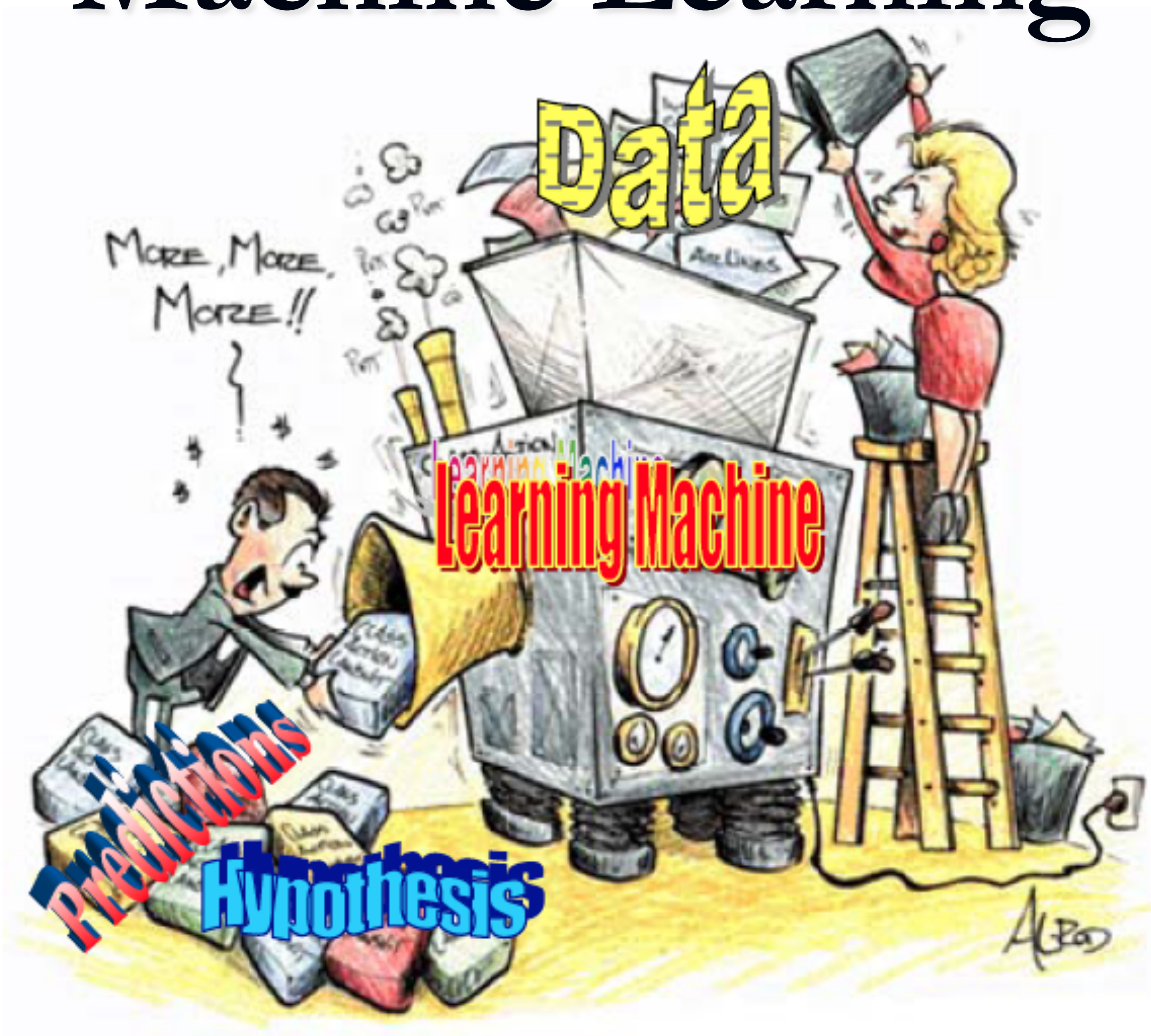
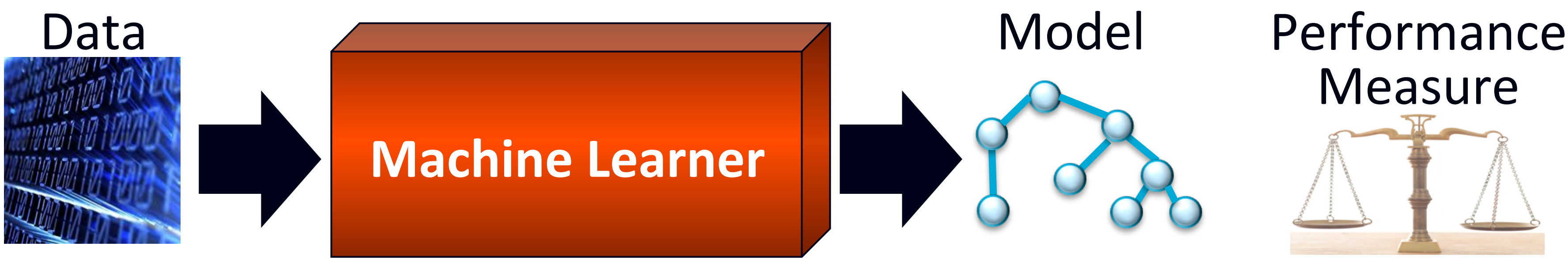
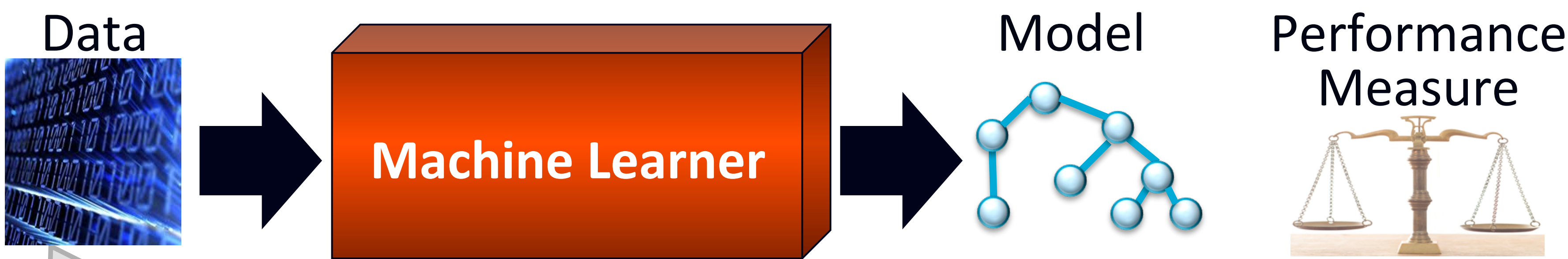
Machine Learning



