Introduction to Artificial Intelligence

Lecture: Basic Machine Learning

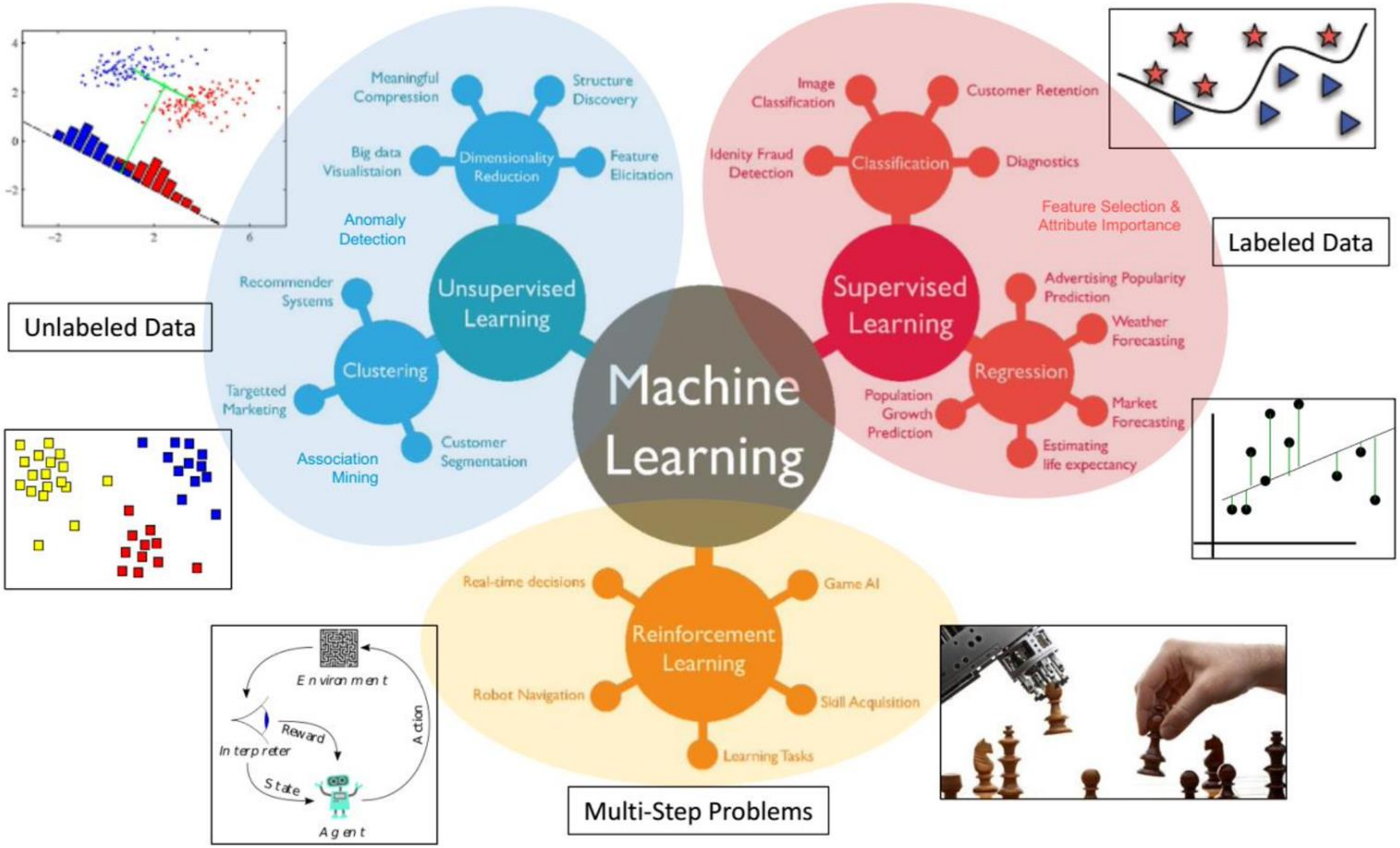
Outline

- Introduction to Machine Learning
- ID3 Decision Tree Learning
- Naïve Bayesian Learning

