

Introduction to AI



**FAIRLEIGH
DICKINSON
UNIVERSITY**

Some slides are adapted from “MIT OpenCourseWare ‘AI 101’ (Fall 2021)”.
1

Artificial Intelligence

AI: A field of computer science focused on creating machines and software that can perform tasks typically requiring human intelligence



Learning

Acquiring knowledge from data
(e.g., machine learning)



Reasoning

Drawing conclusions from available information



Problem-Solving

Identifying solutions to complex issues



Perception

Interpreting sensory data to understand the environment



Language Understanding

Processing and generating human language



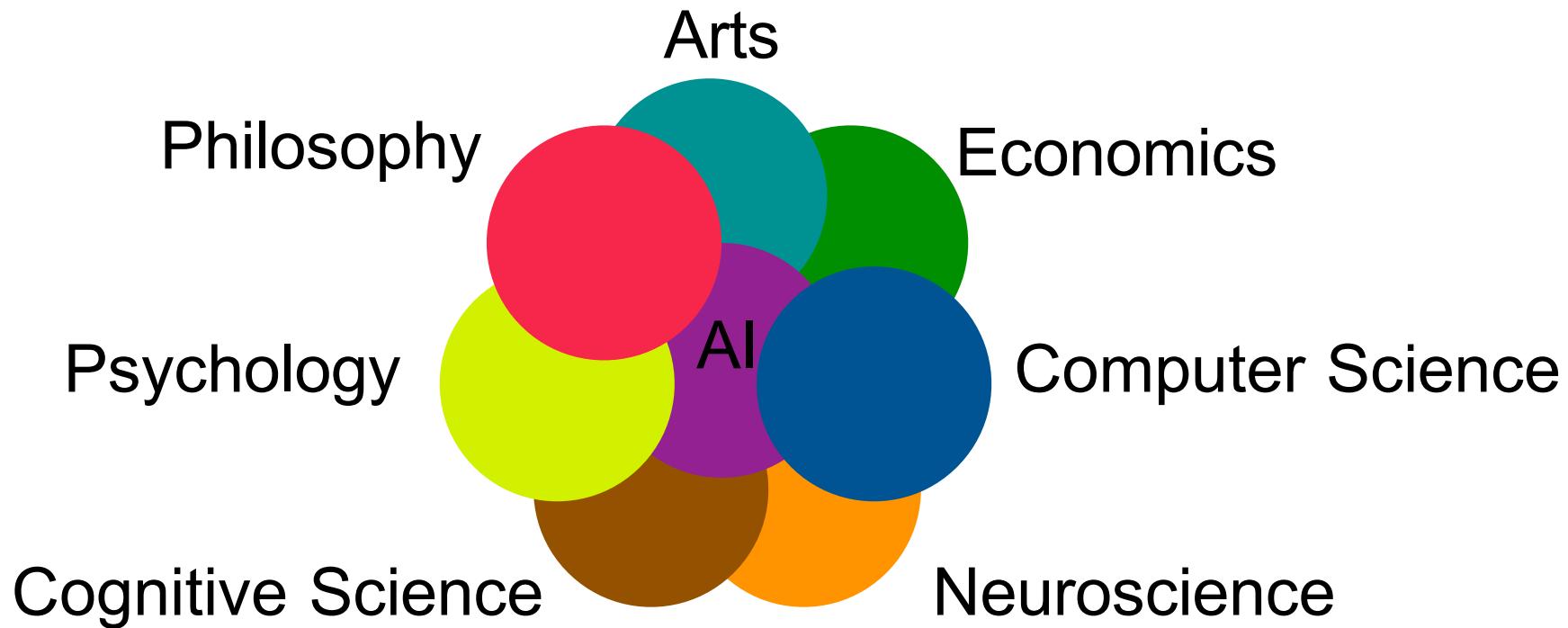
Decision-Making

Making choices based on analysis



**FAIRLEIGH
DICKINSON
UNIVERSITY**

Artificial Intelligence in Context



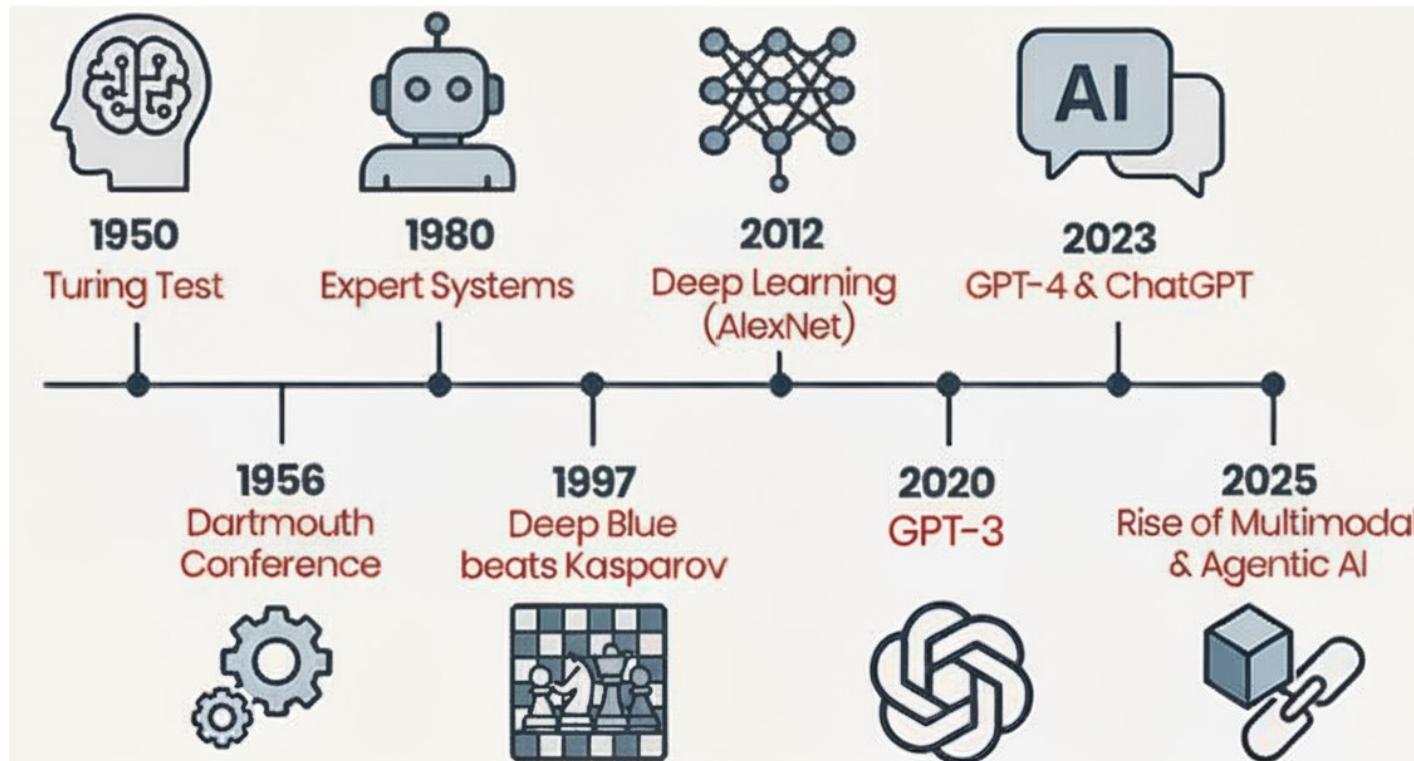
**FAIRLEIGH
DICKINSON
UNIVERSITY**

History of AI



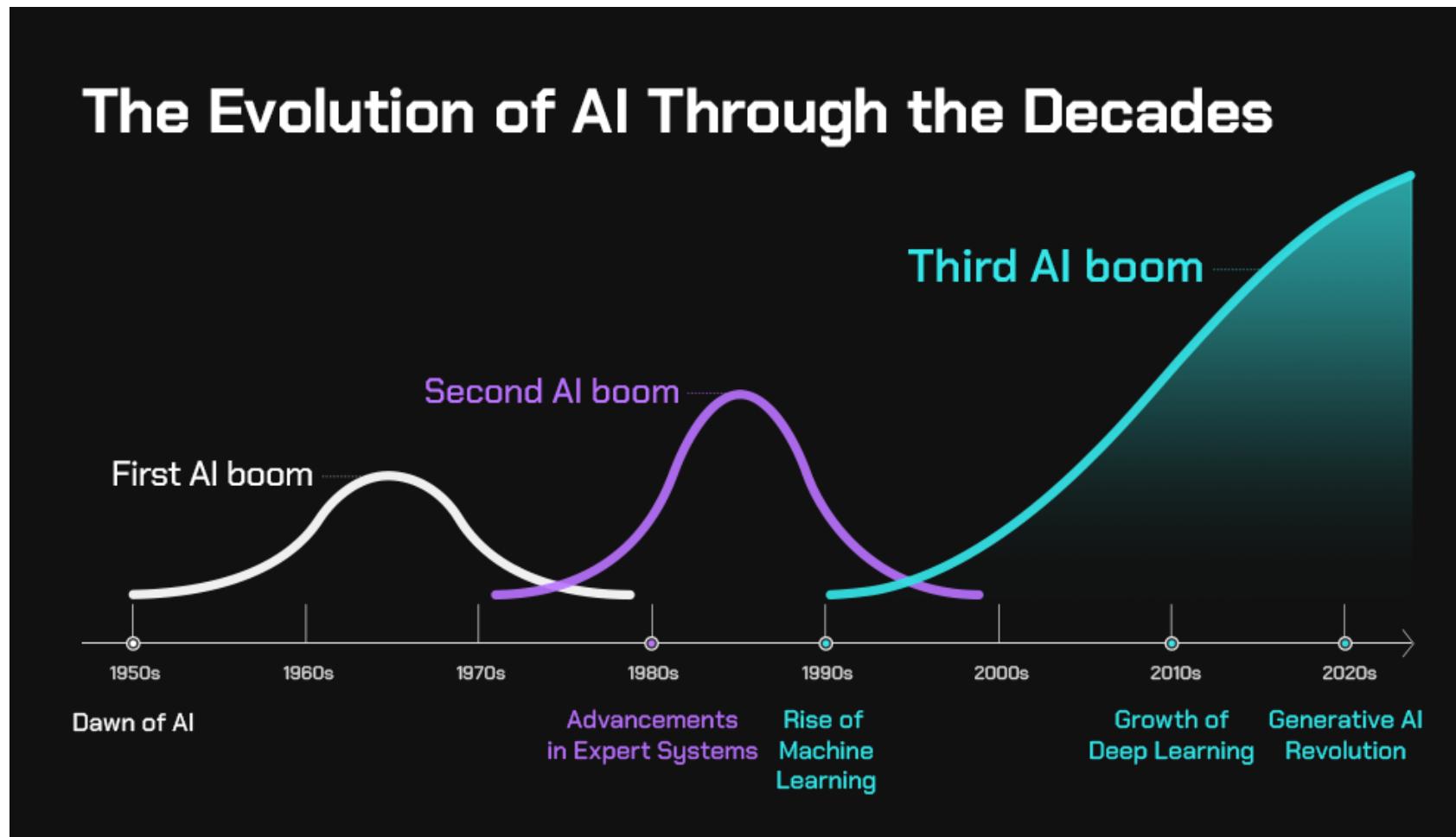
**FAIRLEIGH
DICKINSON
UNIVERSITY**

History of AI



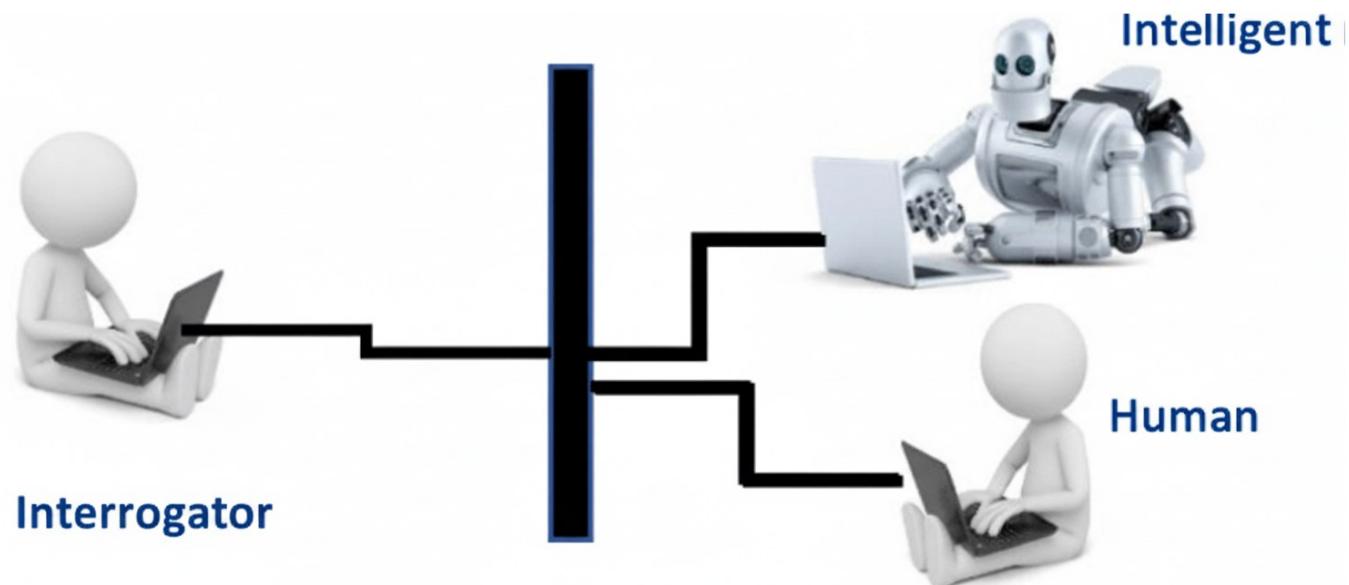
**FAIRLEIGH
DICKINSON
UNIVERSITY**

History of AI



**FAIRLEIGH
DICKINSON
UNIVERSITY**

Turing Test



In the test, an **interrogator (human judge)** communicates with both a **human** and a **machine** (usually through text). The interrogator's goal is to figure out which one is the machine. If the interrogator cannot reliably tell the difference between the human and the machine, then the machine is said to have **passed the Turing Test**.



**FAIRLEIGH
DICKINSON
UNIVERSITY**

Dartmouth Conference

A PROPOSAL FOR THE
DARTMOUTH SUMMER RESEARCH PROJECT
ON ARTIFICIAL INTELLIGENCE

**1956 Dartmouth Conference:
The Founding Fathers of AI**



J. McCarthy, Dartmouth College
M. L. Minsky, Harvard University
N. Rochester, I.B.M. Corporation
C.E. Shannon, Bell Telephone Laboratories

1. Automatic Computers

If a machine can do a job, then an automatic calculator can be programmed to simulate the machine. The speeds and memory capacities of present computers may be insufficient to simulate many of the higher functions of the human brain, but the major obstacle is not lack of machine capacity, but our inability to write programs taking full advantage of what we have.

2. How Can a Computer be Programmed to Use a Language

It may be speculated that a large part of human thought consists of manipulating words according to rules of reasoning and rules of conjecture. From this point of view, forming a generalization consists of admitting a new word and some rules whereby sentences containing it imply and are implied by others. This idea has never been very precisely formulated nor have examples been worked out.

3. Neuron Nets

