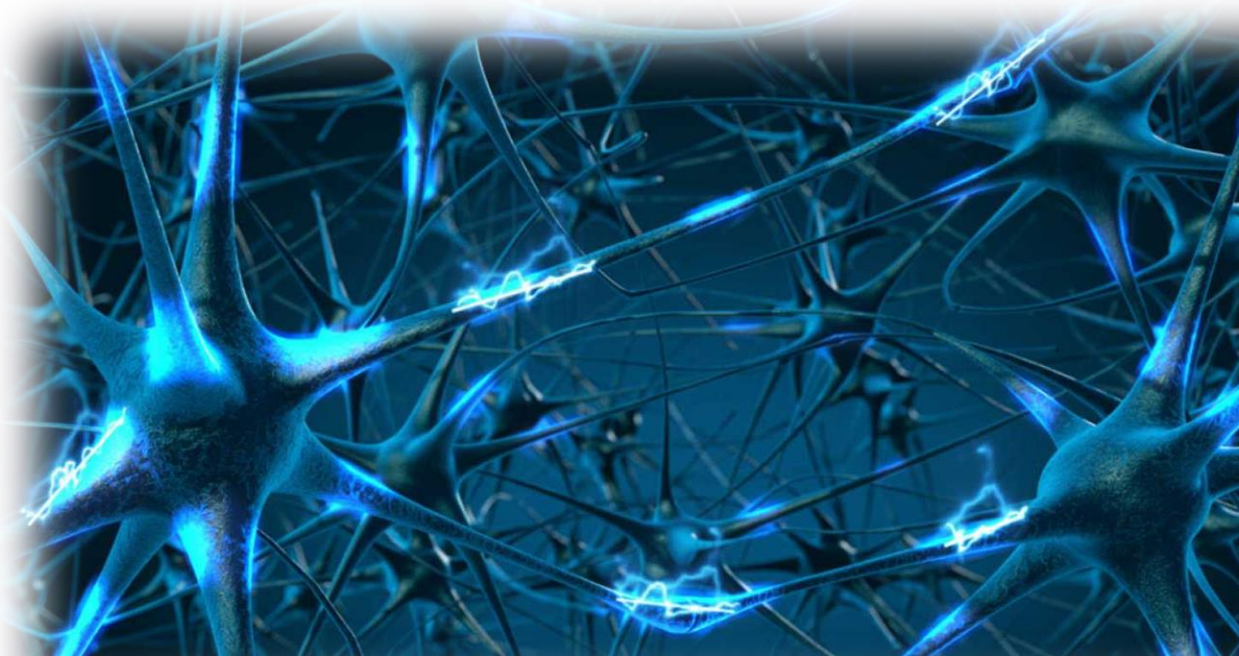
APPRENDIMENTO AUTOMATICO DA ESEMPI

Corsi di Laurea in Informatica, Ing. Gestionale, Ing. Informatica, Ing. di Internet

(a.a. 2022-2023)

Roberto Basili

(*) dalle *slides* di S. Russel



Overview (AIMA chpt. 18.1-18.4)

- Agents & machine learning
- Learning from examples:
  - Complexity and Expressiveness
  - The definition of model selection
- Performance Evaluation
- Learning methodology: design, experiment/ evaluation and model selection
  - Cross validation
- An example: Decision Tree learning
  - Recursive search among Boolean formulas
  - Attribute Selection in DT: Information Gain

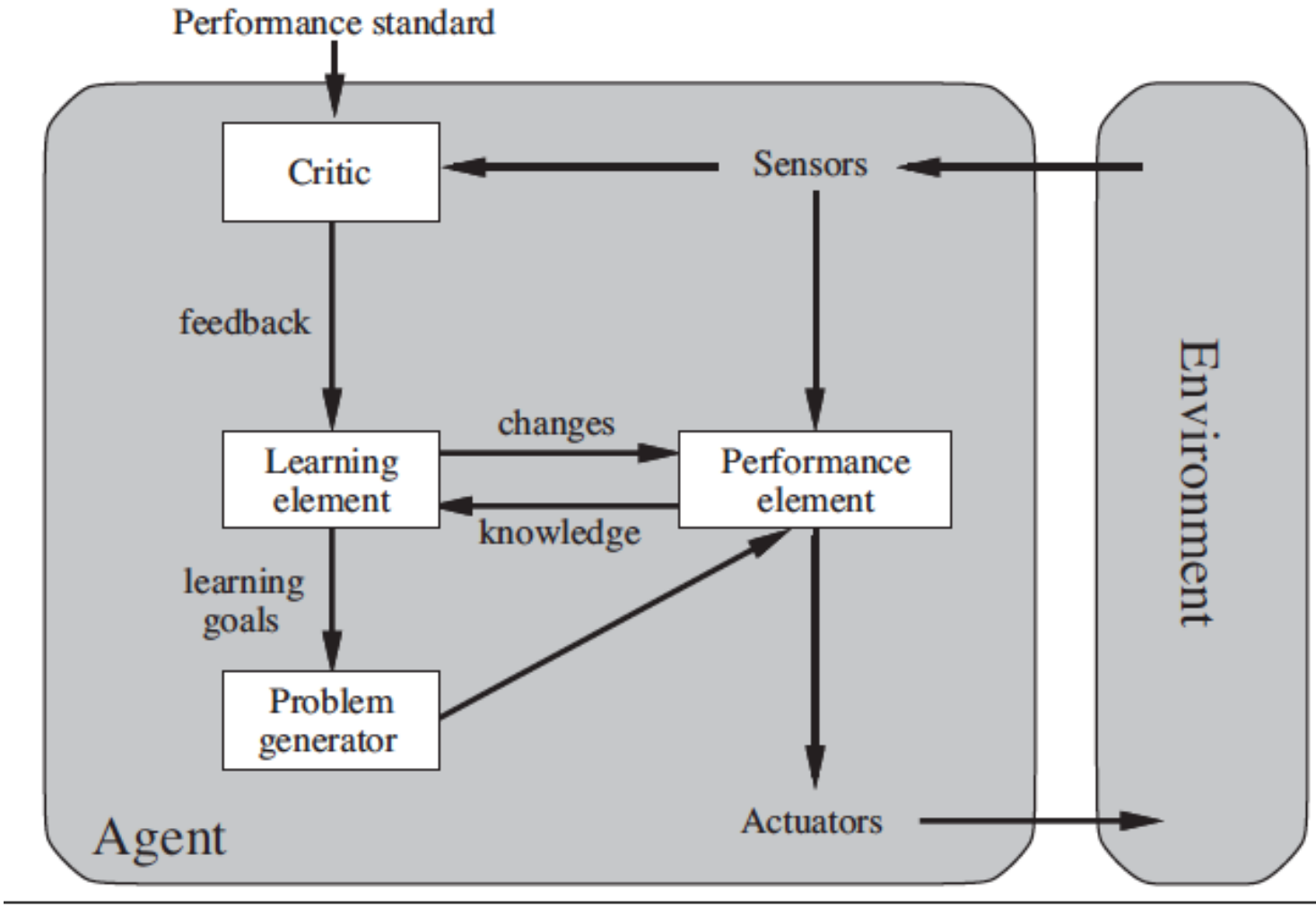
Introduction to machine learning

- Introduction to machine learning
  - When appropriate and when not appropriate
  - Task definition