Machine Learning in a Nutshell



Machine Learning in a Nutshell

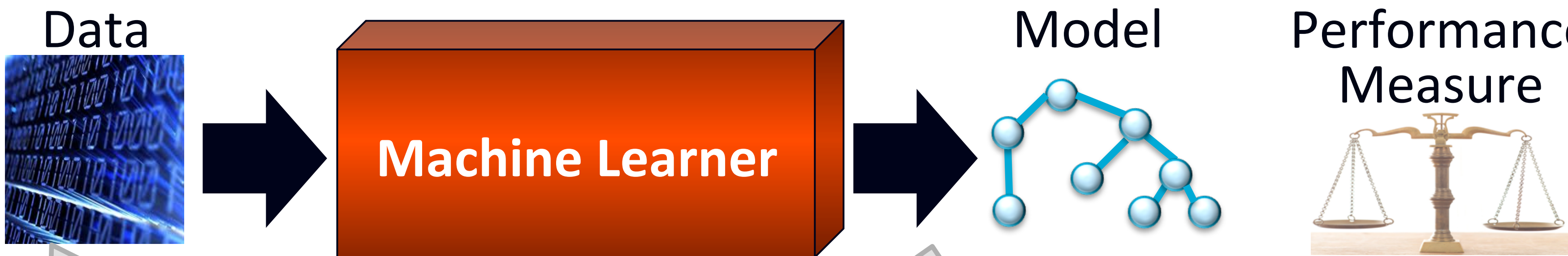


Data with attributes

ID	A1	Reflex	RefLow	RefHigh	Label
1	5.6	Normal	3.4	7	No
2	5.5	Normal	2.4	5.7	No
3	5.3	Normal	2.4	5.7	Yes
4	5.3	Elevated	2.4	5.7	No
5	6.3	Normal	3.4	7	No
6	3.3	Normal	2.4	5.7	Yes
7	5.1	Decreased	2.4	5.7	Yes
8	4.2	Normal	2.4	5.7	Yes
…	…	…	…	…	…