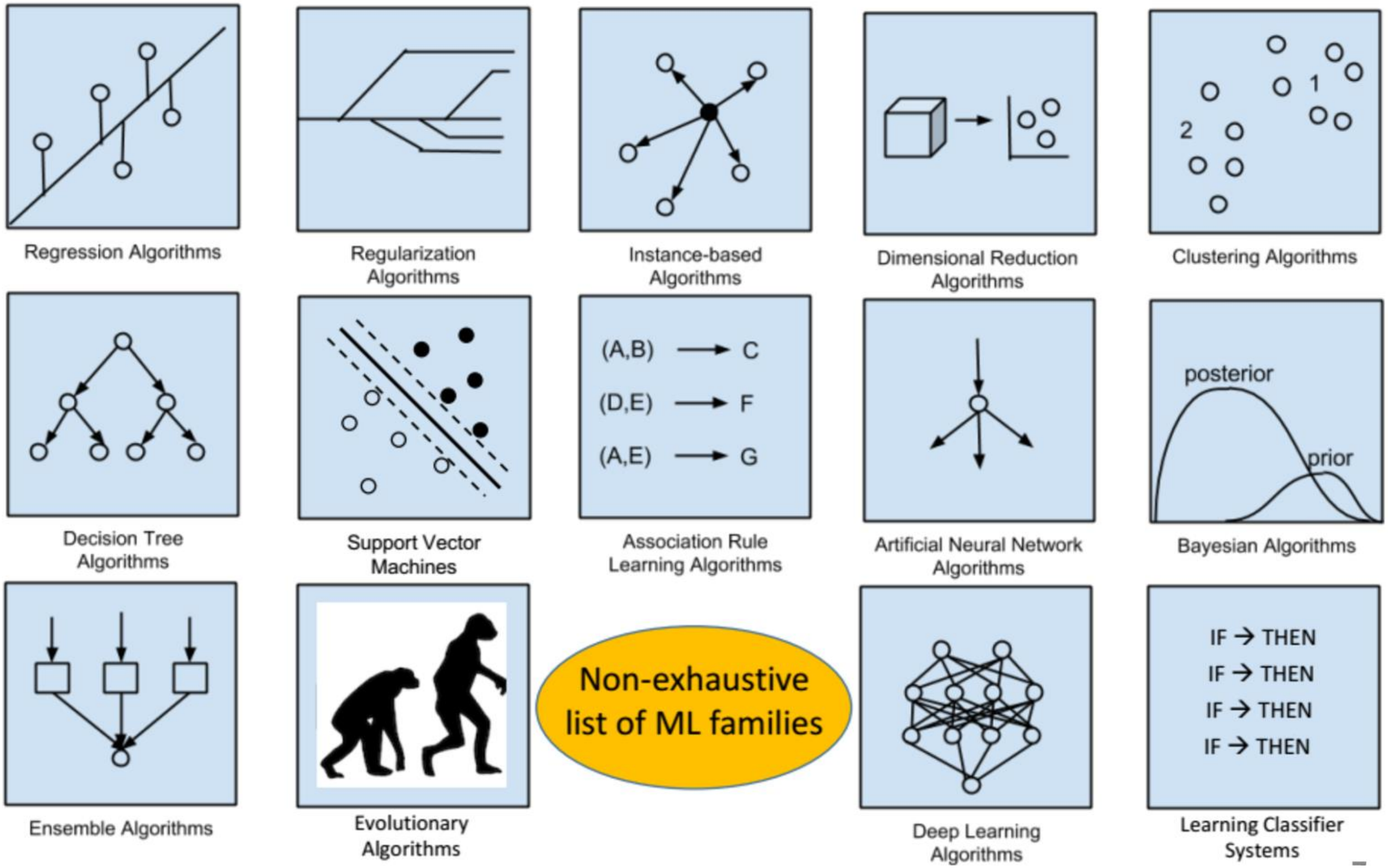
Machine Learning

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

Machine Learning



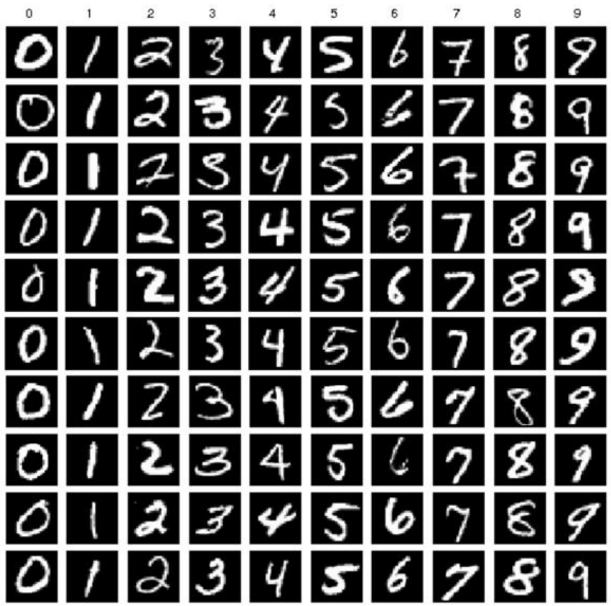
Machine Learning

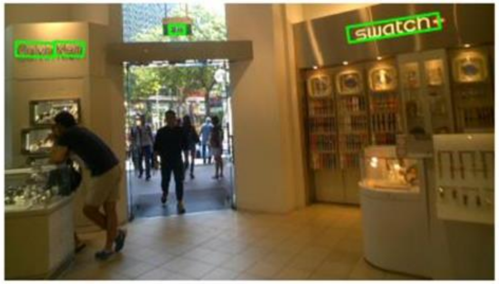


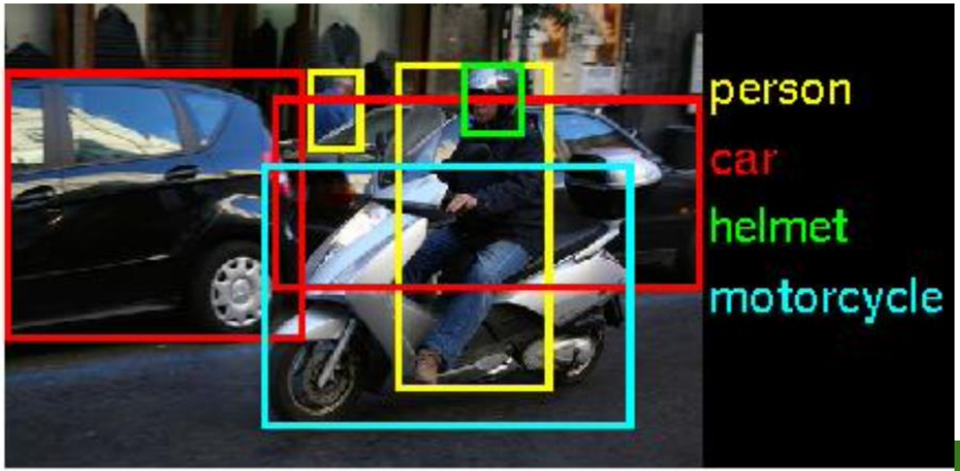
Supervised learning

- Learn a function that maps an input to an output based on example input-output pairs.
- Example,
  - Spam detection: Decide which emails are spam and which are important
