CSCU9YE - Artificial Intelligence

Lecture 8: Supervised Machine Learning

Prof. Gabriela Ochoa, University of Stirling

Sources

- Chapter 18 Learning from Examples from Artificial Intelligence: A Modern Approach,
- A Course in Machine Learning by Hal Daumé III (http://ciml.info/)
- The Hundred-Page Machine Learning Book by Andriy Burkov (http://themlbook.com/wiki/doku.php)
- Scikit-learn (http://scikit-learn.org/stable/documentation.html)
- Online courses: Udacity (https://www.udacity.com/)

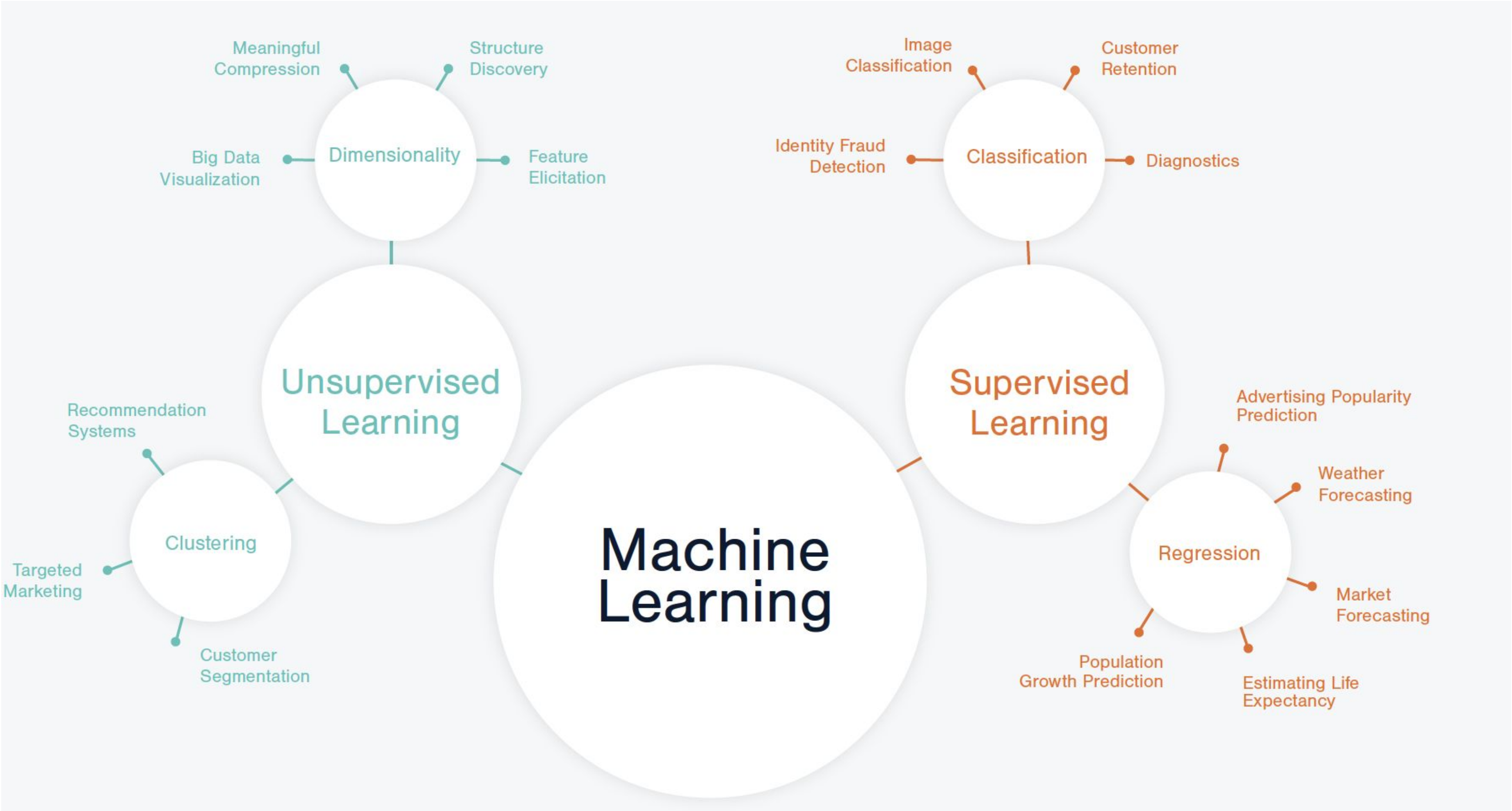
What is machine learning?

Machine learning is the science of getting computers to act without being explicitly programmed.

A system is learning if it improves its performance on future tasks after making observations about the world

Machine learning is the process of solving a practical problem by:

1. Gathering a dataset
2. Algorithmically building a statistical model based on that dataset



Supervised vs. Unsupervised Learning https://content.alegion.com/supervised-vs-unsupervised

Supervised learning

Data a list of observations <X,y>

Learn a function from examples

f is the target function - unknown!