Instance $x_i \in \mathcal{X}$
with label $y_i \in \mathcal{Y}$

Machine Learning in a Nutshell



Data with attributes

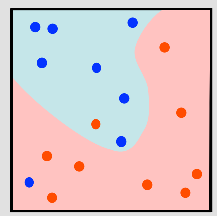
ID	A1	Reflex	RefLow	RefHigh	Label
1	5.6	Normal	3.4	7	No
2	5.5	Normal	2.4	5.7	No
3	5.3	Normal	2.4	5.7	Yes
4	5.3	Elevated	2.4	5.7	No
5	6.3	Normal	3.4	7	No
6	3.3	Normal	2.4	5.7	Yes
7	5.1	Decreased	2.4	5.7	Yes
8	4.2	Normal	2.4	5.7	Yes
…	…	…	…	…	…

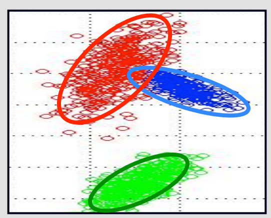
Instance $x_i \in \mathcal{X}$
with label $y_i \in \mathcal{Y}$

Model $f : \mathcal{X} \mapsto \mathcal{Y}$

Logistic regression

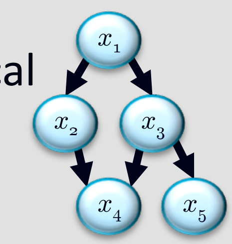
Support vector machines



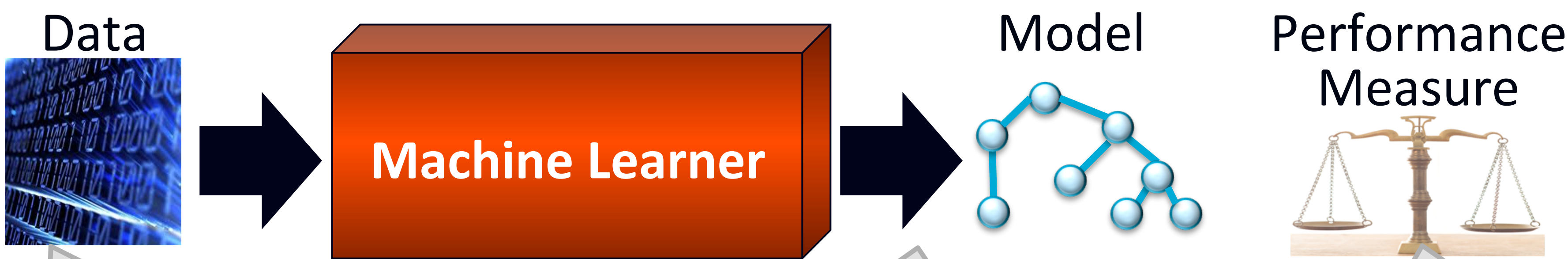


Mixture Models

Hierarchical Bayesian Networks



Machine Learning in a Nutshell



Data with attributes

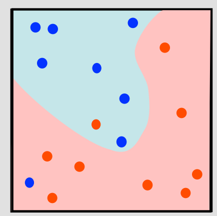
ID	A1	Reflex	RefLow	RefHigh	Label
1	5.6	Normal	3.4	7	No
2	5.5	Normal	2.4	5.7	No
3	5.3	Normal	2.4	5.7	Yes
4	5.3	Elevated	2.4	5.7	No
5	6.3	Normal	3.4	7	No
6	3.3	Normal	2.4	5.7	Yes
7	5.1	Decreased	2.4	5.7	Yes
8	4.2	Normal	2.4	5.7	Yes
…	…	…	…	…	…

