



# CS 6375

## Introduction to Machine Learning

Rishabh Iyer

University of Texas at Dallas

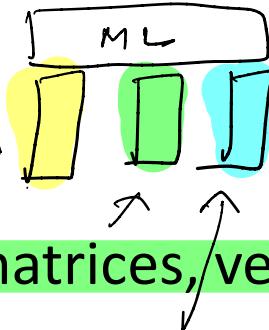
## Grading

- 4-5 problem sets (50%)
  - learning
  - Mix of theory + programming
  - 1 assignment every 2-3 weeks
- Mid term (20%)
- Final exam (30%)

(subject to change)

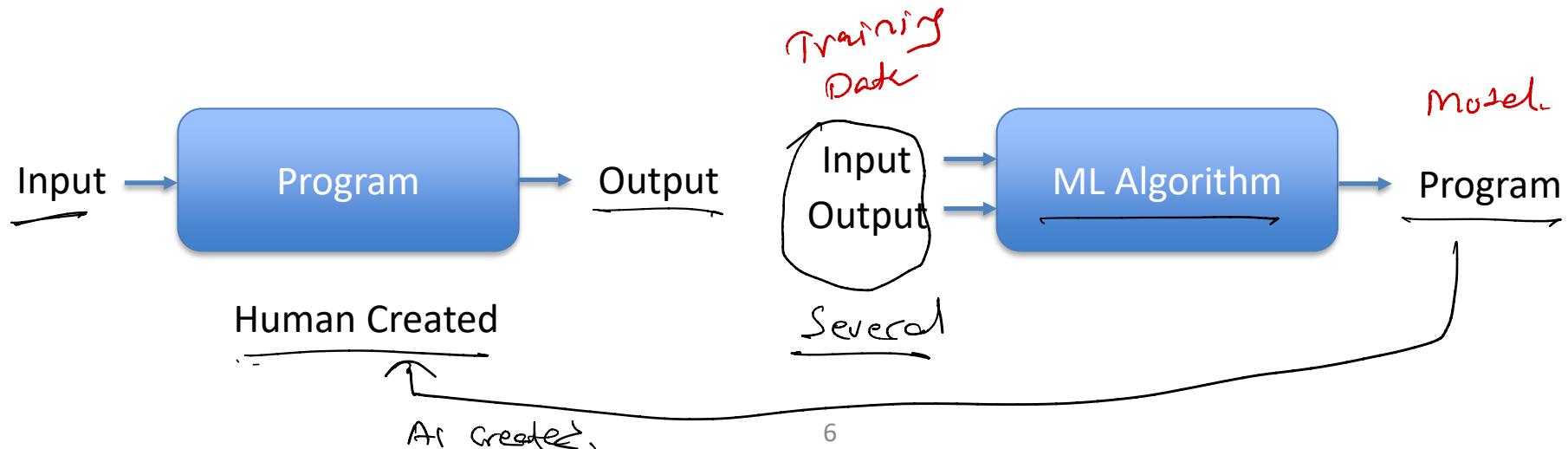
# Prerequisites

- CS3345, Data Structures and Algorithms
- CS3341, Probability and Statistics in Computer Science
- “Mathematical sophistication”
  - Basic probability
  - Linear algebra: eigenvalues/vectors, matrices, vectors, etc.
  - Multivariate calculus: derivatives, gradients, etc.
- I’ll review some concepts as we come to them, but **you should brush up on areas that you aren’t as comfortable**
- Take prerequisite “quiz” on eLearning

