Machine Learning

A Brief Introduction

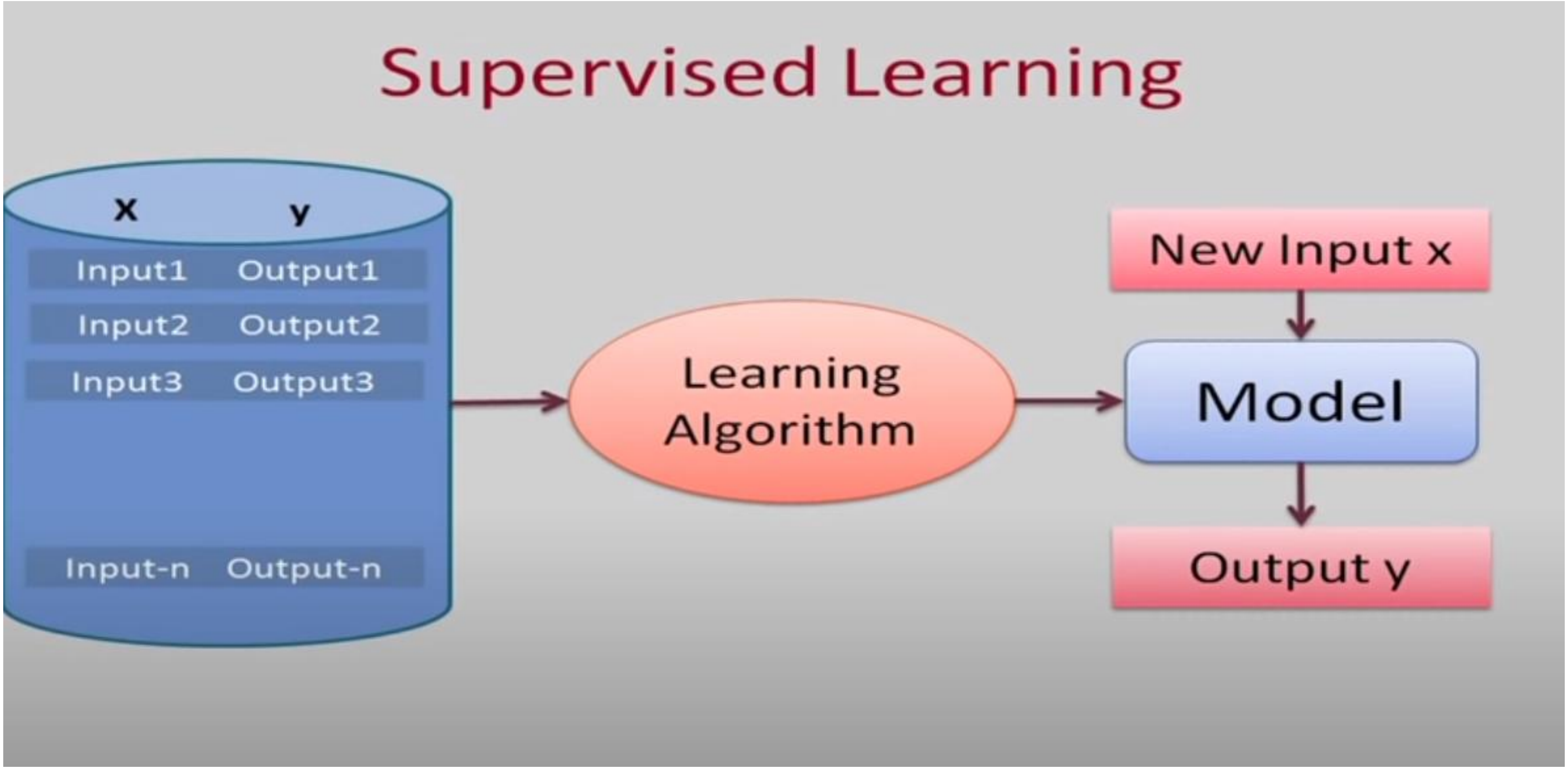
"… said to learn from experience with respect to some class of tasks, and a performance measure P, if [the learner' s] performance at tasks in the class, as measured by P, improves with experience."

Tom Mitchell 1997.