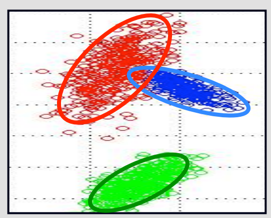
Instance $x_i \in \mathcal{X}$
with label $y_i \in \mathcal{Y}$

Model $f : \mathcal{X} \mapsto \mathcal{Y}$

Logistic regression

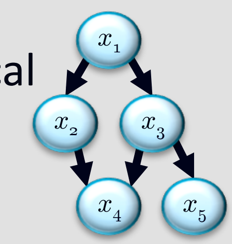
Support vector machines





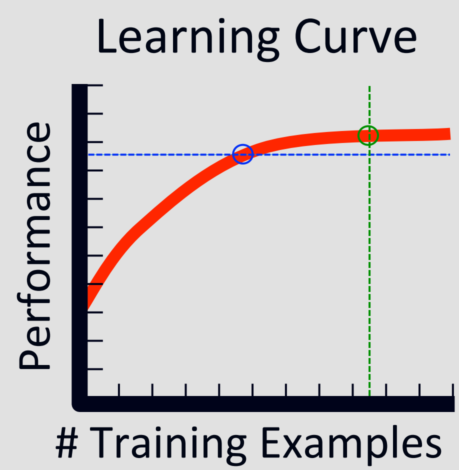
Mixture Models

Hierarchical Bayesian Networks



Evaluation

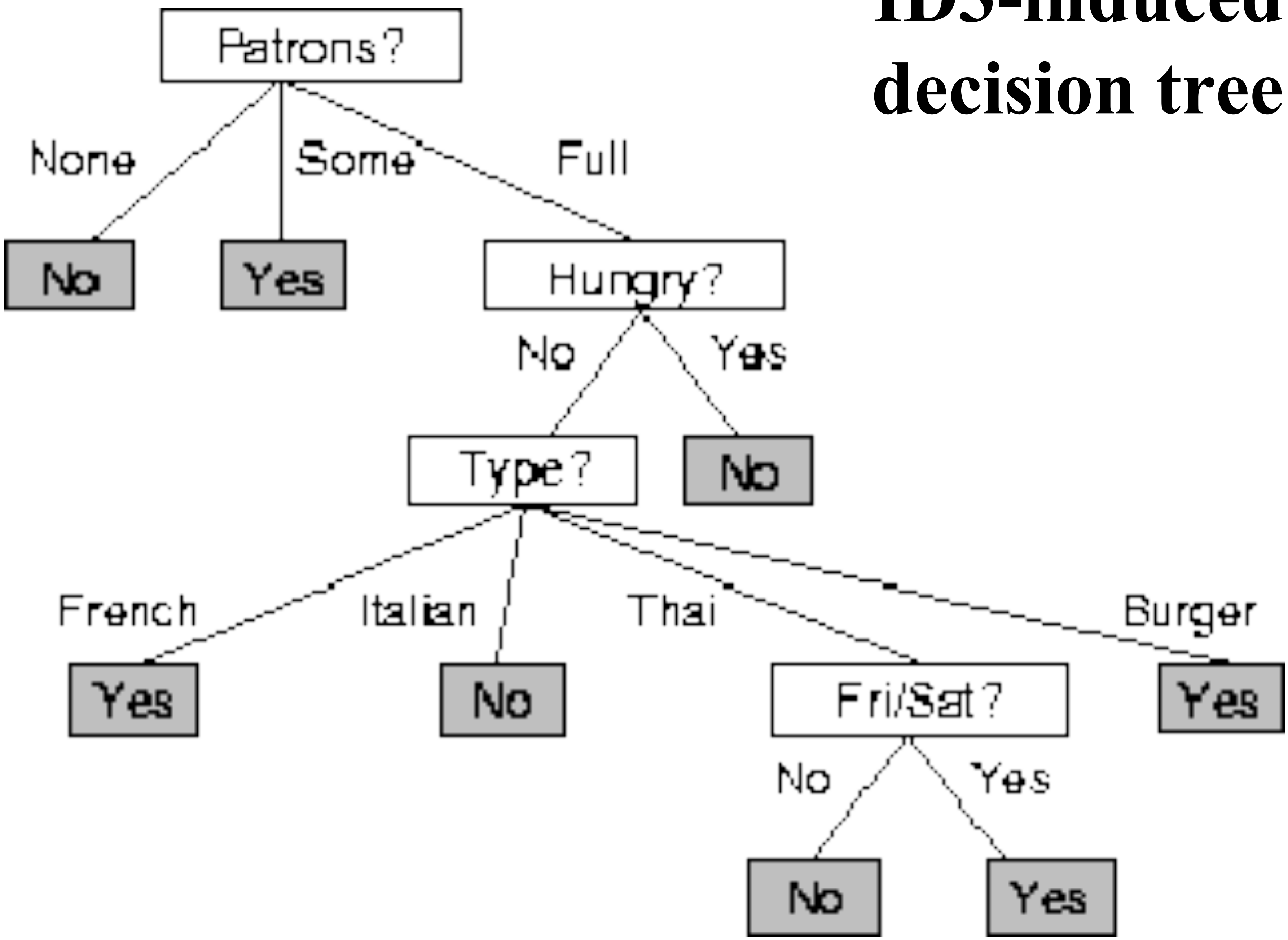
Measure predicted labels vs actual labels on test data