  - Object detection and recognition: Localize and identify instances of semantic objects of a certain class (e.g., humans, buildings, or cars) in digital images and videos.

Supervised learning







Classification vs. Regression

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

Unsupervised learning

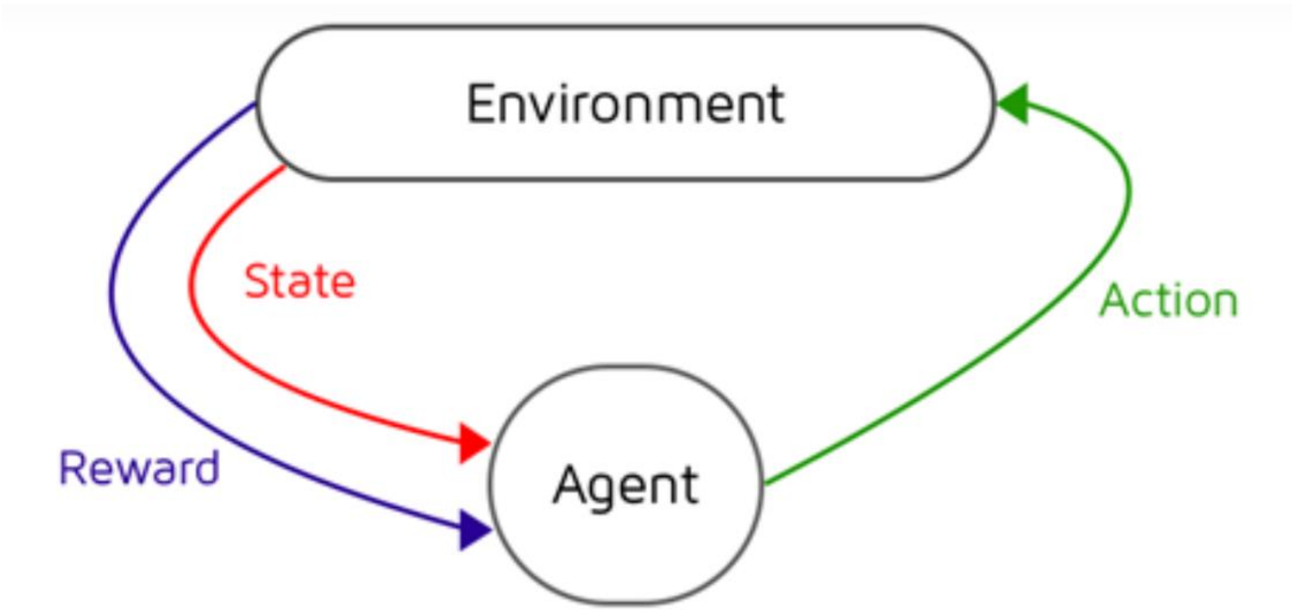
- Infer a function to describe hidden structure from "unlabeled" data.
- Example,
  - Social network analysis: cluster users of social networks by interest (community detection)