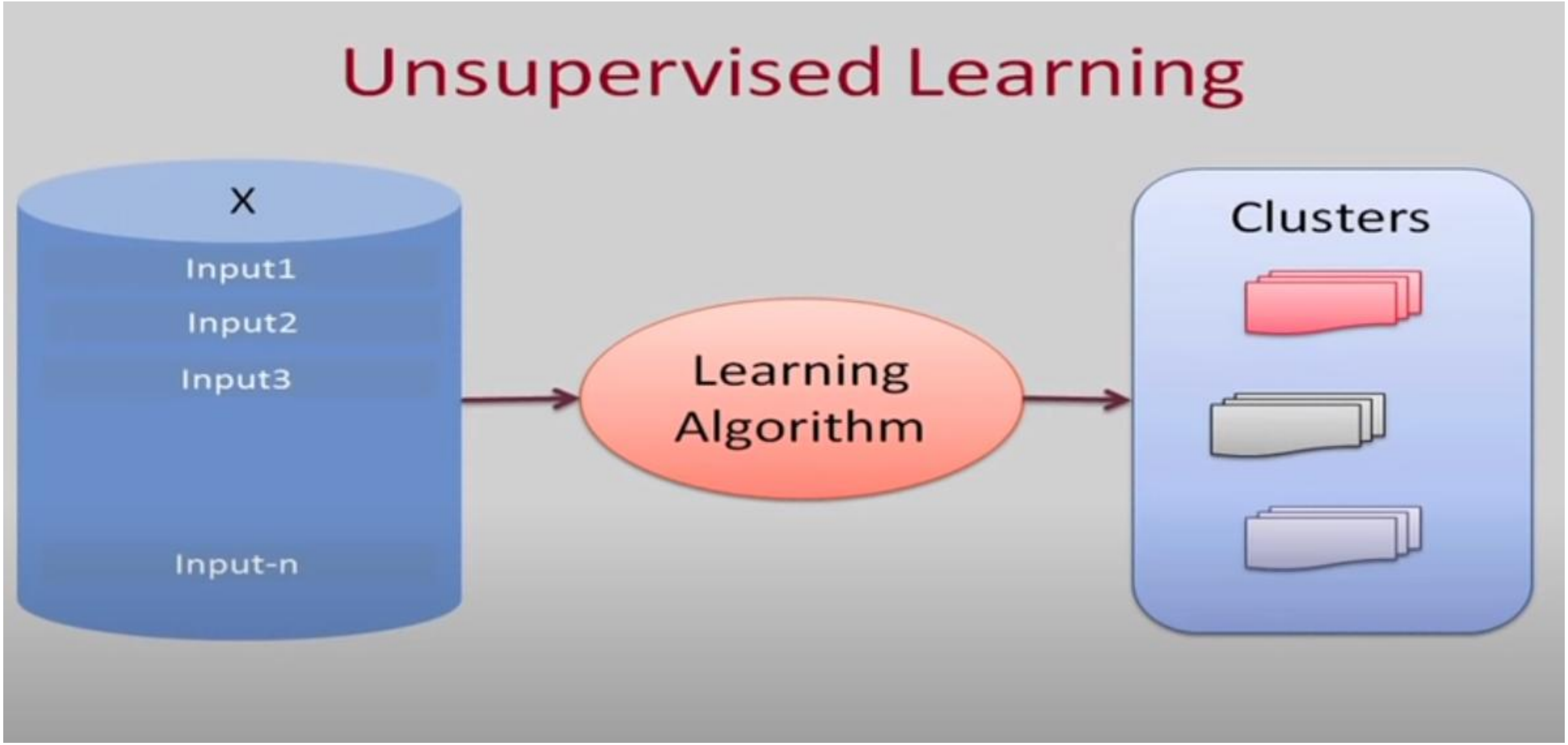
Inductive Learning

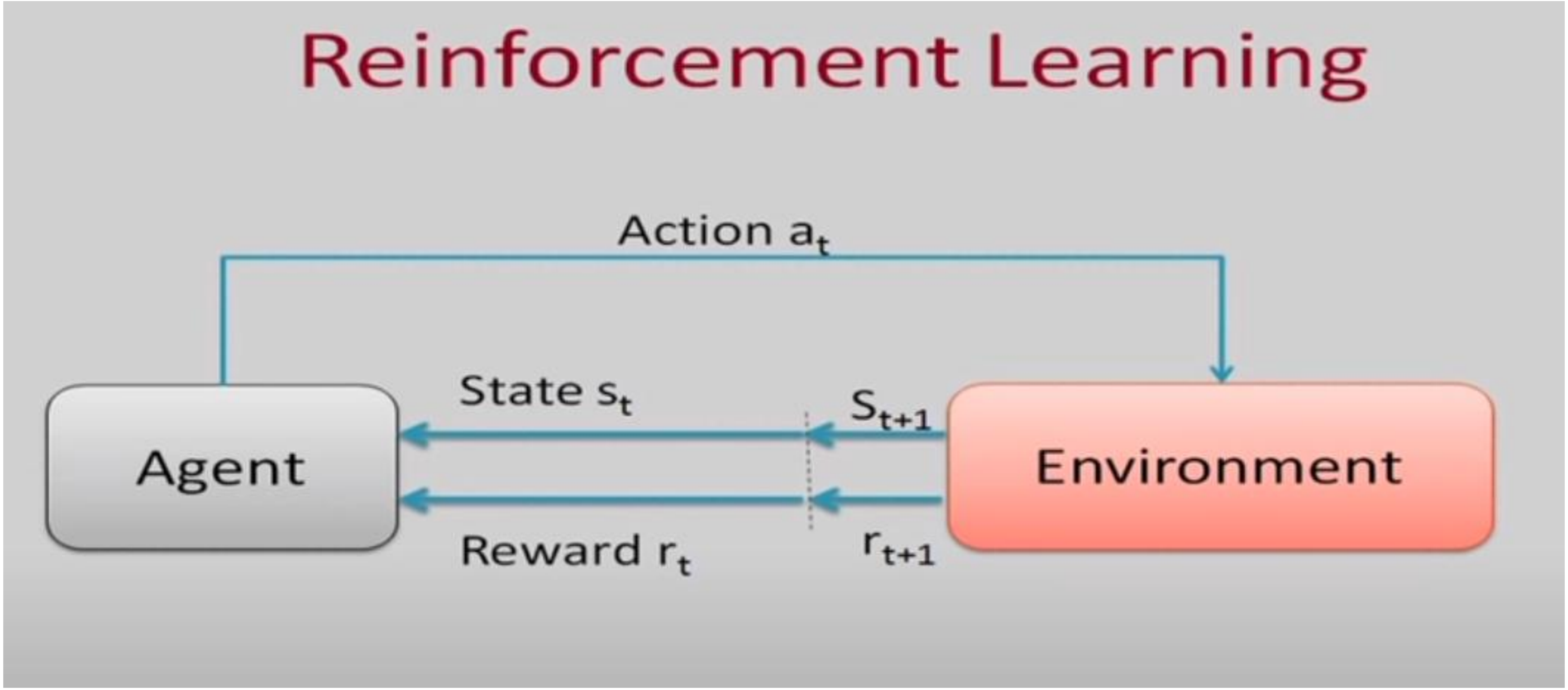
Broad types of machine learning

- Supervised Learning
  - X,y (pre-classified training examples)
  - Given an observation x, what is the best label for y?
- Unsupervised learning
  - X
  - Given a set of x's, cluster or summarize them
- Reinforcement Learning
  - Determine what to do based on rewards and punishments.



Supervised Learning
x
y
Input1
Output1
Input2
Output2
Input3
Output3
Input-n
Output-n
Learning Algorithm
New Input x
Model
Output y


Unsupervised Learning
X
Input1
Input2
Input3
Input-n
Learning Algorithm
Clusters


Reinforcement Learning
Action a_t
Agent
State s_t
Reward r_t
S_{t+1}
r_{t+1}
Environment

Supervised Learning

Given:

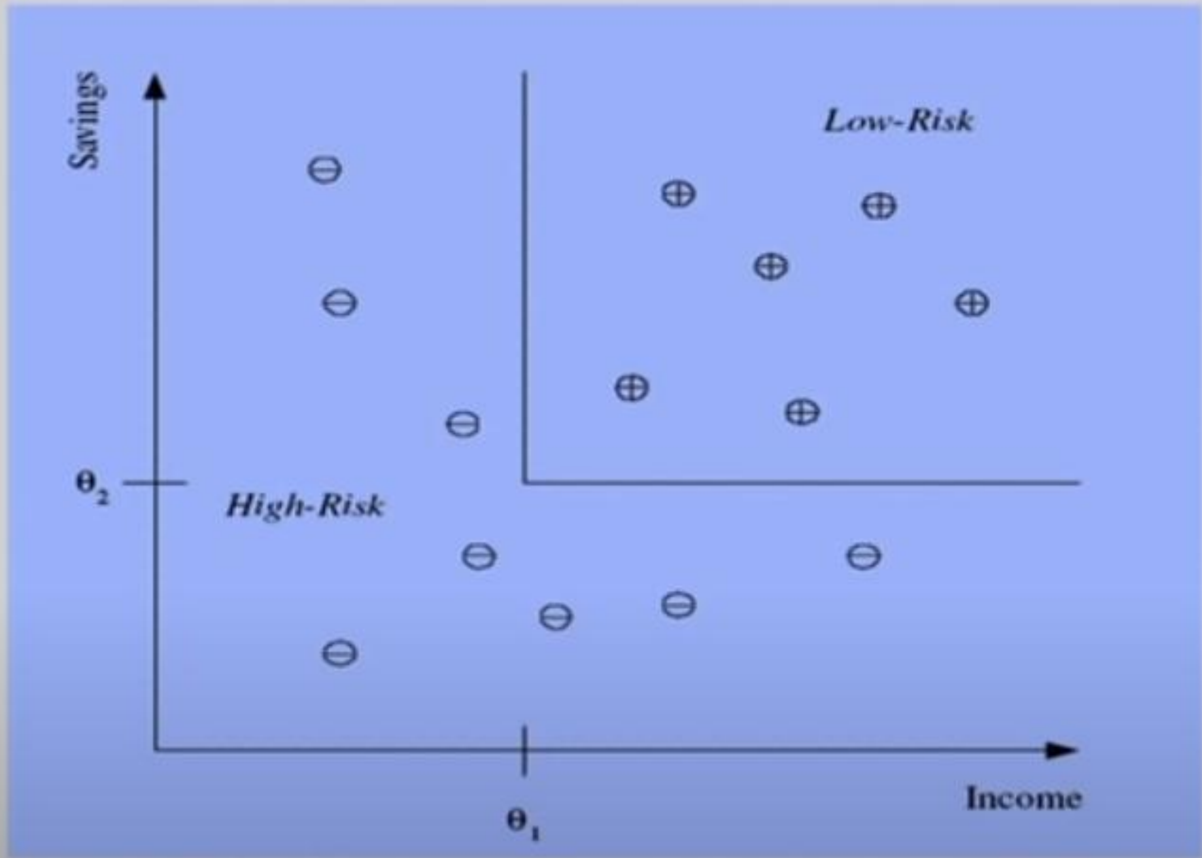
- a set of input features $X_1, \ldots, X_n$
- A target feature Y
- a set of training examples where the values for the input features and the target features are given for each example
- a new example, where only the values for the input features are given

Predict the values for the target features for the new example.

