

Lecture 04: Representing the World, Classification, and Decision Trees

Instructor: Sergei V. Kalinin

What are our objects?



Quantitative Variables:

- 3 apples, 2 oranges

Categorical Variables:

- Apple and Orange

Schemas:

- "Fruit," "Type," and "Quantity"

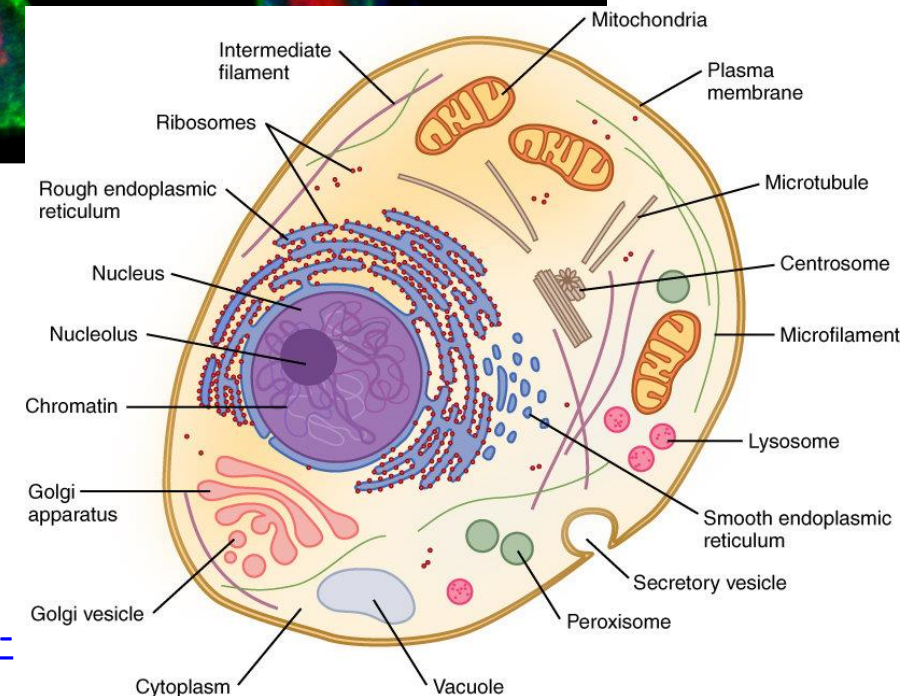
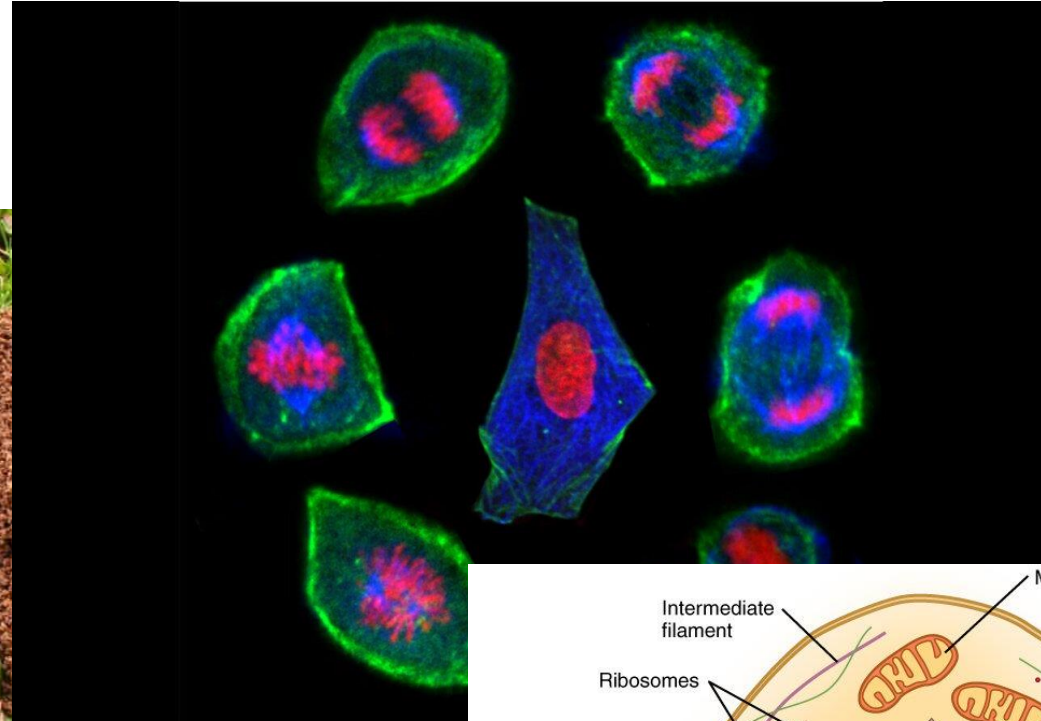
Ontologies:

- "Apple is a type of Fruit"

Knowledge Graphs:

- Connection between "Apple," "Fruit," "Vitamin C content," and "Growth Regions."

No equations (or ML) without representation!



<https://phys.org/news/2021-10-cell-life-optical-tweezers.html>

<https://doi.org/10.1080/00150193.2020.1722012>

<https://www.pointepestcontrol.net/inside-the-ant-hill/>

https://www.researchgate.net/figure/Schematic-representation-of-a-cell-20-with-organelles-delimited-or-made-of-lipid_fig5_350133282

What are our objects?

Alphabet Pronunciation			
A	B	C	D
[eɪ]	[bi:]	[si:]	[di:]
F	G	H	I
[ef]	[dʒi:]	[ertʃ]	[aɪ]
K	L	M	N
[keɪ]	[el]	[em]	[en]
P	Q	R	S
[pi:]	[kju:]	[a:]	[es]
U	V	W	
[ju:]	[vi:]	['dʌbəlju:]	
X	Y	Z	
[eks]	[waɪ]	[zed/zi:]	

Engli

我	的	你	是	了	不	们	这	一	他	么
shǎo	qǐe	shī	suǎn	xìng	cǐ	mén	zhè	yī	tā	me
few; little	to cut; to slice	to lose	to calculate	nature; gender	this; these	plural p.	this	one; a	he; him	interrogative p.
下	真	现	做	大	因	用	打	地	再	正
xià	zhēn	xiàn	zuò	dà	yīn	yòng	dǎ	de	zài	zhèng
elbow; down	really; truly	present	to do	big	reason	to use	to hit	-ly structural p.	again; then	just (right); correct
候	次	信	欢	白	吃	亲	种	者	嘿	队
hòu	cì	xìn	huān	bái	chī	qīn	zhǒng	zhě	hēi	duì
wait; season	mw. for time	letter; to trust	joyous	white; pure	to eat	rent; relative	kind	one who (is)	hey	team; group
嗯	计	任	确	德	队	必	备	合	德	队
en	jì	rèn	què	dé	duì	bì	bèi	hé	dé	duì
roval interjection	to plan	to appoint; office	solid; real	virtue; ethics	team; group	must; will	get ready	to close; together	virtue; ethics	team; group

Haiku by Matsuo Bashō reading "Quietly, quietly, / yellow
mountain roses fall – / sound of the rapids"



Schemas and SQL

A **schema** is a blueprint that defines the structure of a database, outlining how data is organized into tables, fields, and relationships. A schema might define tables for Students, Courses, and Enrollments, showing how they relate to each other.

SQL (Structured Query Language) is a standard language used to manage and query data in relational databases. SQL relies on schemas to define and enforce the structure of data, ensuring consistency and integrity.

We can represent data with schemas in Python using:

- **Dictionaries:** Flexible, key-value pairs.

```
student = {"StudentID": 1, "Name": "Alice", "Major": "Physics"}
```

- **Pandas DataFrame:** Powerful structure for handling tabular data

```
import pandas as pd
```

```
df = pd.DataFrame([{"StudentID": 1, "Name": "Alice", "Major": "Physics"}])
```

- **JSON:** For storing data in a structured text format.

```
import json  
json_data = json.dumps(student)
```

Ontologies

- A structured framework that defines a set of concepts and categories within a domain, along with the relationships between them.
- **Vocabulary:** A set of standardized terms used to describe concepts and relationships within an ontology. Terms like "Material," "Property," and "Process" in a materials science ontology.
- **Semantics:** The meaning and interpretation of the terms and relationships in an ontology. "Material has Property" describes the relationship between a material and its attributes.
- **Syntax:** The formal structure or rules that govern how information is represented and encoded in an ontology. OWL (web ontology language) syntax defines how classes and properties are written, RDF (resource description framework) syntax uses triples (subject, predicate, object).
- **Classes:** Categories or types of things within an ontology. "Mammal" as a class in a biological ontology.
- **Instances:** Specific examples or members of a class. "Dog" as an instance of the class "Mammal."
- **Relations:** The ways in which concepts or instances are connected within an ontology. "Material undergoes Process" to describe how a material is involved in a manufacturing step.