How can a set of (hypothetical) neurons be arranged so as to form concepts. Considerable theoretical and experimental work has been done on this problem by Uttley, Rashevsky and his group, Farley and Clark, Pitts and McCulloch, Minsky, Rochester and Holland, and others. Partial results have been obtained but the problem needs more theoretical work.

4. Theory of the Size of a Calculation

If we are given a well-defined problem (one for which it is possible to test mechanically whether or not a proposed answer is a valid answer) one way of solving it is to try all possible answers in order. This method is inefficient, and to exclude it one must have some criterion for efficiency of calculation. Some consideration will show that to get a measure of the efficiency of a calculation it is necessary to have on hand a method of measuring the complexity of calculating devices which in turn can be done if one has a theory of the complexity of functions. Some partial results on this problem have been obtained by Shannon, and also by McCarthy.

5. Self-Improvement

Probably a truly intelligent machine will carry out activities which may best be described as self-improvement. Some schemes for doing this have been proposed and are worth further study. It seems likely that this question can be studied abstractly as well.

6. Abstractions

A number of types of "abstraction" can be distinctly defined and several others less distinctly. A direct attempt to classify these and to describe machine methods of forming abstractions from sensory and other data would seem worthwhile.

7. Randomness and Creativity

A fairly attractive and yet clearly incomplete conjecture is that the difference between creative thinking and unimaginative competent thinking lies in the injection of some randomness. The randomness must be guided by intuition to be efficient. In other words, the educated guess or the hunch include controlled randomness in otherwise orderly thinking.

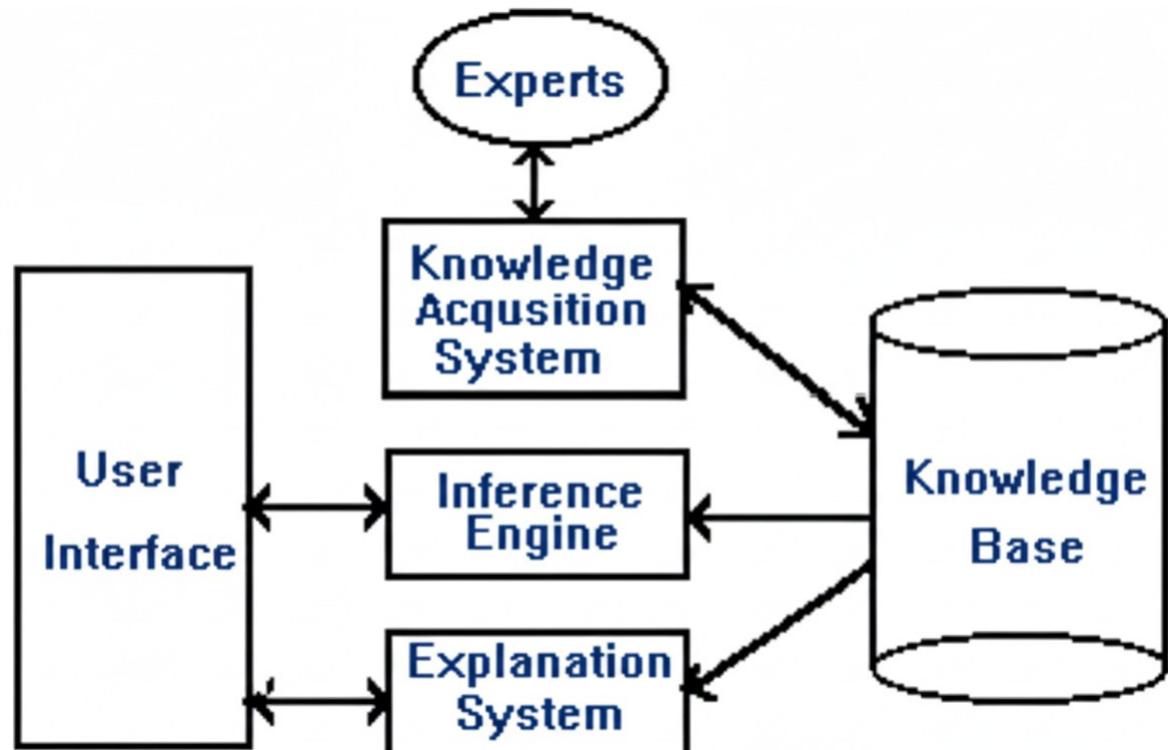
Dartmouth Conference (1956)

- First formal workshop on AI
- Coined the term “Artificial Intelligence”
- Aim: Explore if machines can simulate human intelligence
- Topics: Learning, reasoning, language, problem solving



**FAIRLEIGH
DICKINSON
UNIVERSITY**

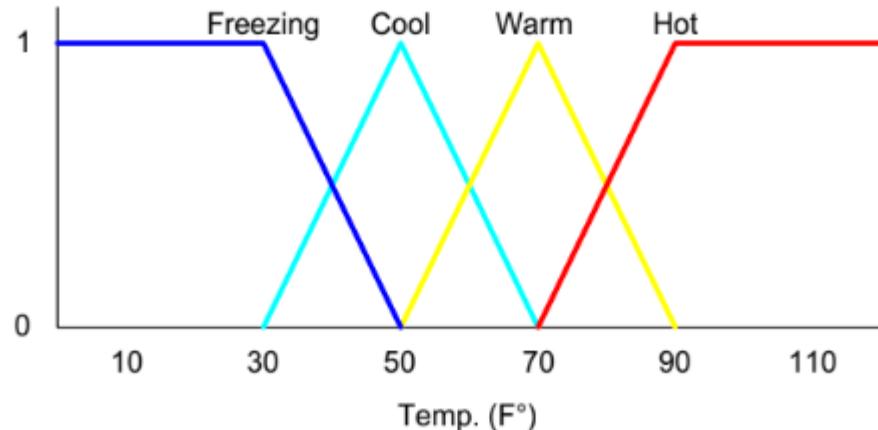
Expert systems



- Early practical AI (1970s–1980s)
- Mimic a **human expert's decision-making**
- Built on:
 - **Knowledge Base** → facts & rules
 - **Inference Engine** → reasoning mechanism
- **Applications:** Medicine (*MYCIN*), engineering, finance, customer support
- **Limitations:** Narrow focus, no common sense, difficult to update
- Pioneered real-world use of AI in problem-solving