- classification when Y is discrete
- regression when **Y** is continuous

Classification

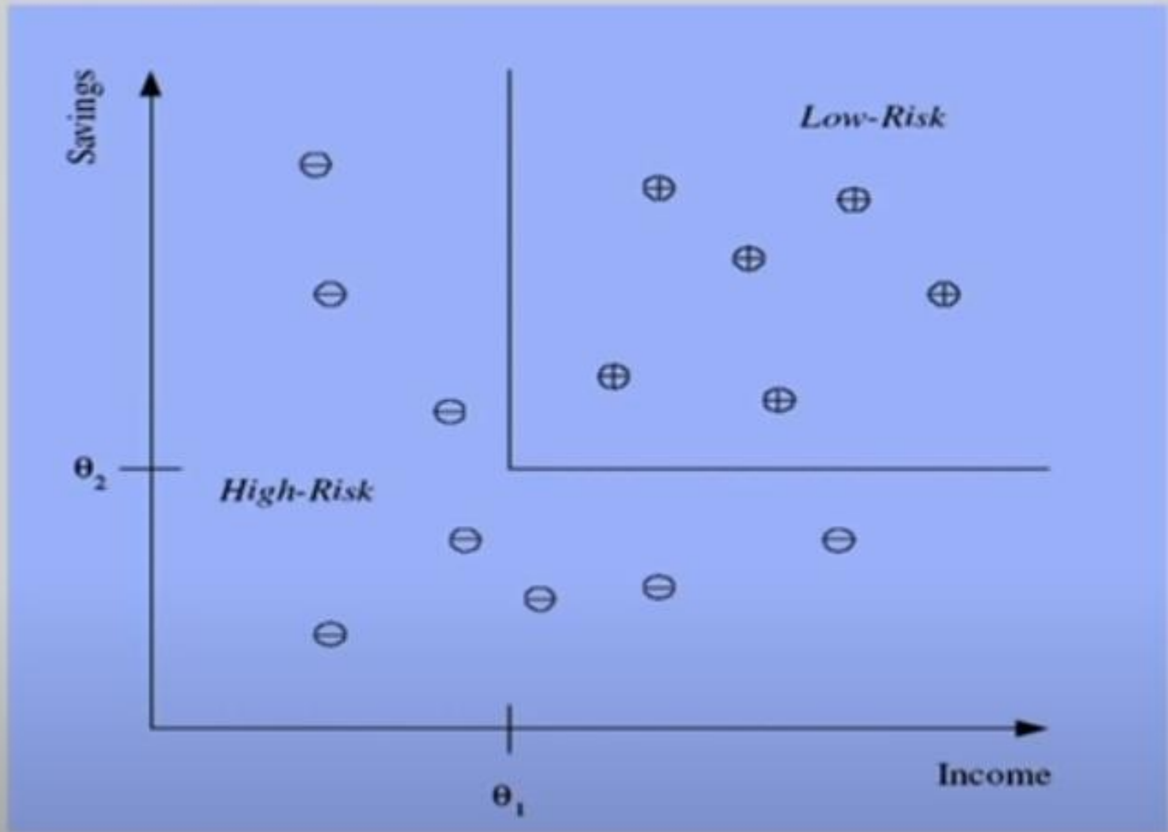
Example: Credit scoring



Classification

Example: Credit scoring

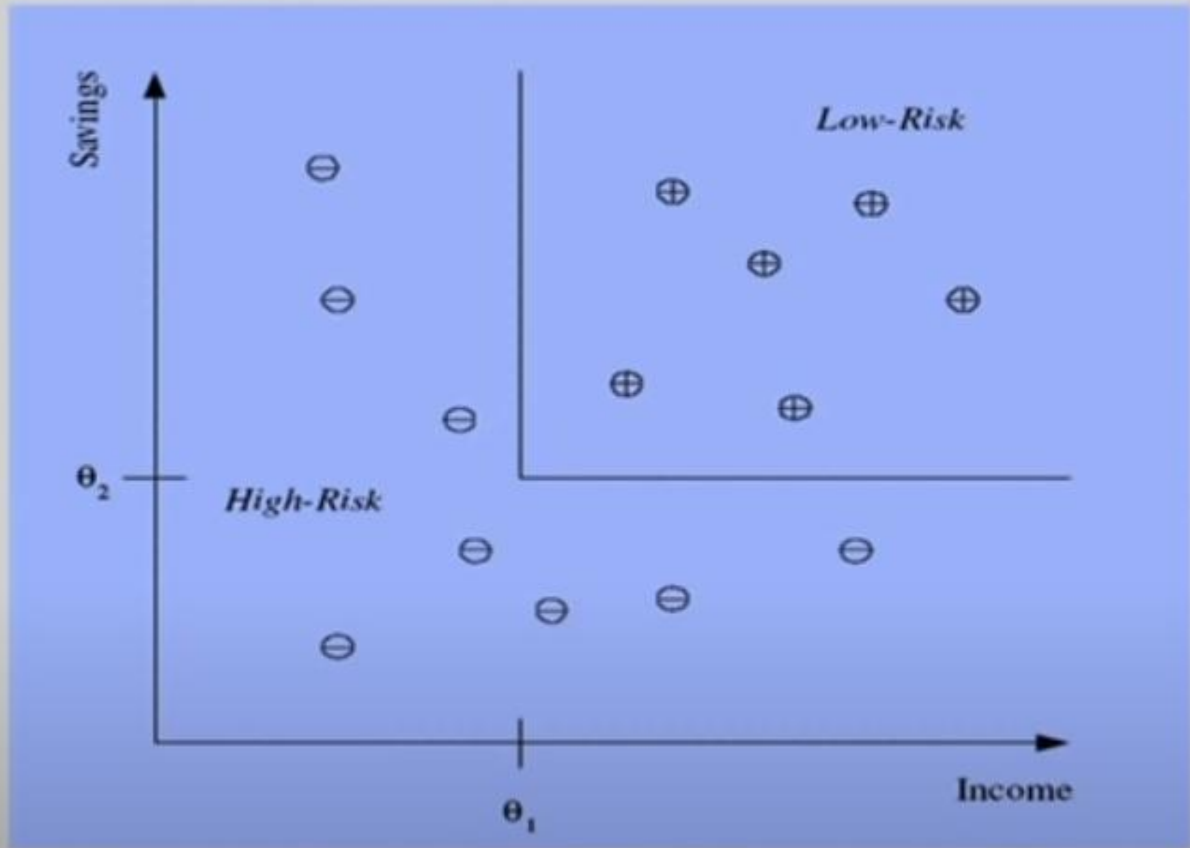
Differentiating between low-risk and high-risk customers from their *income* and *savings*



Classification

Example: Credit scoring

Differentiating between low-risk and high-risk customers from their *income* and *savings*