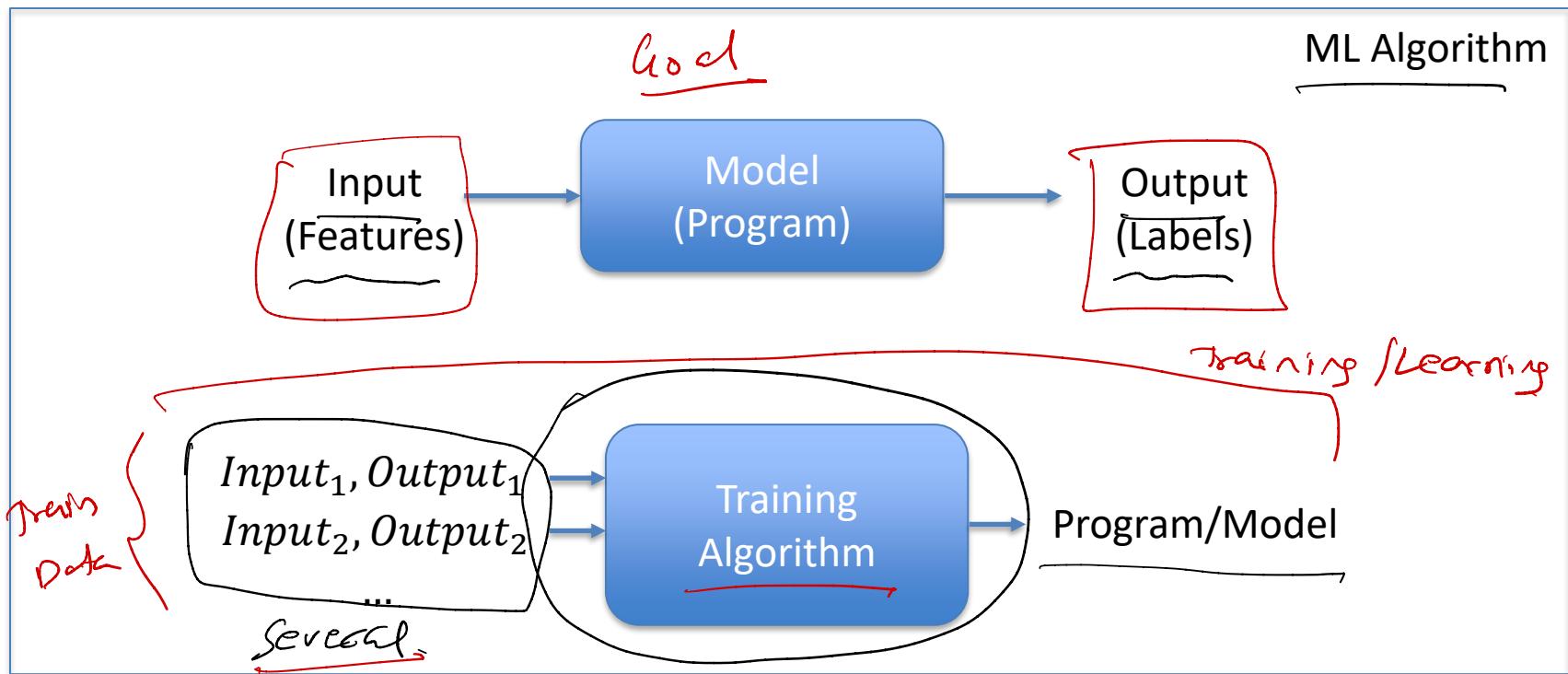


# What is Machine Learning?

- ❑ Programming: (Software Engineering)
  - ❑ A human writes a program (set of rules/conditions/algorithm) to do a specific task
  - ❑ For a given input, the program generates an output
- ❑ Machine Learning Paradigm:
  - ❑ Generate training data consisting of (“input”, “output”) pairs
  - ❑ The “ML Model” automatically generates a program (set of rules/conditions) to generate an output for a new (unseen) input



# Basic Machine Learning Paradigm



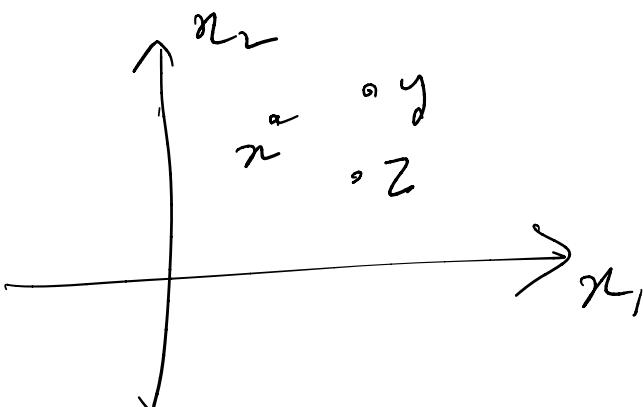
# Basics: Vectors

$x \in \mathbb{R}^d$  ( $d$ -dimensional vector)

