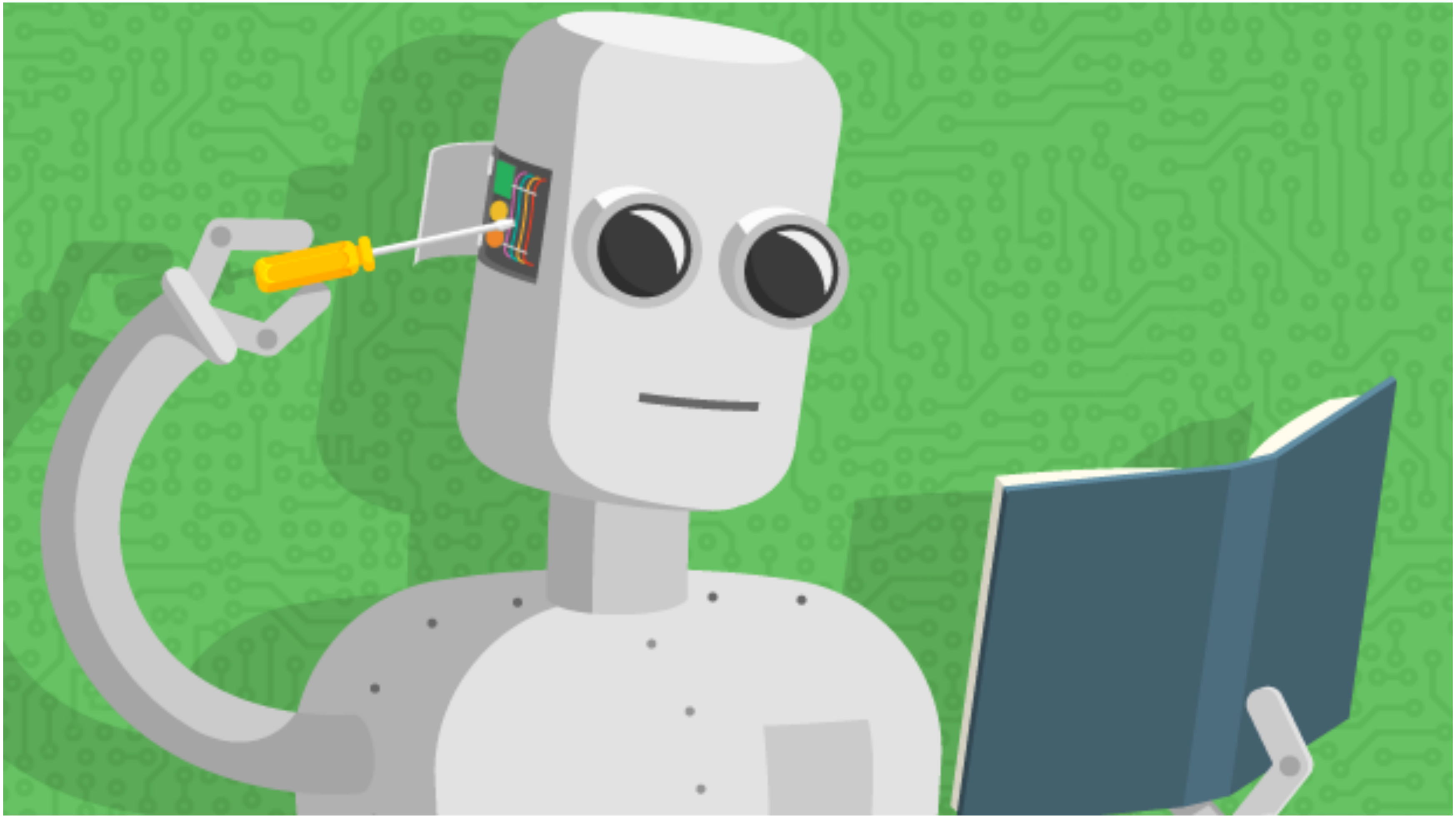
Machine Learning overview

Chapter 18, 21



What is learning?

- Learning denotes changes in a system that ... enable a system to do the same task more efficiently the next time – Herbert Simon
- Learning is constructing or modifying representations of what is being experienced – Ryszard Michalski
- Learning is making useful changes in our minds – Marvin Minsky

Why study learning?

- Understand and improve efficiency of **human learning**
  - Use to improve methods for teaching and tutoring people (e.g., better computer-aided instruction)
- **Discover** new things or structure previously unknown
  - Examples: data mining, scientific discovery
- Fill in skeletal or **incomplete specifications in** a domain
  - Large, complex systems can't be completely built by hand & require dynamic updating to incorporate new information
  - Learning new characteristics expands the domain or expertise and lessens the "brittleness" of the system
- Build agents that can **adapt** to users, other agents, and their environment

