# Lecture 15: Supervised learning

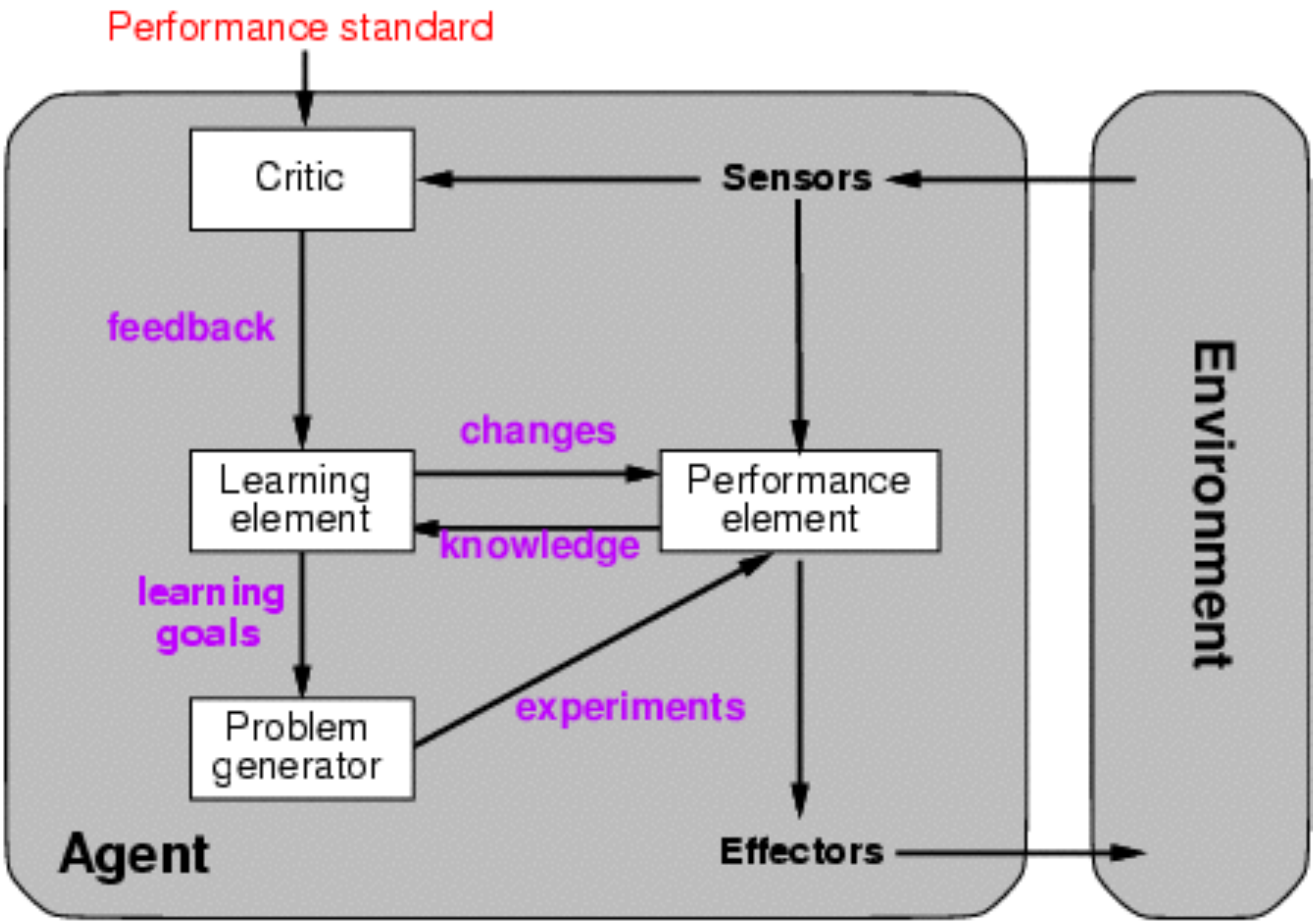
Artificial Intelligence
CS-GY-6613

Julian Togelius
julian.togelius@nyu.edu

# Why learning?

- So far in the course: we have specified the mechanism by which the agent should decide how to act
- So far in the course: the world is mostly known
- The agent might not know what the world is like, or what policies work well in the world
- The world may change
- You don't want to do all the programming