Knowledge Graphs

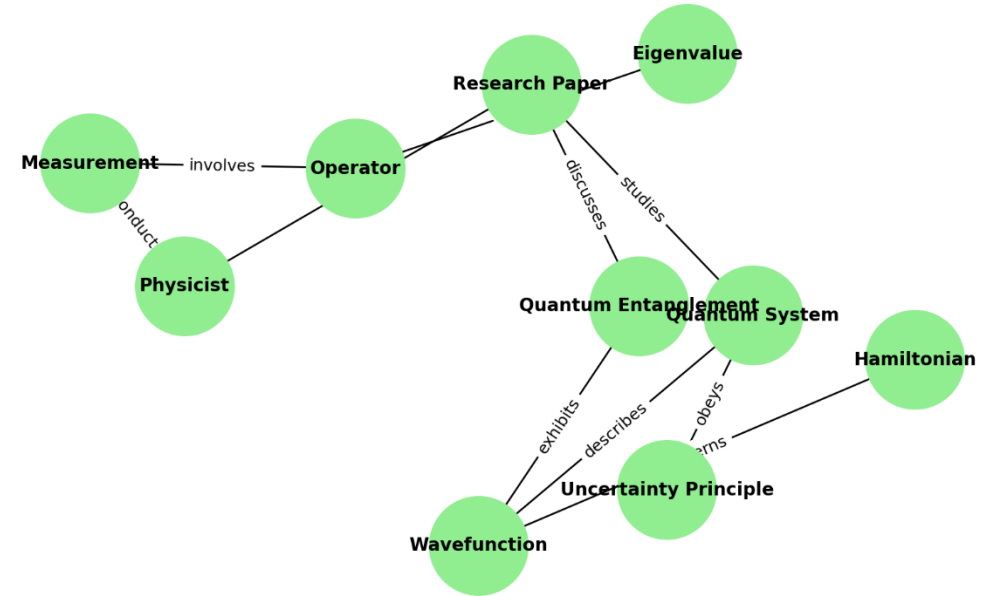
A knowledge graph is a network of interconnected entities (nodes) and their relationships (edges), often built upon ontologies.

- Nodes represent entities or concepts (e.g., "Person," "Material," "Document").
- Edges represent relationships between entities (e.g., "authored," "relatedTo," "hasProperty").
- Knowledge graphs are often constructed using ontologies to define the types of nodes and edges, ensuring consistency and meaning across the graph.