AI & Learning Today

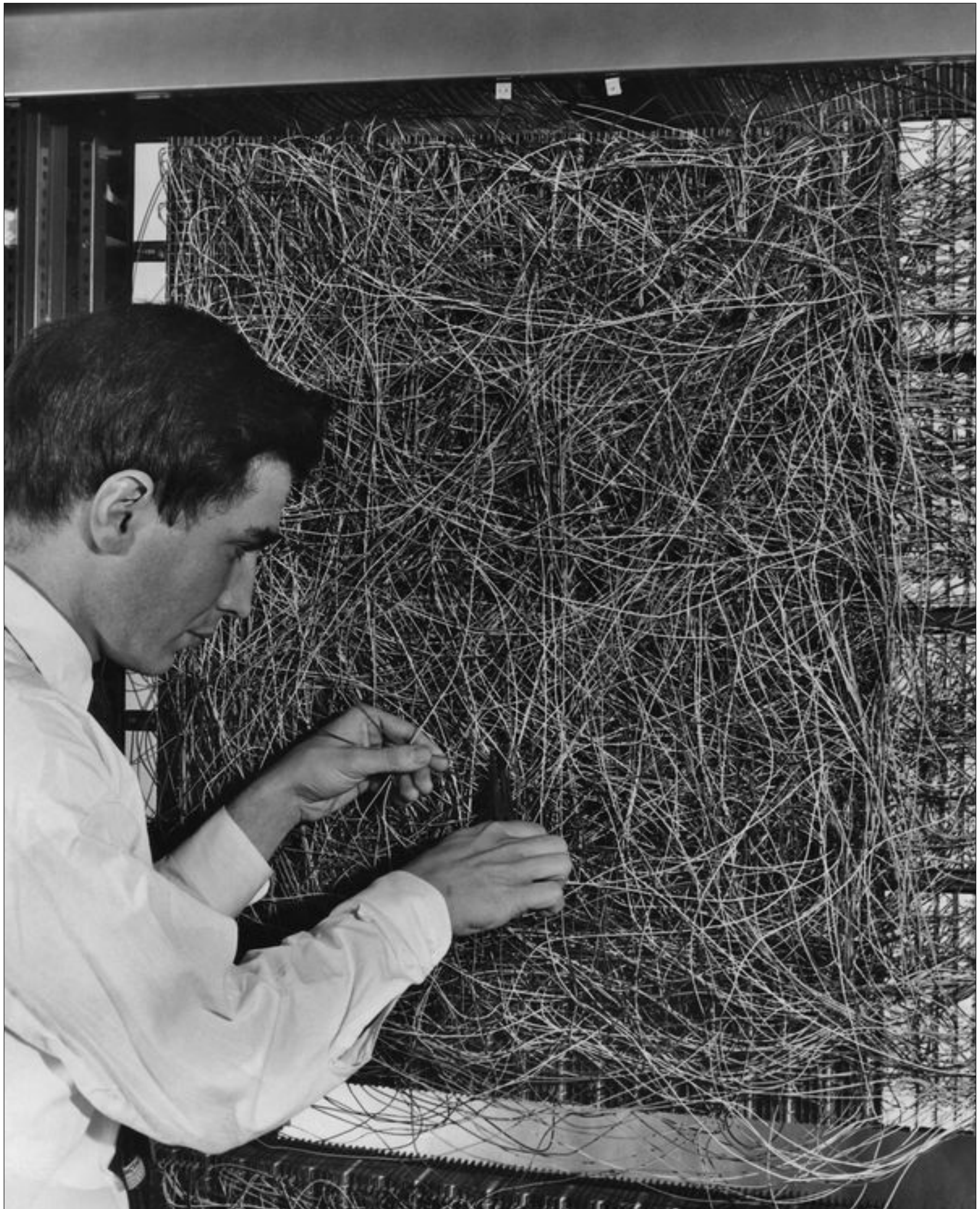
- 50s&60s: neural network learning popular

  Marvin Minsky did neural networks for his dissertation

- Mid 60s: replaced by paradigm of manually encoding & using symbolic knowledge

  Cf. Perceptrons, Minsky & Papert book showing limitations of perceptron model of neural networks