# A (complex) learning agent

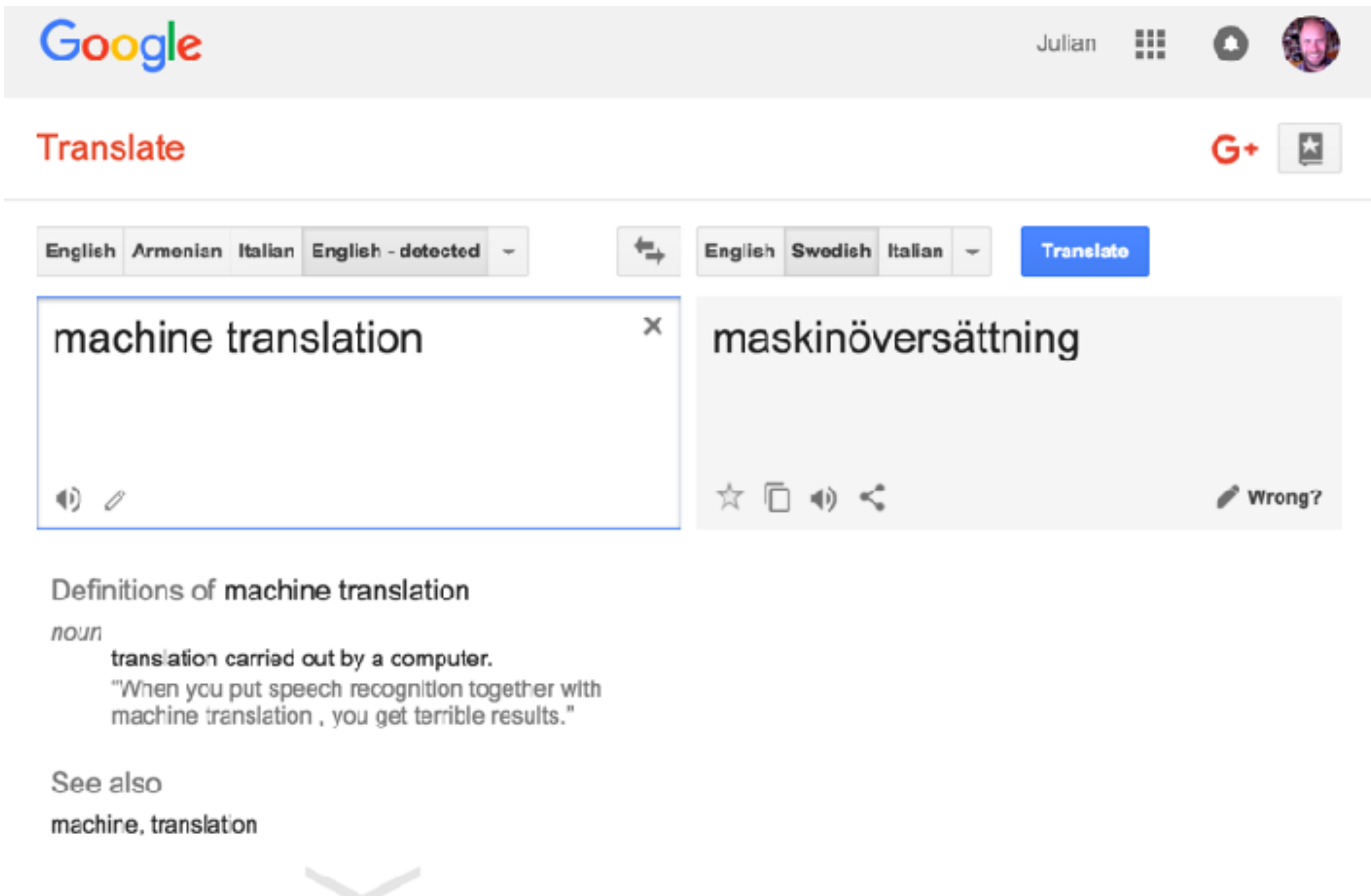


# What could be learned?

# What could be learned?

- How to drive from point A to point B, without hitting pedestrians
- What a human (or a cat, or bush) looks like
- Hand gestures
- How to drive in the style of a particular human (or according to that human's preferences)
- Which routes from A to B are actually fastest
- How far back you want your seat, temperature for the AC, favorite radio channel…
- Estimating distances