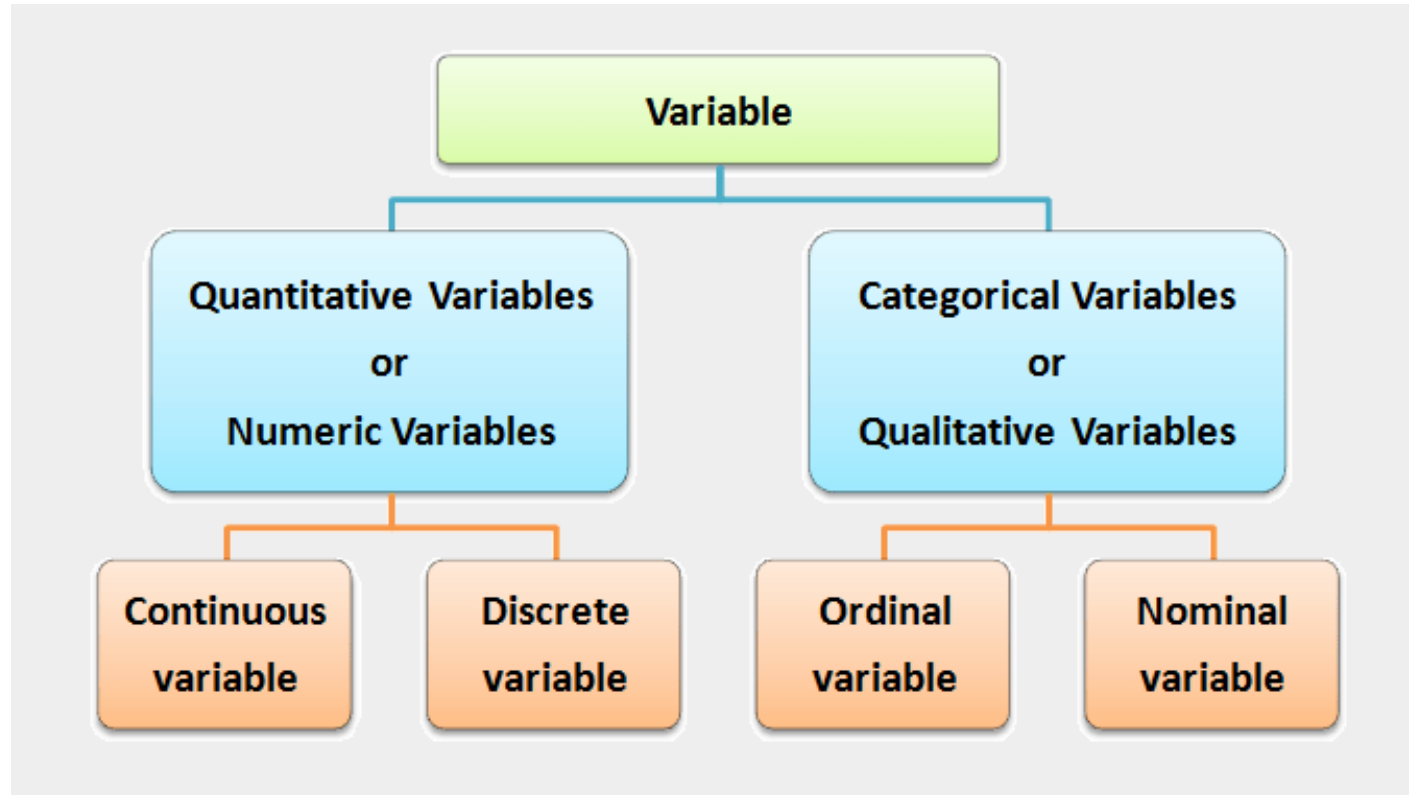
How Knowledge Graphs are Used:

- Data Integration: Combine and link data from multiple sources, making it easier to query and analyze complex information.
- Semantic Search: Enable more precise and context-aware search capabilities by understanding the relationships between entities.
- Inference and Reasoning: Use the relationships in the graph to infer new knowledge or validate existing relationships.

Example of a Knowledge Graph in Quantum Mechanics



Type of variables



Nominal variable: no order
Colors: blue, red, yellow, white
Dog breed: Labrador, German shepherd, poodle

Ordinal variable: there is order
Letter grades: A, B, C, ...
Severity of the software bug: mild, serious, critical

<https://prinsli.com/categorical-variables/>

We have got the data... Now what?

- Classification is the process of identifying and grouping objects or ideas into predetermined categories.
- In data management, classification enables the separation and sorting of data according to set requirements for various business or personal objectives.
- In machine learning (ML), classification is used in predictive modeling to assign input data with a class label.

Classification Workflow

1. Selecting features and collecting labeled training examples
2. Choosing a performance metric
3. Choosing a learning algorithm and training a model
 - Can we understand how it works?
 - Does it have any hyperparameters
 - How well does it generalize to new data?
 - How expensive is it?
4. Evaluating the performance of the model
5. Changing the settings of the algorithm and tuning the model.

There are many classifiers:

Choosing an appropriate classification algorithm for a particular problem task requires practice and experience; each algorithm has its own quirks and is based on certain assumptions.

To paraphrase the **no free lunch theorem** by David H. Wolpert, no single classifier works best across all possible scenarios (*The Lack of A Priori Distinctions Between Learning Algorithms*, Wolpert, David H, *Neural Computation* 8.7 (1996): 1341-1390).