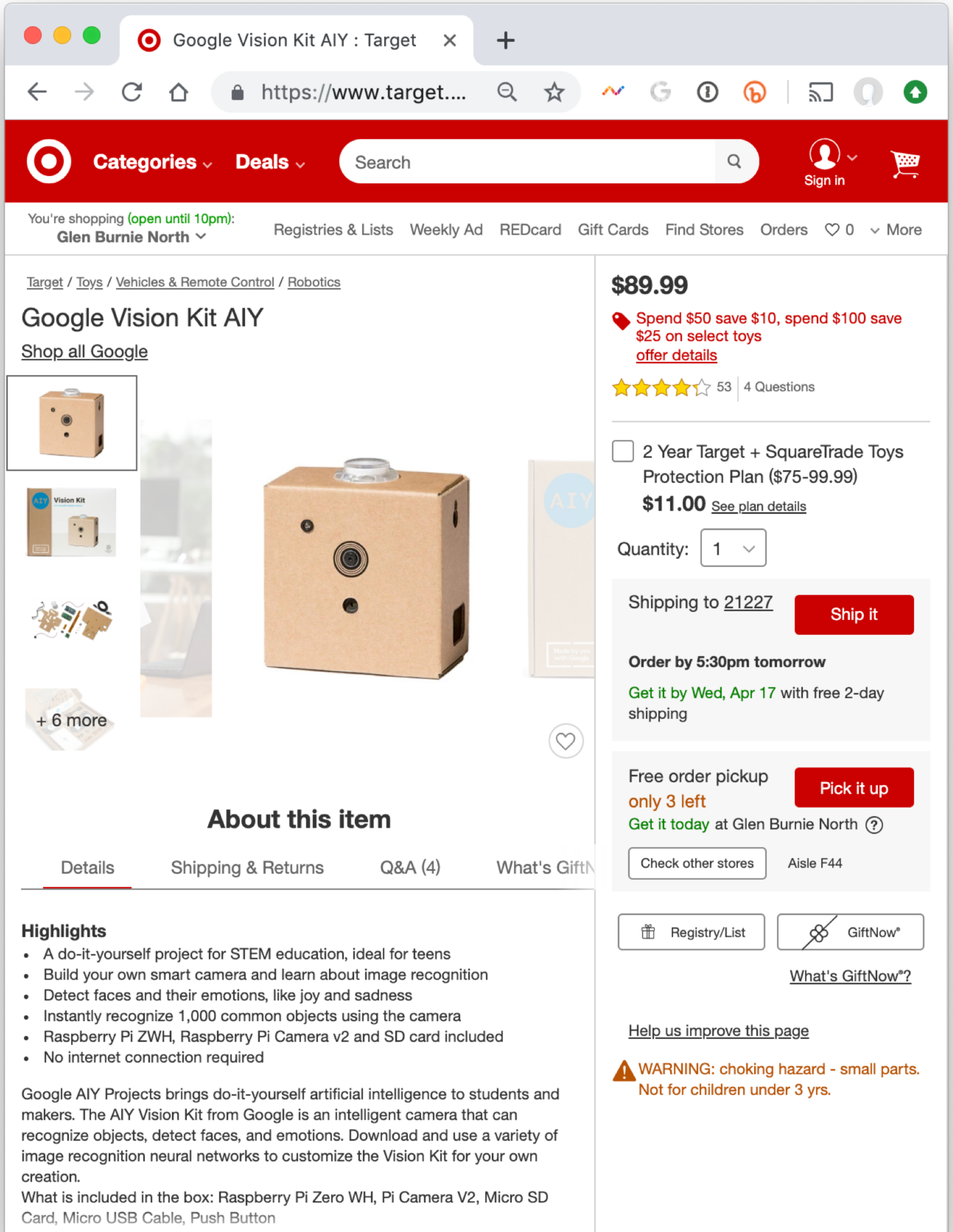
- 90s: more data and the Web drove interest in new statistical machine learning (ML) techniques and new data mining applications
- Now: ML techniques & big data play biggest driver in almost all successful intelligent systems



Neural Networks 1960

A man adjusting the random wiring network between the light sensors and association unit of scientist Frank Rosen-blatt's Perceptron, or MARK 1 computer, at the Cornell Aeronautical Laboratory, Buffalo, New York, circa 1960. The machine is designed to use a type of artificial neural network, known as a perceptron.