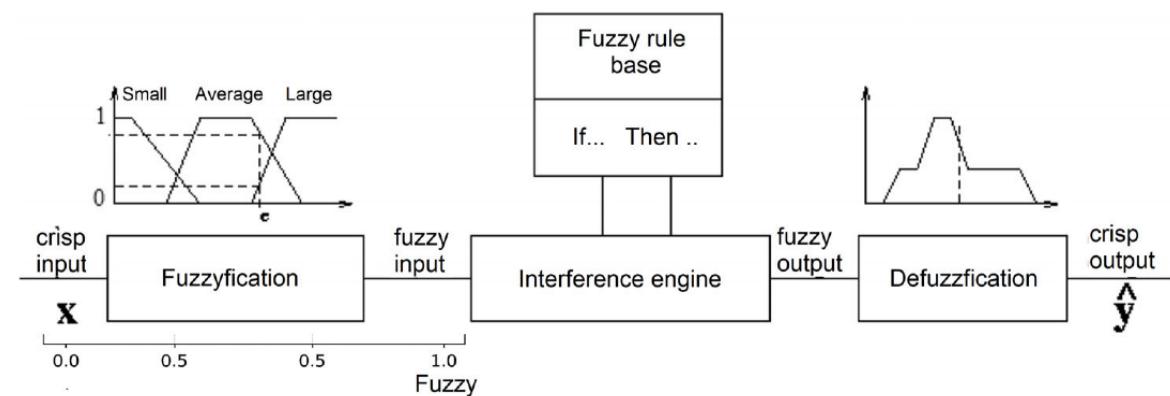


Fuzzy Logic

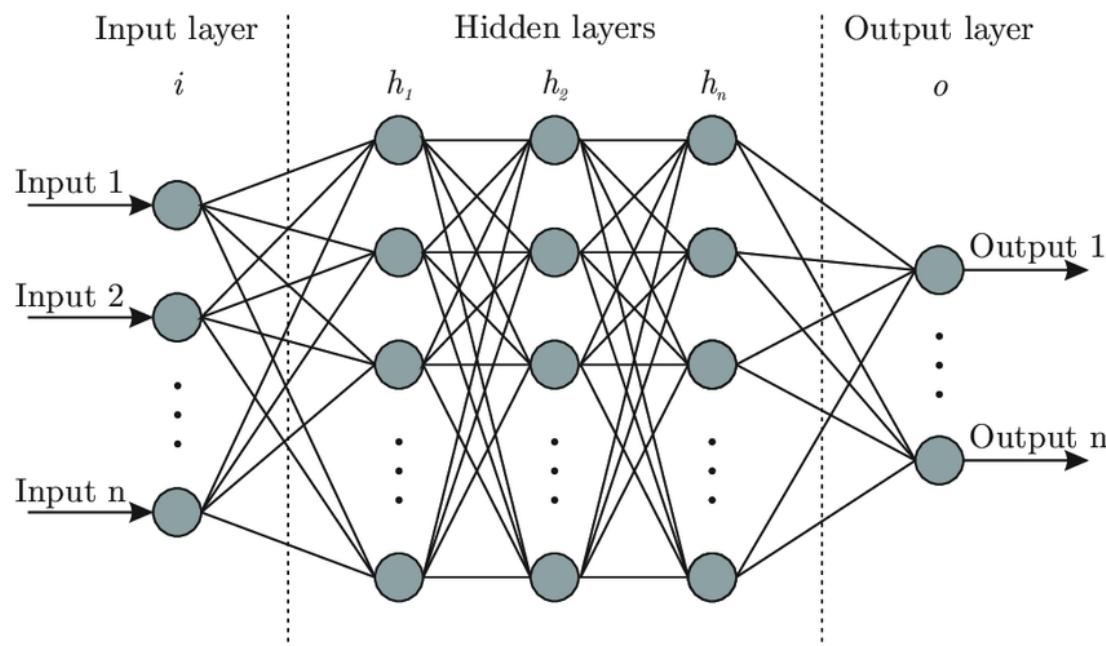


- **Fuzzy Logic:**
- Extends classical logic (True/False) to handle **degrees of truth** (e.g., *0.2 Cool*, *0.8 warm*).
- Useful for reasoning under **uncertainty** and **vague concepts**.

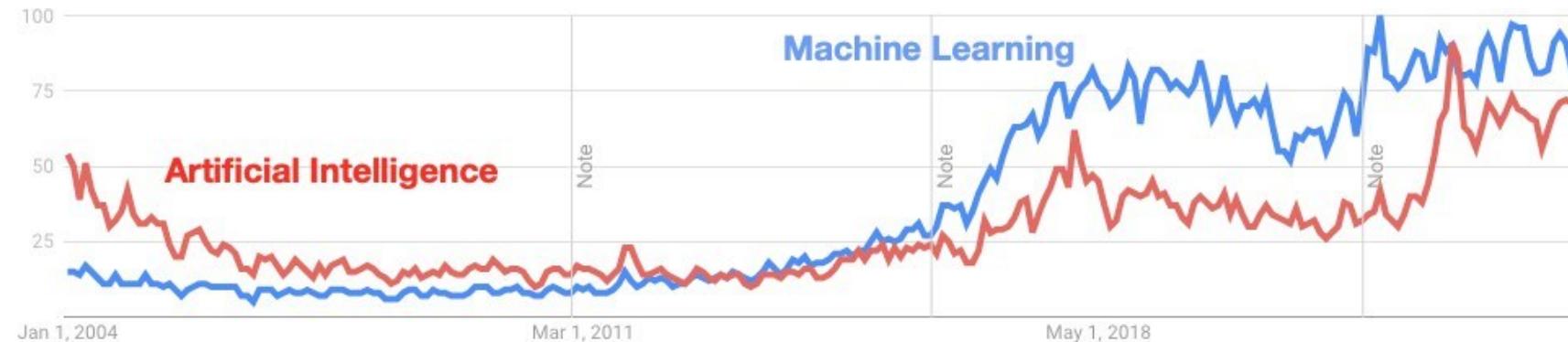
- **Fuzzy Rule-Based Systems:**
- Knowledge represented as **IF-THEN rules** with fuzzy terms (e.g., "warm", "slightly warm," "very high").
- Example: *IF temperature is high THEN fan speed is high.*
- Inference mechanism combines fuzzy rules → produces approximate, human-like reasoning.
- **Applications:** Control systems (washing machines, AC), medical diagnosis, decision support



- Rediscovery of neural networks
 - initially proposed in the 1960s
 - very popular in 1980s / early 1990s ("connectionism")
- Why now? Better hardware
 - typically run on GPUs
 - 5000+ compute cores per processor
- Many toolkits available (Tensorflow, pyTorch, ...)



Shift to Machine Learning (since 1990s)



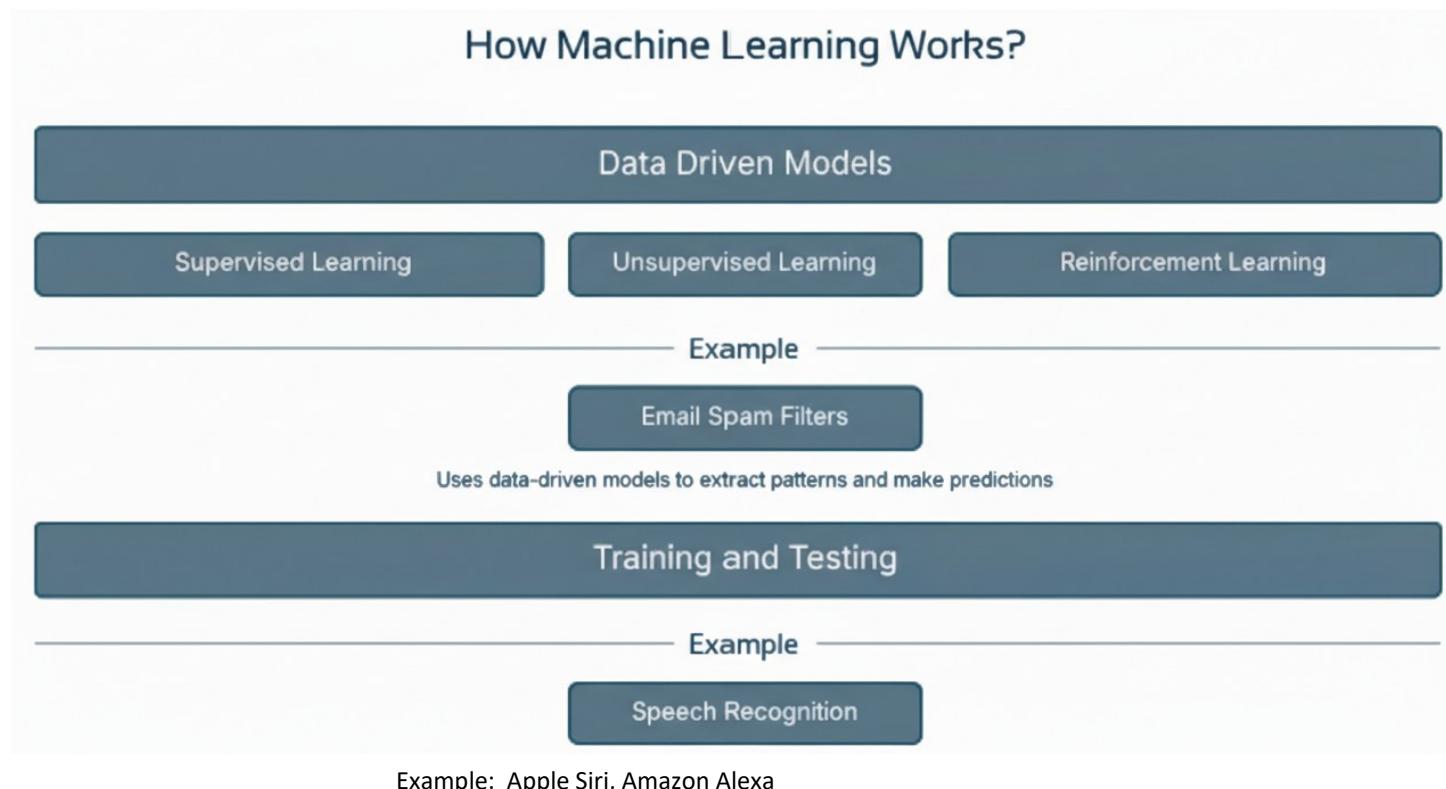
- Idea
 - collect data, maybe annotate data
 - learn patterns automatically
- Many approaches
 - is the truth known? maybe delayed? partially?
 - are we predicting a class or complex structure?
 - is the input/output continuous or discrete?
 - how much of the structure of the problem is known and can be used?
- Dominant paradigm in language and speech processing and many other fields