Hello, I am Akinator



Think about a real or
fictional character.
I will try to guess who it
is

akinator®

☐ Inactive
Sensitive content



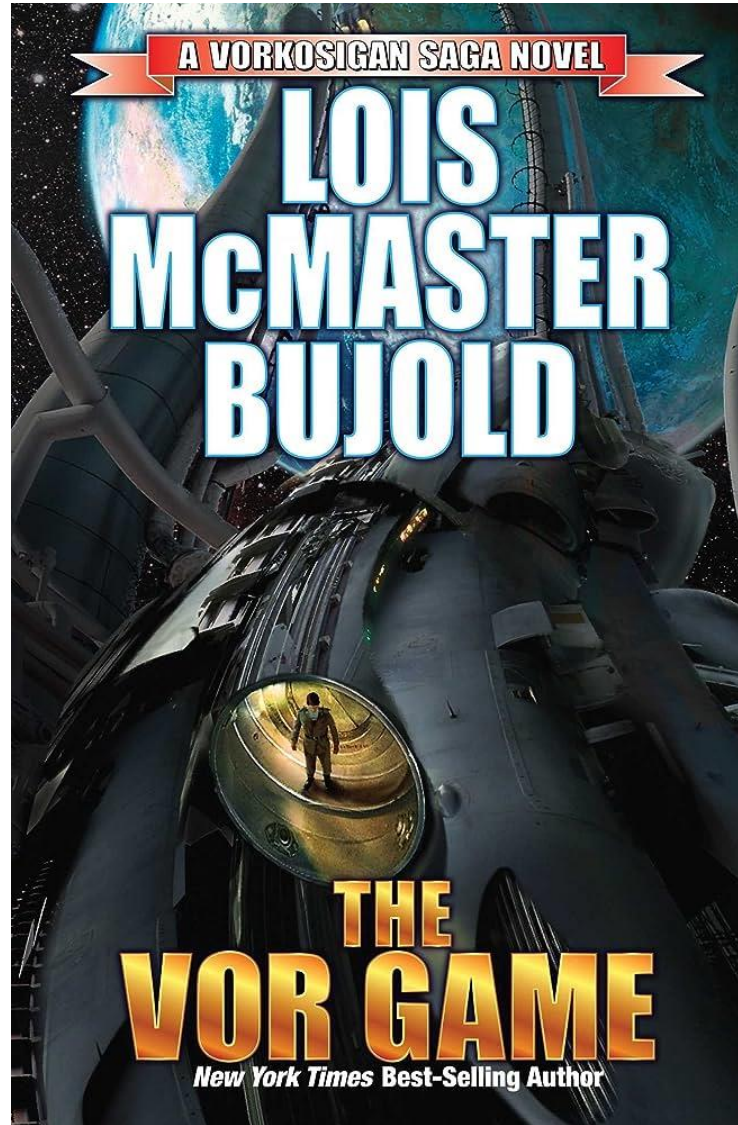
824 people are playing right now.
631563588 games played 30599 today.

Famous sportsmen Akinator can guess

Playing a classifier game:

1. Is your character a pokemon: No
2. Does your character have legs: Yes
3. Is your character a girl: No
4. Is your character famous youtuber: No
5. Is your character real: No
6. Does your character have human head: Yes
7. Is your character from Japanese anime: No
8. Wear a mask: No
9. Originally from video game: No
10. Animated: No
11. Played in superhero movie: No
12. From a book: Yes
13. Have been in movie: No
14. Have powers: No

15. Popular tv show: No
16. Have phone: yes
17. Adult: Yes
18. Killed people: Yes
19. Live in future: Yes
20. Been in space: Yes
21. Wife dead: No
22. Linked with metal suit: No
23. Short: Yes
24. Father famous: Yes