  - Learning methodology: design, experiment, evaluation
  - Learning issues: representing hypothesis

  - Learning paradigms
    - Supervised learning
    - Unsupervised learning
    - Reinforcement learning

AIMA learning architecture



Machine learning: definition

- *A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E* [Mitchell]
- Problem definition for a learning agent
  - Task T
  - Performance measure P
  - Experience E

Designing a learning system

1. Choosing the training experience
   - Examples of best moves, games outcome …
2. Choosing the target function
   - board-move, board-value, …
3. Choosing a representation for the target function
   - linear function with weights (hypothesis space)
4. Choosing a learning algorithm for approximating the target function
   - A method for parameter estimation

Inductive learning

- Simplest form: learn a function from examples

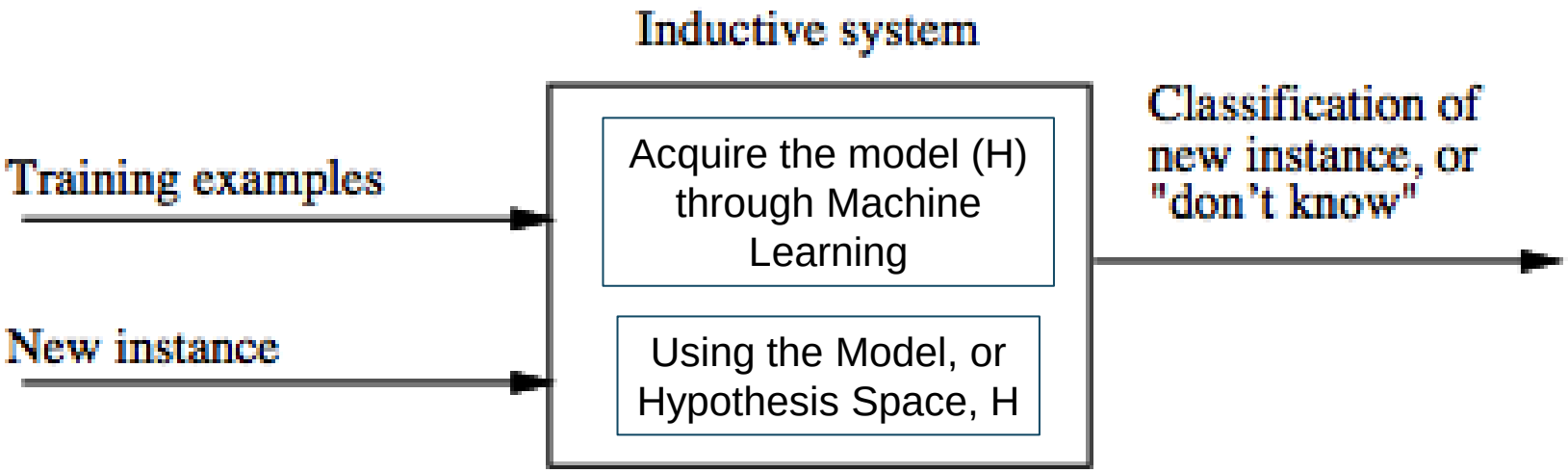
f is the target function

An example is a pair (*x*, *f(x)*)

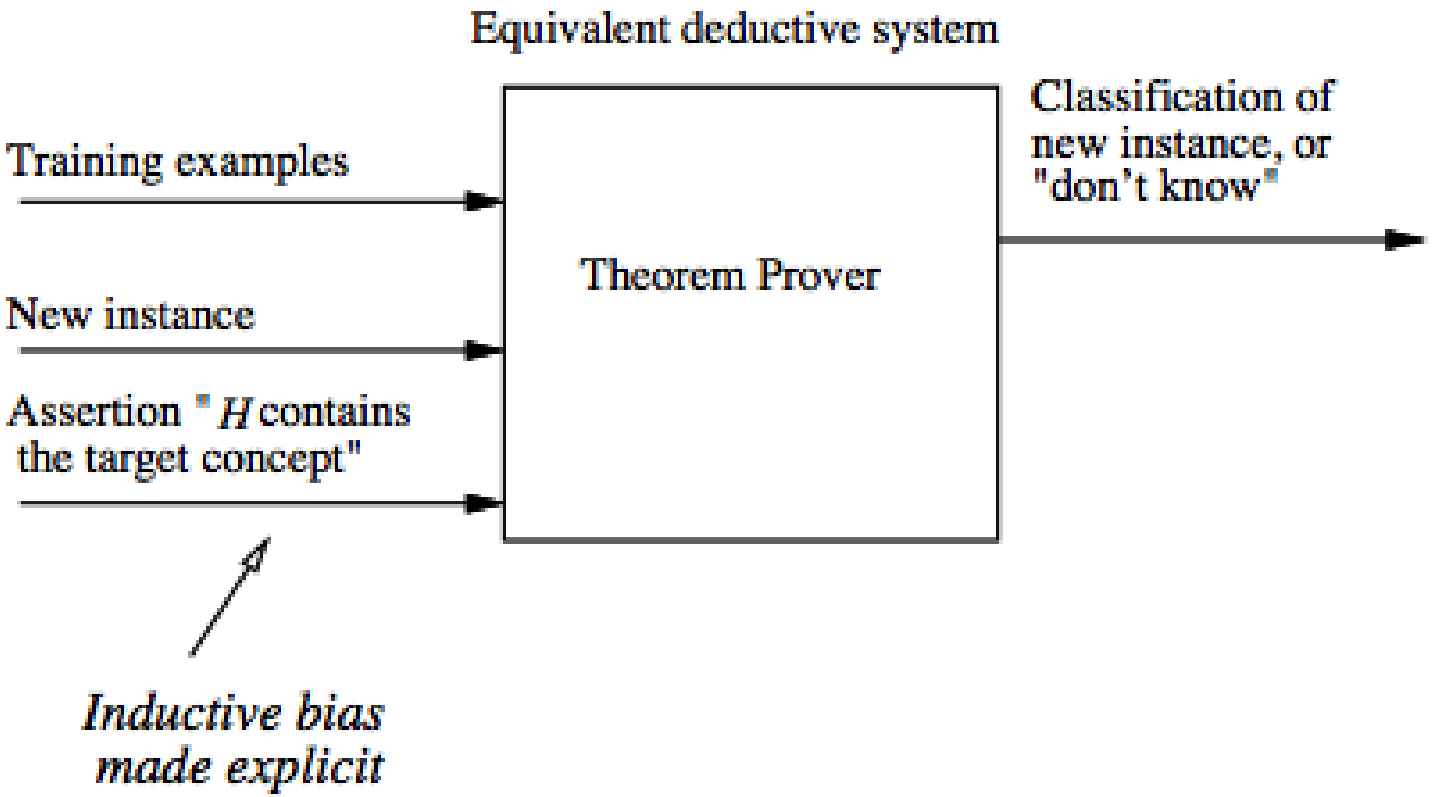
Problem: find a hypothesis *h*
such that $h \approx f$
given a training set of examples