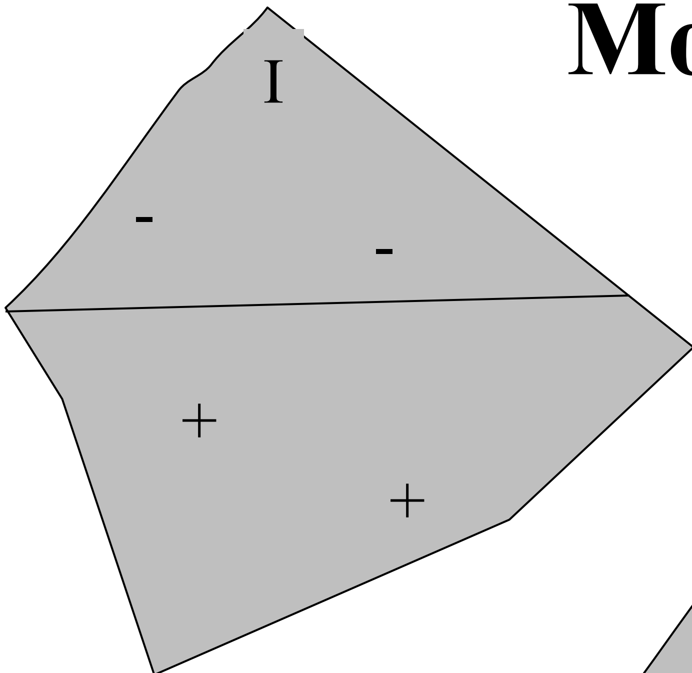
A training set

Example	Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0–10	Yes
X_2	Yes	No	No	Yes	Full	$	No	No	Thai	30–60	No
X_3	No	Yes	No	No	Some	$	No	No	Burger	0–10	Yes
X_4	Yes	No	Yes	Yes	Full	$	No	No	Thai	10–30	Yes
X_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
X_6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0–10	Yes
X_7	No	Yes	No	No	None	$	Yes	No	Burger	0–10	No
X_8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0–10	Yes
X_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
X_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10–30	No
X_{11}	No	No	No	No	None	$	No	No	Thai	0–10	No
X_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30–60	Yes