Semi-supervised learning

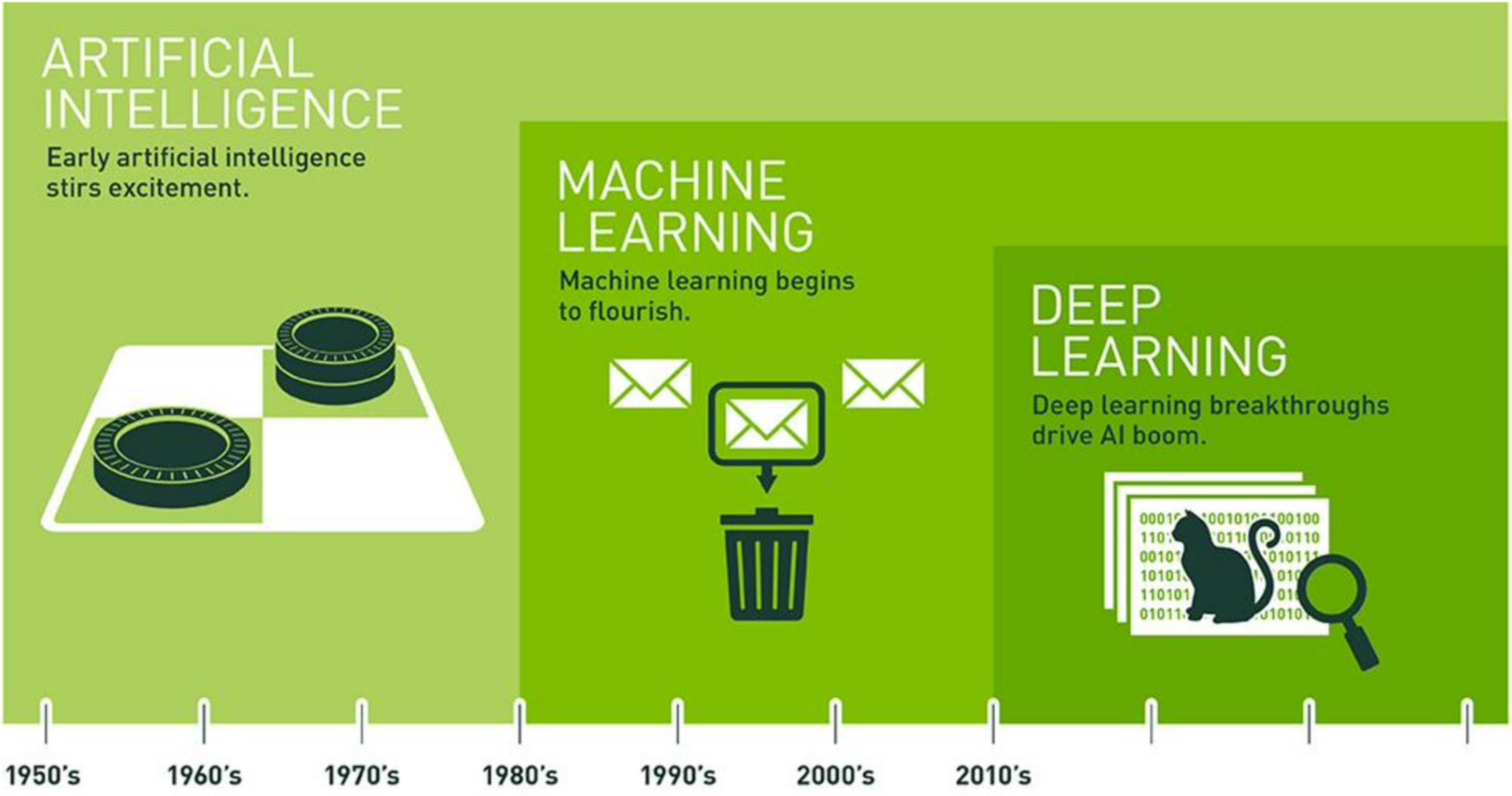
- The model is initially trained with a small amount of labeled data and a large amount of unlabeled data.

Reinforcement learning

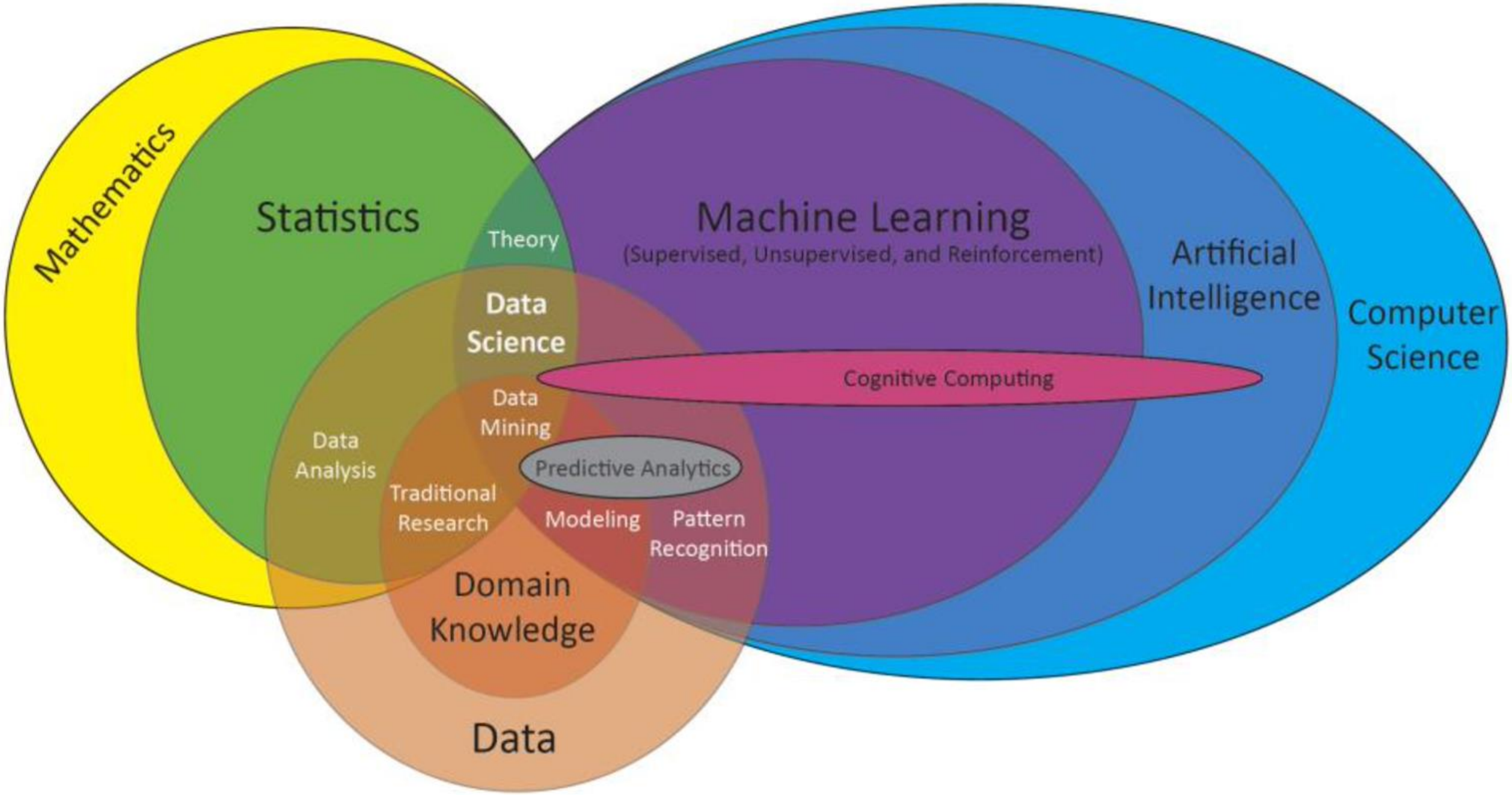
- The agent learns from the environment by interacting with it and receives rewards for performing actions.