Discriminant: IF *income* > θ_1 AND *savings* > θ_2 THEN low-risk ELSE high-risk

Regression

Example: Price of a used car

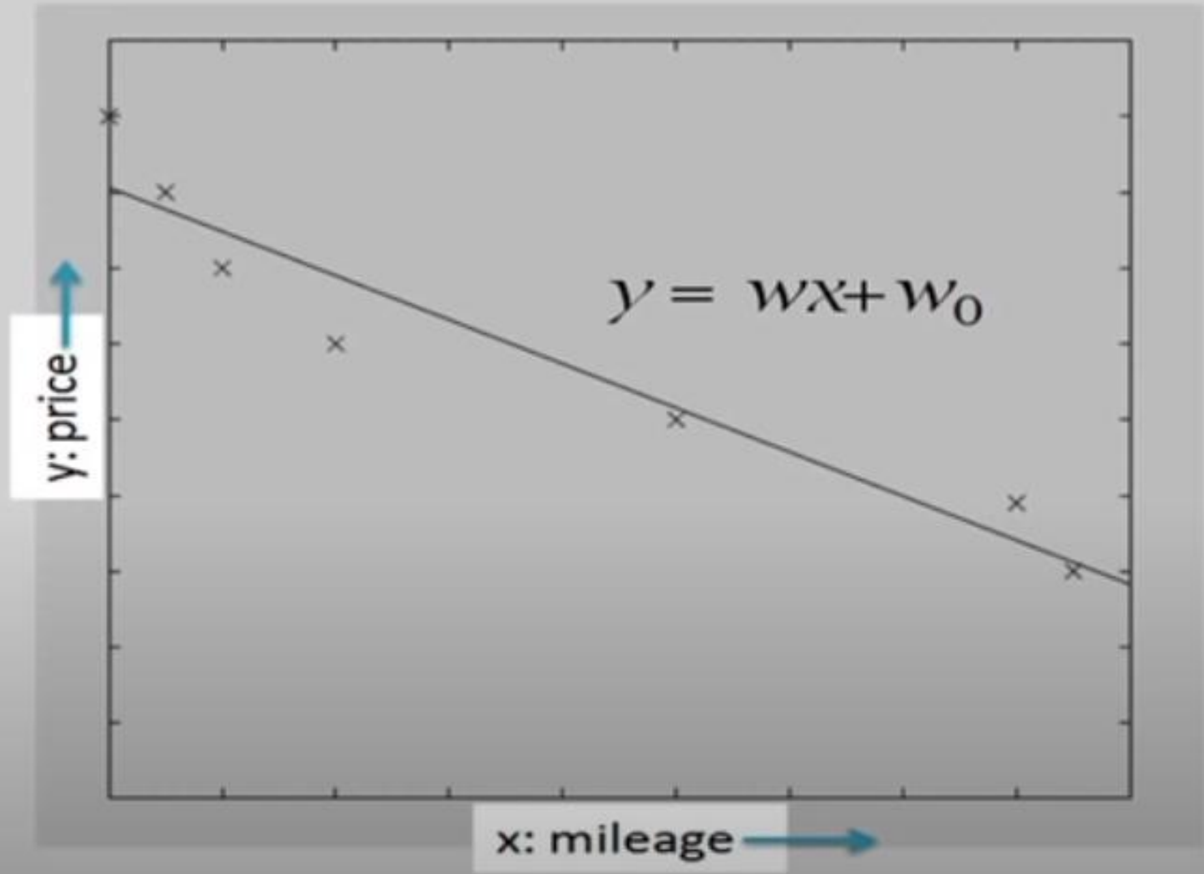
x : car attributes

y : price

$$y = g(x, \theta)$$

$g()$ model,

θ parameters



Features

- Often, the individual observations are analyzed into a set of quantifiable properties which are called features. May be
  - categorical (e.g. "A", "B", "AB" or "O", for blood type)
  - ordinal (e.g. "large", "medium" or "small")
  - integer-valued (e.g. the number of words in a text)
  - real-valued (e.g. height)

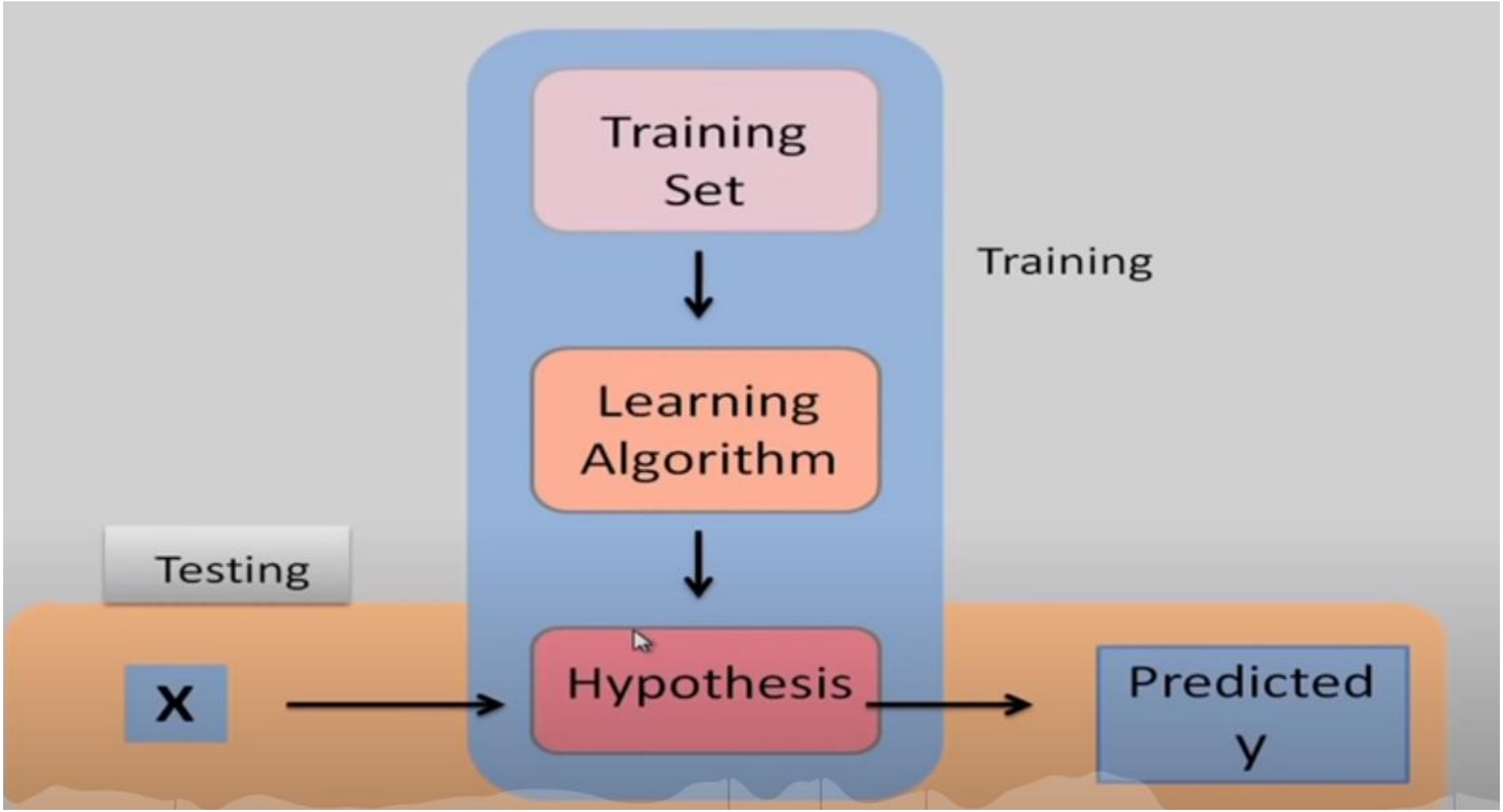
Example Data

Training Examples:

	Action	Author	Thread	Length	Where
e1	skips	known	new	long	Home
e2	reads	unknown	new	short	Work
e3	skips	unknown	old	long	Work
e4	skips	known	old	long	home
e5	reads	known	new	short	home
e6	skips	known	old	long	work

New Examples:

e7	???	known	new	short	work
e8	???	unknown	new	short	work


Training
Set
Learning
Algorithm
Hypothesis
Training
Testing
X
Predicted
y

Classification learning

- Task *T:*
  - input: a set of *instances* $d_1,\ldots,d_n$
    - an instance has a set of *features*
    - we can represent an instance as a vector $\mathbf{d}=<x_1,\ldots,x_n>$
  - output: a set of *predictions* $\hat{y}_1,\ldots,\hat{y}_n$
    - one of a fixed set of constant values:
      - *{+1,-1}* or *{cancer, healthy}*, or *{rose, hibiscus, jasmine, ...}*, or ...
- Performance metric *P:*
- Experience *E:*

Classification learning

- Task *T:*
  - input: a set of *instances* $d_1, \ldots, d_n$
  - output: a set of *predictions* $\hat{y}_1, \ldots, \hat{y}_n$