Neural Networks 2019

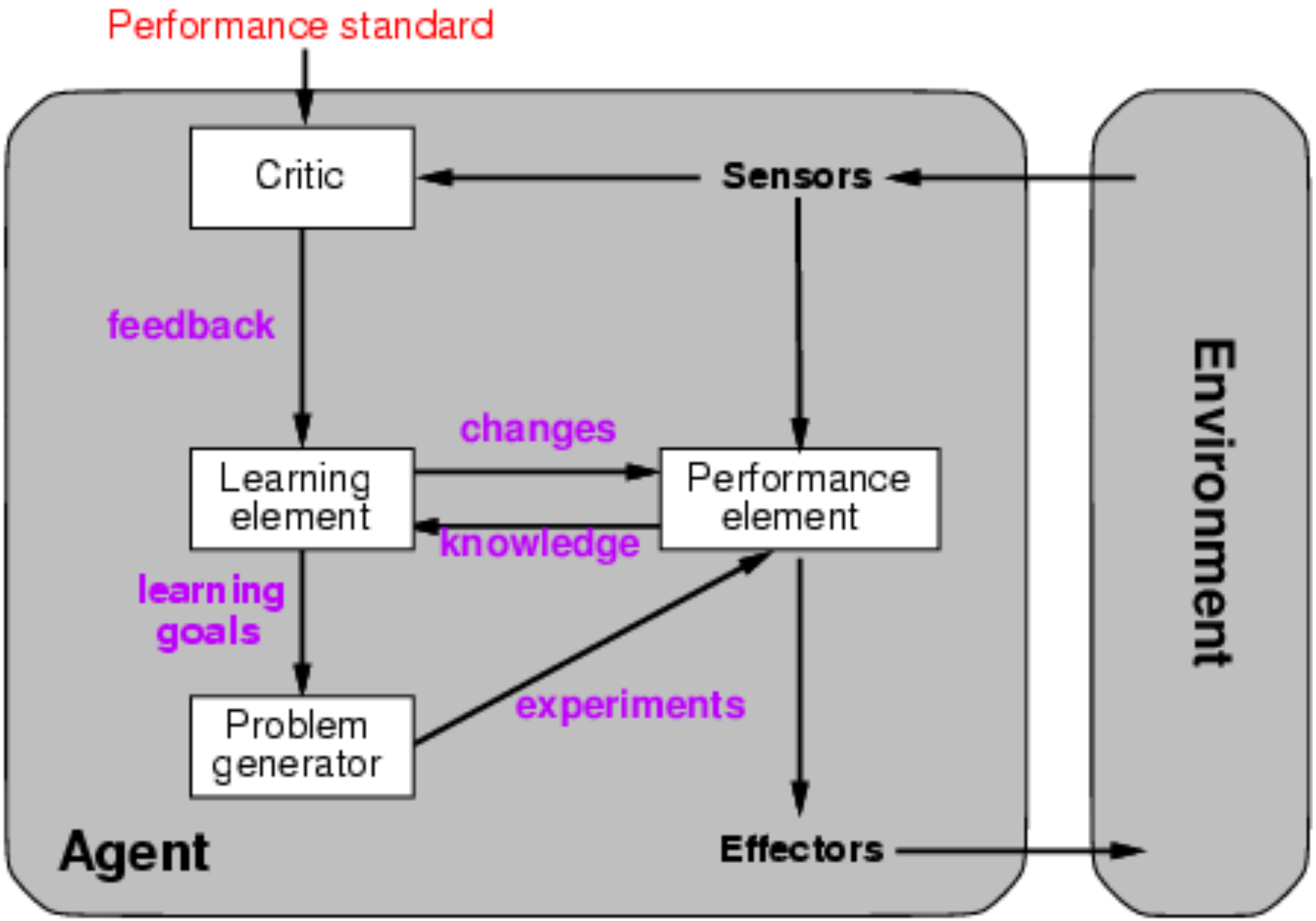


Google's AIY Vision Kit ($89.99 at Target) is an intelligent camera that can recognize objects, detect faces and emotions. Download and use a variety of image recognition neural networks to customize the Vision Kit for your own creation. Included in the box: Raspberry Pi Zero WH, Pi Camera V2, Micro SD Card, Micro USB Cable, Push Button.

Machine Leaning Successes

- Games: chess, go, poker
- Text: sentiment analysis
- Email: spam detection
- Machine translation
- Spoken language understanding
- named entity detection
- Autonomous vehicles
- Individual face recognition
- Understanding digital images
- Recommender systems (Netflix, Amazon)
- Credit card fraud detection
- Showing annoying ads

A general model of learning agents



The Big Idea and Terminology

Given some data, learn a model of how the world works that lets you predict new data

- **Training Set:** Data from which you learn initially
- **Model:** What you learn; a "model" of how inputs are associated with outputs
- **Test set:** New data you test your model against
- **Corpus:** A body of data (pl.: corpora)
- **Representation:** The computational expression of data

Major Machine learning paradigms (1)

- **Rote learning**: 1-1 mapping from inputs to stored representation, learning by memorization, association-based storage & retrieval
- **Induction:** Use specific examples to reach general conclusions
- **Clustering**: Unsupervised discovery of natural groups in data

Major Machine learning paradigms (2)

- **Analogy:** Find correspondence between different representations
- **Discovery**: Unsupervised, specific goal not given
- **Genetic algorithms:** *Evolutionary* search techniques, based on an analogy to *survival of the fittest*
- **Reinforcement** – Feedback (positive or negative reward) given at the end of a sequence of steps

What we will and won't cover

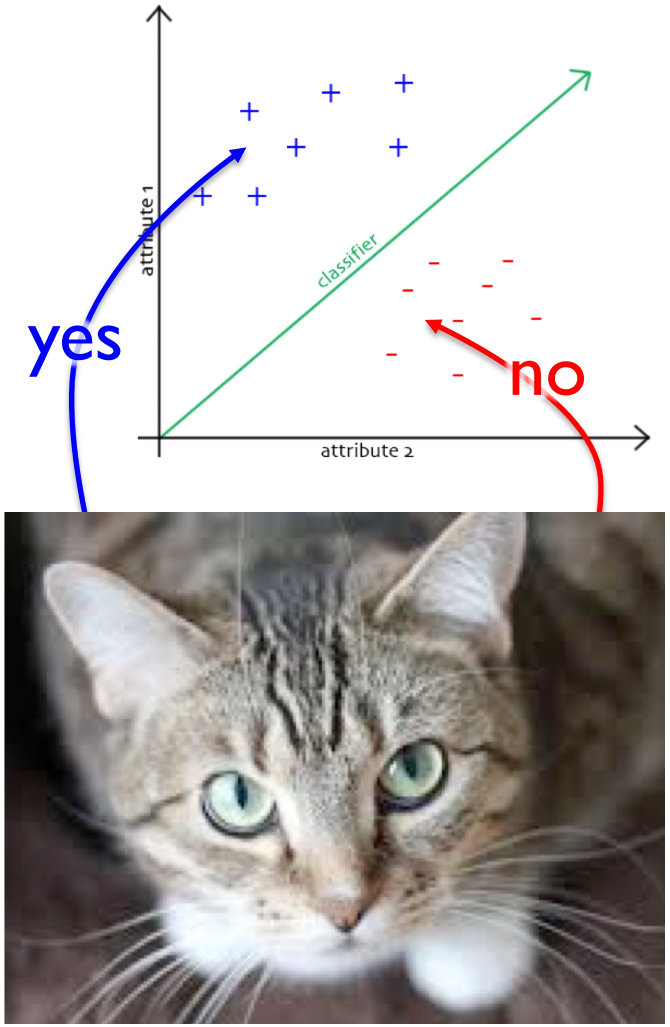
- We'll look at a few popular machine learning problems and algorithms
  - Take CMSC 478/678 Machine Leaning for more
  - Use online resources & experiment on your own
- We'll focus on when/how to use techniques and only touch on how/why they work
- We'll cover basic methodology and evaluation
- We'll use various platform for examples & demos (e.g., scikit-learn, Weka)
  - Great for exploration and learning