Shift to Machine Learning (since 1990s)

Primary ML paradigms:

Approach	Description	Example
Supervised Learning	Learns from labeled input–output pairs	Email spam filter
Unsupervised Learning	Discovers patterns in unlabeled data	Customer segmentation (clustering)
Reinforcement Learning	Learns via rewards/penalties in dynamic environments	Game-playing agents (e.g., AlphaGo)



Big Data (since 2000s)

- Computers have became bigger
 - Vast amounts of stored data available (e.g., the Internet)
 - Better sensor systems allow collection of rich information about the environment
- ⇒ AI is big business



Microsoft

facebook

Google™

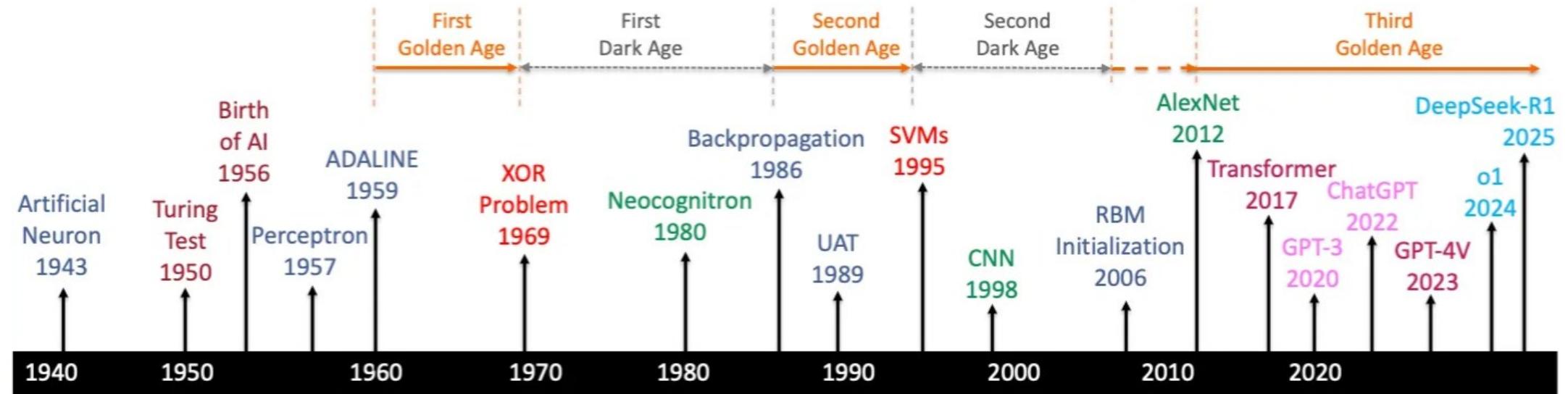


amazon



FAIRLEIGH
DICKINSON
UNIVERSITY
1947

Neural Networks/Deep Learning



McCulloch-Pitts



Rosenblatt



Widrow-Hoff



Minsky-Papert



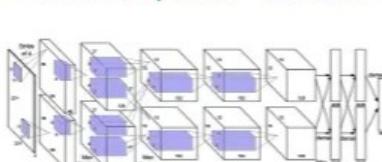
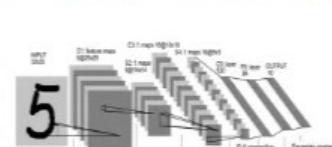
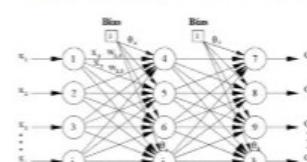
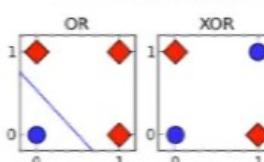
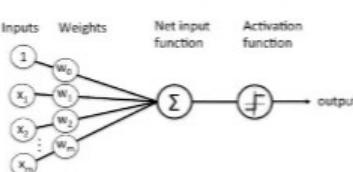
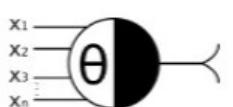
Rumelhart, Hinton et al.



LeCun

Hinton-Ruslan
Krizhevsky et al.

Vaswani



FAIRLEIGH
DICKINSON
UNIVERSITY
19

Deep Learning



2024 Nobel Prizes in Physics & Chemistry: AI at the Forefront

Physics: Neural Network Foundations

- **Laureates:** John Hopfield & Geoffrey Hinton
- **Contribution:** Developed core neural network methods
 - Hopfield: Associative memory systems
 - Hinton: Boltzmann machines for autonomous feature detection
- **Impact:** Laid groundwork for modern AI and deep learning

Chemistry: AI for Protein Science

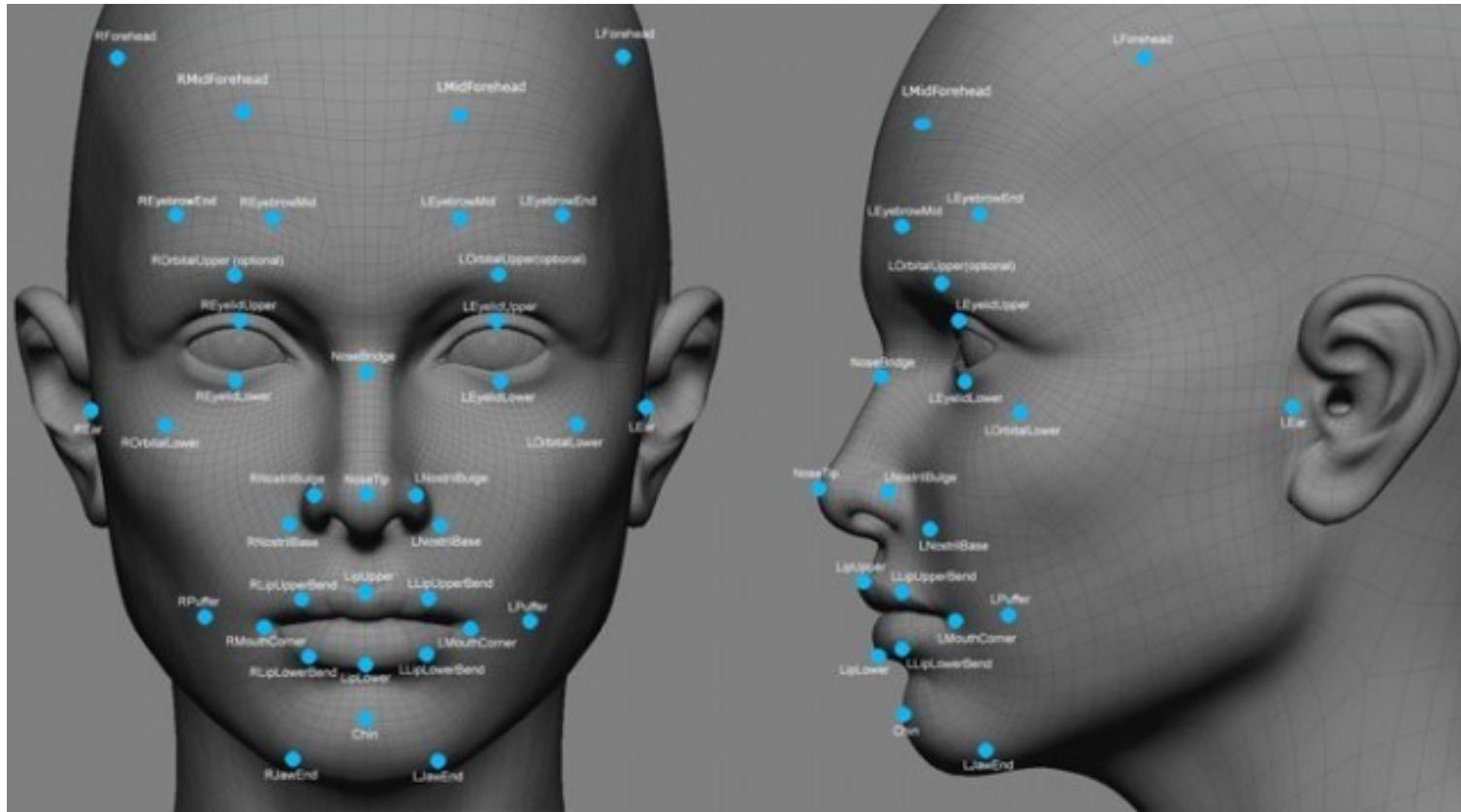
- **Laureates:** David Baker, Demis Hassabis & John Jumper
- **Contribution:** AI-driven protein structure prediction(deep learning/transfomers)
 - Hassabis & Jumper: AlphaFold for accurate 3D protein structures
 - Baker: Designing novel proteins with therapeutic potential
- **Impact:** Accelerated drug discovery and biological research