(This is a highly simplified model of real learning:
- Ignores prior knowledge
- Assumes examples are given)

Inductive system



Equivalent deductive system



Learning decision trees

Problem: decide whether to wait for a table at a restaurant, 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)

Attribute-based representations

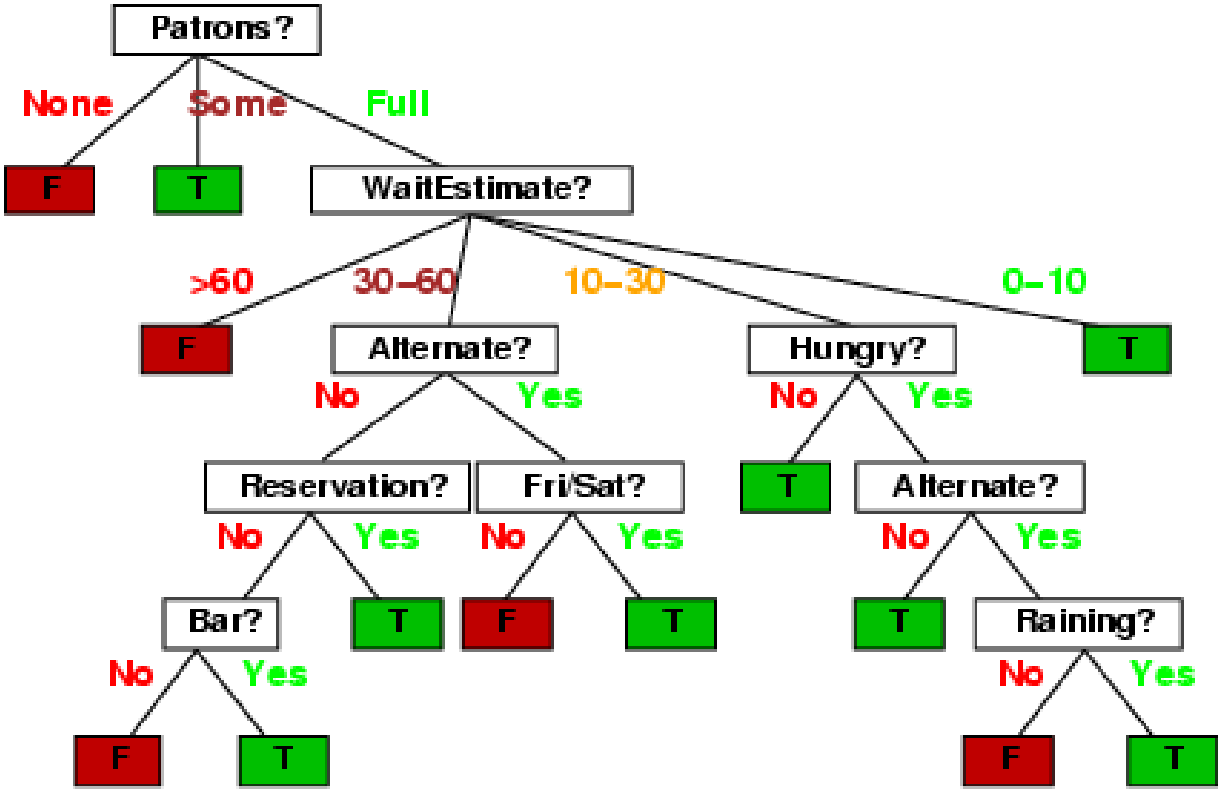
- Examples described by attribute values (Boolean, discrete, continuous)
- E.g., situations where I will/won't wait for a table:

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
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

- Classification of examples is positive (T) or negative (F)

Decision trees

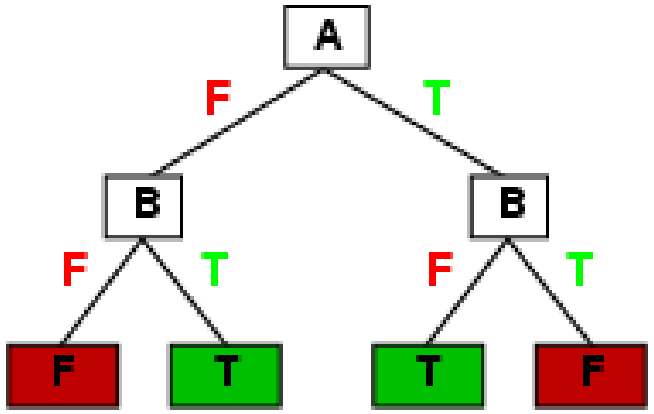
- One possible representation for hypotheses
- E.g., here is the “true” tree for deciding whether to wait:



Expressiveness

- Decision trees can express any function of the input attributes.
- E.g., for Boolean functions, truth table row → path to leaf:

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



- Trivially, there is a consistent decision tree for any training set with one path to leaf for each example (unless *f* nondeterministic in *x*) but it probably won't generalize to new examples

- Prefer to find more compact decision trees

Hypothesis spaces

How many distinct decision trees with *n* Boolean attributes?

= number of Boolean functions

= number of distinct truth tables with 2^n rows = 2^{2^n}

- E.g., with 6 Boolean attributes, there are 2^{64}=18,446,744,073,709,551,616 trees

Hypothesis spaces

How many distinct decision trees with *n* Boolean attributes?