# What could be learned?

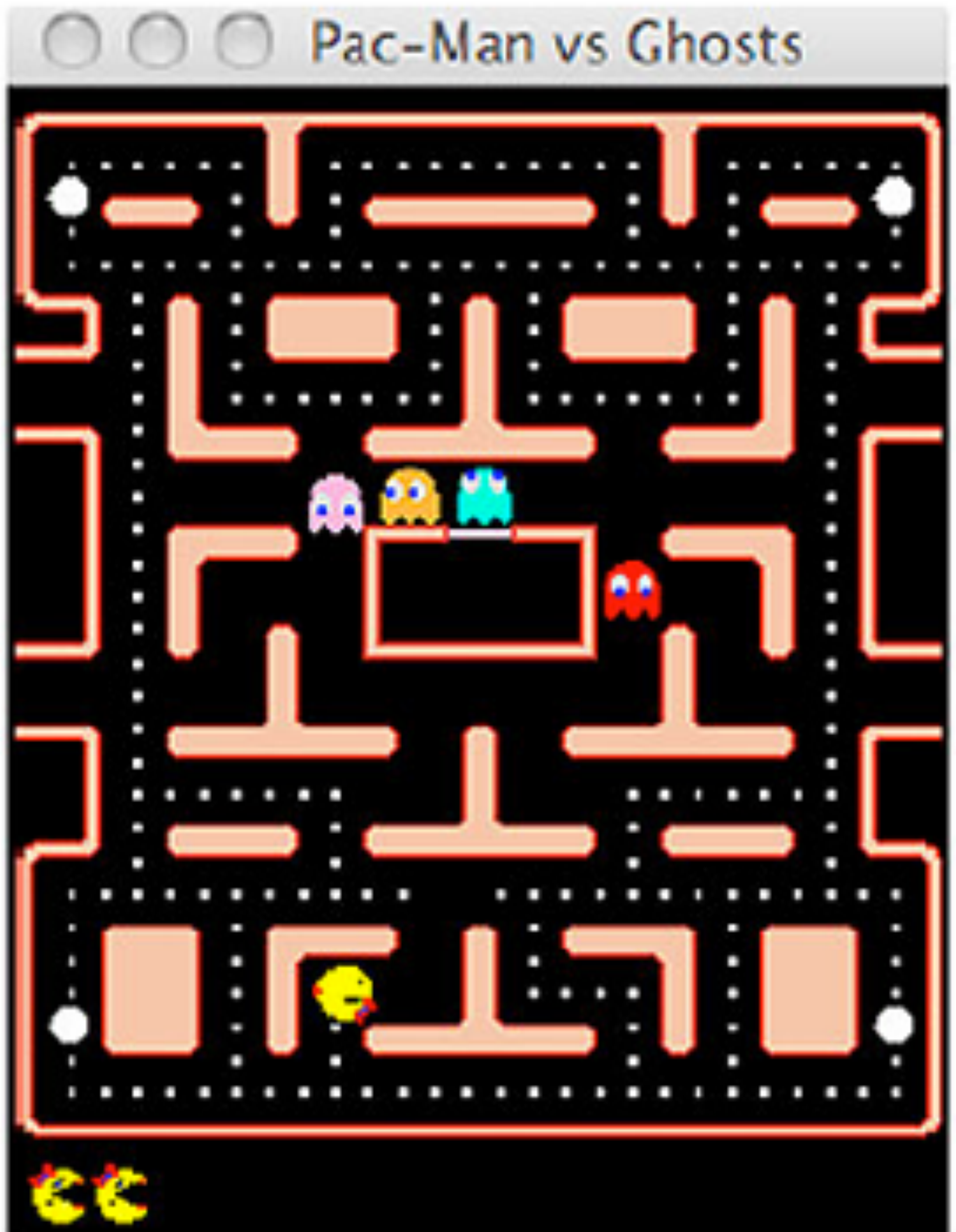