ML/ DL Applications



**FAIRLEIGH
DICKINSON
UNIVERSITY**

Face Recognition

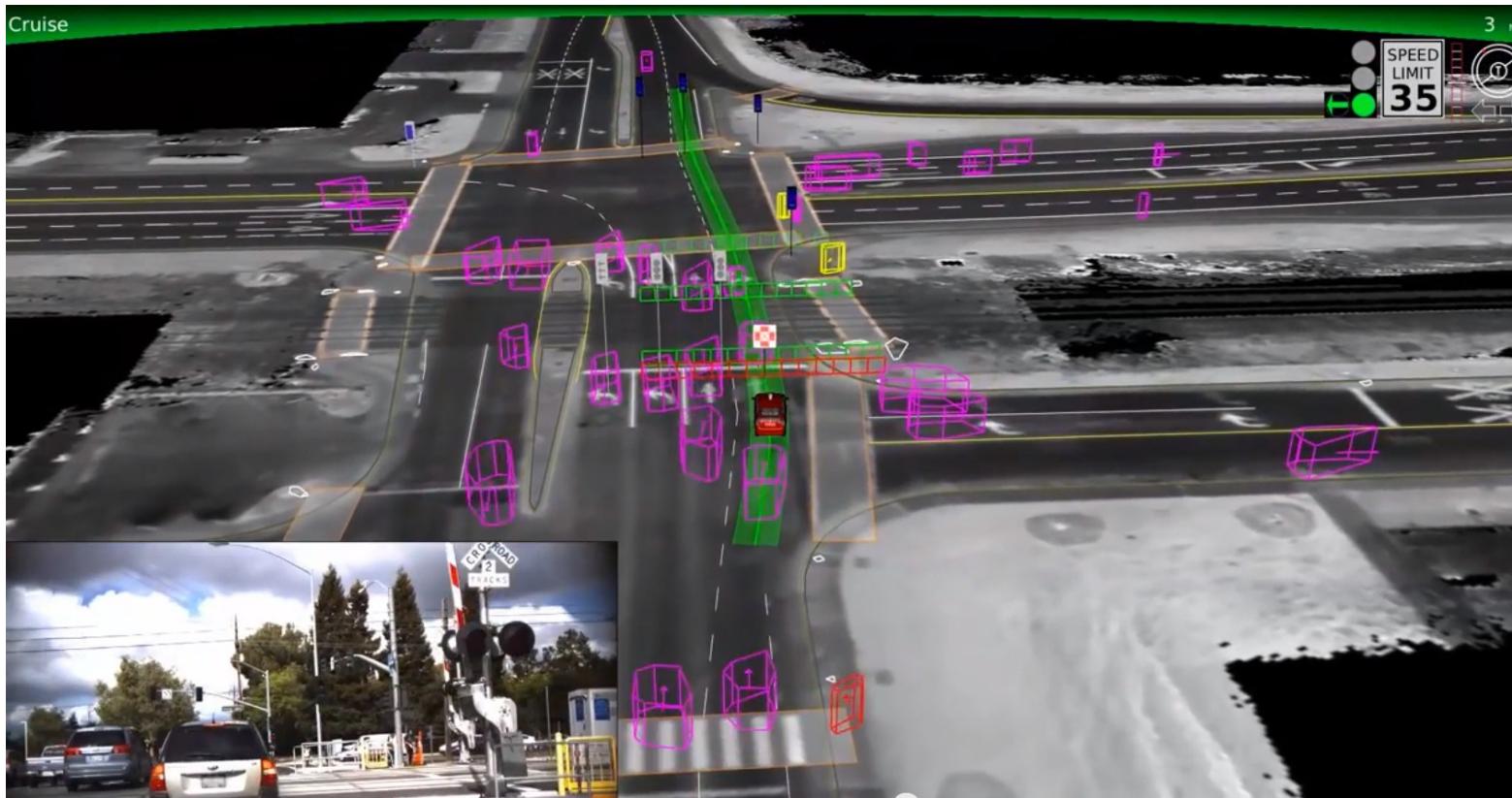


- Used by Facebook to tag photos
- Also used in border control automation



FAIRLEIGH
DICKINSON
UNIVERSITY
18

Self-Driving Cars



- Google self-driving car (Waymo) available as taxi service in selected cities
- Currently uses special high-precision maps



**FAIRLEIGH
DICKINSON
UNIVERSITY**
19

Walking Robots



- Boston Dynamics walking robot: Spot, 2015
- Can walk across difficult terrain, stabilizes itself



FAIRLEIGH
DICKINSON
UNIVERSITY

Generative AI: Images

- Image manipulation
 - converting style
 - swapping faces
- Stable Diffusion:
image from text
- Video?



Generative AI: Language

- November 2022: ChatGPT
 - explosion of use of large language models
 - bigger and better
 - still hard to separate hype from reality
- Cutting edge: generative speech and language models
 - speech synthesis
 - speech translation
 - speech chat

You
I need an example of what you can do for my class today.

ChatGPT
Sure, I can provide examples and assistance across various subjects. Could you please specify the topic or subject for which you'd like an example or help?

You
I want to impress the students in my Artificial Intelligence class.

ChatGPT
Certainly! Here's a simple example related to artificial intelligence:

```
python
# Python code for a basic sentiment analysis using a pre-trained model

from transformers import pipeline

def perform_sentiment_analysis(text):
    sentiment_classifier = pipeline('sentiment-analysis')
    result = sentiment_classifier(text)
    return result[0]['label']

# Example usage
text_to_analyze = "I absolutely loved the movie! The acting was incredible"
sentiment = perform_sentiment_analysis(text_to_analyze)

print(f"Sentiment: {sentiment}")
```

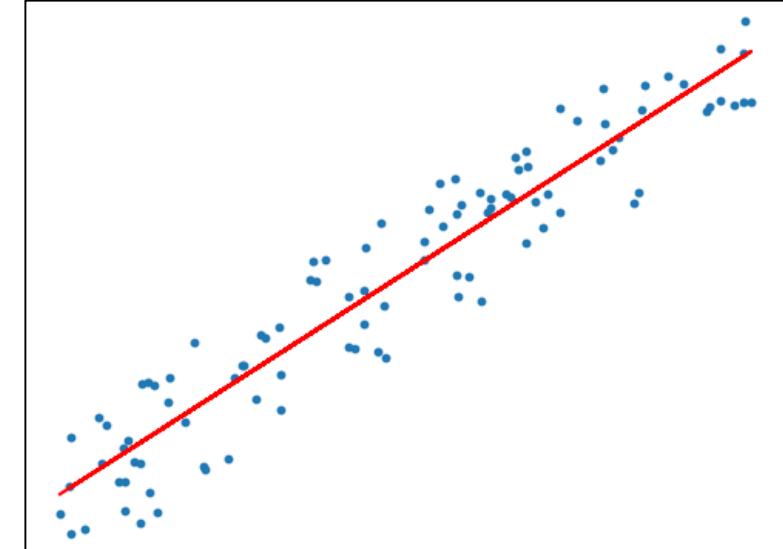
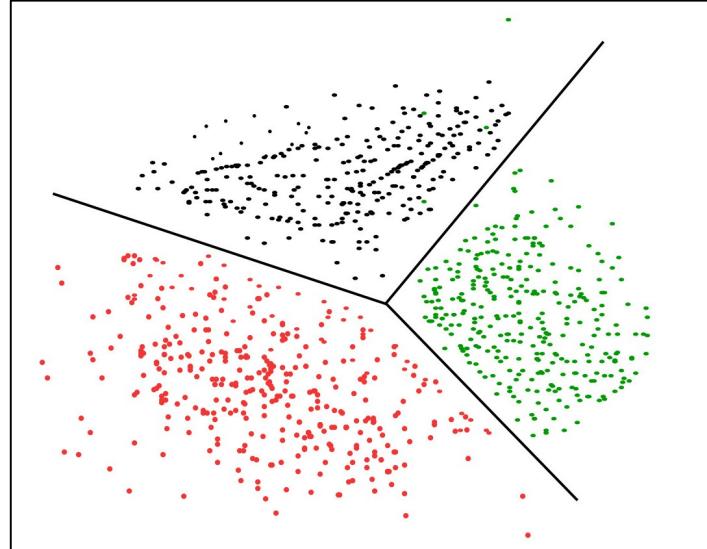
In this example, I'm using the Hugging Face Transformers library to perform sentiment analysis. The code defines a function `perform_sentiment_analysis` that takes a text input, analyzes its sentiment, and returns the result. The example text is then analyzed, and the sentiment label (e.g., 'POSITIVE', 'NEGATIVE') is printed.

You can explain to your students how this simple AI model uses pre-trained language models to understand the sentiment behind a given text. Feel free to modify or expand upon this example based on the level and interests of your class.