= number of Boolean functions

= number of distinct truth tables with 2^n rows = 2^{2^n}

- E.g., with 6 Boolean attributes, there are 18,446,744,073,709,551,616 trees

How many purely conjunctive hypotheses (e.g., *Hungry* $\wedge \neg$*Rain*)?

- Each attribute can be in (positive), in (negative), or out
  $\Rightarrow 3^n$ distinct conjunctive hypotheses
- More expressive hypothesis space
  - increases chance that target function can be expressed
  - increases number of hypotheses consistent with training set
    $\Rightarrow$ may get worse predictions

Decision tree learning

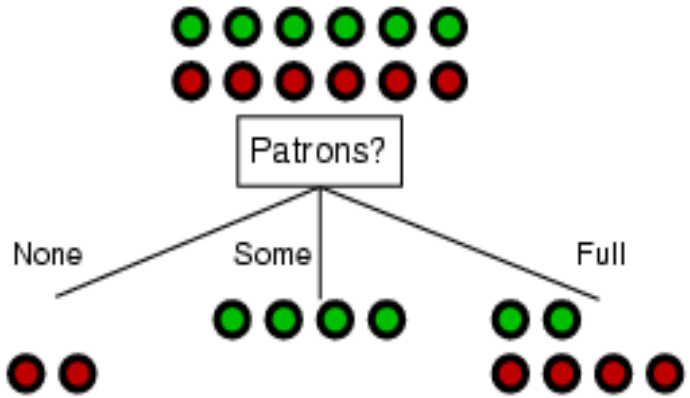
- Aim: find a *small* tree consistent with the training examples
- Idea: (recursively) choose "most significant" attribute as root of (sub)tree

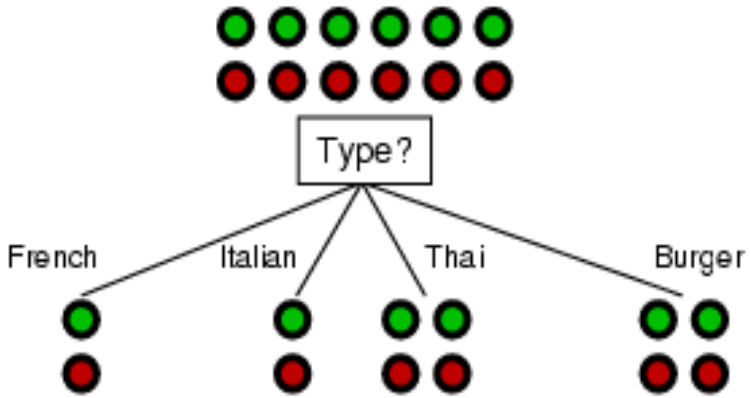
```
function DTL(examples, attributes, default) returns a decision tree
    if examples is empty then return default
    else if all examples have the same classification then return the classification
    else if attributes is empty then return MODE(examples)
    else
        best ← CHOOSE-ATTRIBUTE(attributes, examples)
        tree ← a new decision tree with root test best
        for each value v_i of best do
            examples_i ← {elements of examples with best = v_i}
            subtree ← DTL(examples_i, attributes − best, MODE(examples))
            add a branch to tree with label v_i and subtree subtree
        return tree
```

Choosing an attribute

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
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

- Idea: a good attribute splits the examples into subsets that are (ideally) "all positive" or "all negative"





- *Patrons?* is a better choice

Using information theory

- To implement CHOOSE-ATTRIBUTE in the DTL algorithm

- Information Content (Entropy):

$$I(P(v_1), \ldots, P(v_n)) = \sum_{i=1} -P(v_i) \log_2 P(v_i)$$

- For a training set containing *p* positive examples and *n* negative examples:

$$I(\frac{p}{p+n}, \frac{n}{p+n}) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Information

Information answers questions

The more clueless I am about the answer initially, the more information is contained in the answer

Scale: 1 bit = answer to Boolean question with prior $\langle 0.5, 0.5 \rangle$

Information in an answer when prior is $\langle P_1, \ldots, P_n \rangle$ is

$$H(\langle P_1, \ldots, P_n \rangle) = \sum_{i=1}^{n} - P_i \log_2 P_i$$

(also called entropy of the prior)

Information gain

- A chosen attribute *A* divides the training set *E* into subsets $E_1, \ldots, E_v$ according to their values for *A*, where *A* has v distinct values.

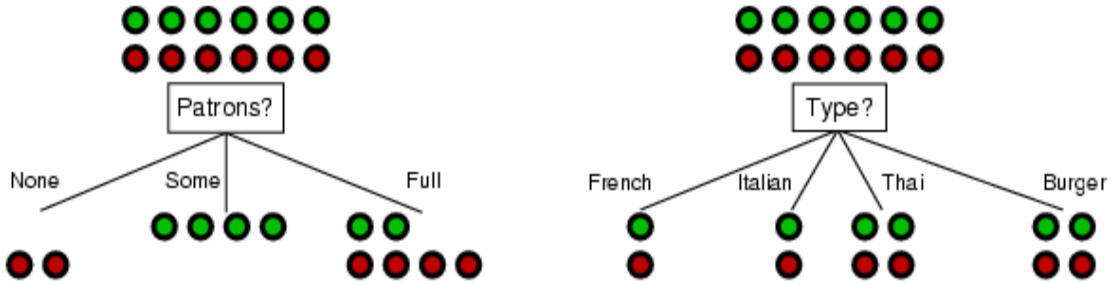
$$remainder(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

- Information Gain (IG) or reduction in entropy from the attribute test:

$$IG(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - remainder(A)$$

- Choose the attribute with the largest IG

Information gain



For the training set, $p = n = 6$, *I(6/12, 6/12) = 1* bit

Consider the attributes *Patrons* and *Type* (and others too):

$$IG(Patrons) = 1 - [\frac{2}{12} I(0,1) + \frac{4}{12} I(1,0) + \frac{6}{12} I(\frac{2}{6}, \frac{4}{6})] = .0541 \text{ bits}$$

$$IG(Type) = 1 - [\frac{2}{12} I(\frac{1}{2}, \frac{1}{2}) + \frac{2}{12} I(\frac{1}{2}, \frac{1}{2}) + \frac{4}{12} I(\frac{2}{4}, \frac{2}{4}) + \frac{4}{12} I(\frac{2}{4}, \frac{2}{4})] = 0 \text{ bits}$$

Patrons has the highest IG of all attributes and so is chosen by the DTL algorithm as the root

Information contd.