Machine Learning



Machine Learning



ID3 Decision Tree Learning

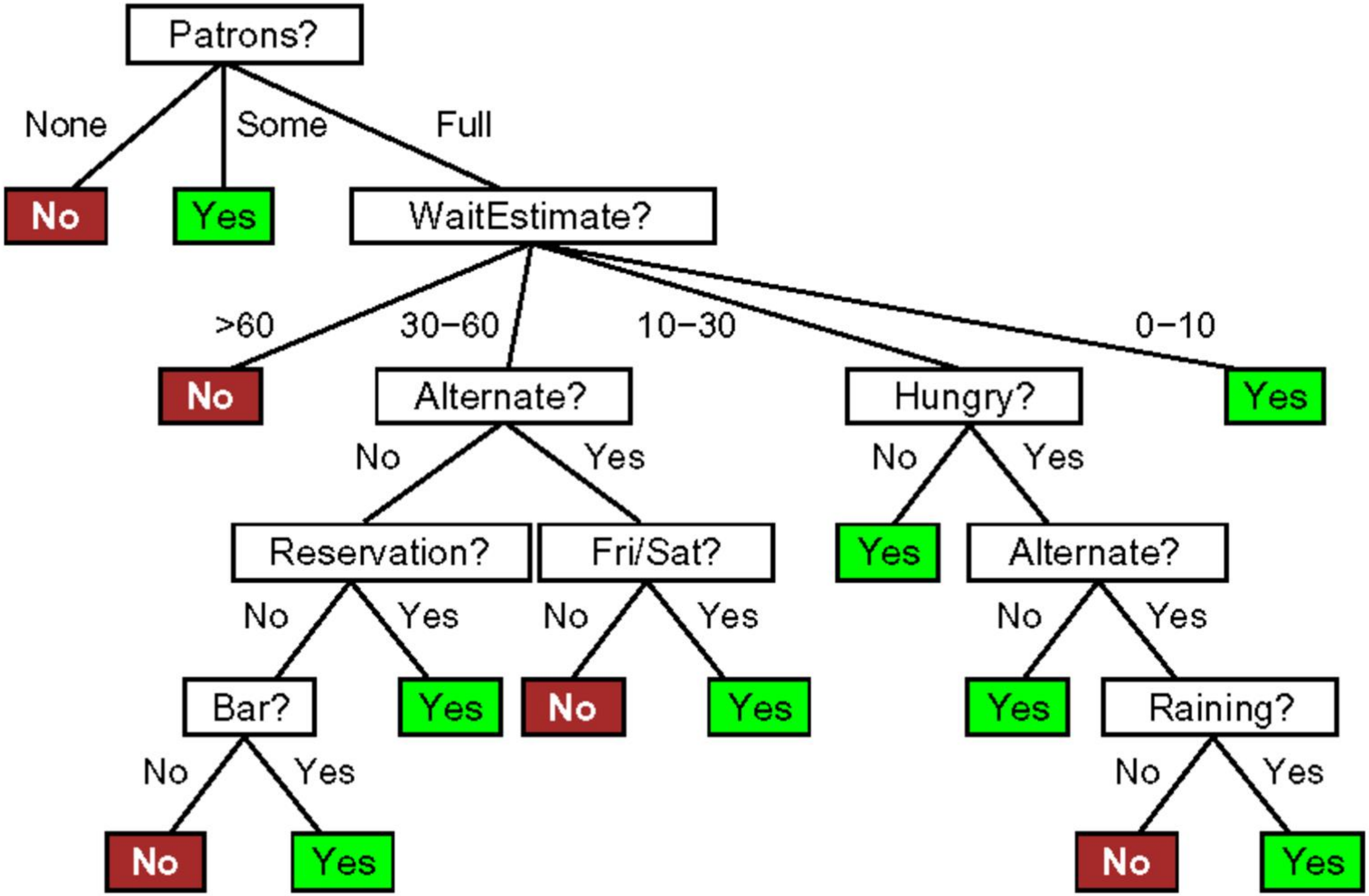
- The decision is based on the following attributes