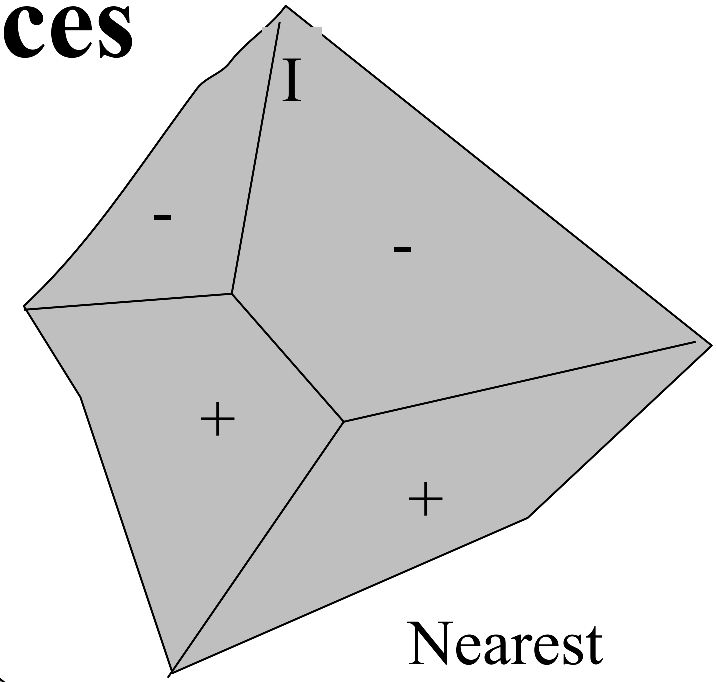
ID3-induced decision tree



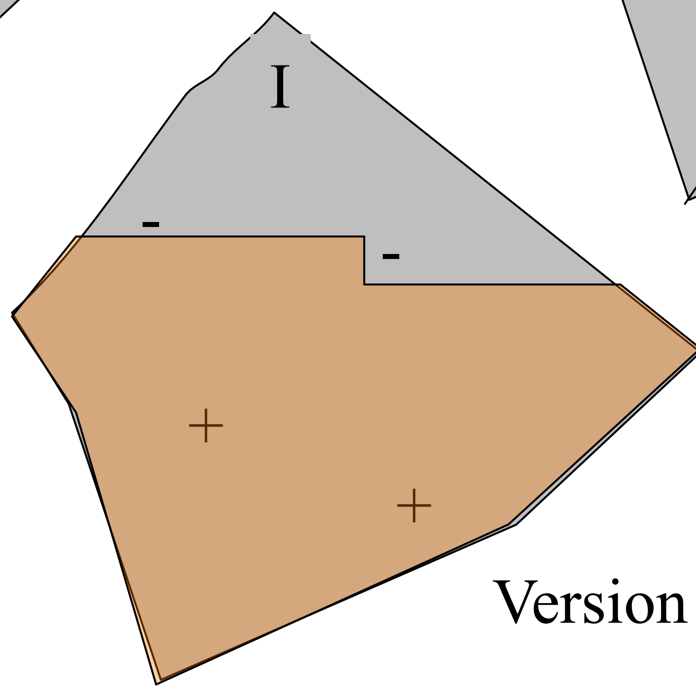
Model spaces



Decision tree

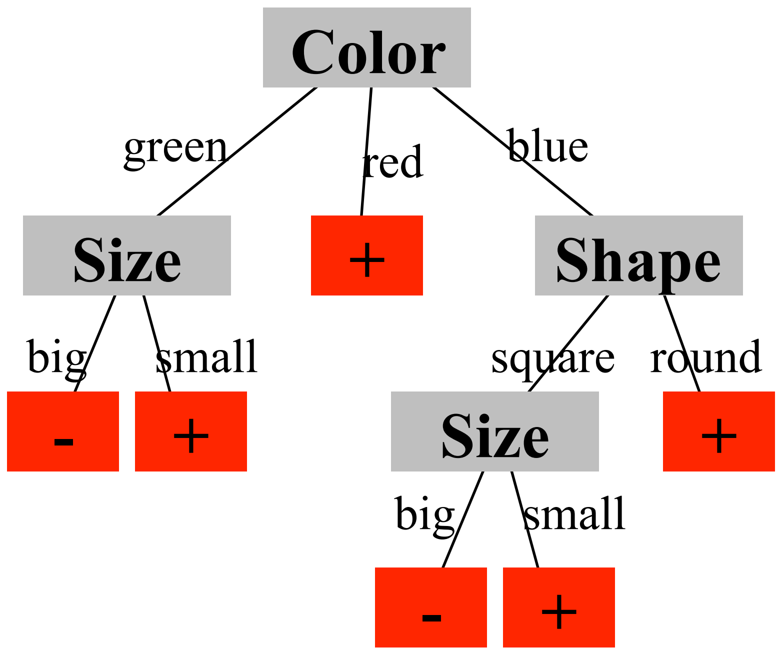


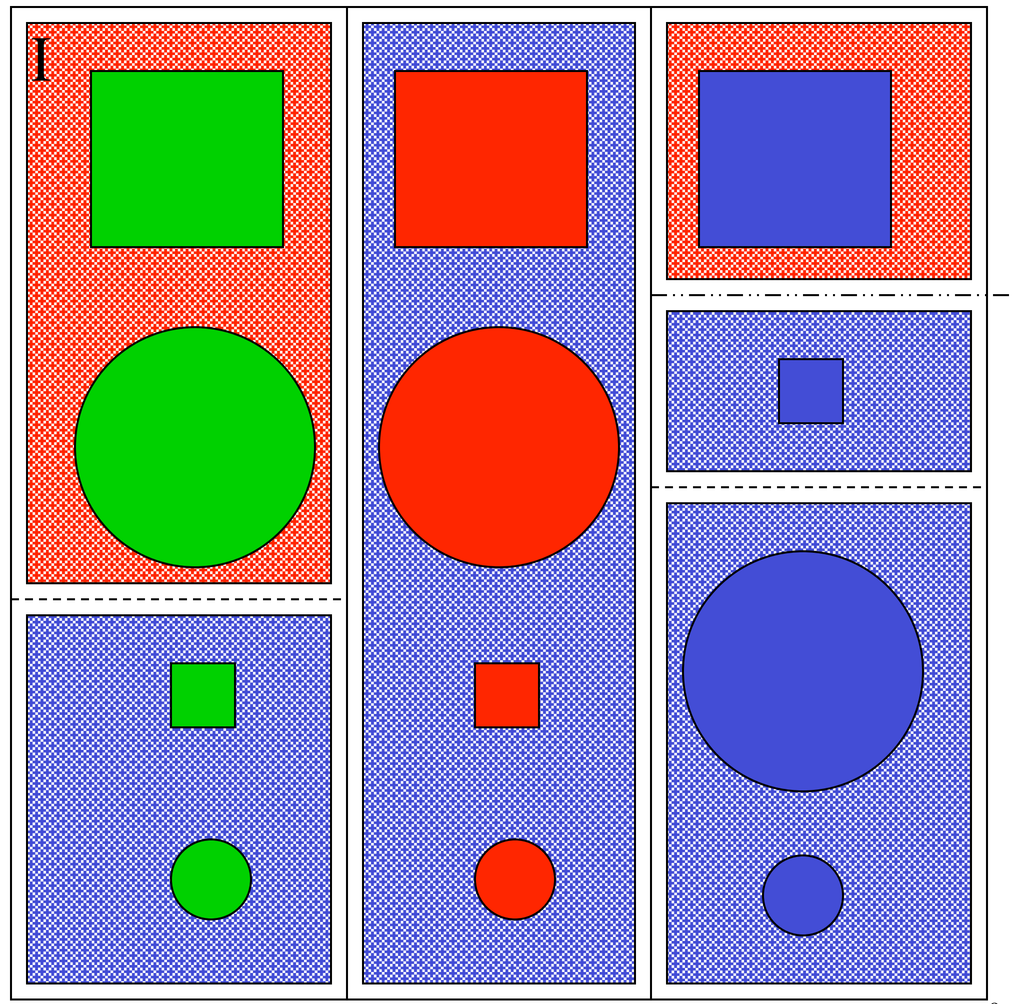
Nearest neighbor



Version space

Decision tree-induced partition – example





The Naïve Bayes Classifier

Some material adapted from slides by Tom Mitchell, CMU.

The Naïve Bayes Classifier

- Recall Bayes rule:

$$P(Y_i \mid X_j) = \frac{P(Y_i)P(X_j \mid Y_i)}{P(X_j)}$$

- Which is short for:

$$P(Y = y_i \mid X = x_j) = \frac{P(Y = y_i)P(X = x_j \mid Y = y_i)}{P(X = x_j)}$$

- We can re-write this as:

$$P(Y = y_i \mid X = x_j) = \frac{P(Y = y_i)P(X = x_j \mid Y = y_i)}{\sum_k P(X = x_j \mid Y = y_k)P(Y = y_k)}$$

Deriving Naïve Bayes

- Idea: use the training data to directly estimate:

$$P(X \mid Y) \quad \text{and} \quad P(Y)$$

- Then, we can use these values to estimate
  $P(Y \mid X_{new})$ using Bayes rule.