Suppose we have p positive and n negative examples at the root
$\Rightarrow$ $H(\langle p/(p+n), n/(p+n)\rangle)$ bits needed to classify a new example
E.g., for 12 restaurant examples, $p = n = 6$ so we need 1 bit

An attribute splits the examples E into subsets E_i, each of which (we hope) needs less information to complete the classification

Let E_i have p_i positive and n_i negative examples
$\Rightarrow$ $H(\langle p_i/(p_i+n_i), n_i/(p_i+n_i)\rangle)$ bits needed to classify a new example
$\Rightarrow$ **expected** number of bits per example over all branches is

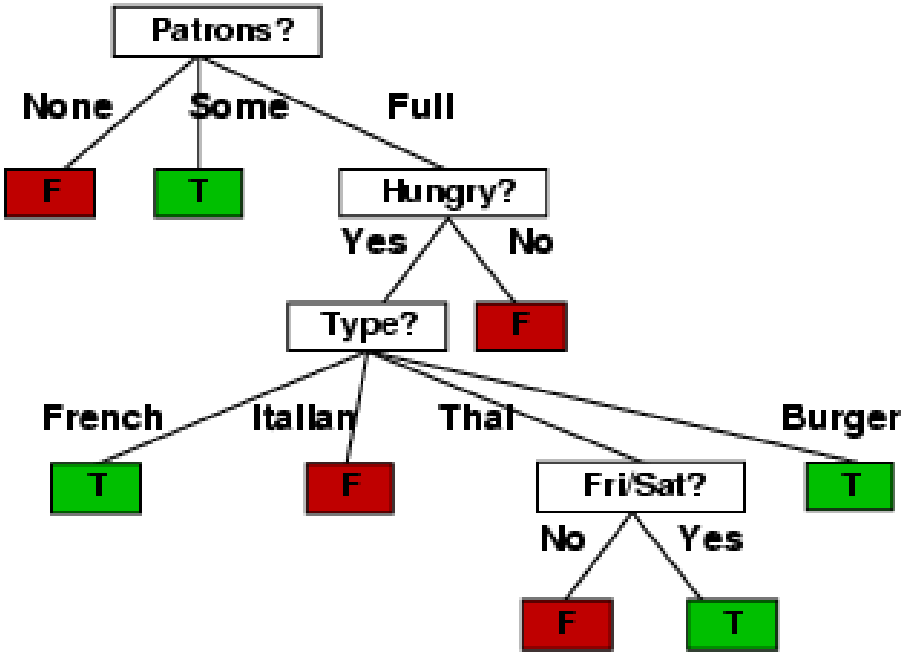
$$\sum_i \frac{p_i + n_i}{p + n} H(\langle p_i/(p_i + n_i), n_i/(p_i + n_i)\rangle)$$

For $Patrons?$, this is 0.459 bits, for $Type$ this is (still) 1 bit

$\Rightarrow$ choose the attribute that minimizes the remaining information needed

Example contd.

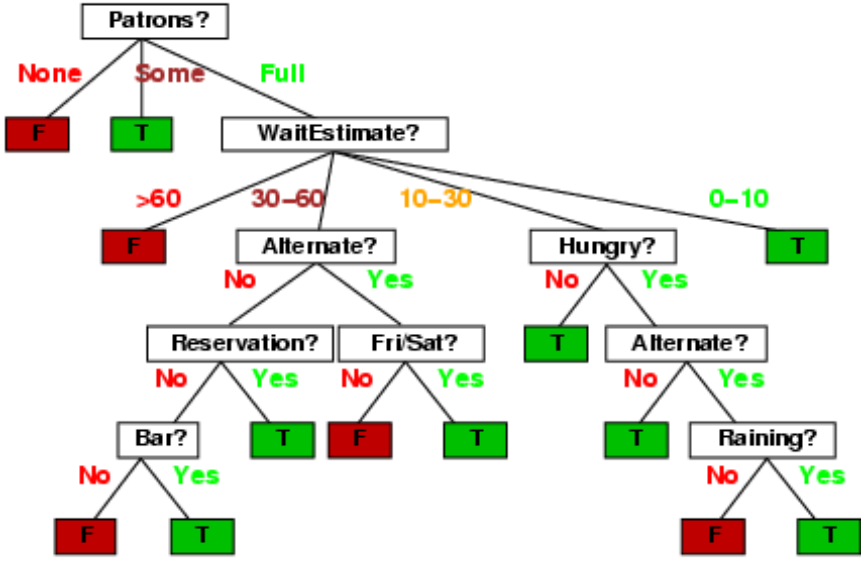
- Decision tree learned from the 12 examples:



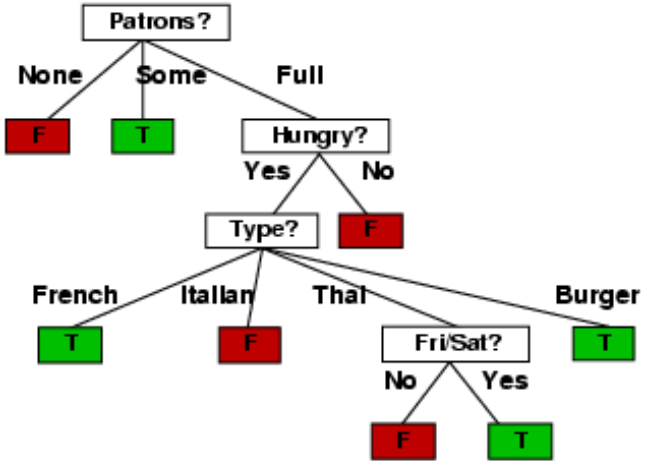
- Substantially simpler than “true” tree ---a more complex hypothesis isn’t justified by small amount of data

Example contd.