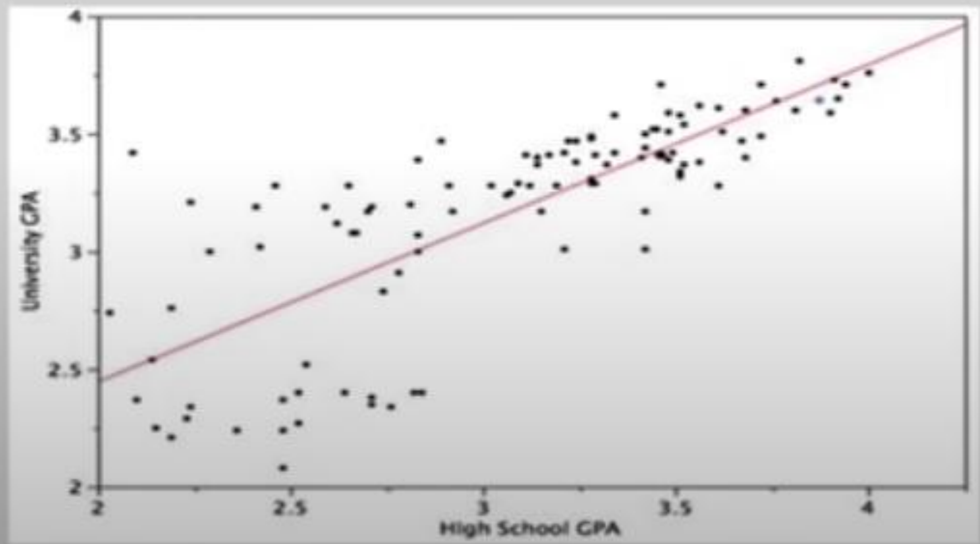
we care about performance on the *distribution*, not the *training data*

- Performance metric *P:*
  - Prob (wrong prediction) on examples from *D*
- Experience *E:*
  - a set of *labeled examples (x,y)* where *y* is the true label for *x*
  - ideally, examples should be *sampled* from some fixed distribution *D*