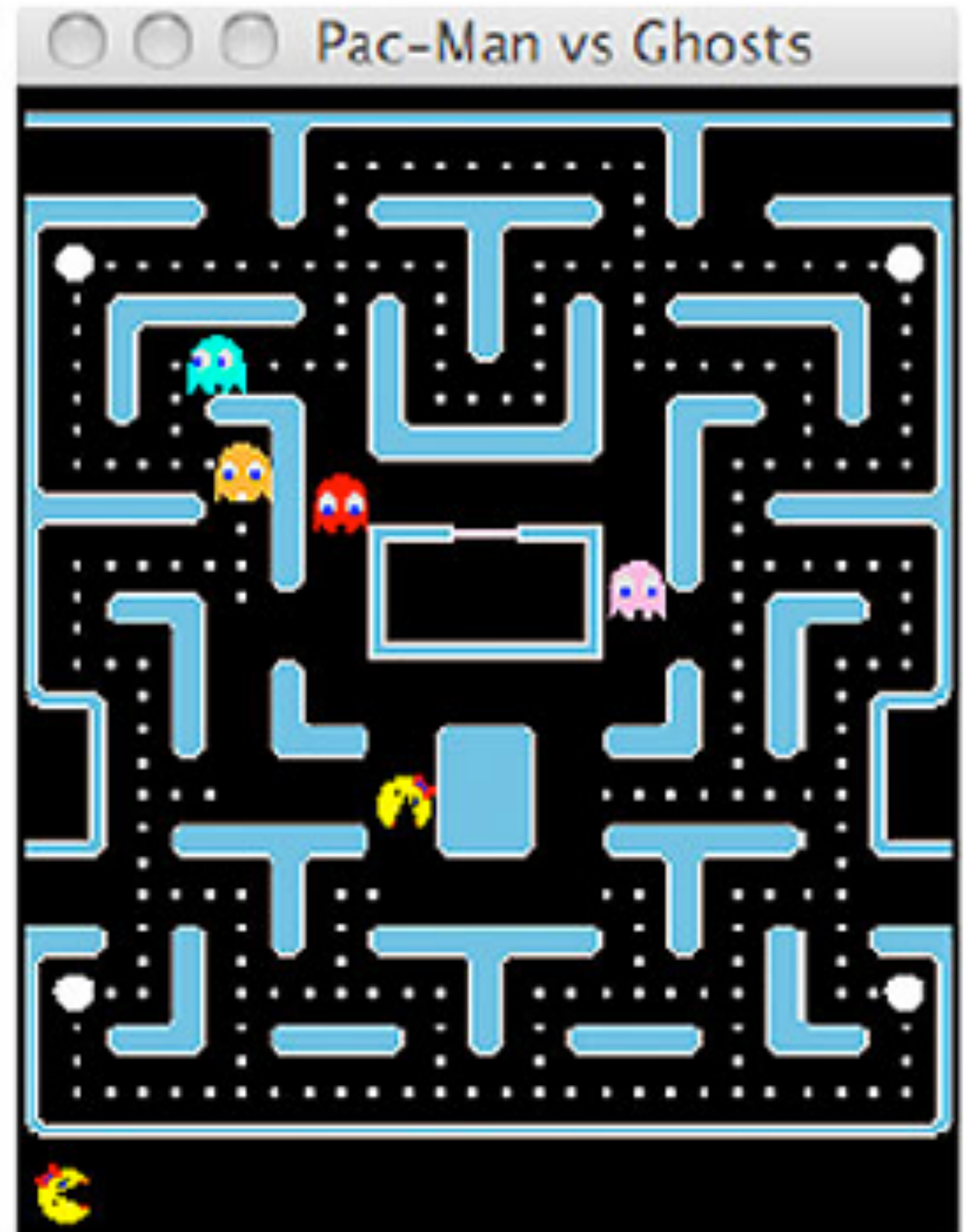
# What could be learned?