- Model selection through IG

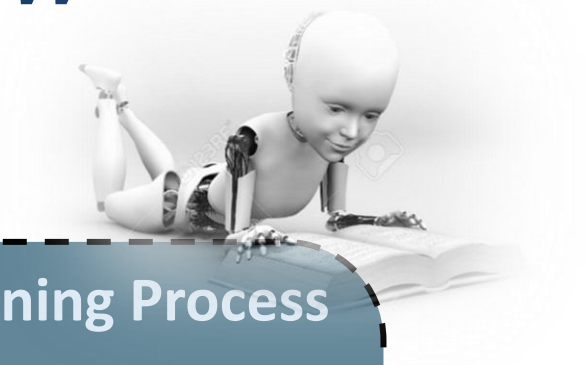


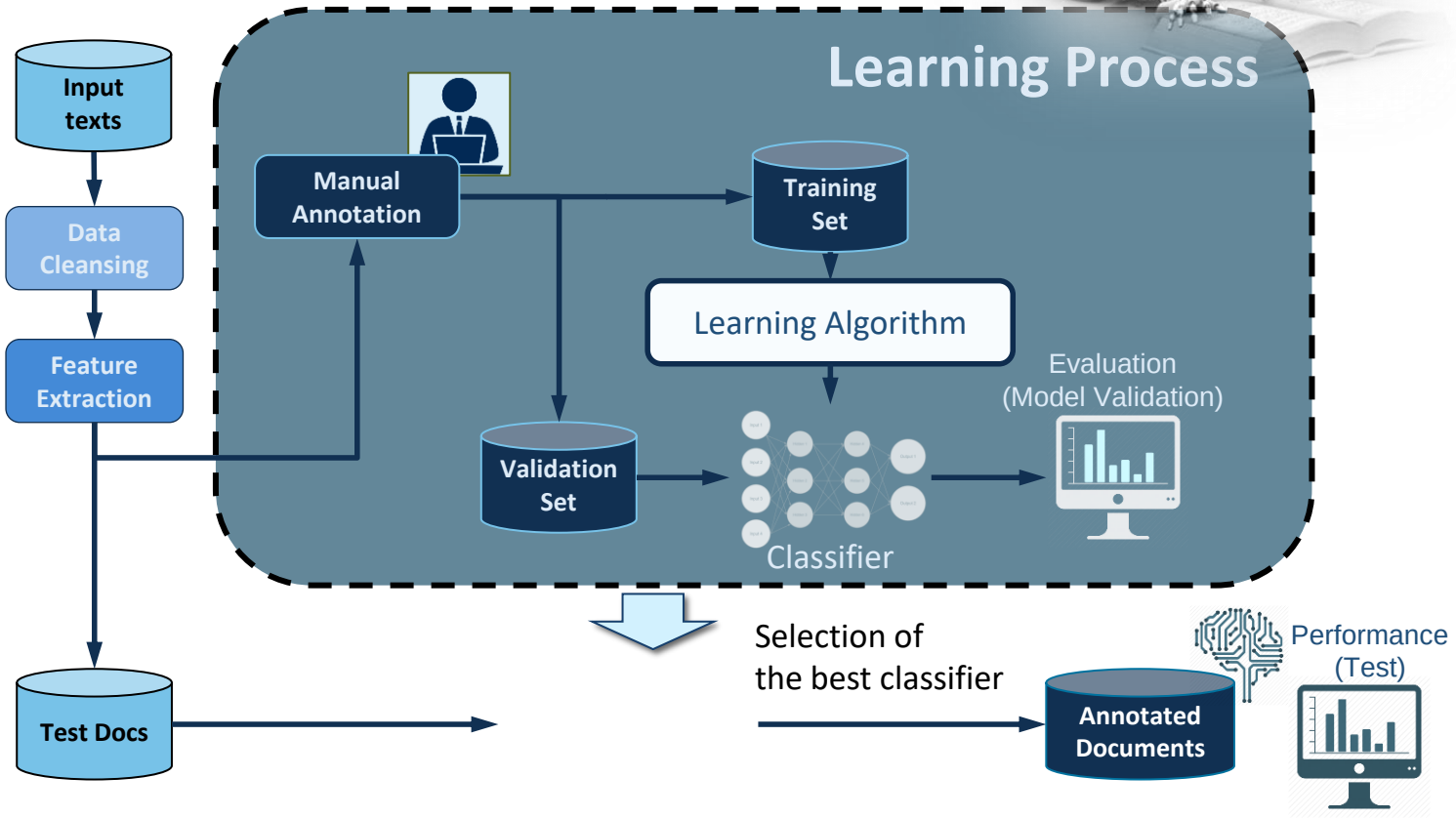
POTENTIAL (CONSISTENT) MODEL



OPTIMAL (CONSISTENT) MODEL

Machine Learning workflow





Evaluation of a ML system

- Performance Evaluation Metrics
  - Classifier Evaluation Metrics
- Tuning and Evaluation Methods

Single Class Metrics

		PREDICTED VALUE	
		Class C	**Not Class C**
ACTUAL VALUE	**Class C**	**TP** True Positive	**FN** False Negative
	Not Class C	**FP** False Positive	**TN** True Negative

$$precision = \frac{TP}{TP + FP}$$

what percentage of instances the classifier labeled as positive are actually positive?

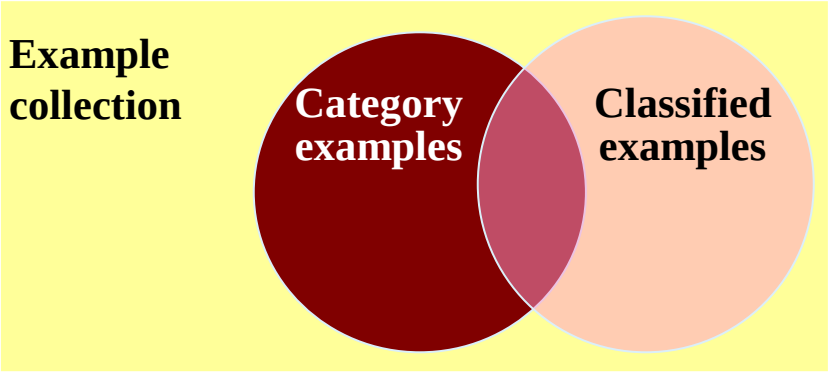
$$recall = \frac{TP}{TP + FN}$$

what percentage of positive instances did the classifier label as positive?