Representations

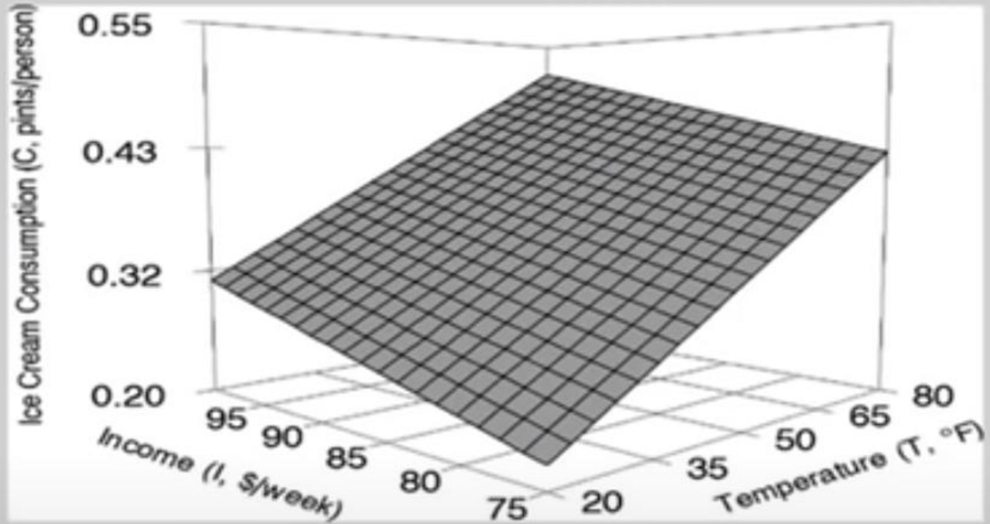
1. Decision Tree

2. Linear function

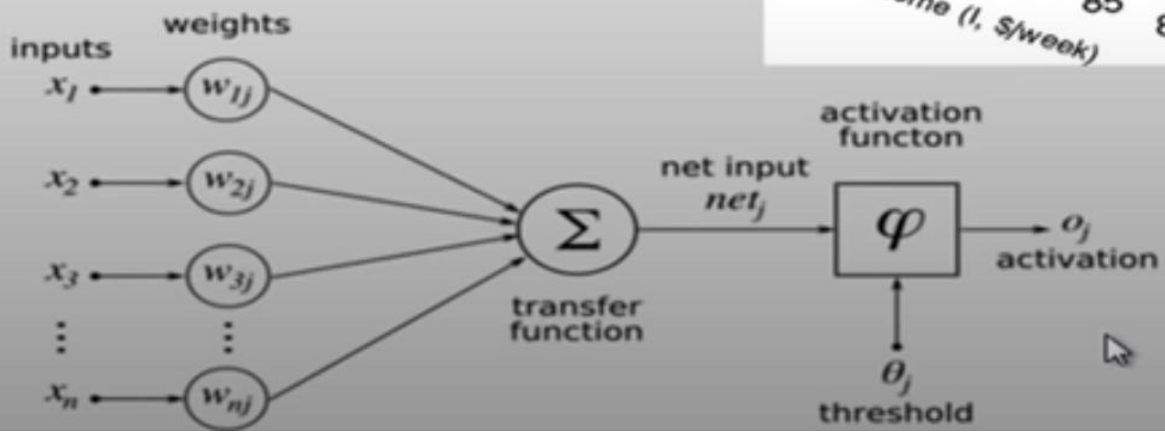


Representations

3. Multivariate linear function

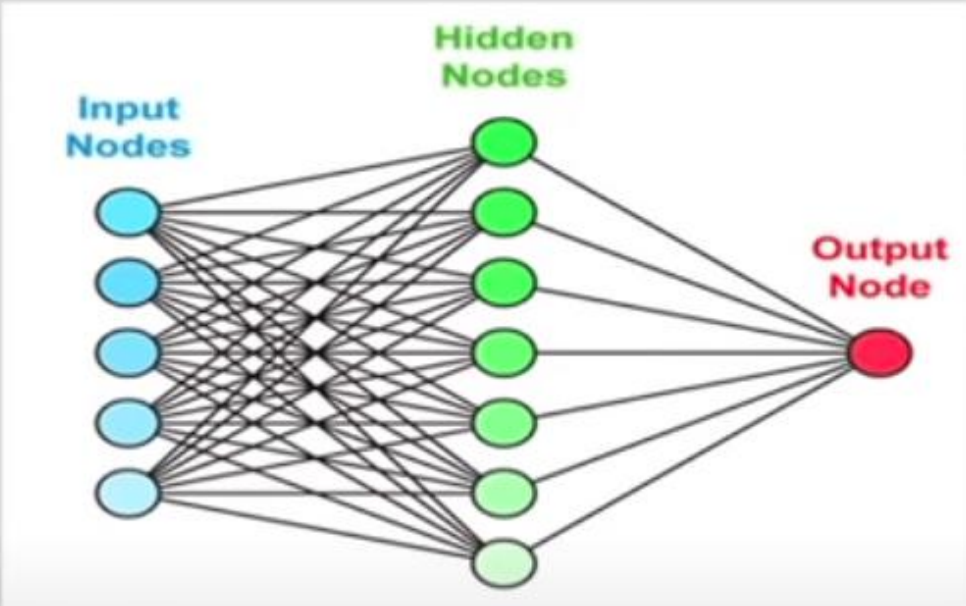


4. Single layer perceptron



Representations

5. Multi-layer neural network



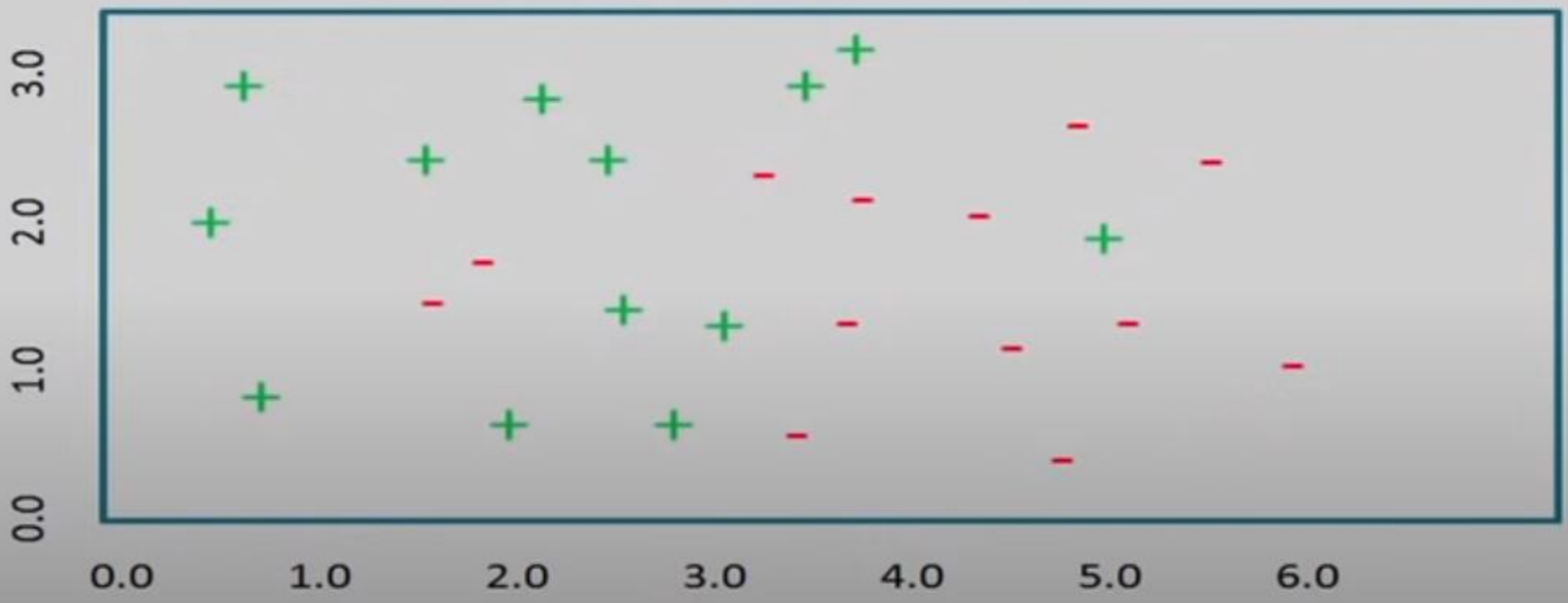
Hypothesis Space

- One way to think about a supervised learning machine is as a device that explores a "hypothesis space".
  - Each setting of the parameters in the machine is a different hypothesis about the function that maps input vectors to output vectors.

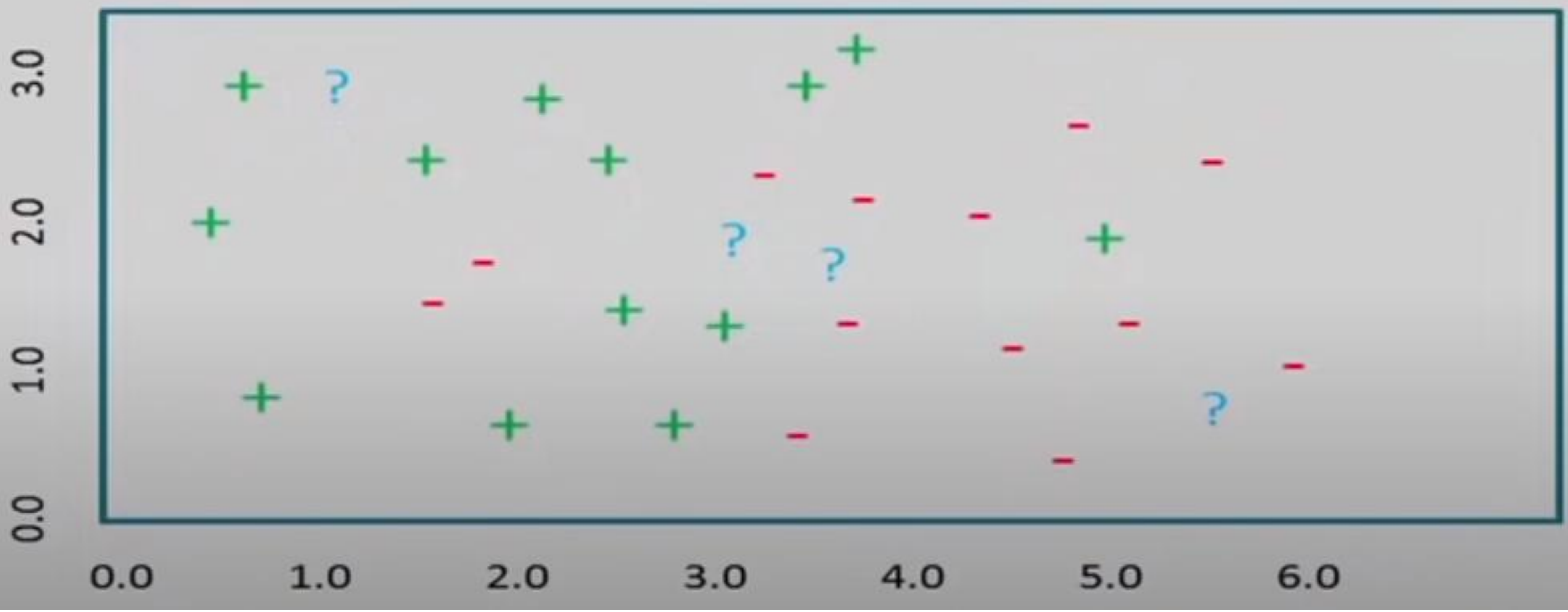
Hypothesis Space

- One way to think about a supervised learning machine is as a device that explores a "hypothesis space".
  - Each setting of the parameters in the machine is a different hypothesis about the function that maps input vectors to output vectors.