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

Problem: find a hypothesis or estimate of *f*, let us call it *h*

such that $h \approx f$

given a training set of examples

$$
\begin{array}{cc}
X & y \\
\begin{array}{|c|}\hline
\langle x_{11}, x_{12}, \dots, x_{1p} \rangle \\
\langle x_{21}, x_{22}, \dots, x_{2p} \rangle \\
\langle x_{31}, x_{32}, \dots, x_{3p} \rangle \\
\dots \\
\dots \\
\dots \\
\langle x_{n1}, x_{n2}, \dots, x_{np} \rangle \\ \hline
\end{array}
&
\begin{array}{|c|}\hline
y_1 \\
y_2 \\
y_3 \\
\dots \\
\dots \\
\dots \\
y_n \\ \hline
\end{array}
\end{array}
$$

Classification example: weather forecast

Given input variables, predict a a category or class

Data:

Input variables

Target variable

Date	Temperature	Humidity	Wind Speed	Weather
01/10/17	22	48	2.7	Sunny
02/10/17	15	80	3.8	Rainy
03/10/17	12	45	17.9	Windy
04/10/17	14	77	4.2	Cloudy

What would be the wheatear in the future?
Weather in 05/10/17?

Two example of supervised algorithms

There are many supervised algorithms. We will study two examples, which are relatively simple yet very useful

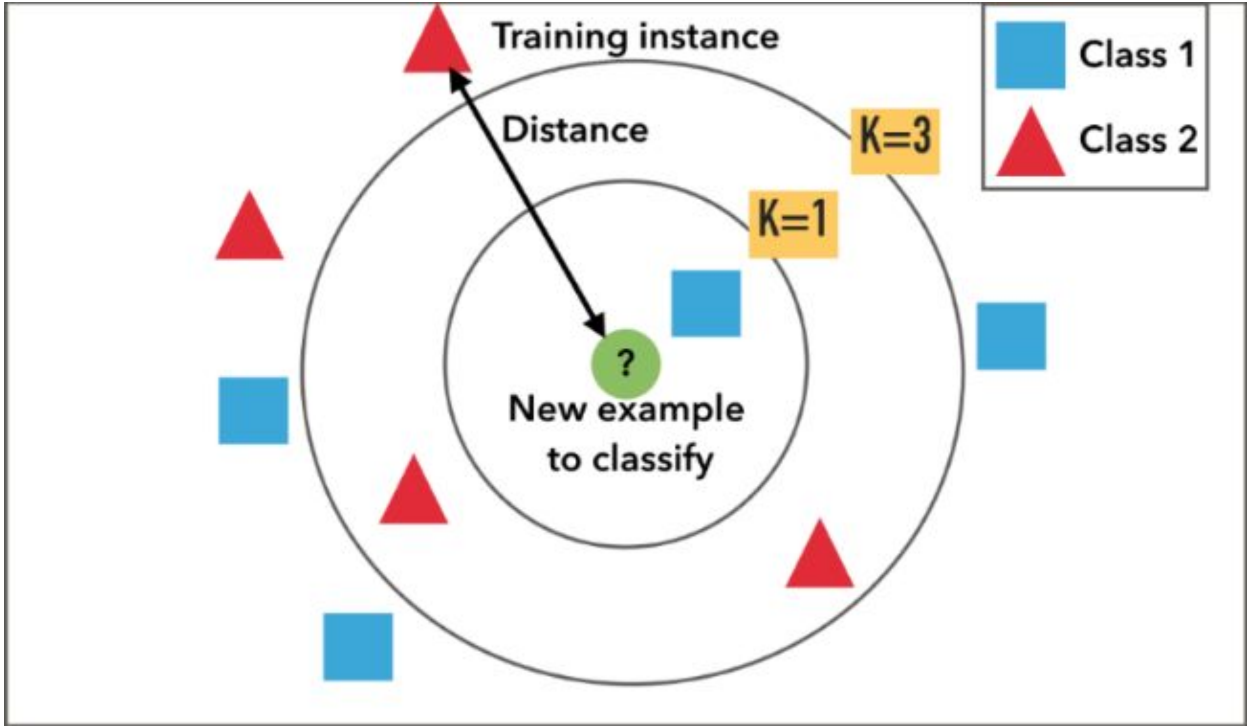
Can be used for both classification and regression

1. K Nearest Neighbour Algorithm
2. Decision Trees

K-Nearest Neighbour (KNN) algorithm

- KNN algorithm is based on feature similarity.
- Given a new point whose class we want to predict, we find a number of close *neighbours* to the new point in the training set.
- We predict the label (class) of the new point according to the class of the neighbours.
- How many neighbours? *K* is a parameter of the algorithms
- In order to find the neighbours, we need a notion of distance
- Commonly used distance: Euclidean distance

K-Nearest Neighbour (KNN) algorithm



K integer number (small)

An object is classified by a plurality vote of its neighbours

Assigned to the class most common among its K nearest neighbours

What is the class of the test sample (Green dot)?