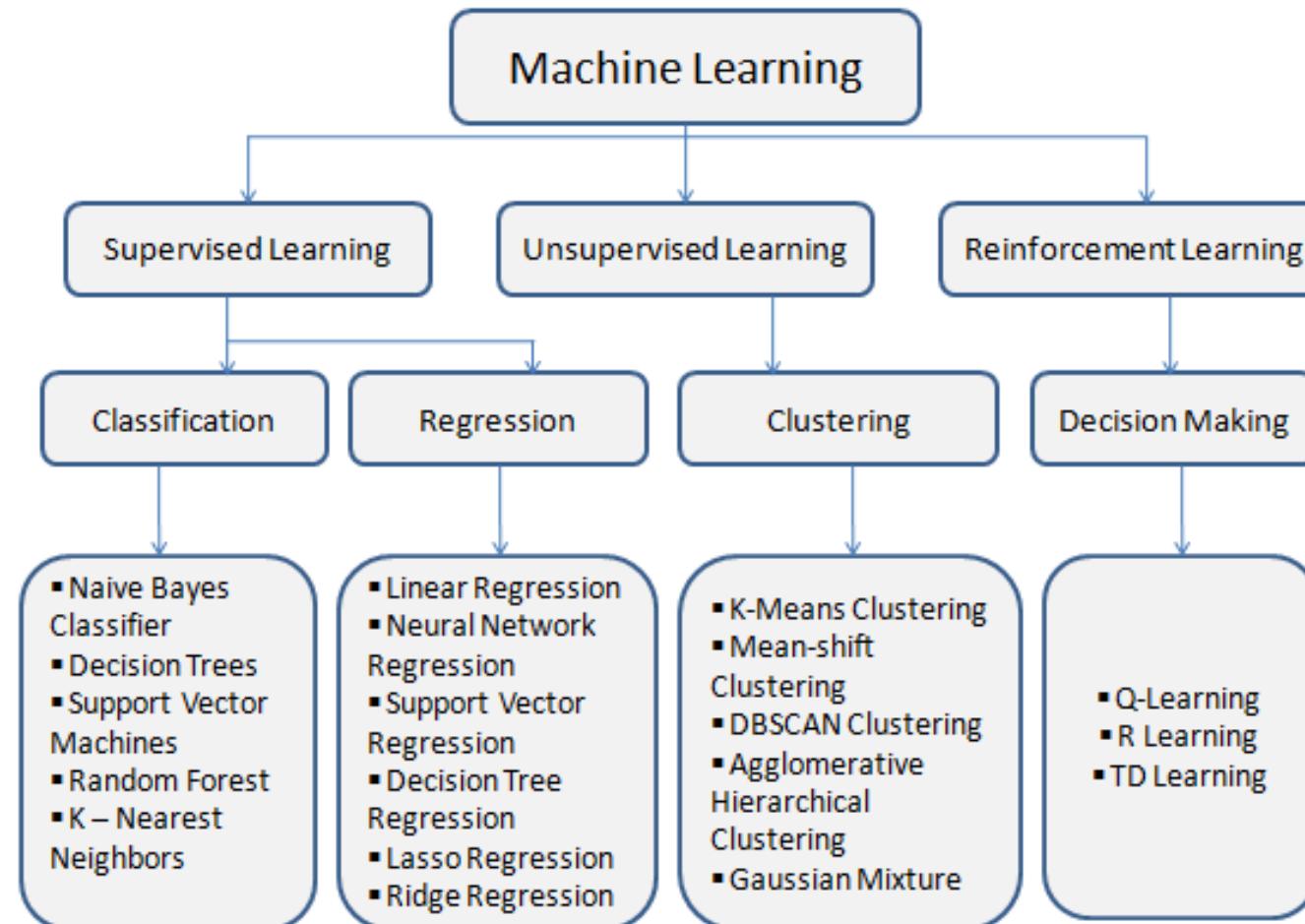


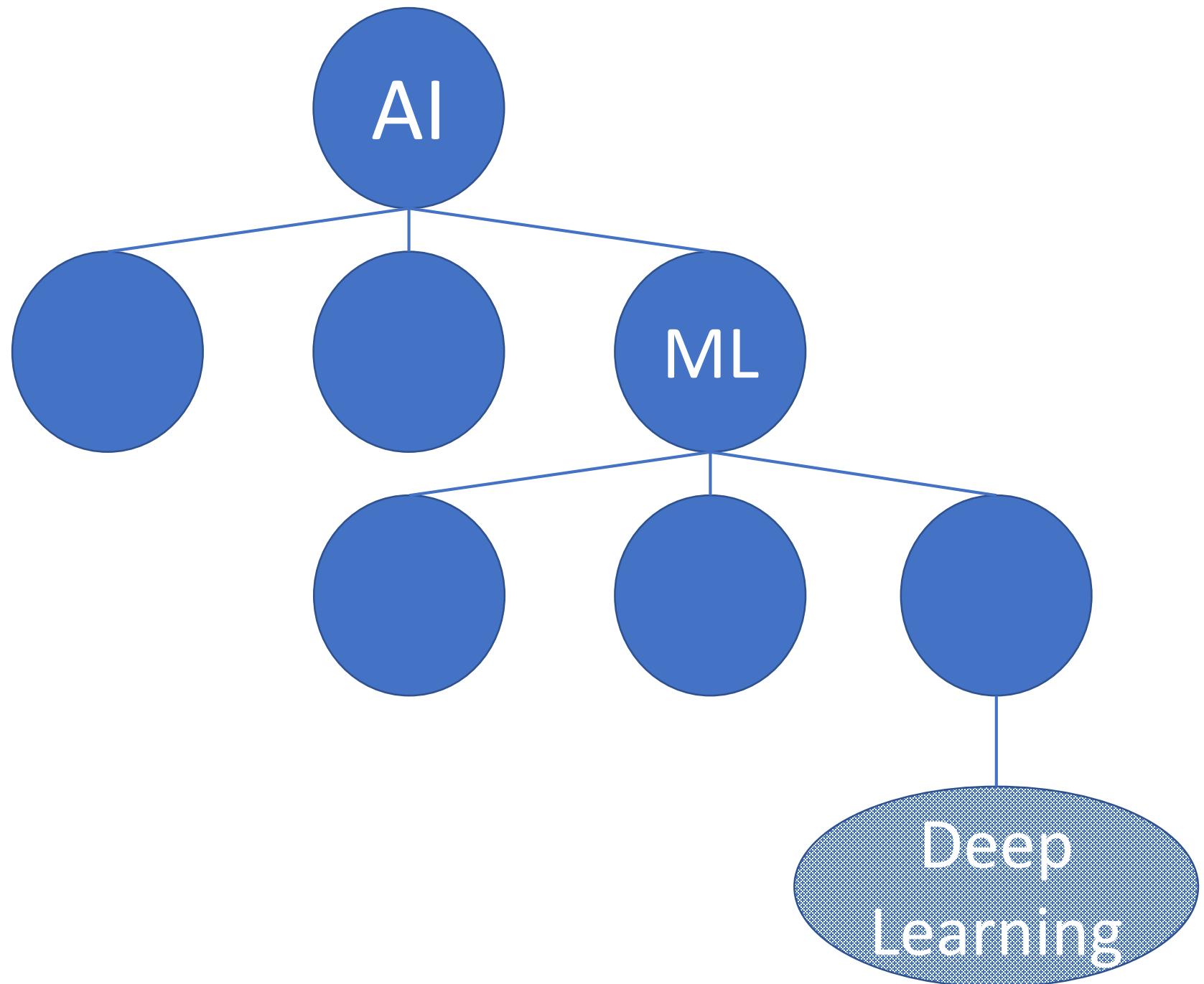
Machine Learning

Machine learning can perform many tasks,
i.e. classification, clustering, and regression.

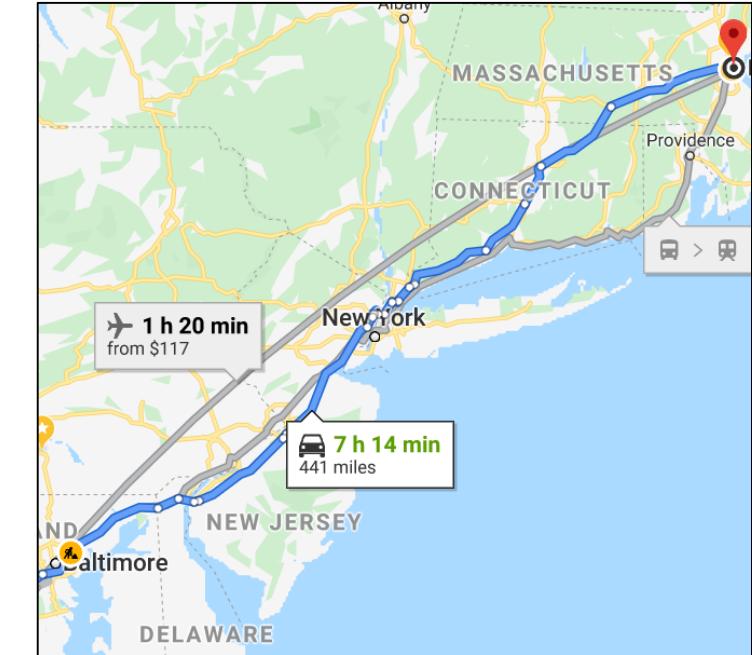
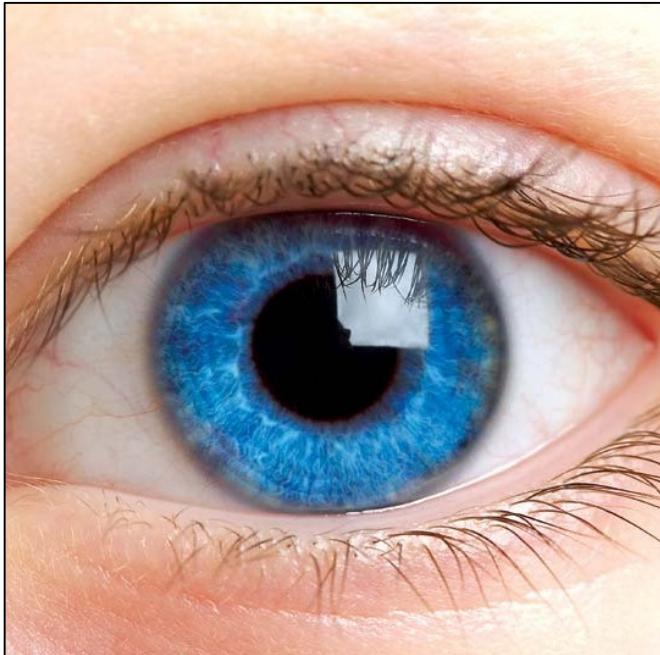