Answer: Miles Vorkosigan

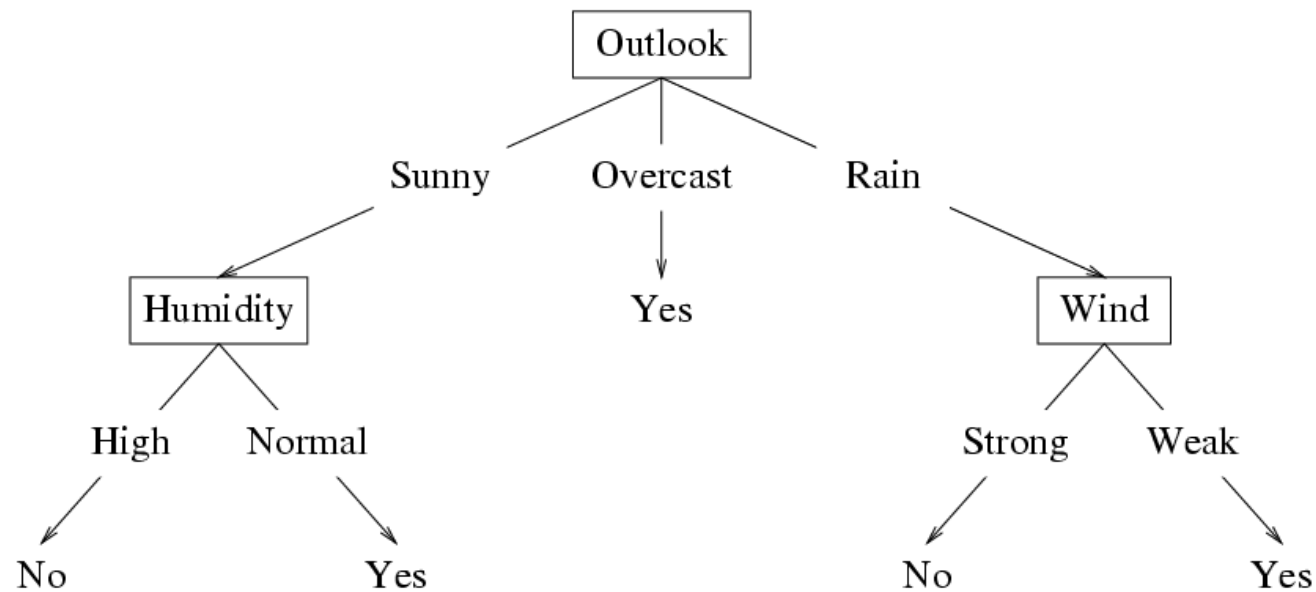
Tennis Player Example

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Decision Tree Hypothesis Space

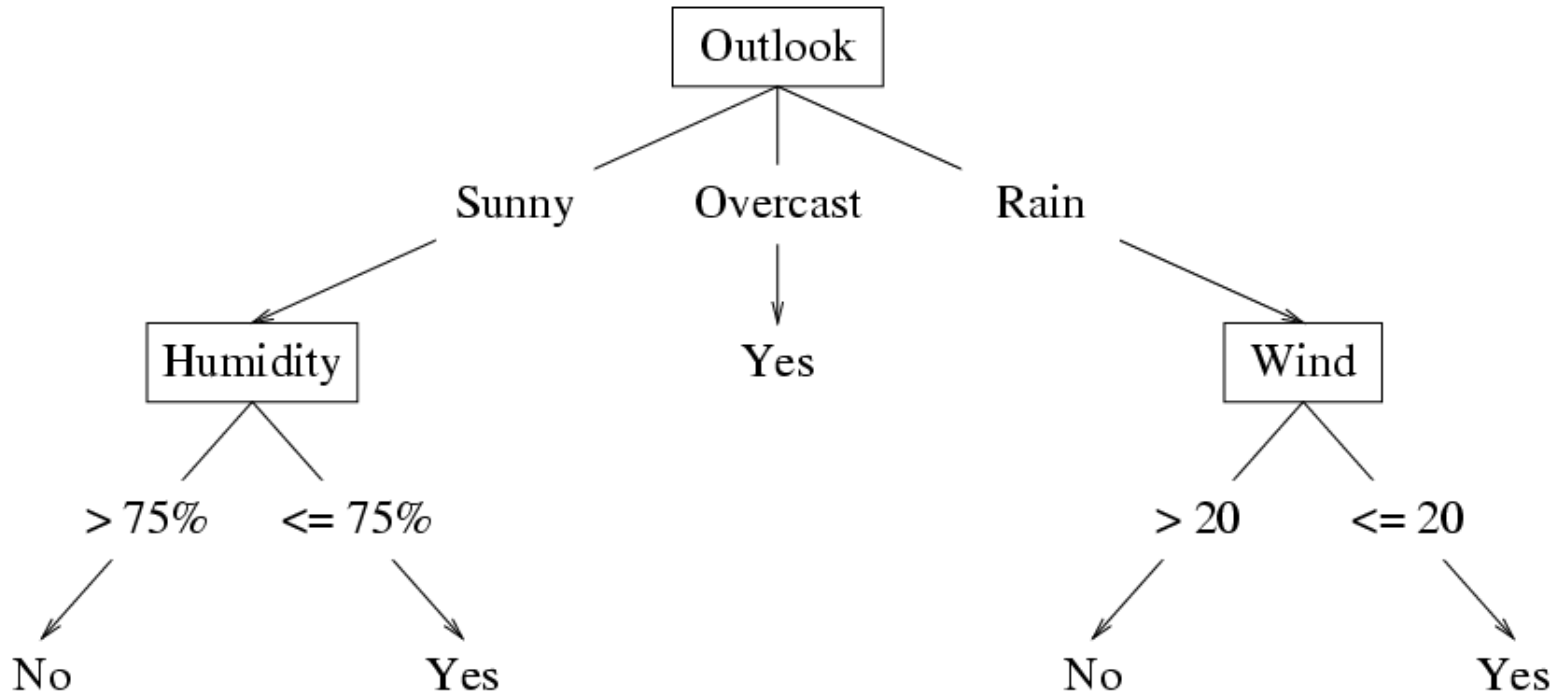
- **Internal nodes** test the value of particular features x_j and branch according to the results of the test.
- **Leaf nodes** specify the class $h(\mathbf{x})$.