# What could be learned?

- How much a board position is worth
- What action to take in a specific situation
- What action a particular person would take in a specific situation
- Who is likely to win a game between two people, and how long it takes

# What could be learned?



# What could be learned?

- What actions a ghost would take
- The value of a state
- Dangerous positions
- What action to take in a specific situation

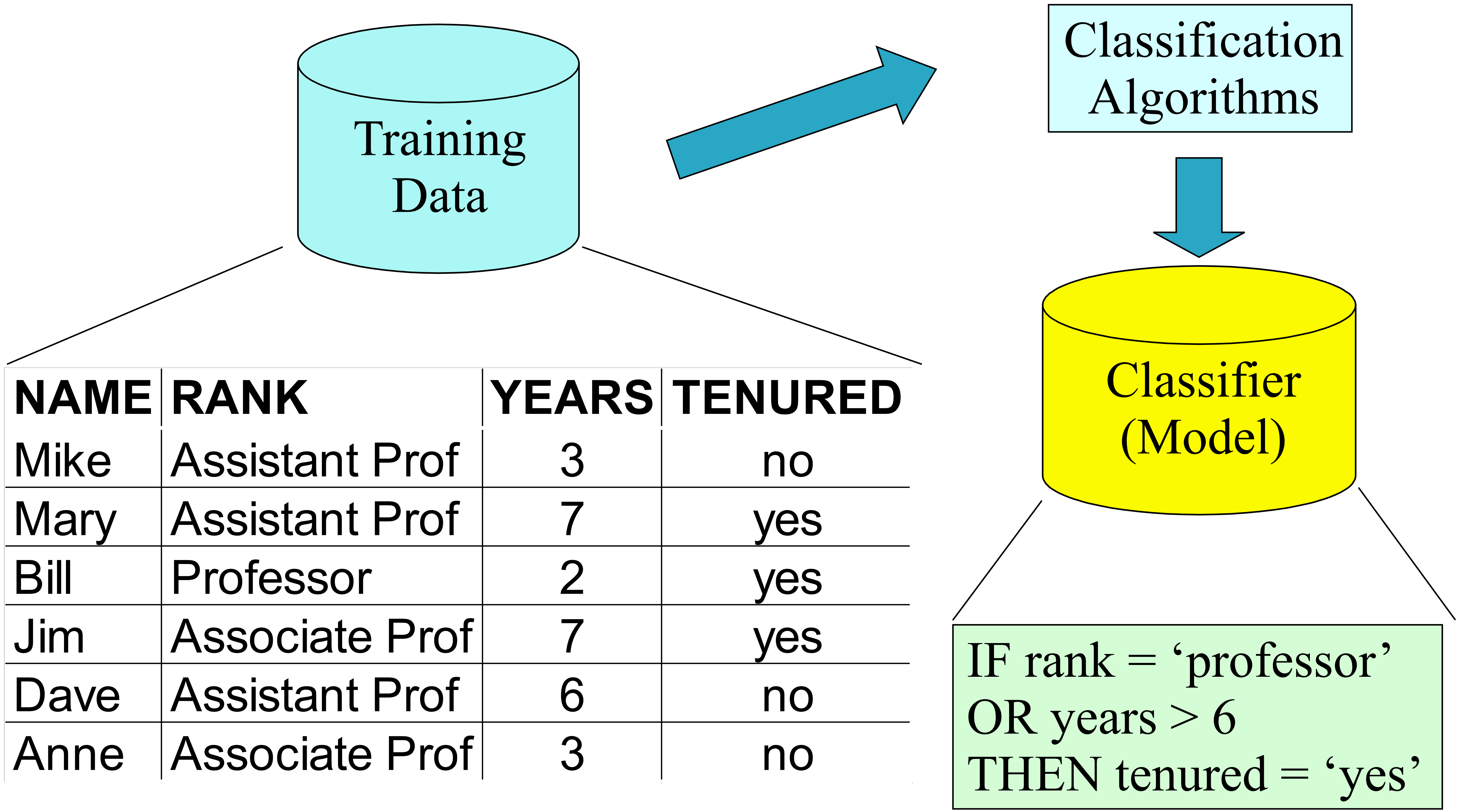
# Types of learning

- **Supervised learning**
  Learning to predict or classify labels based on labeled input data

- **Unsupervised learning**
  Finding patterns in unlabeled data

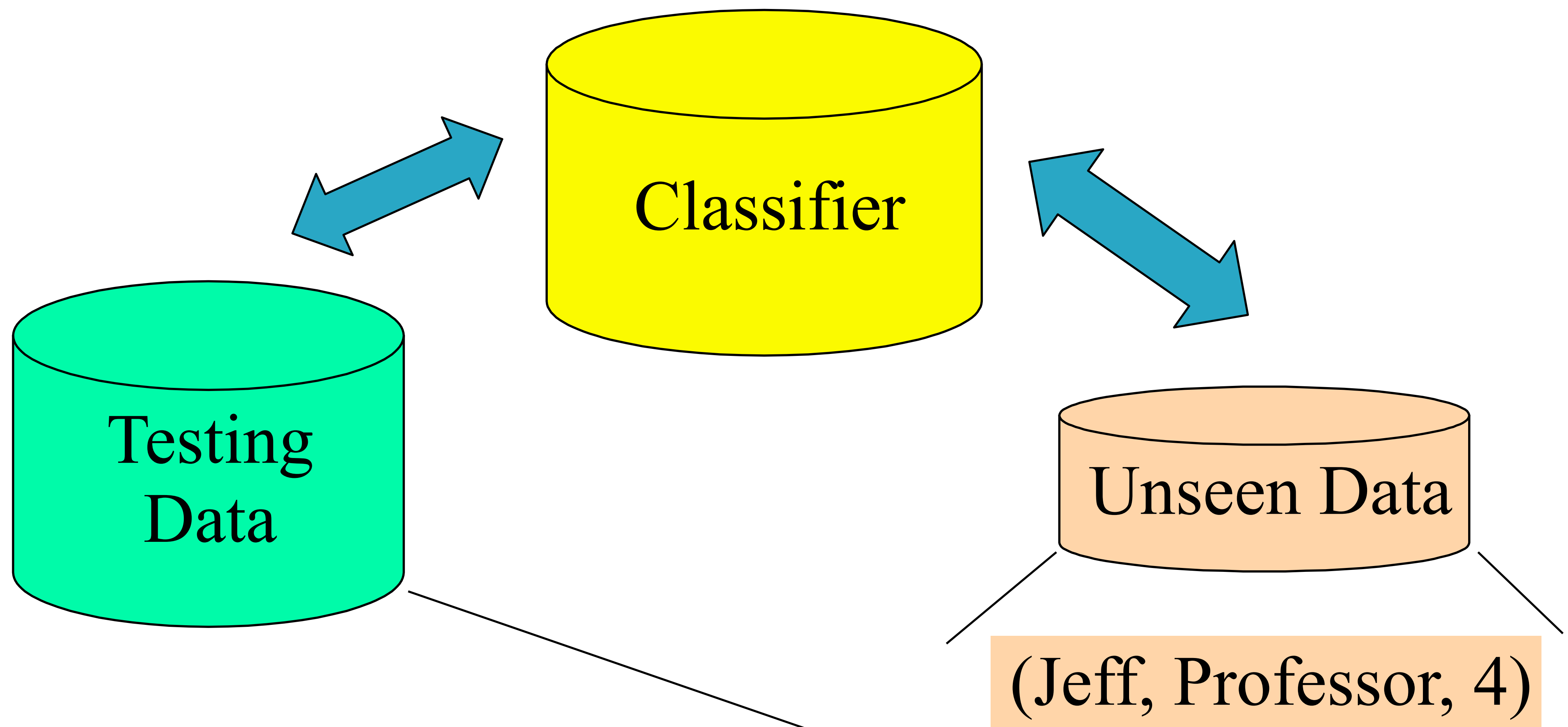
- **Reinforcement learning**
  Learning well-performing behavior from state observations and rewards