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)

ID3 Decision Tree Learning

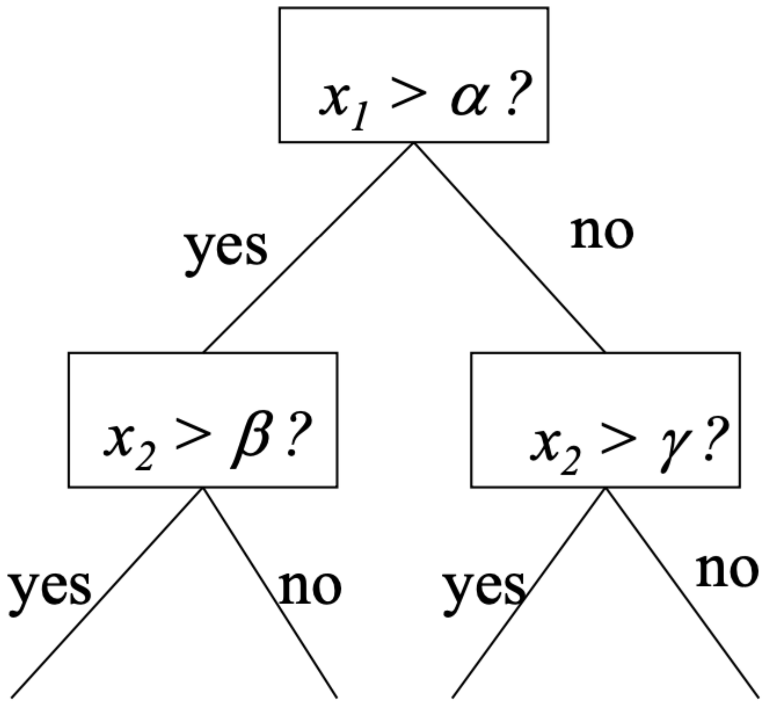
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	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

ID3 Decision Tree Learning



ID3 Decision Tree

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



ID3 Decision Tree

- The remaining examples are all positive (or all negative)

  → DONE, it is possible to answer Yes or No
- There are some positive and some negative examples

  → choose the “best” attribute to split them
- No examples left at a branch

  → return a default value
- No attributes left but both positive and negative examples

  → return the plurality classification of remaining ones.

ID3 Decision Tree

function DECISION-TREE-LEARNING ($examples$, $attributes$, $parent\ examples$) **returns** a tree

if $examples$ is empty

  then return PLURALITY-VALUE($parent\ examples$)

else if all $examples$ have the same classification

  then return the classification

else if $attributes$ is empty

  then return PLURALITY-VALUE($examples$)

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 $tree$

A purity measure with entropy

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

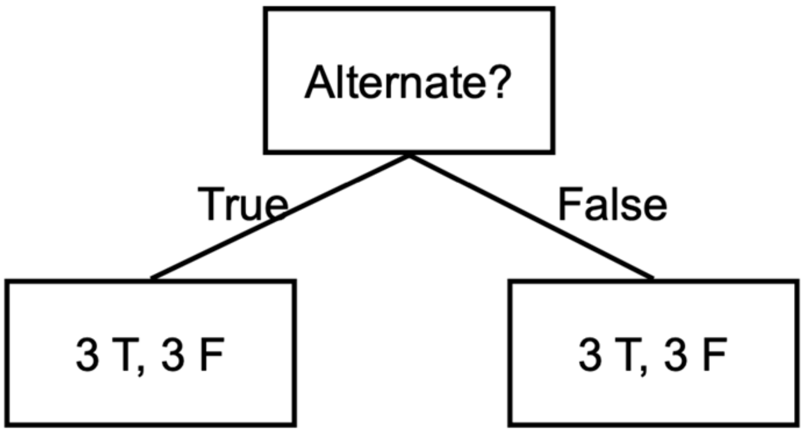
$$\boldsymbol{H(V) = \sum_k P(v_k)\, log_2 \frac{1}{P(v_k)} = -\sum_k P(v_k)\, log_2 P(v_k)}$$