Suppose the features are **Outlook** (x_1), **Temperature** (x_2), **Humidity** (x_3), and **Wind** (x_4). Then the feature vector $\mathbf{x} = (\text{Sunny}, \text{Hot}, \text{High}, \text{Strong})$ will be classified as **No**. The **Temperature** feature is irrelevant.

What if we have continuous variables?

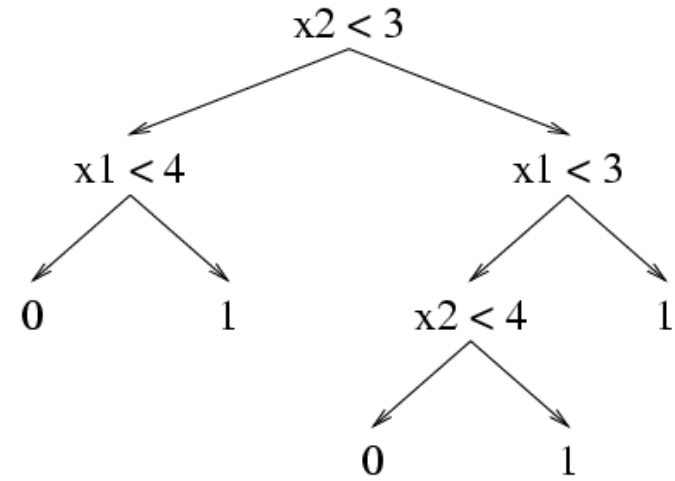
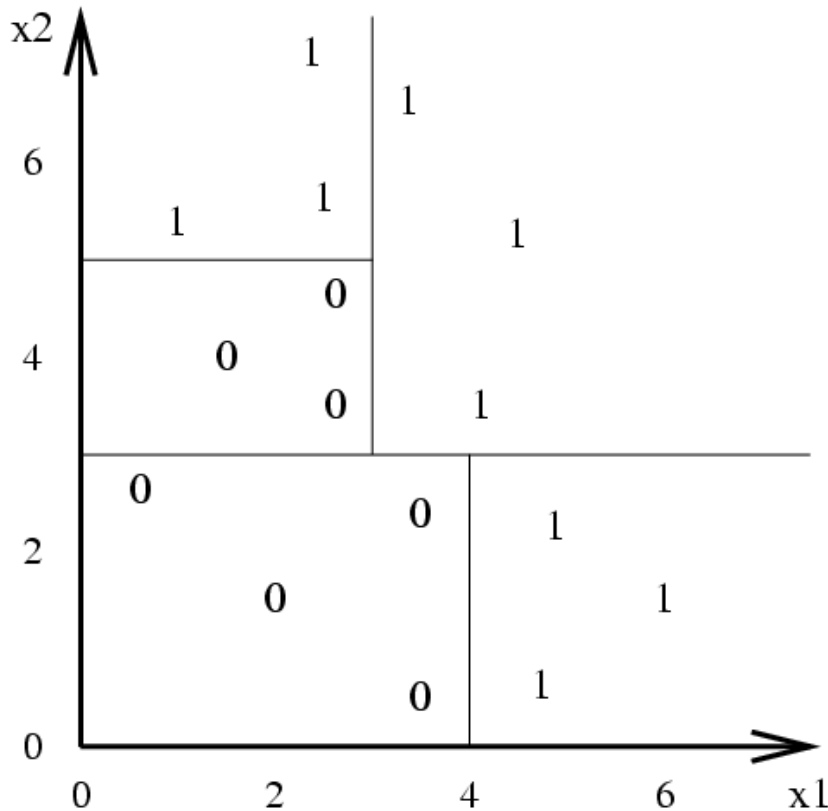
If the features are continuous, internal nodes may test the value of a feature against a threshold.