Retrieved from medium.com

Euclidean distance in n-dimensional space

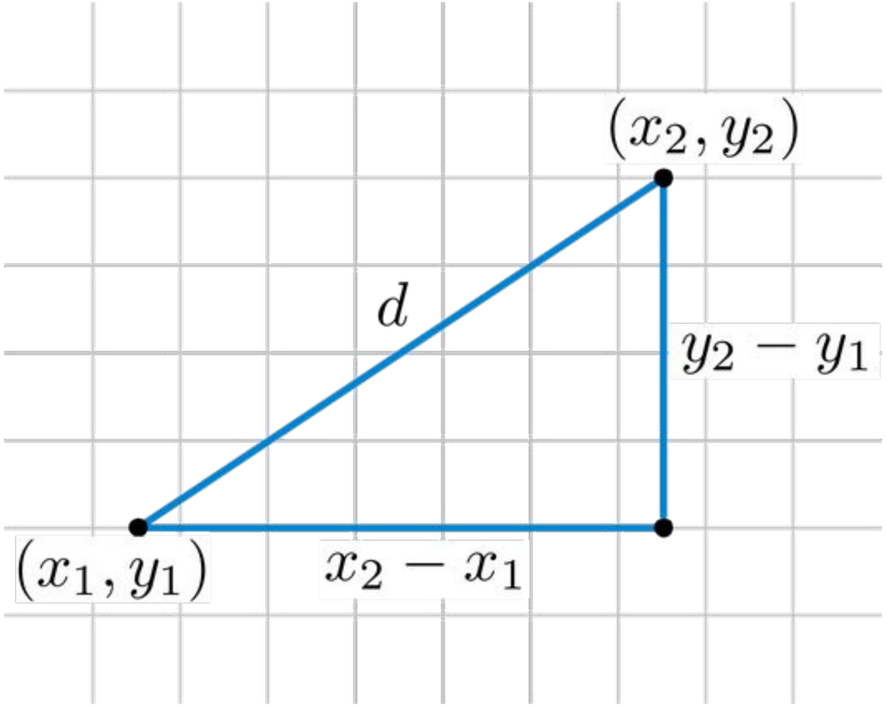
A measure of the ‘straight’ line between two points

Pythagorean Theorem

$$dx,y = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$ 2D

$$d_{\mathbf{x},\mathbf{y}} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}$$ 3D

$$d_{\mathbf{x},\mathbf{y}} = \sqrt{\sum_{j=1}^{J} (x_j - y_j)^2}$$ n-dimensions



Summary of KNN algorithm

1. A positive integer K is specified, along with a new sample
2. We select the K entries in our dataset which are closest to the new sample, according to a given distance metric
3. We find the most common classification of these entries
4. This is the classification we give to the new sample

Characteristics of KNN algorithm

KNN is a type of *instance-based* learning or *non-generalizing* learning: it does not attempt to construct a general internal model, but simply stores instances of the training data.

PROS

- Simple and versatile: used for both classification and regression
- Non-parametric: useful for non-linear data

CONS

- High memory usage — because the algorithm stores all of the training data
- Prediction time can be slow if dataset is large

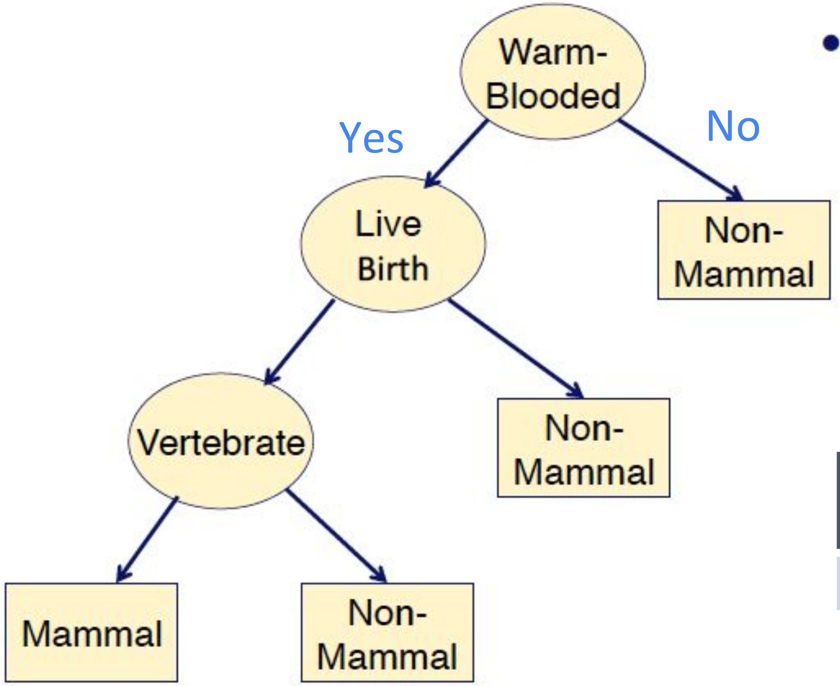
Decision trees

Decisions are reached by performing a number of tests (questions) on the attributes

Let us consider problems where

- Inputs have discrete values
- Output has two possible values. Boolean classification
    - True (positive example)
    - False (negative example)