Types of learning problems

- **Supervised**: learn from training examples
  - Regression
  - Classification: Decision Trees, SVM
- **Unsupervised**: learn w/o training examples
  - Clustering
  - Dimensionality reduction
- **Reinforcement learning:** improve performance using feedback from actions taken
- Lots more we won't cover
  - Hidden Markov models, Learning to rank, Semi-supervised learning, Active learning …

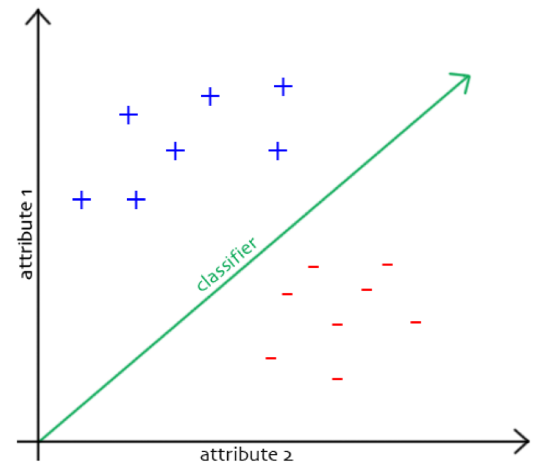
Classification Problem

- Extrapolate from **examples** to make accurate **predictions** about future data points
  - Examples are **training data**
- Predict into **classes**, based on attributes ("**features**")
  - Example: it has tomato sauce, cheese, and no bread. Is it pizza?
  - Example: does this image contain a cat?



Supervised Learning

- Goal: Learn an unknown function $f(X) = Y$, where
  - X is an input example
  - Y is the desired output. (f is the..?)
- **Supervised learning:** given a training set of (X, Y) pairs by a "teacher"

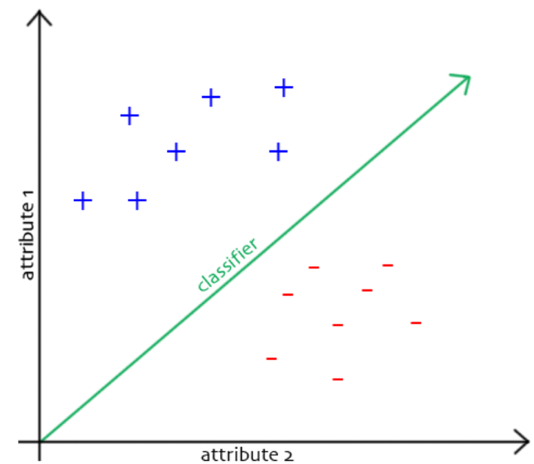


X			*Y*
bread	cheese	tomato sauce	**pizza**
¬ bread	¬ cheese	tomato sauce	**¬ not pizza**
bread	cheese	¬ tomato sauce	**pizza** (gross pizza)
lots more rows…			

"class labels" provided

Unsupervised Learning

- Goal: Learn an unknown function $f(X) = Y$, where
  - X is an input example
  - Y is the desired output. (f is the..?)
- **Unsupervised learning:** only given Xs and some (eventual) feedback



X		
bread	cheese	tomato sauce
¬ bread	¬ cheese	tomato sauce
bread	cheese	¬ tomato sauce
lots more rows…		

<u>I think:</u>
pizza,
¬ pizza,
¬ pizza

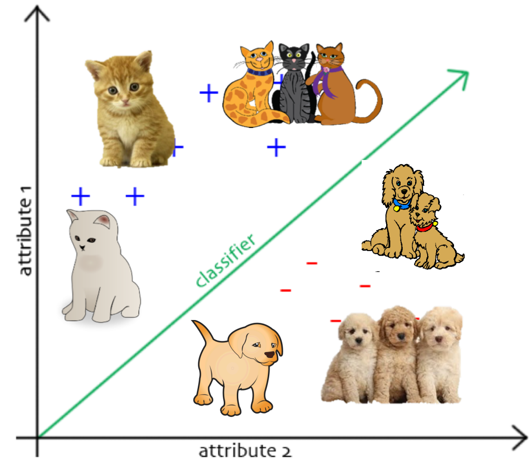
67%
right

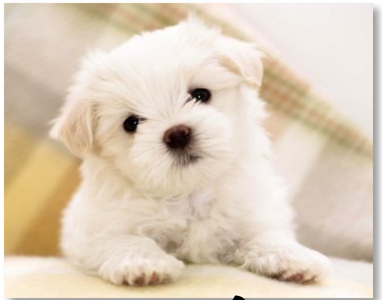
Classification

- Classification or concept learning (aka "induction")