# Model construction




| NAME | RANK | YEARS | TENURED |
|---|---|---|---|
| Mike | Assistant Prof | 3 | no |
| Mary | Assistant Prof | 7 | yes |
| Bill | Professor | 2 | yes |
| Jim | Associate Prof | 7 | yes |
| Dave | Assistant Prof | 6 | no |
| Anne | Associate Prof | 3 | no |

# Using the model




| NAME | RANK | YEARS | TENURED |
|---|---|---|---|
| Tom | Assistant Prof | 2 | no |
| Merlisa | Associate Prof | 7 | no |
| George | Professor | 5 | yes |
| Joseph | Assistant Prof | 7 | yes |

Tenured?

Yes

# Classification vs prediction

- Classification: binary or nominal labels
    - Examples: pregnant or not, from which country, which type of road sign
- Prediction: continuous labels
    - Examples: future stock price, life expectancy, distance to obstacle

# Terminology (supervised learning)

- Each line of data: instance / data point / tuple
- The features of each instance: features / attributes
- That which should be learned: labels / targets
- Each instance has features and a label
- We train on the training set…
- …and test on the testing set

# What's desirable?

- Accuracy
  classifier accuracy: predicting class label
  predictor accuracy: guessing value of predicted attributes
- Speed
  time to construct the model (training time)
  time to use the model (classification/prediction time)
- Robustness: handling noise and missing values
- Interpretability
- Other measures, e.g., goodness of rules, such as decision tree size or compactness of classification rules

# Lazy vs Eager learning

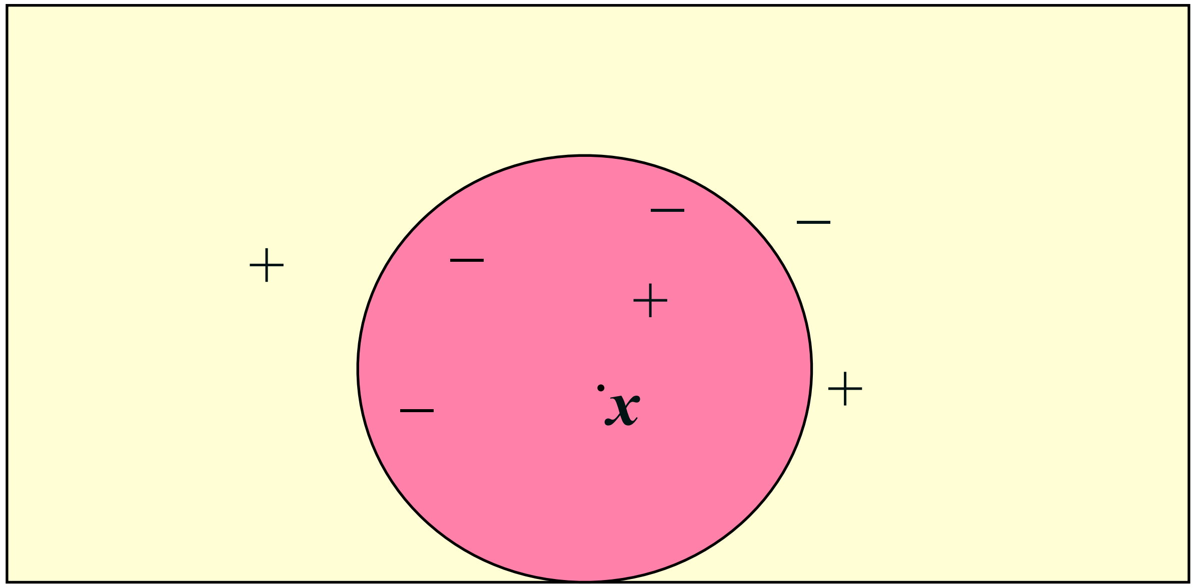
- Lazy learning: Simply stores training data (or only minor processing) and waits until it is given a test tuple
- Eager learning: Given a training set, constructs a classification model (smaller than the data) before receiving new data to classify
- Lazy: less time in training but more time in predicting