Example: Decision Tree for classifing animals

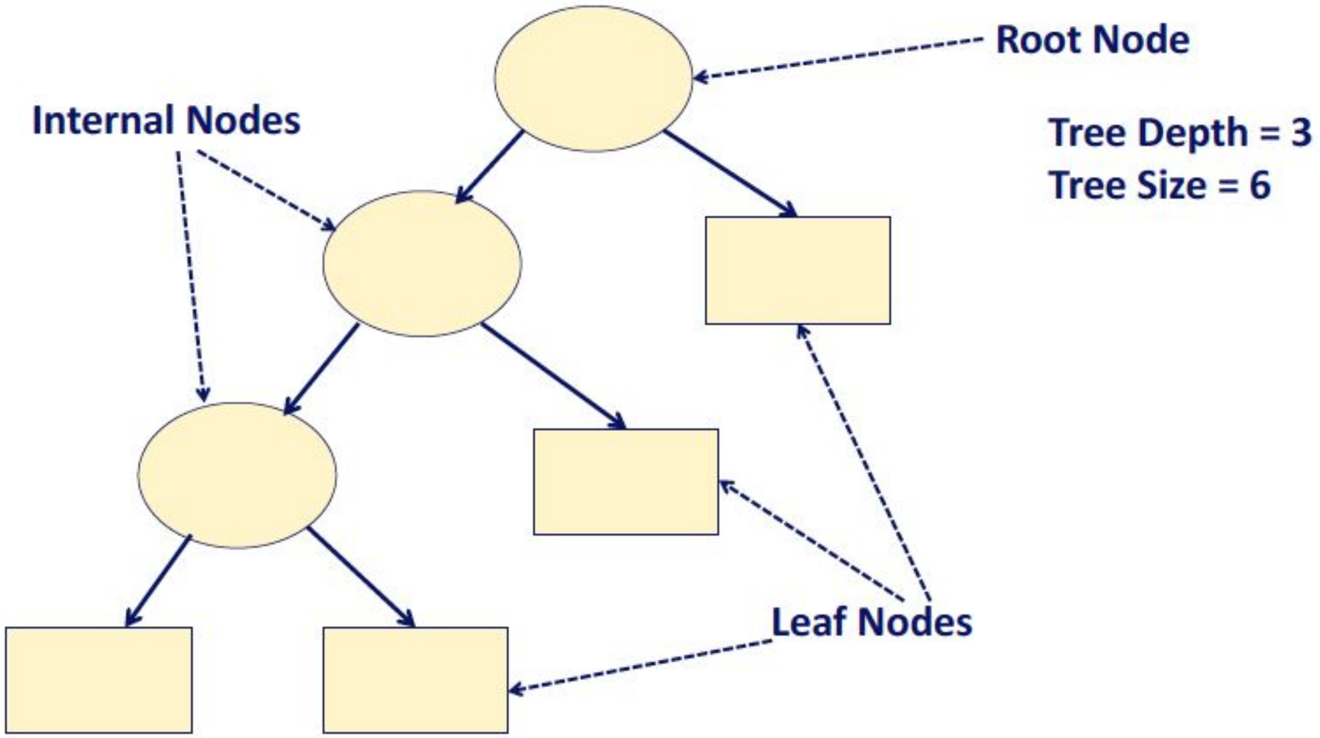


- Tree captures multiple classification decision paths

Warm Blooded	Live Birth	Vertebrate	Target
yes	yes	yes	?

Gabriela Ochoa, goc@stir.ac.uk

Decision Trees



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)

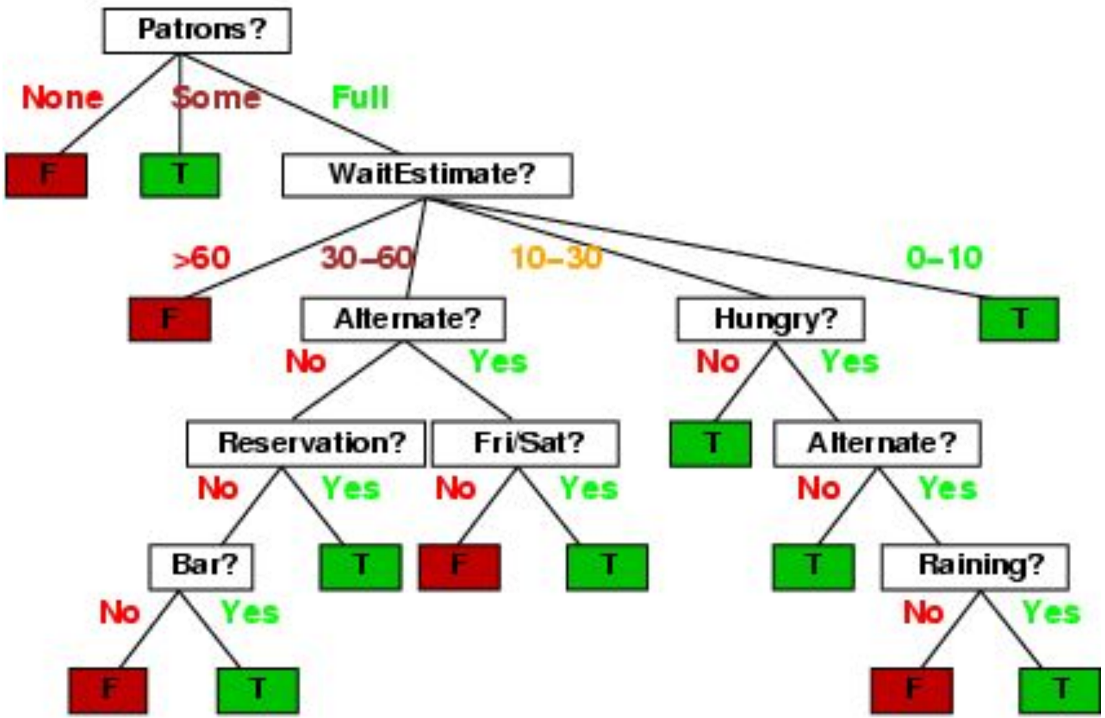
Dataset: attribute-based representations

- Examples described by attribute values (Boolean, discrete values or ranges)
- Dataset: 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

Decision trees: example

Manually constructed (book authors) tree for deciding whether to wait:



Internal nodes: test of the value of an attribute

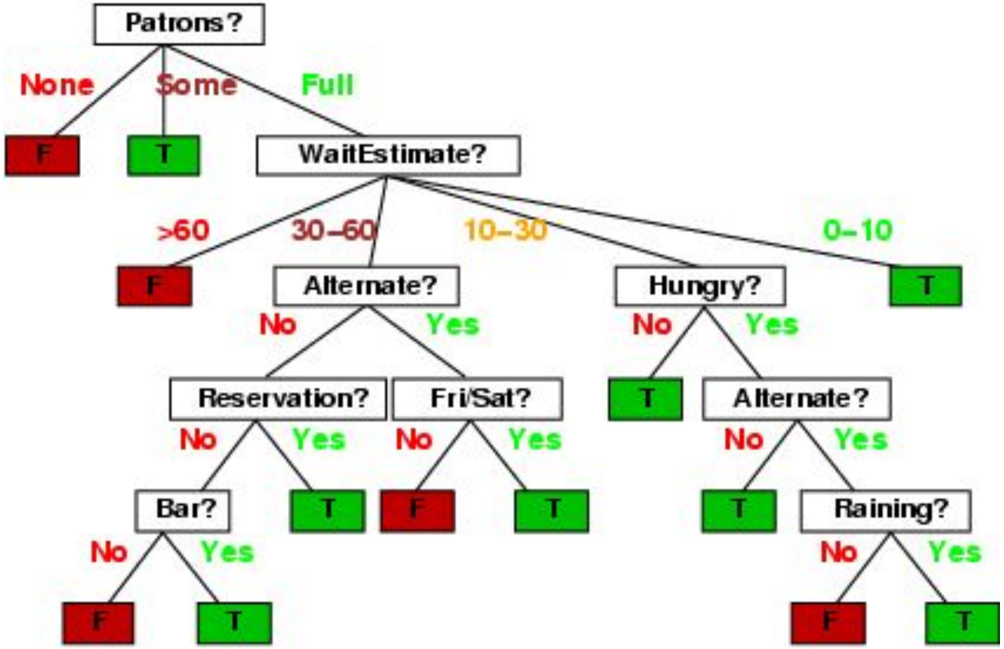
Branches: labelled with the attribute value

Leaves: value to be returned by the function

- T: Wait
- F: Not to wait

Decision trees: Example

Calculate decision for "unseen" example



Patrons: Full

Wait Estimate: 30-60

Alternate: yes

Bar: yes

Fri/Sat: Saturday

Hungry: yes

Price: ££

Raining: yes

Reservation: no

Type: Burger

Decision tree learning

- Aim: find a small tree consistent with the training examples
- Greedy divide-and-conquer strategy: always test the most *important* attribute first.
- This test divides the problem up into smaller subproblems that the can be solved recursively
- "Most important attribute": the one that makes the most *difference* to the classification of an example.
- We hope to get the classification with a small number of tests, meaning that all paths in the tree will be short and the tree as a whole will be shallow

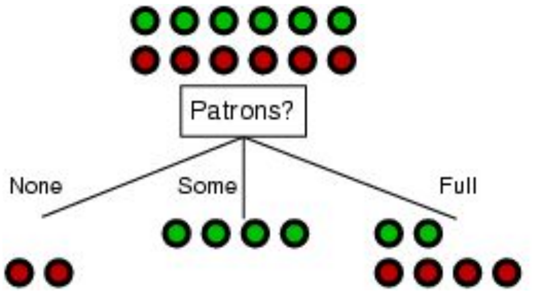
Dataset: attribute-based representations

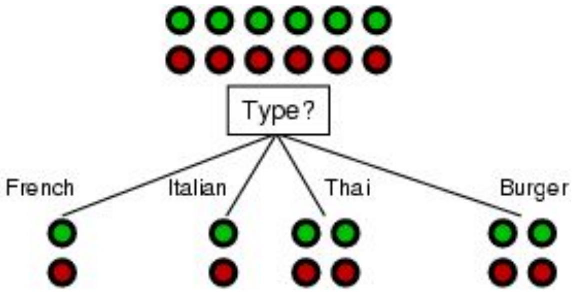
- Examples described by attribute values (Boolean, discrete values or ranges)
- Dataset: situations where I will/won't wait for a table:

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

Choosing an attribute

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

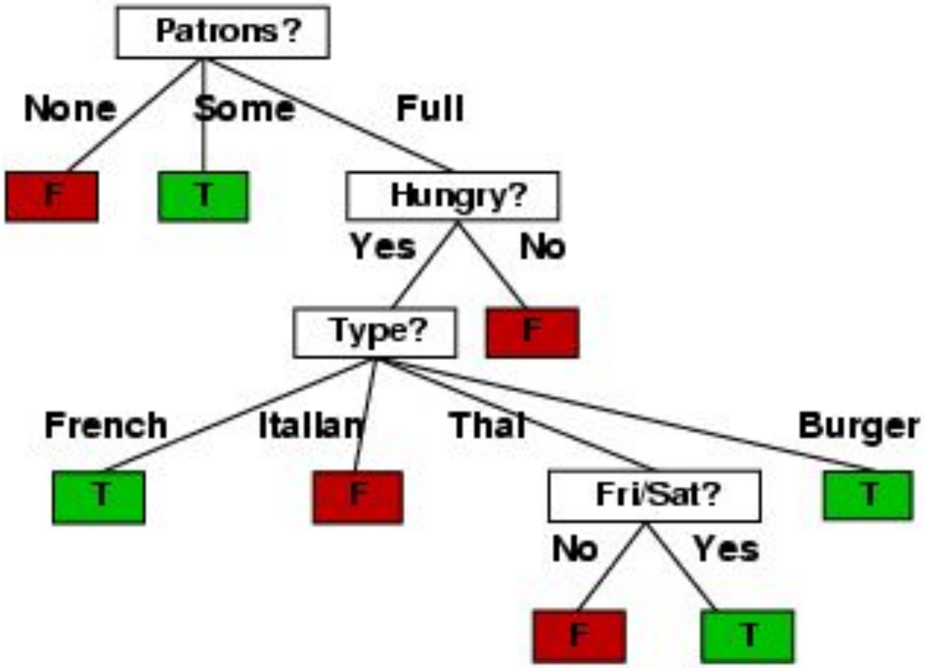




- Which attribute is better? *Patrons?* is a better choice, because it produces a purest classification (all positive or all negative)
- The *importance or goodness* of an attribute is measured using information Theory (information gain)
- The idea is to complete a classification with the smaller number of tests
- The tree as a whole is shallow

Example contd.

Decision tree learned from the dataset



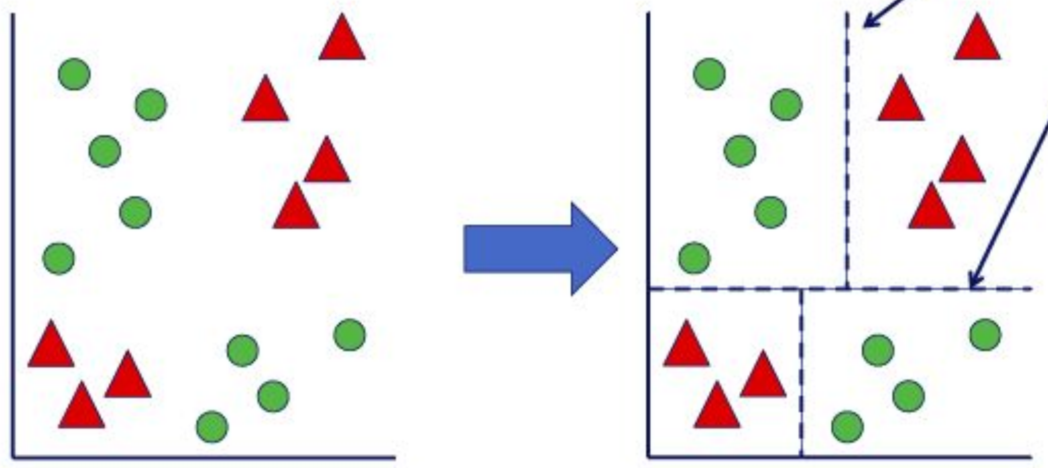
Substantially simpler than manually constructed tree

More complex tree is not justified by small amount of data

Some attributes are not used

Decision Tree: example with numerical values

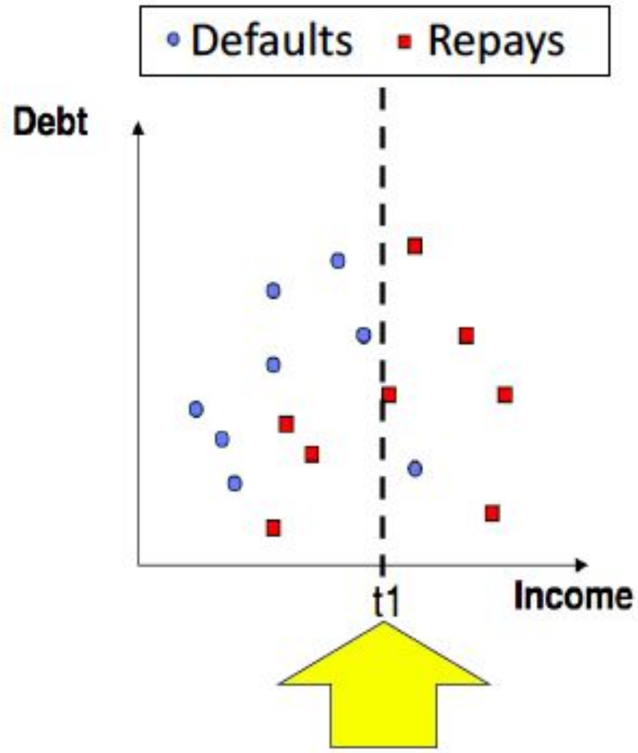
- Idea: Split data into "pure" regions

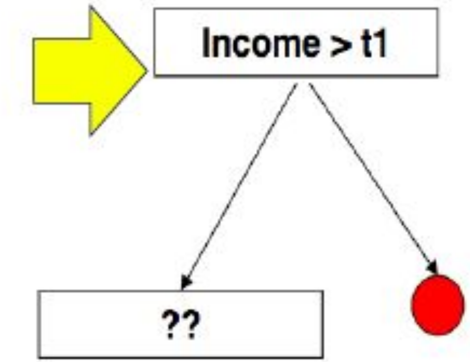


Let us consider two variables X and Y and two categories: Green dots and Red Triangles