  Given a set of examples of some concept/class/category:

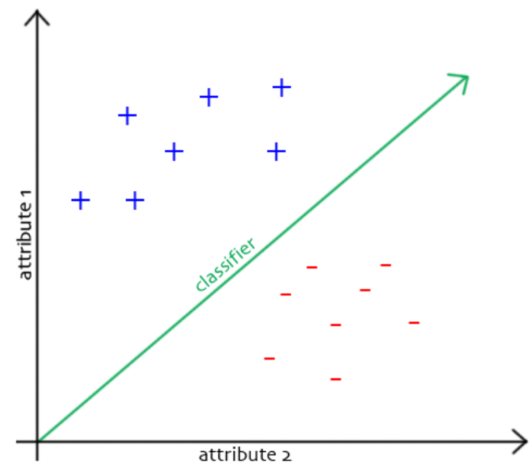
  1. Determine if given example is an instance of concept (class member)
  2. If it is: positive example
  3. If it is not: negative example
  4. Or we can make a probabilistic prediction (e.g., using a Bayes net)





cat?

Supervised Concept Learning



- Given a training set of positive and negative examples of a concept
- Construct a description (model) that will accurately classify whether future examples are positive or negative
- I.e., learn estimate of function f given a training set:

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

where each y_i is either + (positive) or - (negative), or a probability distribution over +/-

Machine Learning Problems

	Supervised Learning	***Unsupervised Learning***
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

Supervised learning

- Given training examples of inputs & corres-ponding outputs, produce "correct" outputs for new inputs
- Two important scenarios:
  - **Classification:** outputs typically labels (goodRisk, badRisk); learn a decision boundary that separates classes
  - **Regression:** aka "curve fitting" or "function approximation." Learn a *continuous* input-output mapping from examples, e.g. for a given zip code, predict house sale price given its square footage

Unsupervised Learning

Given only *unlabeled* data as input, learn some sort of structure, e.g.:

- **Clustering**: group Facebook friends based on similarity of posts and friends
- **Embeddings**: Find sets of words whose meanings are related (e.g., doctor, hospital)
- **Topic modelling**: Induce N topics and words most common in documents about each

Inductive Learning Framework

- Raw input data from sensors preprocessed to obtain **feature vector**, **X**, of **relevant** features for classifying examples
- Each **X** is a list of (attribute, value) pairs
- ***n*** attributes (a.k.a. features): fixed, positive, and finite
- Features have fixed, finite number # of possible values
  - Or continuous within some well-defined space, e.g., "age"
- Each example is a point in an *n*-dimensional feature space
  - X = [Person:Sue, EyeColor:Brown, Age:Young, Sex:Female]
  - X = [Cheese:*f*, Sauce:*t*, Bread:*t*]
  - X = [Texture:Fuzzy, Ears:Pointy, Purrs:Yes, Legs:4]

Inductive Learning as Search

- Instance space, I, is set of all possible examples
  - Defines the language for the training and test instances
  - Usually each instance $i \in I$ is a feature vector
  - Features are also sometimes called attributes or variables

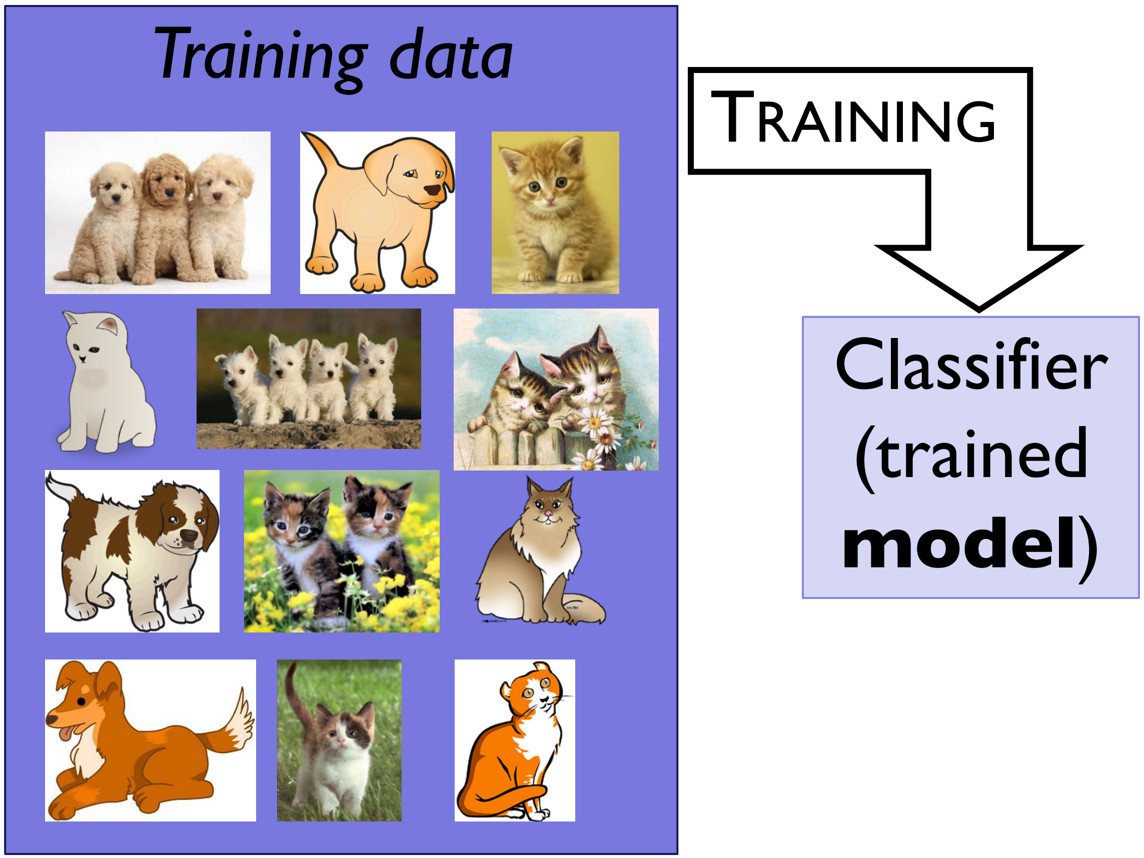
    $I: V_1 \times V_2 \times \ldots \times V_k, \; i = (v_1, v_2, \ldots, v_k)$
- Class variable C gives an instance's class (to be predicted)