Machine Learning Algorithms

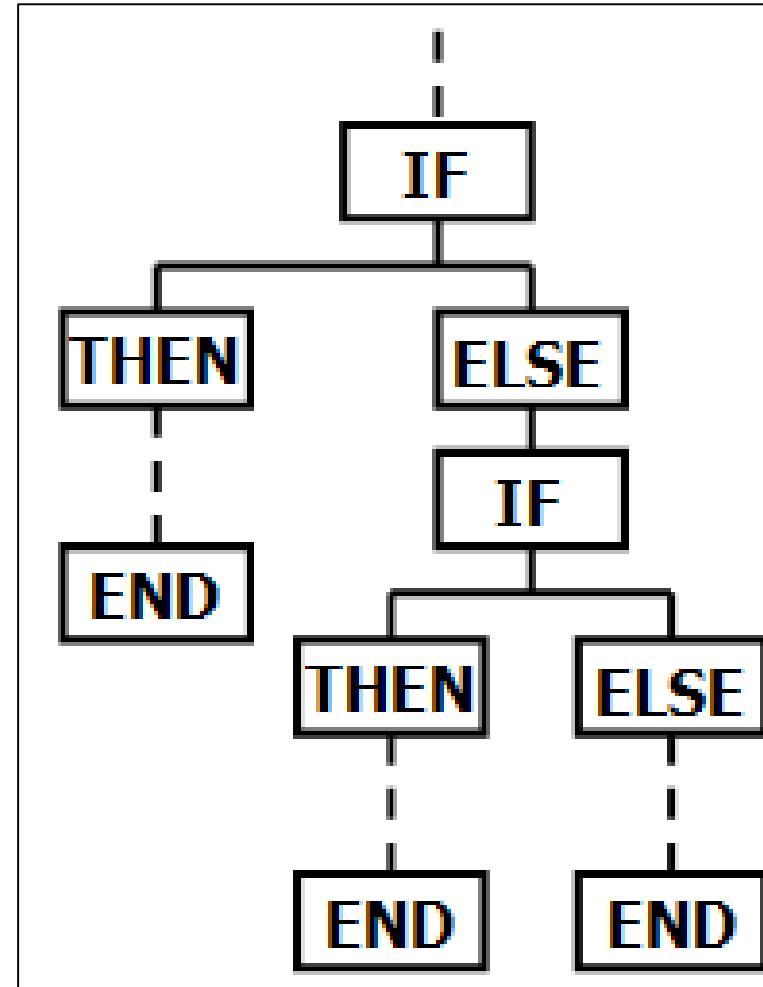




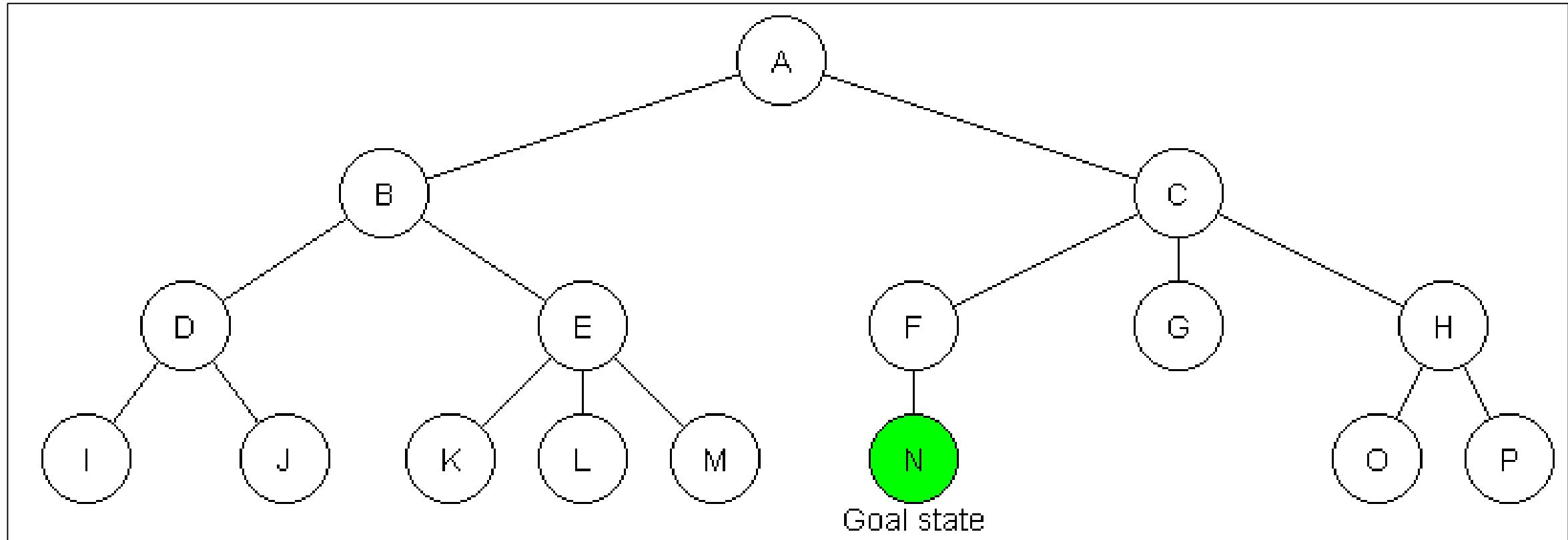
Typical “narrow” tasks include vision, language processing, and planning.



There are many ways to build AI,
including **expert systems** and tree search.



There are many ways to build AI,
including expert systems and **tree search**.

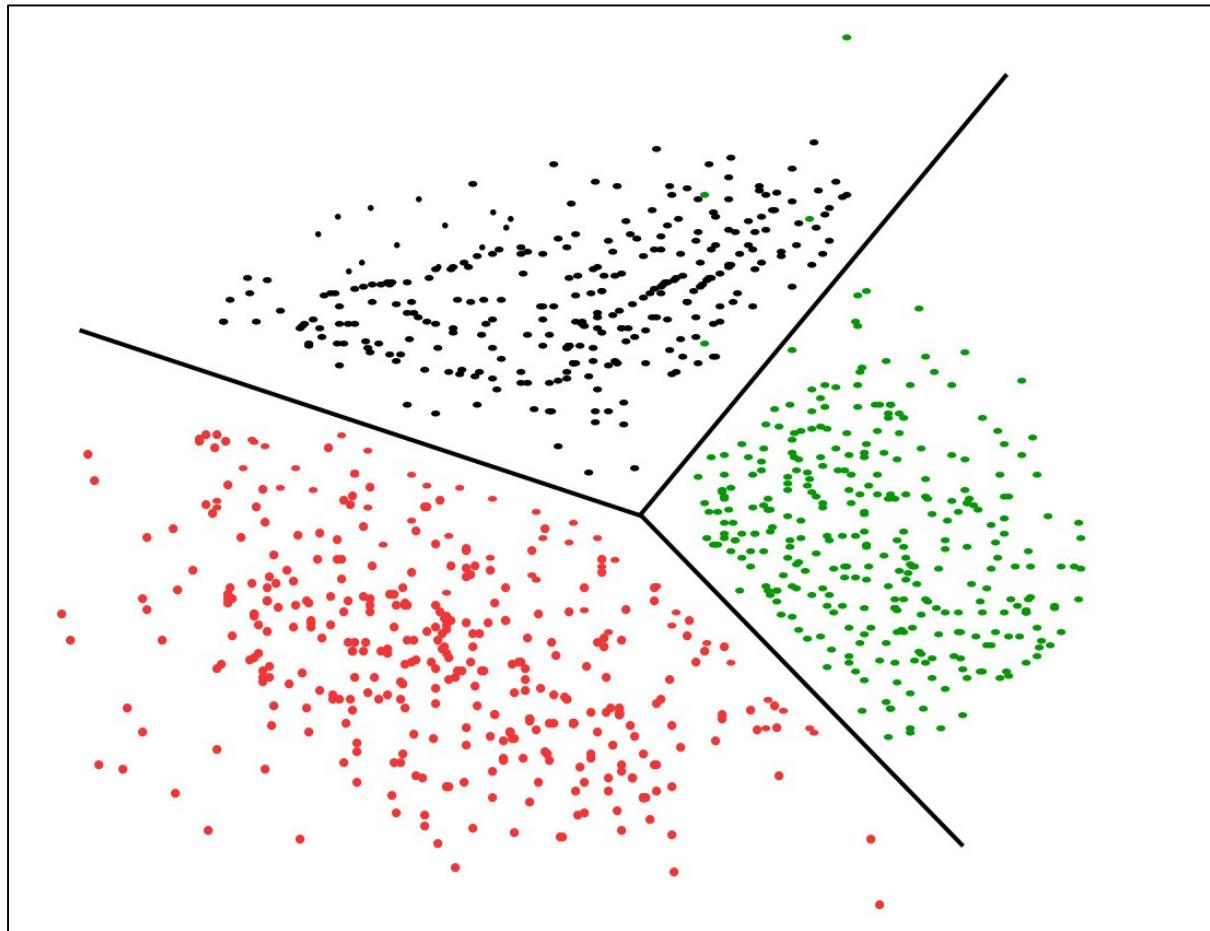


Machine learning can perform many tasks,
i.e. **classification**, clustering, and regression.

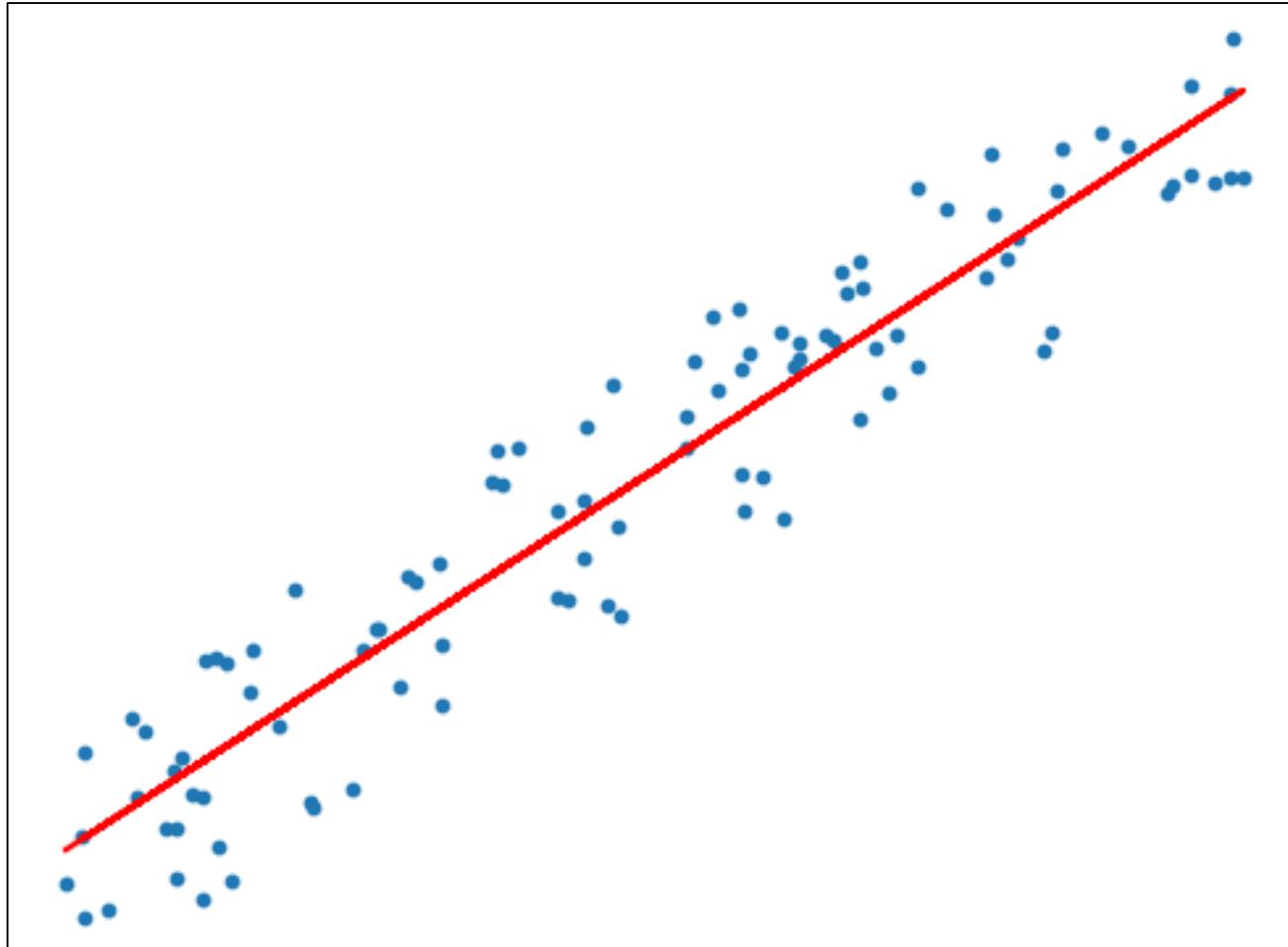


CAT

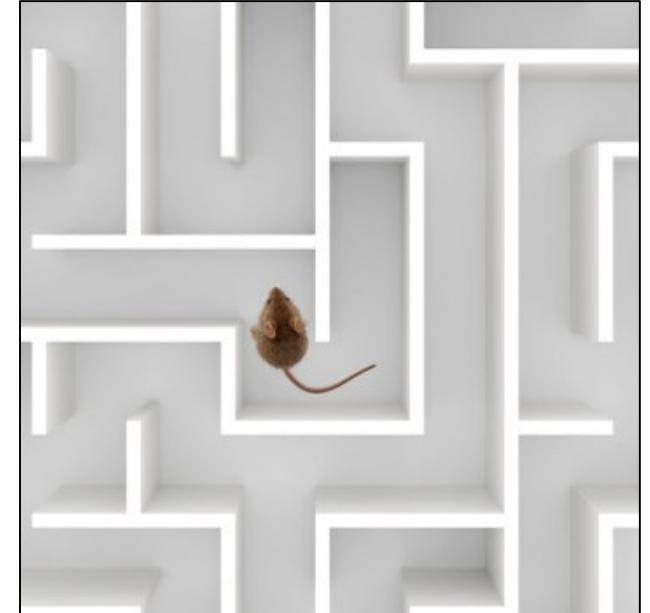
Machine learning can perform many tasks,
i.e. classification, **clustering**, and regression.