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

Average Entropy



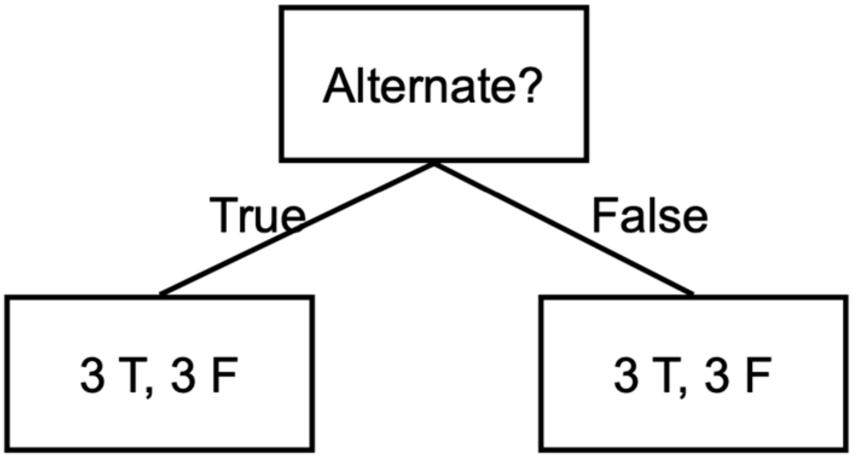
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	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

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

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

Information Gain



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.

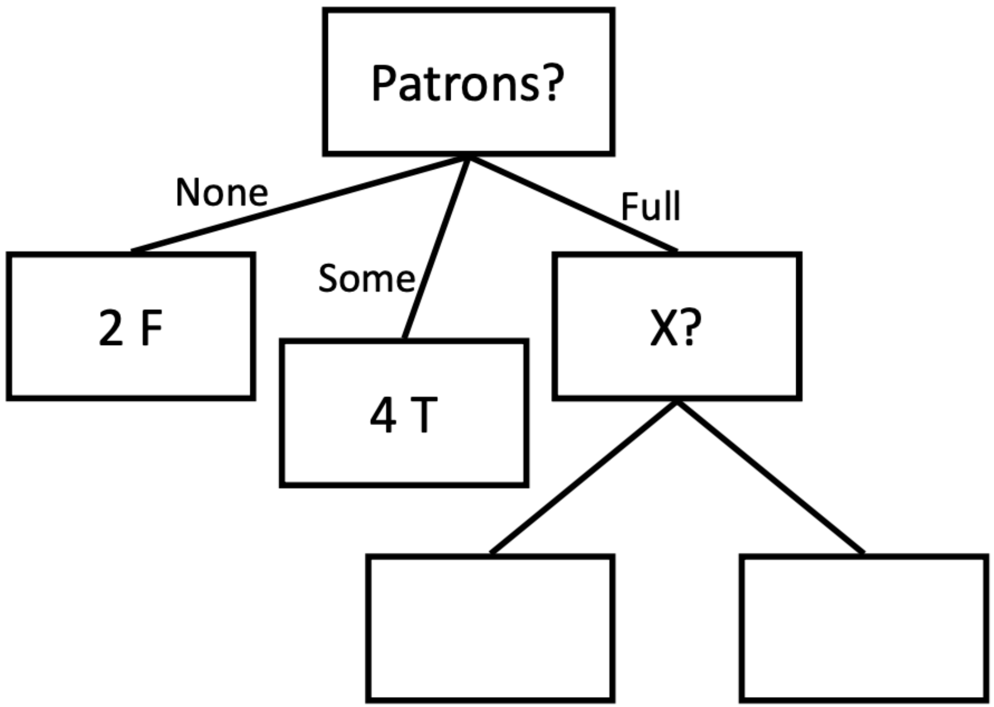
$$IG(Alternate, S) = H(S) - AE_{Alternate} = 1 - 1 = 0$$

Information Gain

- IG(Alternate, S) = 0
- IG(Bar, S) = 0
- IG(Sat/Fri?, S) = 0.021
- IG(Hungry, S) = 0.196
- IG(Raining, S) = 0
- IG(Reservation, S) = 0.021
- IG(Patron, S) = 0.541
- IG(Price, S) = 0.23
- IG(Type, S) = 0
- IG(Est. waiting time, S) = 0.208