- Recall that representing the full joint probability
  $P(X_1, X_2, \ldots, X_n \mid Y)$ is not practical.

Deriving Naïve Bayes

- However, if we make the assumption that the attributes are independent, estimation is easy!

$$P(X_1, \ldots, X_n \mid Y) = \prod_i P(X_i \mid Y)$$

- In other words, we assume all attributes are conditionally independent given Y.
  - Often this assumption is violated in practice, but more on that later…

Deriving Naïve Bayes

- Let $X = \langle X_1, \ldots, X_n \rangle$ and label Y be discrete.
- Then, we can estimate $P(X_i \mid Y_i)$ and $P(Y_i)$ directly from the training data by counting!

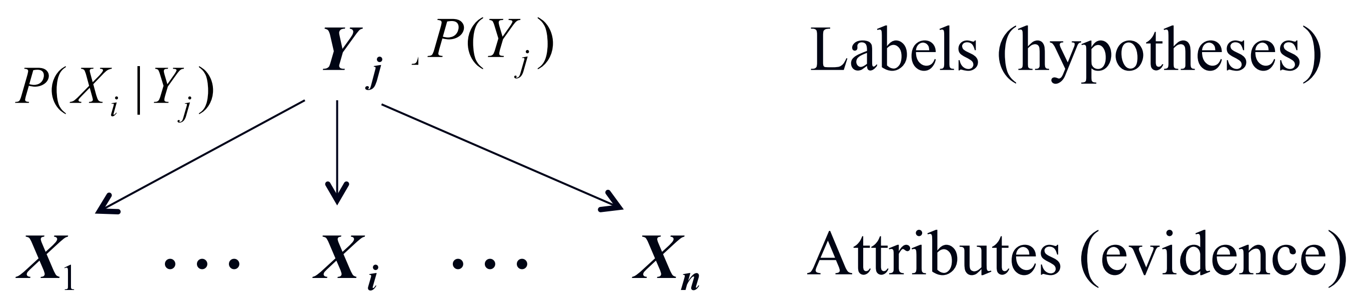
Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

The Naïve Bayes Classifier

- Now we have:

$$P(Y = y_j \mid X_1, \ldots, X_n) = \frac{P(Y = y_j)\prod_i P(X_i \mid Y = y_j)}{\sum_k P(Y = y_k)\prod_i P(X_i \mid Y = y_k)}$$

which is just a one-level Bayesian Network



- To classify a new point X_{new}:

$$Y_{new} \leftarrow \arg\max_{y_k} P(Y = y_k)\prod_i P(X_i \mid Y = y_k)$$

The Naïve Bayes Algorithm

- For each value y_k
  - Estimate $P(Y = y_k)$ from the data.
  - For each value x_{ij} of each attribute X_i
    - Estimate $P(X_i = x_{ij} \mid Y = y_k)$
- Classify a new point via:

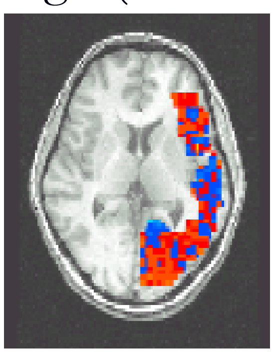
$$Y_{new} \leftarrow \arg\max_{y_k} P(Y = y_k) \prod_i P(X_i \mid Y = y_k)$$

- In practice, the independence assumption doesn't often hold true, but Naïve Bayes performs very well despite it.

Naïve Bayes Applications

- Text classification
  - Which e-mails are spam?
  - Which e-mails are meeting notices?
  - Which author wrote a document?
- Classifying mental states

Learning P(BrainActivity | WordCategory)



People Words

Animal Words

Pairwise Classification Accuracy: 85%