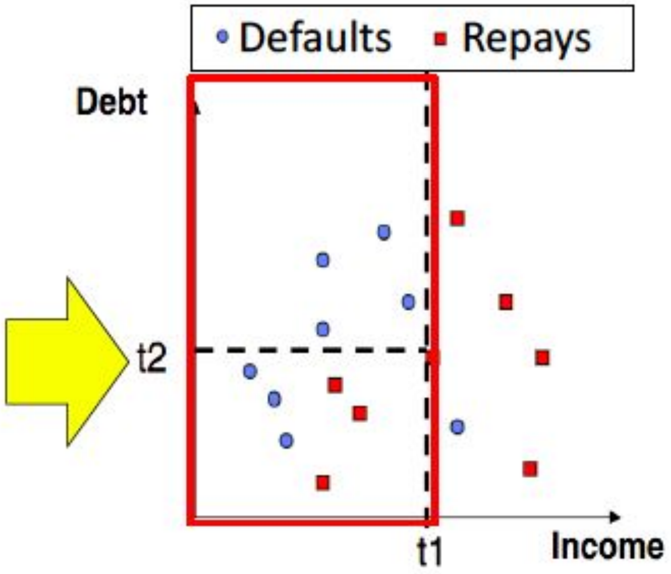
We can create decision boundaries withlines

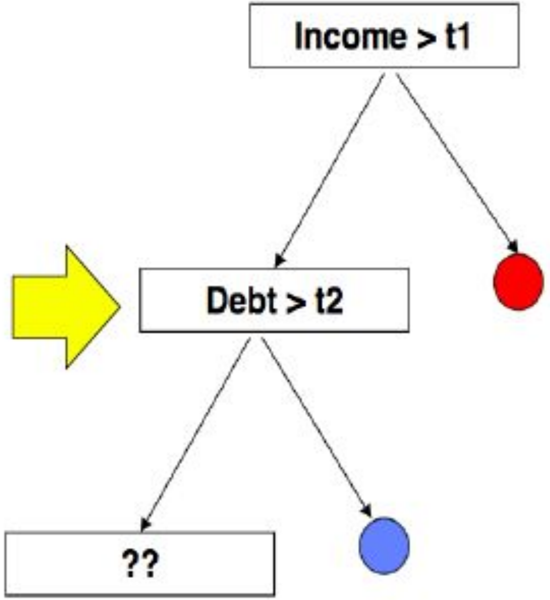
Example loan payment: split 1



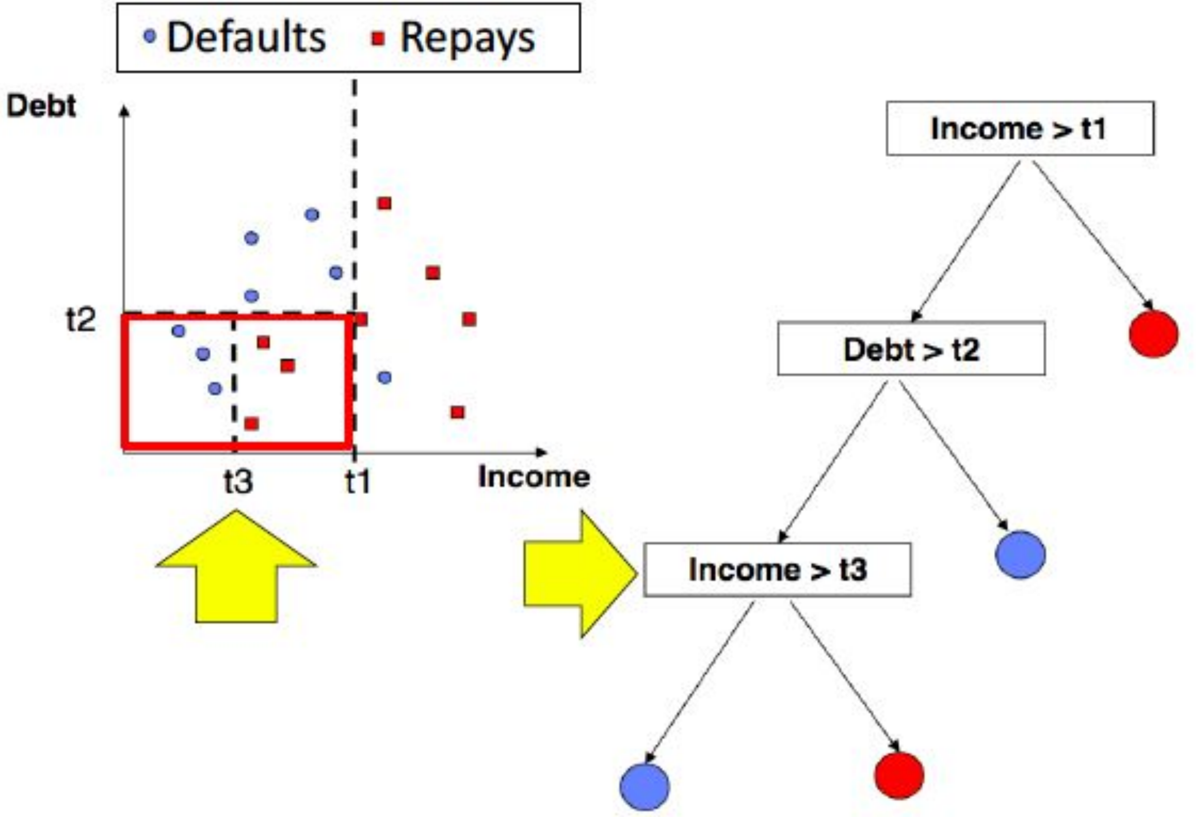


Example loan payment: split 2





Example loan payment: split 3



Summary

- Machine learning is the science of getting computers to act without being explicitly programmed.
- ML builds a *model* that can be used for predicting the future or making decisions
- Supervised ML Tasks: Classification, Regression
- Today we covered Classification tasks using a Supervised methods: KNN, Decision Trees
- scikit-learn: A powerful ML library for Python

Labs using Python library: scikit-learn

- ML Lab 1: Classification with KNN and Decision Trees
- ML Lab 2: Clustering with K-means and hierarchical clustering
- Famous example dataset: IRIS flowers (https://gist.github.com/netj/8836201)
- The data set consists of 50 samples from each of three species of Iris (*setosa*, *virginica* and *versicolor*).
- Four features from each sample: the length and the width of the sepals and petals, in centimeters.