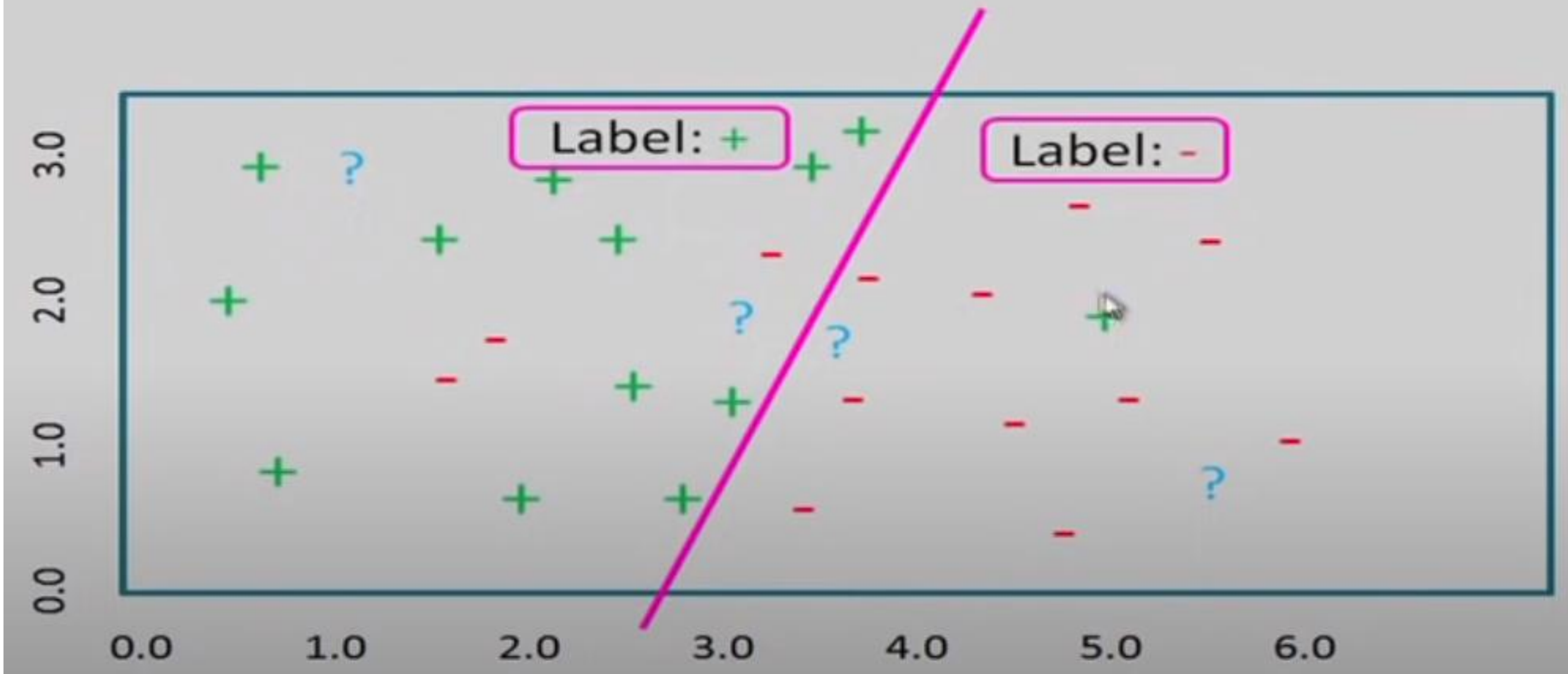
Feature Space



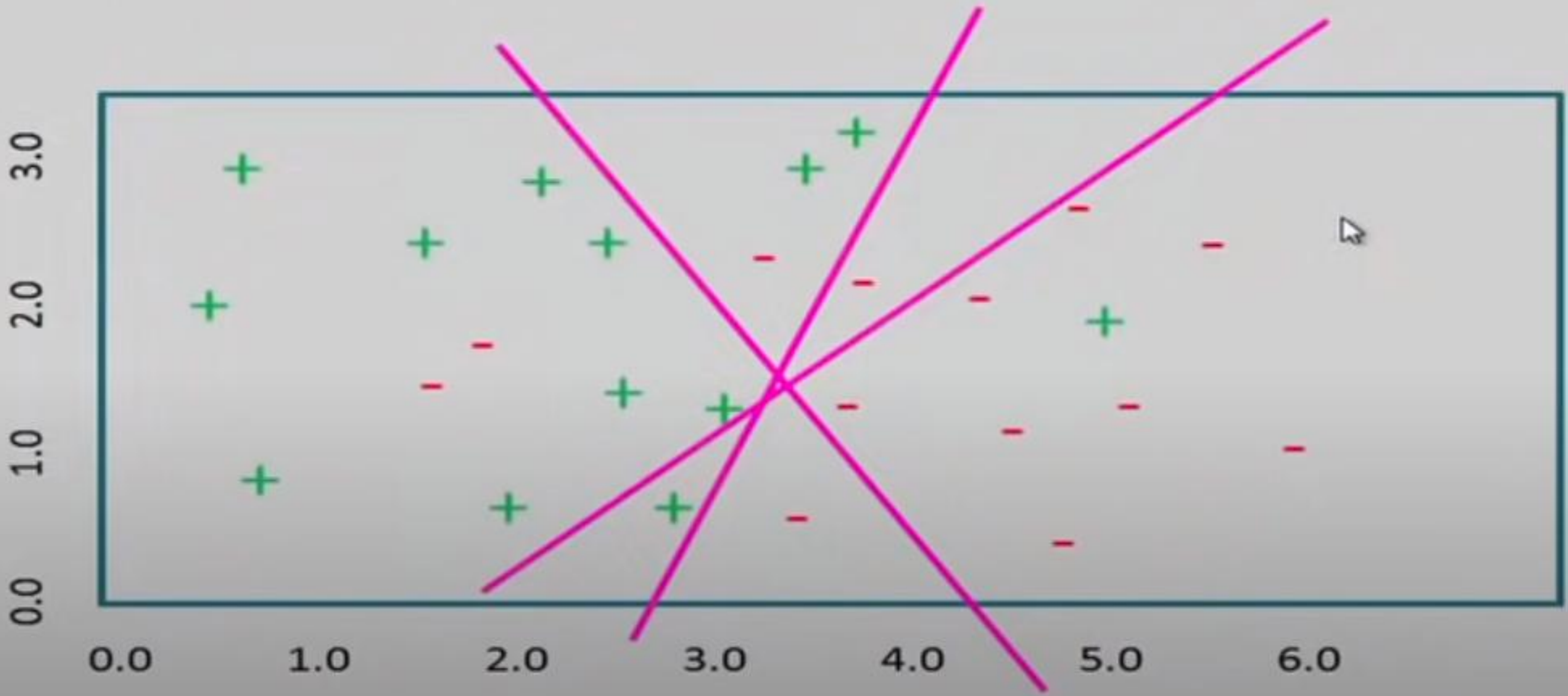
Terminology



Terminology



Terminology



Hypothesis Space

- The space of all hypotheses that can, in principle, be output by a learning algorithm.
- We can think about a supervised learning machine as a device that explores a "hypothesis space".
  - Each setting of the parameters in the machine is a different hypothesis about the function that maps input vectors to output vectors.

Classifier

- Hypothesis h: Function that approximates f.
- Hypothesis Space $\mathcal{H}$: Set of functions we allow for approximating f.
- The set of hypotheses that can be produced, can be restricted further by specifying a language bias.
- Input: Training set $\mathcal{S} \subseteq X$
- Output: A hypothesis $h \in \mathcal{H}$

Hypothesis Spaces

- If there are 4 (N) input features, there are 2^{16} $\left(2^{2^N}\right)$ possible Boolean functions.
- We cannot figure out which one is correct unless we see every possible input-output pair $2^4(2^N)$

Inductive Bias

- Need to make assumptions
  - Experience alone doesn't allow us to make conclusions about unseen data instances
- Two types of bias:
  - Restriction: Limit the hypothesis space
  - Preference: Impose ordering on hypothesis space

Inductive learning

- Inductive learning: Inducing a general function from training examples
  - Construct hypothesis *h* to agree with *c* on the training examples.
  - A hypothesis is consistent if it agrees with all training examples.
  - A hypothesis said to generalize well if it correctly predicts the value of *y* for novel example.
- *Inductive Learning is an Ill Posed Problem*:
  Unless we see all possible examples the data is not sufficient for an inductive learning algorithm to find a unique solution.

Inductive Learning Hypothesis

- Any hypothesis h found to approximate the target function c well over a sufficiently large set of training examples $\mathcal{D}$ will also approximate the target function well over other unobserved examples.

Learning as Refining the Hypothesis Space

- Concept learning is a task of searching an hypotheses space of possible representations looking for the representation(s) that best fits the data, given the bias.
- The tendency to prefer one hypothesis over another is called a **bias**.
- Given a representation, data, and a bias, the problem of learning can be reduced to one of search.

Occam's Razor

- A classical example of Inductive Bias

- the simplest consistent hypothesis about the target function is actually the best

Important issues in Machine Learning

- What are good hypothesis spaces?
- Algorithms that work with the hypothesis spaces
- How to optimize accuracy over future data points (overfitting)
- How can we have confidence in the result? (How much training data – statistical qs)
- Are some learning problems computationally intractable?