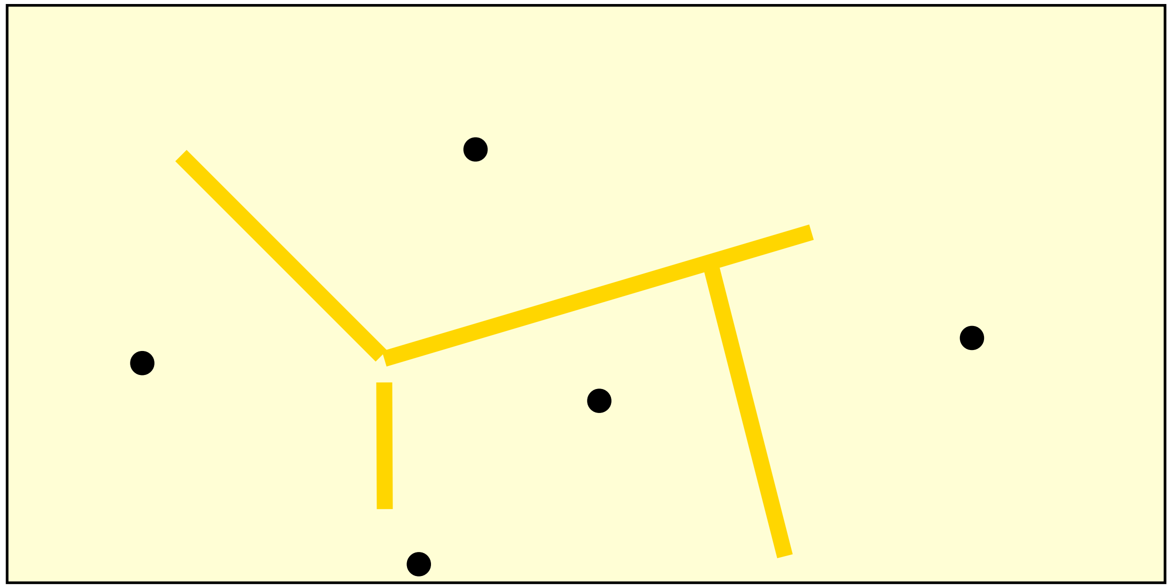
What’s the simplest imaginable working classifier?

# k-Nearest Neighbor Classification

- Simply look at the k instances in the training data which are closest to the instance you want to classify

- Choose the median/mean/mode of those values



# k-Nearest Neighbor Classification

- All instances correspond to points in the n-D space
- The nearest neighbor is defined in terms of Euclidean distance, dist($X_1$, $X_2$)
- Target function could be discrete- or real-valued
- For discrete-valued, k-NN returns the most common value among the k training examples nearest to xq
- Vonoroi diagram: the decision surface induced by 1-NN for a typical set of training examples

# k-Nearest Neighbor Classification

- Distance-weighted nearest neighbor algorithm: Weigh the contribution of each of the k neighbors according to their distance to the query $x_q$, and give greater weight to closer neighbors $w \equiv \frac{1}{d(x_q, x_i)^2}$
- Robust to noisy data by averaging k-nearest neighbors
- *Curse of dimensionality:* distance between neighbors could be dominated by irrelevant attributes
- To overcome it, stretch or shrink axes or eliminate the least relevant attributes

# Distances

- Euclidean distance for continuous attributes

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \ldots + |x_{i_p} - x_{j_p}|^2)}$$

- Hamming distance for binary/nominal attributes: how many of the attributes differ