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

F-measure is the harmonic mean of precision and recall

Class-based evaluation



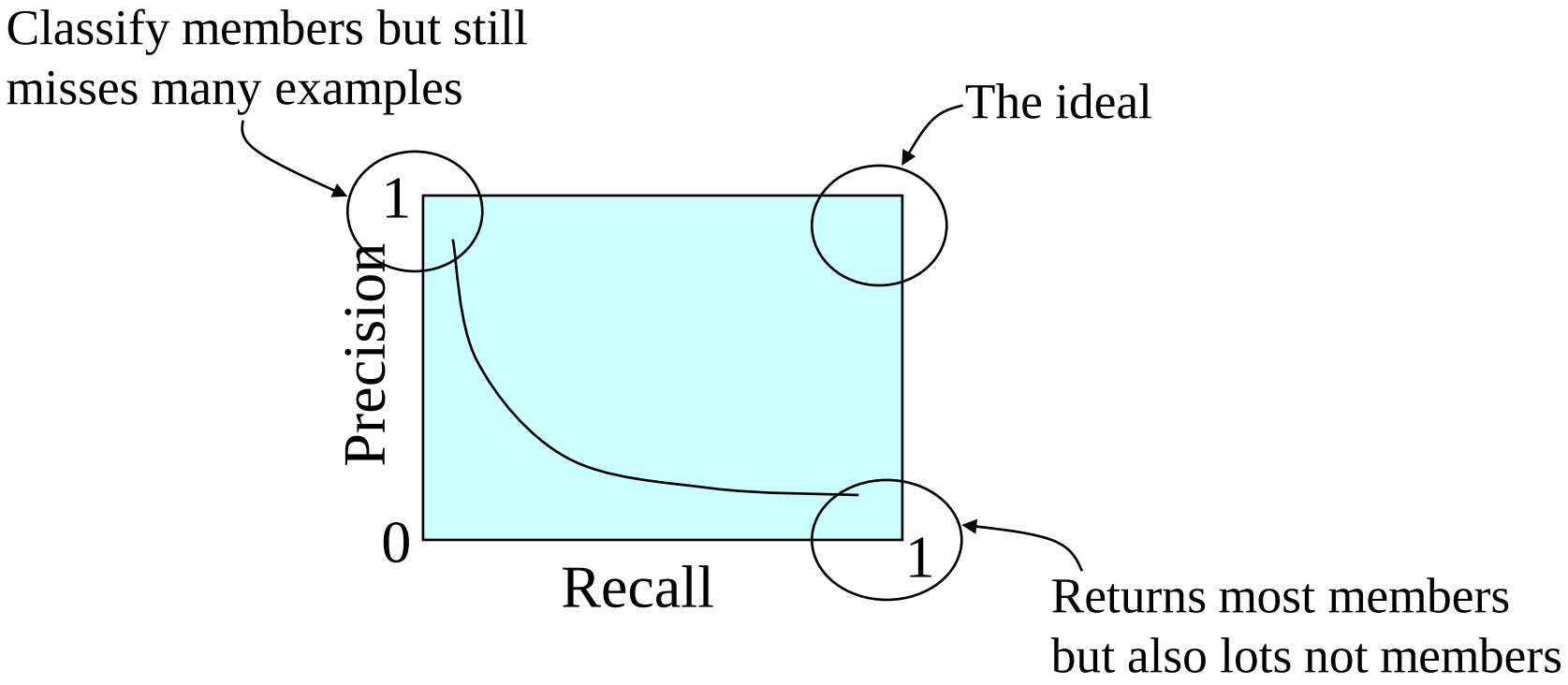
	Classified	Rejected
Not Members	Classified but not Members	Rejected & not Members
Members	Classified & Members	Rejected but Members

$$precision = \frac{\#\ of\ Members\ Classified}{\#\ of\ Members\ Classified\ +\ \#\ of\ Classified\ not\ Members}$$

$$recall = \frac{\#\ of\ Members\ Classified}{\#\ of\ Members\ Classified\ +\ \#\ of\ Rejected\ Members}$$

What about accuracy???

Trade-off between Precision and Recall



Other class based measures

Precision and Recall of C_i

- a, corrects (TP_i)
- b, mistakes (FP_i)
- c, instances of a $Class_i$ that are not actually retrieved, (FN_i)

The *Precision* and *Recall* are defined by the above counts:

$$Precision_i = \frac{a_i}{a_i + b_i}$$

$$Recall_i = \frac{a_i}{a_i + c_i}$$

		PREDICTED VALUE		
ACTUAL VALUE		Class A	Class B	Class C
	Class A	38	12	0
	Class B	5	43	2
	Class C	6	0	44

- $\text{Precision}_A = 38/(38+5+6)=38/49$
- $\text{Recall}_A = 38/(38+12)=38/50$

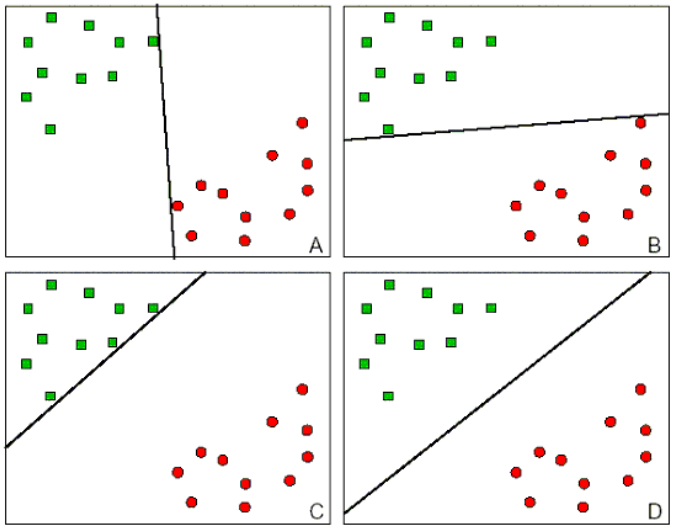
- $\text{Precision}_B = 43/(43+12)=43/55$
- $\text{Recall}_C = 44/(44+6)=44/50$

Machine Learning Tasks

- Supervised learning da esempi
  - Classification
    - Approcci dicriminativi
    - Approcci generativi
    - Problemi: Outlier and deviation detection
  - Regression
  - Dependency modeling
    - Riconoscimento di Associazioni/Relazioni, Rel. Implicazione
  - Sequence Classification
    - Language Learning
    - Temporal learning
    - Trend analysis and change/anomaly detection
- Unsupervised learning
  - Clustering
  - Embedding ottimo: Enconding/Decoding
    - Representation Learning for Images
    - PreTraining as optimal encoding

Metodi di ML: selezione dei modelli

- Approcci discriminativi
  - Lineari
  - *h(x) = sign(* ***W*** *·* ***x*** *+ b)*