Inductive Learning as Search

- C gives an instance's class
- Model space M defines the possible classifiers
  - $M: I \rightarrow C$, $M = \{m_1, \ldots m_n\}$ (possibly infinite)
  - Model space is sometimes defined using same features as instance space (not always)
- Training data lets us search for a good (consistent, complete, simple) hypothesis in the model space
- The learned model is a classifier

Inductive Learning Pipeline

Puppy classifier



Inductive Learning Pipeline

Puppy classifier

Training data

Test data

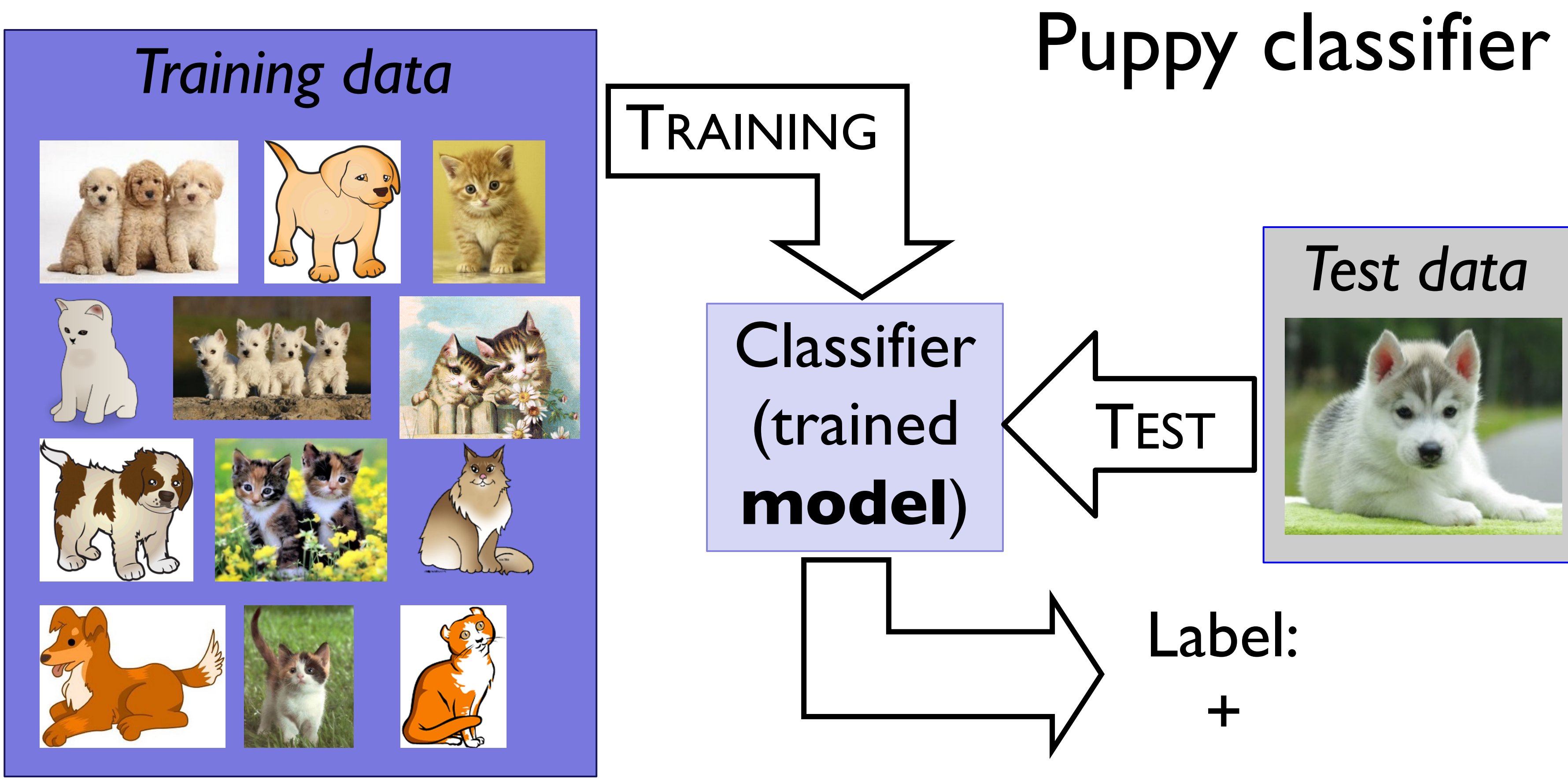
Classifier (trained **model**)