Generalization

- Components of generalization error
  - Bias: how much the average model over all training sets differ from the true model?
    - Error due to inaccurate assumptions/simplifications made by the model
  - Variance: how much models estimated from different training sets differ from each other

Underfitting and Overfitting

- Underfitting: model is too "simple" to represent all the relevant class characteristics
  - High bias and low variance
  - High training error and high test error
- Overfitting: model is too "complex" and fits irrelevant characteristics (noise) in the data
  - Low bias and high variance
  - Low training error and high test error

Experimental Evaluation of Learning Algorithms

- Evaluating the performance of learning systems is important because:
  - Learning systems are usually designed to predict the class of "future" unlabeled data points.
- Typical choices for Performance Evaluation:
  - Error
  - Accuracy
  - Precision/Recall
- Typical choices for Sampling Methods:
  - Train/Test Sets
  - K-Fold Cross-validation

Evaluating predictions

- Suppose we want to make a prediction of a value for a target feature on example **x**:
  - y is the observed value of target feature on example ***x***.
  - $\hat{y}$ is the predicted value of target feature on example ***x***.
  - How is the error measured?

Sample Error and True Error

- The ***sample error*** of hypothesis *f* with respect to target function *c* and data sample *S* is:

$$error_s(f)= 1/n \ \Sigma_{x \in S} \delta(f(x),c(x))$$

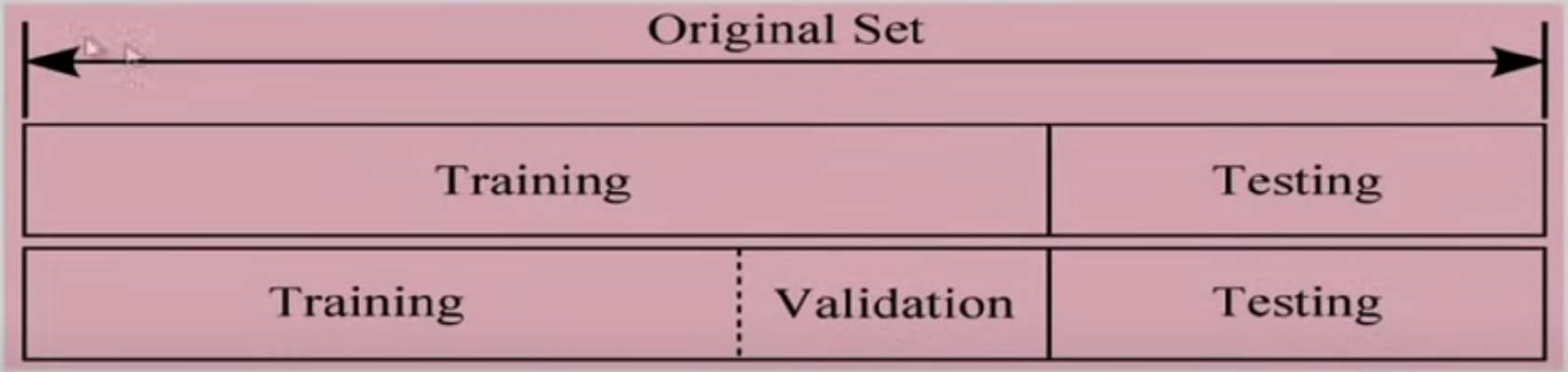
- The ***true error*** (denoted $error_D(f)$) of hypothesis *f* with respect to target function *c* and distribution *D*, is the probability that *h* will misclassify an instance drawn at random according to *D*.

$$error_D(f)= Pr_{x \in D}[f(x) \neq c(x)]$$

Difficulties in evaluating hypotheses with limited data

- Bias in the estimate: The sample error is a poor estimator of true error
  - ==> test the hypothesis on an independent test set
- We divide the examples into:
  - **Training examples** that are used to train the learner
  - **Test examples** that are used to evaluate the learner
- Variance in the estimate: The smaller the test set, the greater the expected variance.

Validation set

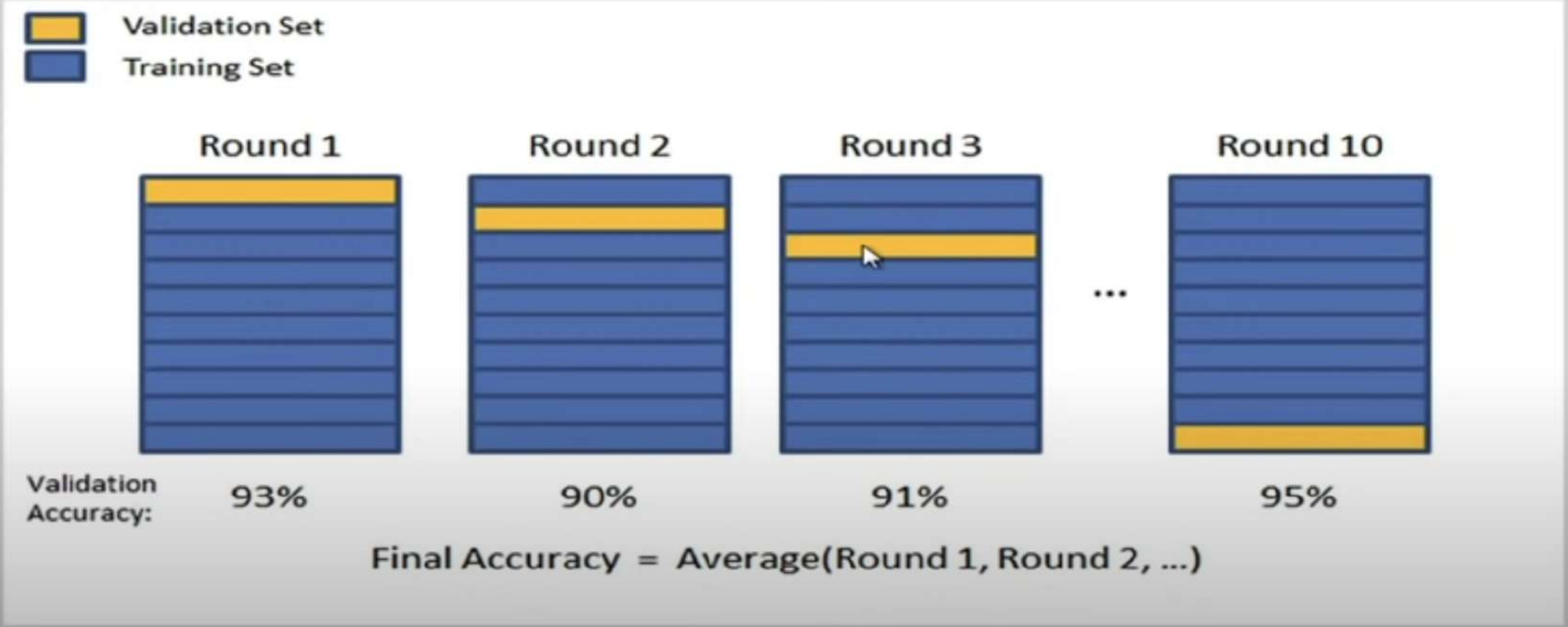


Validation fails to use all the available data

k-fold cross-validation

1. Split the data into k equal subsets
2. Perform k rounds of learning; on each round
   - 1/k of the data is held out as a test set and
   - the remaining examples are used as training data.
3. Compute the average test set score of the k rounds

K-fold cross validation



Trade-off

- In machine learning, there is always a trade-off between
  - complex hypotheses that fit the training data well
  - simpler hypotheses that may generalise better.
- As the amount of training data increases, the generalization error decreases.

- How good is a model?
- How do I choose a model?
- Do I have enough data?
- Is the data of sufficient quality?
  - Errors in data. Ex: Age=225; noise in low resolution images
  - Missing Values
- How confident can I be of the results?
- Am I describing the data correctly?
  - Are Age and Income enough? Should I look at Gender also?
  - How should I represent age? As a number, or as young, middle age,