Decision Tree Learning

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



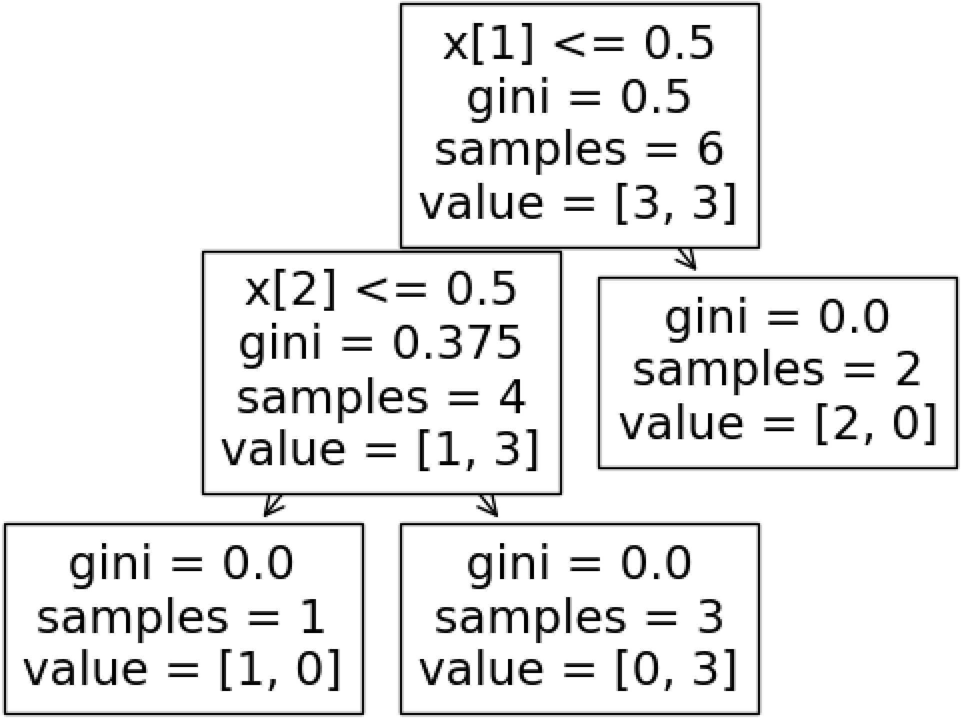
Homework

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

Homework

```
from sklearn.tree import DecisionTreeClassifier, plot_tree
import pandas as pd
df = pd.DataFrame(
    [{'W':1, 'U':0, 'S':0, 'L':0},
     {'W':1, 'U':1, 'S':2, 'L':0},
     {'W':0, 'U':1, 'S':1, 'L':0},
     {'W':0, 'U':0, 'S':1, 'L':1},
     {'W':1, 'U':0, 'S':2, 'L':1},
     {'W':0, 'U':0, 'S':2, 'L':1},
     ]
    )
clf = DecisionTreeClassifier()
clf.fit(df[['W', 'U', 'S']], df['L'])
plot_tree(clf)
```

Homework


x[1] <= 0.5
gini = 0.5
samples = 6
value = [3, 3]
x[2] <= 0.5
gini = 0.375
samples = 4
value = [1, 3]
gini = 0.0
samples = 2
value = [2, 0]
gini = 0.0
samples = 1
value = [1, 0]
gini = 0.0
samples = 3
value = [0, 3]

Bayesian classification

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

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

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

The 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:

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

- Classification is to determine $P(H \mid \mathbf{X})$, i.e. the probability that the hypothesis H holds given the observed data sample $\mathbf{X}$.

Example data set

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

The 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

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

- E.g., given that $\mathbf{X}$ is 31..40 and has a medium income, the probability that $\mathbf{X}$ will buy computer

The Bayes' Theorem

- Informally,

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

$$posteriori = likelihood * prior \,/\, evidence$$

- $\mathbf{X}$ 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.

Classification with Bayes' Theorem

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

Naïve Bayesian classifier

- Class-conditional independence: **<u>There are no dependence relationships among the attributes</u>**.
- 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: $P(x_k \mid C_i) = g\left(x_k, \mu_{C_i}, \sigma_{C_i}\right)$ with the Gaussian distribution $g(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

Naive Bayesian for the example dataset

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

Naive Bayesian for the example dataset

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

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

Therefore, X belongs to class ("buys_computer = yes")

Avoiding the zero-probability problem

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 problem

- 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.

Naive Bayesian for the example dataset

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

Naive Bayesian for the example dataset

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

$P(\mathbf{X}|\ C_i)$

$$P(\mathbf{X}\,|\,buys\_computer\ =\ \text{“}yes\text{”})\ = 5/12 * 5/12 * 7/11 * 7/11\ =\ 0.070$$

$$P(\mathbf{X}\,|\,buys\_computer\ =\ \text{“}no\text{”})\ = 1/8\ * 3/8\ * 2/7 * 3/7 =\ 0.006$$

$P(\mathbf{X}|\ C_i) * P(C_i)$

$$P(\mathbf{X}\,|\,buys\_computer\ =\ \text{“}yes\text{”}) * P(buys\_computer\ = \text{“}yes\text{”}) = 0.044$$

$$P(\mathbf{X}|\,buys\_computer\ =\ \text{“}no\text{”}) * P(buys\_computer\ =\ \text{“}no\text{”}) = 0.002$$

$P(C_i\ |\,\mathbf{X})$

$$P(buys\_computer\ =\ \text{“}yes\text{”}\,|\,\mathbf{X}) = 0.953$$

$$P(buys\_computer\ =\ \text{“}no\text{”}\,|\,\mathbf{X}) = 0.047$$

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	?

Homework

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

References

- Stuart Russell and Peter Norvig. 2009. Artificial Intelligence: A Modern Approach (3rd ed.). Prentice Hall Press, Upper Saddle River, NJ, USA.
- Lê Hoài Bắc, Tô Hoài Việt. 2014. Giáo trình Cơ sở Trí tuệ nhân tạo. Khoa Công nghệ Thông tin. Trường ĐH Khoa học Tự nhiên, ĐHQG-HCM.
- Nguyễn Ngọc Thảo, Nguyễn Hải Minh. 2020. Bài giảng Cơ sở Trí tuệ Nhân tạo. Khoa Công nghệ Thông tin. Trường ĐH Khoa học Tự nhiên, ĐHQG-HCM.