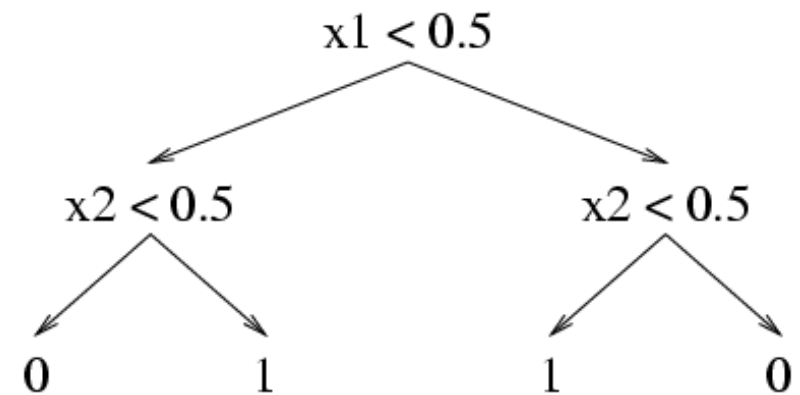
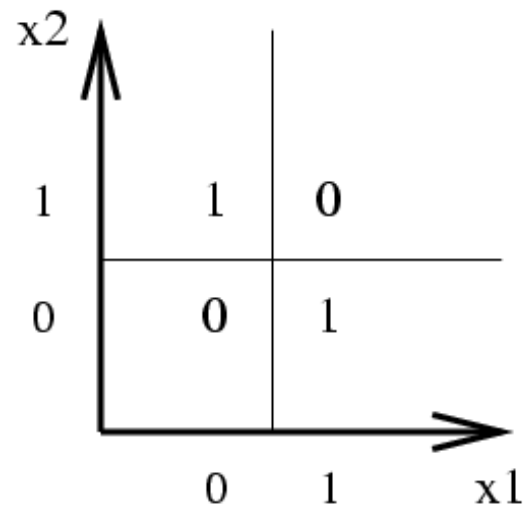
Also: regression trees

Decision Tree Boundaries

Decision trees divide the feature space into axis-parallel rectangles, and label each rectangle with one of the K classes.



Can Represent Any Boolean Function



The tree will in the worst case require exponentially many nodes, however.

Classification and Regression Tree (CART)

- Create a set of questions that consists of all possible questions about the measured variables
- Select a splitting criterion (likelihood).
 - Initialization: create a tree with one node containing all the training data.
 - Splitting: find the best question for splitting each terminal node. Split the one terminal node that results in the greatest increase in the likelihood.
 - Stopping: if each leaf node contains data samples from the same class, or some pre-set threshold is not satisfied, stop. Otherwise, continue splitting.
 - Pruning: use an independent test set or cross-validation to prune the tree.
- There are ways to estimate and incorporate priors into the decision tree (though these methods somewhat predate Bayesian methods).
- Computational complexity is very low for both evaluation and training.