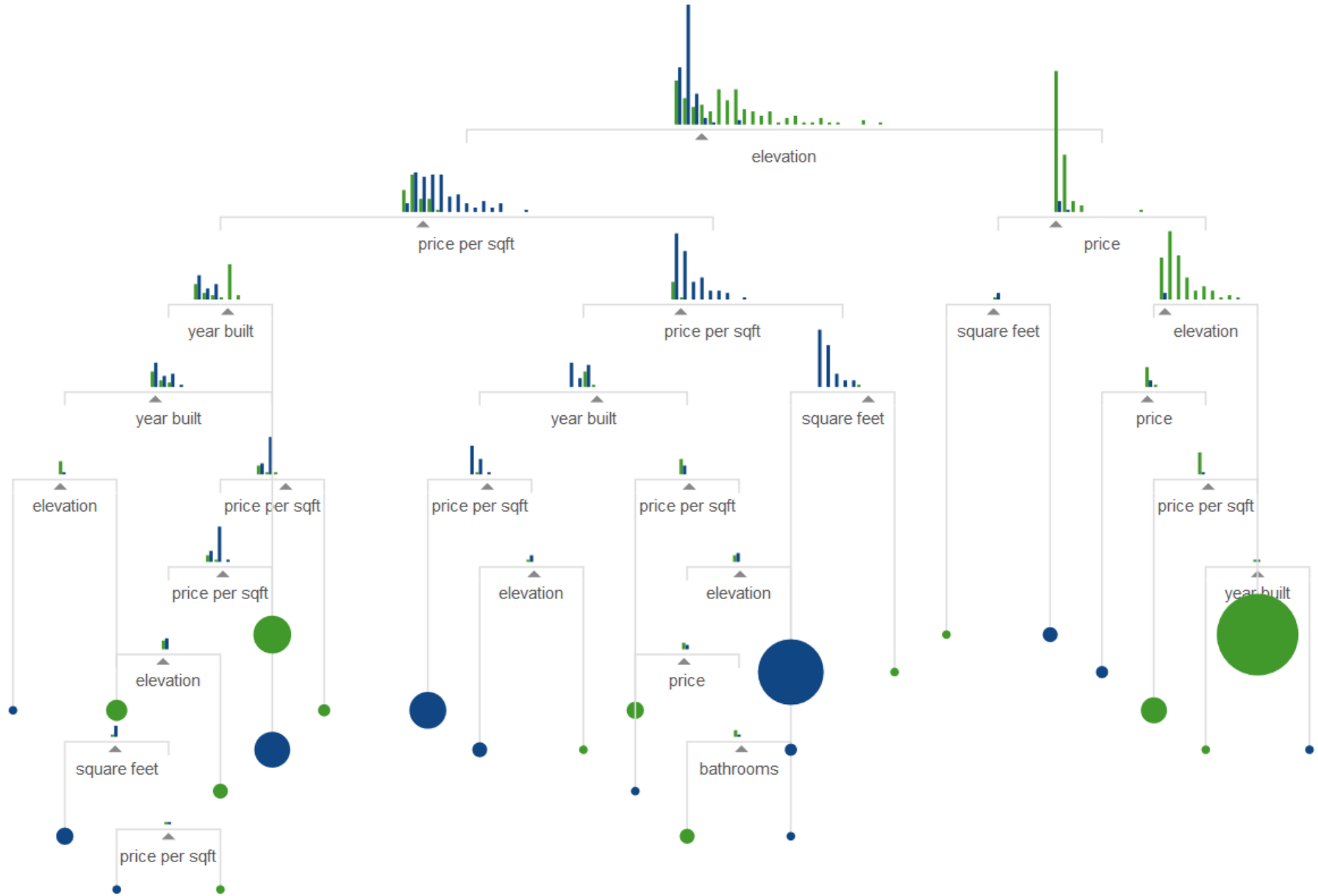
  - Approcci probabilistici
    - Stima delle probabilità $p(\mathcal{C}_k|\mathbf{x})$ attraverso un training set
    - Modello generativo ed uso della inversione Bayesiana

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})}.$$

ML: una introduzione visuale

- See URL: http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?imm_mid=0d76b4&cmp=em-data-na-na-newsltr_20150826

Es. apprendimento alberi di decisione



DTs: Extensions

- **Missing data**: In many domains, not all the attribute values will be known for every example. Two problems:
  - First, given a complete decision tree, how should one classify an example that is missing one of the test attributes?
  - Second, how should one modify the information-gain formula when some examples have unknown values for the attribute? (see Exercise 18.9)

- **Multivalued attributes**: When an attribute has many possible values, the information gain measure gives an inappropriate indication of the attribute's usefulness.
  - There can be one different value for every example, (i.e. each subset of examples is a singleton with a unique classification)
  - One possibility is to allow a Boolean test of the form $A=v_k$, that is, picking out just one of the possible values for an attribute, leaving the remaining values to possibly be tested later in the tree.

- **Continuous and integer-valued input attributes**: Continuous or integer-valued attributes such as Height and Weight , have an infinite set of possible values.
  - Rather than generate infinitely many branches, decision-tree learning algorithms typically find the SPLIT POINT split point that gives the highest information gain

- **Continuous-valued output attributes**: If we are trying to predict a numerical output REGRESSION TREE value, such as the price of an apartment, then we need a regression tree rather than a classification tree.
  - A regression tree has at each leaf a linear function of some subset of numerical attributes, rather than a single value.
  - For example, the branch for two bedroom apartments might end with a linear function of square footage, number of bathrooms, and average income for the neighborhood.
  - The learning algorithm must decide when to stop splitting and begin applying linear regression (see Section 18.6) over the attributes.

Apprendimento e Classi di Algoritmi

- Acquisizione di:
  - Funzioni logiche booleane, (ad es., alberi di decisione)
  - Induzione: determinazione ricorsiva delle CNES che caratterizzano i diversi sottogruppi .
- Approcci probabilistici:
  - Funzione target di Probabilità, (ad es., classificatore Bayesiano)
  - Induzione: Stima delle probabilità (in quanto parametri).
- Approcci geometrici
  - Funzioni di separazione in spazi vettoriali (lineari e non)
    - KNN
    - Funzioni Lineari, percettroni, Neural Networks, Support Vector Machines,…
    - Embeddings, analisi spettrale (trasformazioni di spazio)
  - Induzione: parametrizzare la funzione appartenente ad una certa classe (ad es. polinomi di grado *n*)

Riferimenti Bibliografici

- *AIMA,* Chapter 18, 18.1-18.4, 18.6-18.7, 18.11

- J. R. Quinlan, Induction of decision trees, Journal of Machine Learning volume 1, pages 81–106 (1986),
  - URL: https://link.springer.com/content/pdf/10.1007/BF00116251.pdf?pdf=inline%20link

- Esposito, F.; Malerba, D.; Semeraro, G. A comparative analysis of methods for pruning decision trees. IEEE Trans. Pattern Anal. Mach. Intell. 1997, 19, 476–491.
  - URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=589207