Machine learning can perform many tasks,
i.e. classification, clustering, and **regression**.



There are three types of learning: supervised, unsupervised, and reinforcement learning.

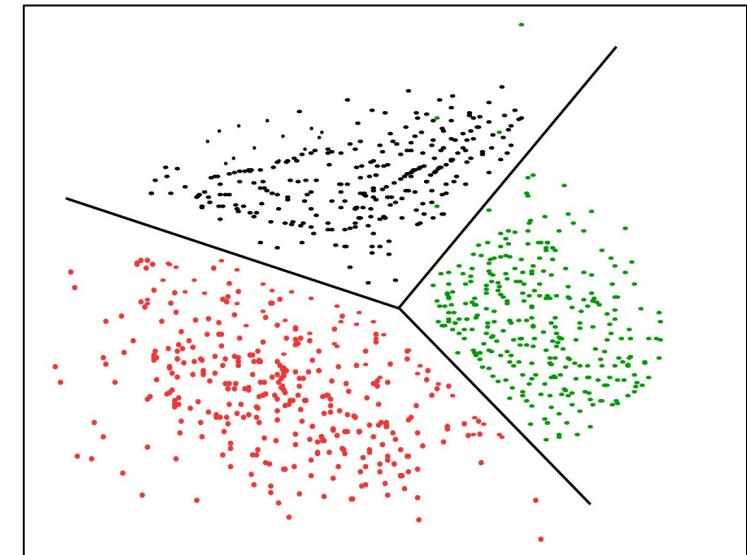


There are three types of learning: supervised, unsupervised, and reinforcement learning.

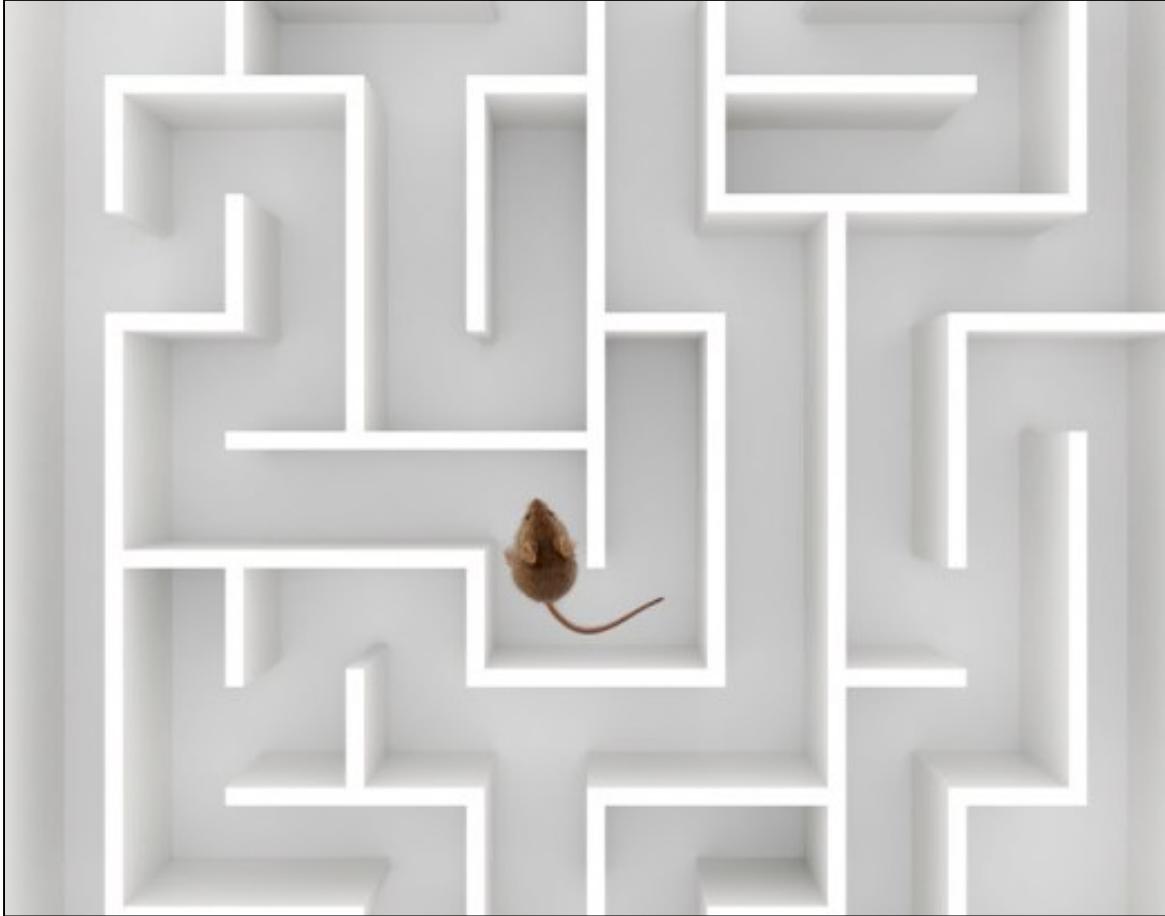


CAT

There are three types of learning: supervised, unsupervised, and reinforcement learning.



There are three types of learning: supervised, unsupervised, and **reinforcement learning**.



With the right data and the right model,
machine learning can solve many problems.

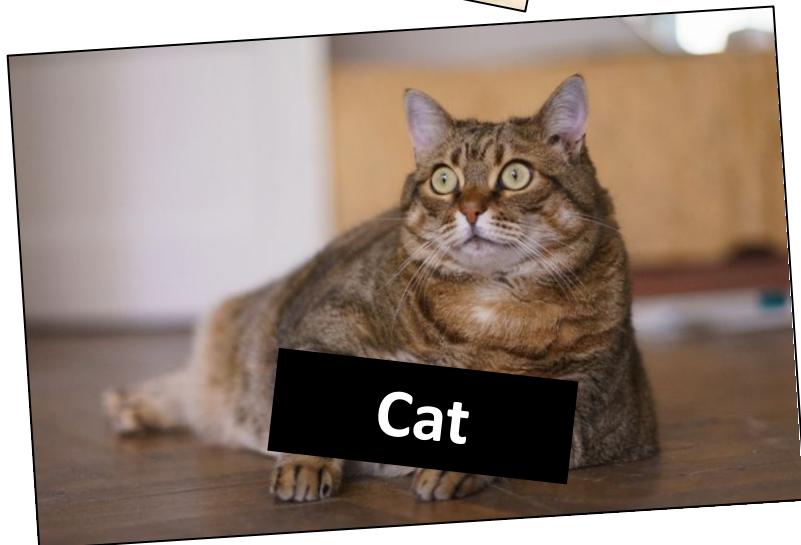
But finding the right data and
training the right model
can be difficult.

1. Goal?
2. Training data?
3. Model?
4. Accuracy?

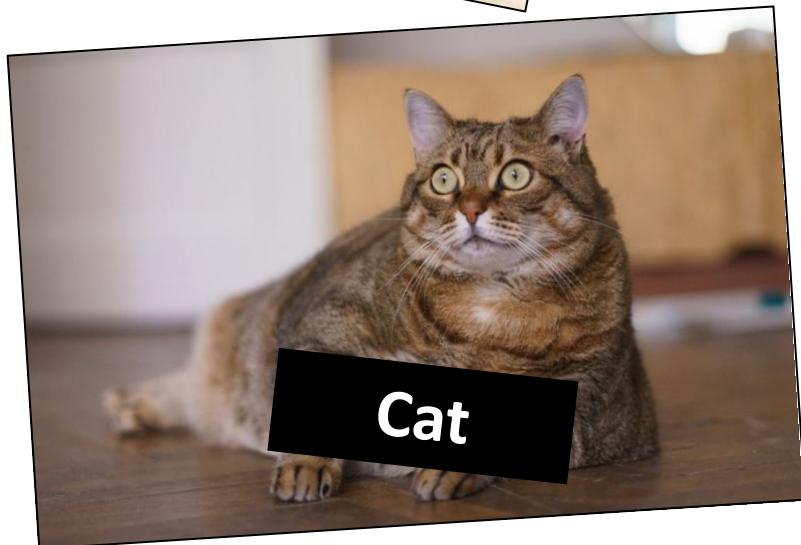
1. Define a problem.



2. Find data.



3. Clean data.



3. Clean data.



4. Choose a model.

Dogs

Always

Sometimes

Cats

Always

Sometimes

5. Train the model.

Cat



5. Train the model.

Cat



5. Train the model.

Dog



5. Train the model.

Dog

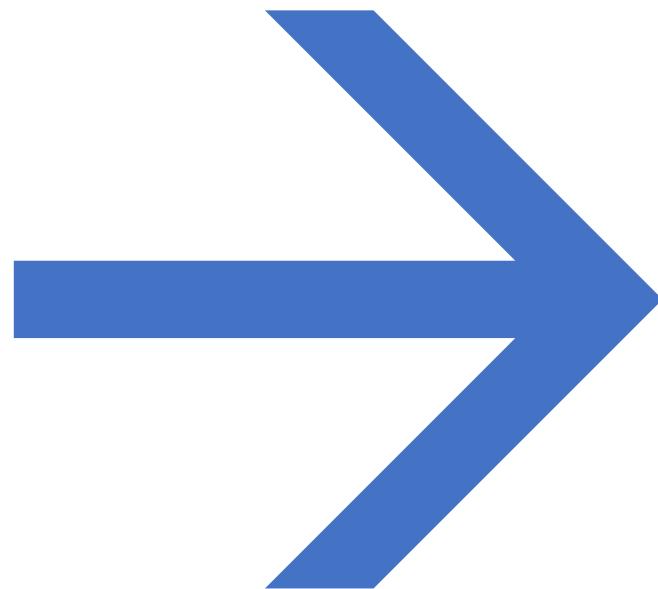


6. Test the model.

Cat



7. Deploy the model.



1. Define a problem.



3. Clean data.



4. Choose a model.

Dogs

Always

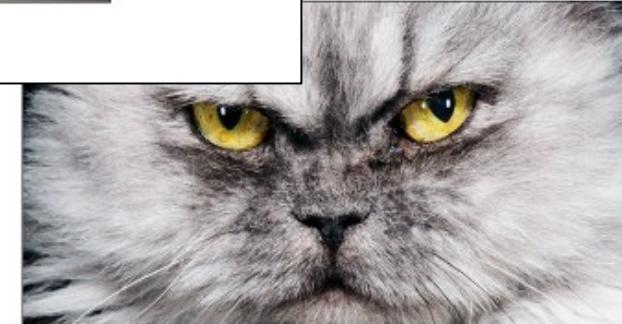
Sometimes

Cat



Sometimes

Cat



Dog



Dog