TEST

Label:

+

Inductive Learning Pipeline

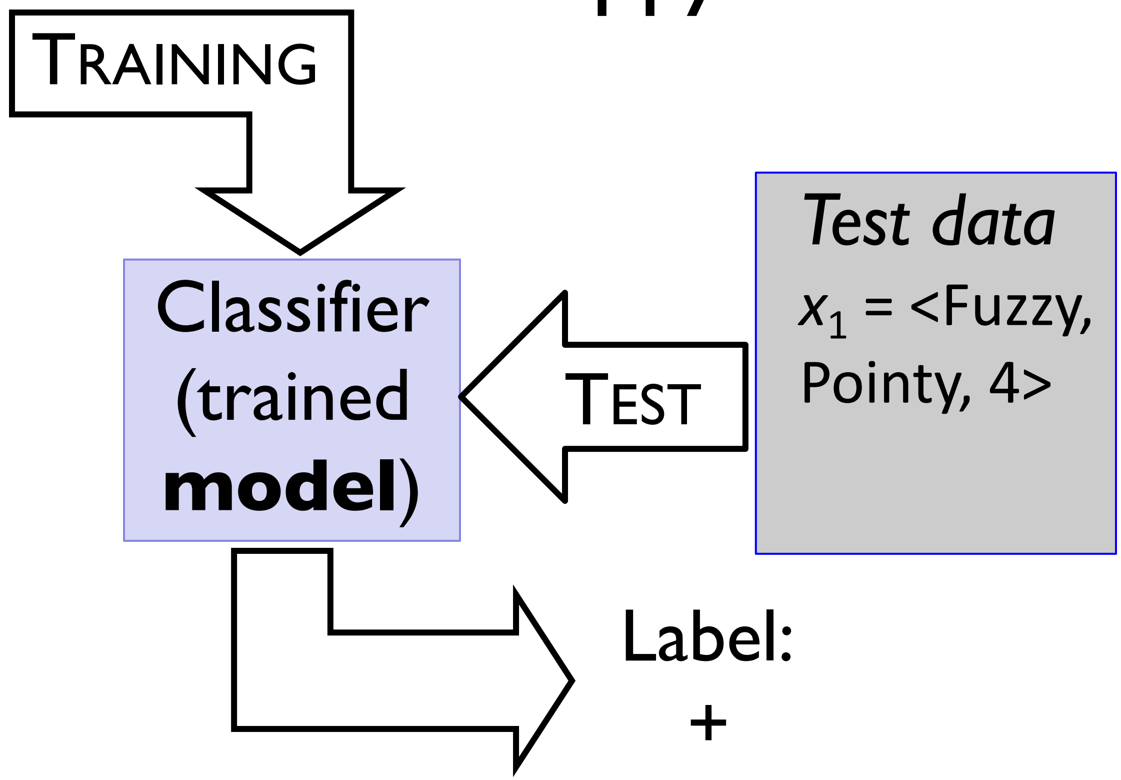


Inductive Learning Pipeline

Puppy classifier

Training data, X

Text-ure	*Ears*	*Legs*	*Class*
Fuzzy	Round	4	+
Slimy	Missing	8	-
Fuzzy	Pointy	4	-
Fuzzy	Round	4	+
Fuzzy	Pointy	4	+
…			



Model Spaces (1)

- Decision trees
  - Partition the instance space **I** into axis-parallel regions
  - Labeled with class value
- Nearest-neighbor classifiers
  - Partition the instance space **I** into regions defined by centroid instances (or cluster of *k* instances)
- Bayesian networks
  - Probabilistic dependencies of class on attributes
  - Naïve Bayes: special case of BNs where class → each attribute

Model Spaces (2)

- Neural networks
  - Nonlinear feed-forward functions of attribute values
- Support vector machines
  - Find a separating plane in a high-dimensional feature space
- Associative rules (feature values → class)
- First-order logical rules

Machine Learning

- ML's significance in AI has gone up and down over the last 75 years
  - Today it's **very** important for AI and data science
- Driving ML are three trends:
  - Cheaper and more powerful computing systems
  - Open-source ML tools (e.g., scikit-learn, TensorFlow)
  - Availability of large amounts of data
- Understanding ML concepts and tools allow many to use them with success