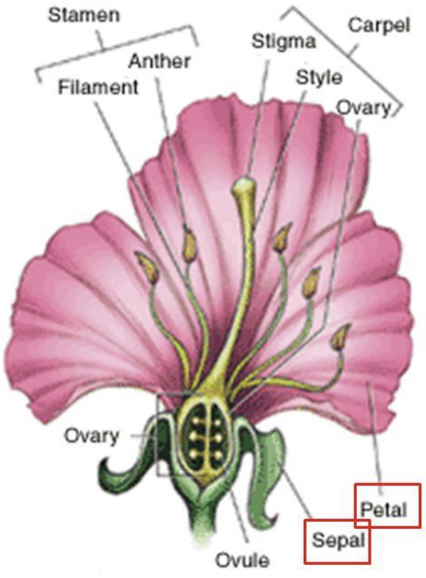
http://scikit-learn.org/stable/documentation.html

Iris flower dataset

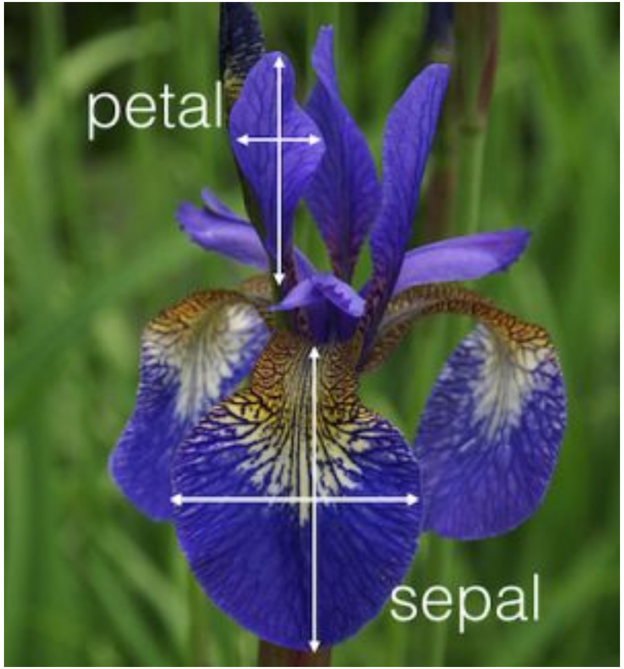
Petals and Sepals

- **<u>Sepals</u>** – <u>outermost circle of flower parts that encloses a bud before it opens</u>

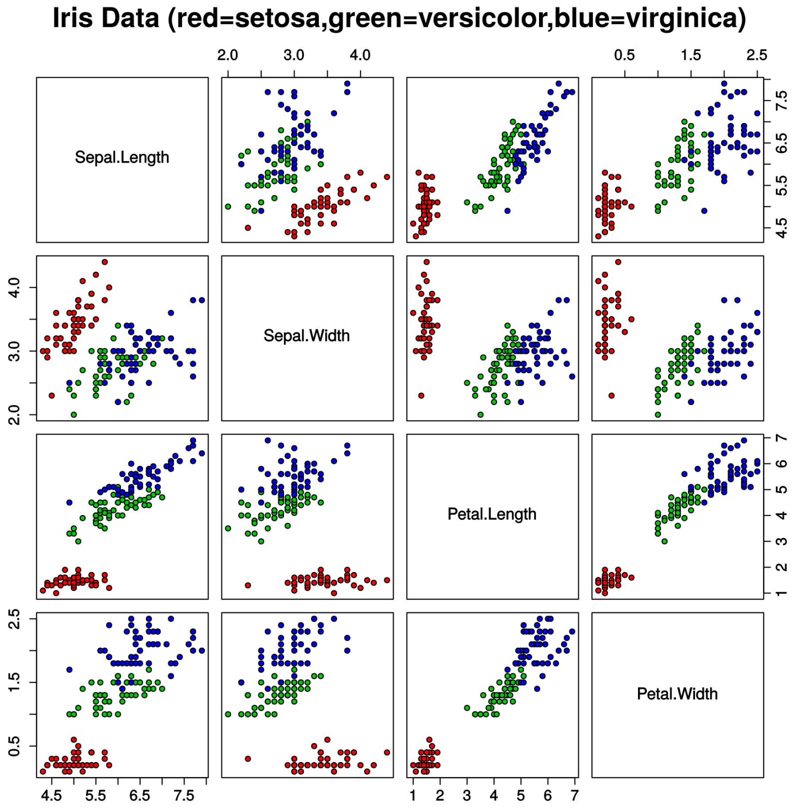
- **<u>Petals</u>** – <u>brightly colored structure just inside the sepals</u> that attracts insects for pollination



Iris flower



Iris flower dataset



Correlation Matrix