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \quad d\text{-dim.} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

e.g.  $x \in \mathbb{R}^2$

$$x = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad x_1, x_2$$



Operations

① vector add<sup>n</sup>

$$z = x + y$$

$$\text{if } z_i = x_i + y_i, \forall i=1:d$$

② vector sub

$$z = x - y$$

(inner product)

③ Dot product.

$$z = \frac{x \cdot y}{d} \quad (\text{scalar})$$

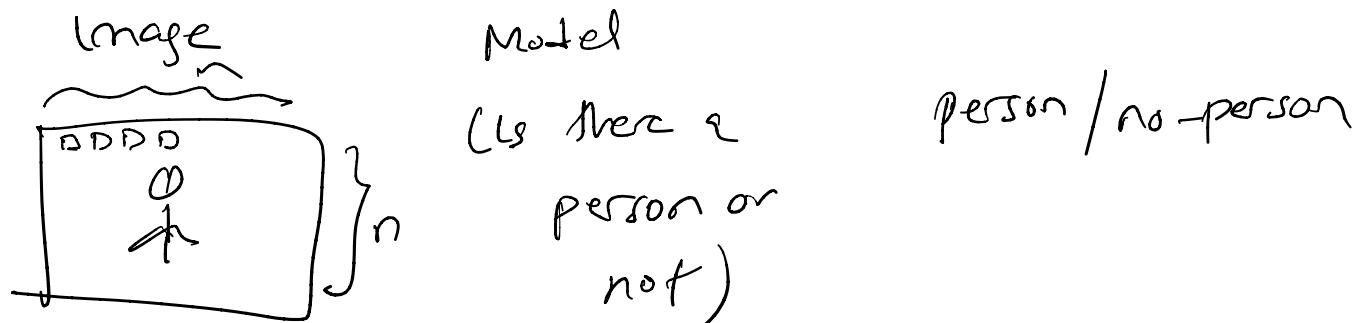
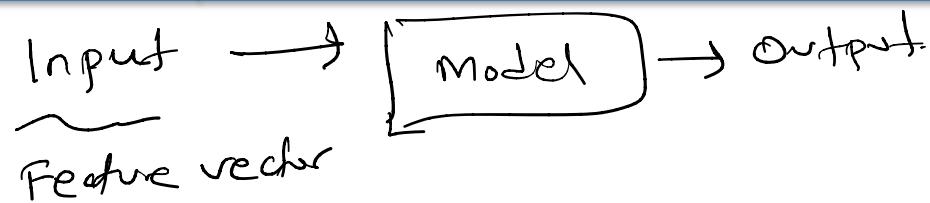
$$= \sum_{i=1}^d x_i y_i$$

$$z = 1 \cdot 3 + 2 \cdot 4$$

$$= 11$$

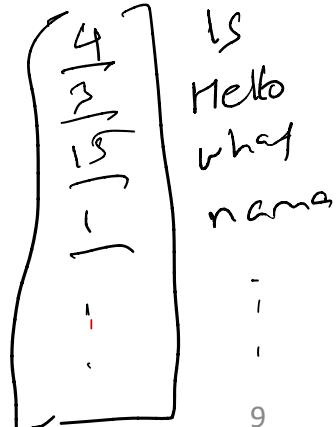
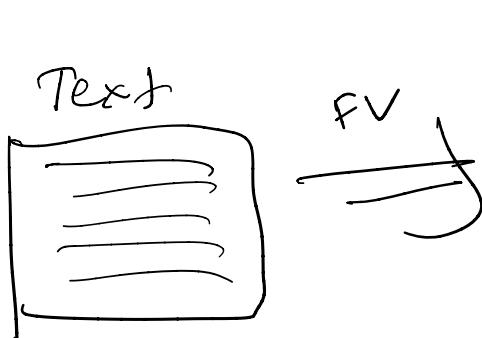
e.g.:

# Basics: Feature Vectors



Feature Vec:  $n \times n \times 3$

RGB



(Bag of words Rep)

- ① Email: spam / Not spam
- ② webpage: sport / news / ..

# Vector Operations

① Addition

② Multiplication (Dot product)

③ Subtraction.

④ Scalar multiplication

$$y = c \cdot x$$

$$= c \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} cx_1 \\ \vdots \\ cx_d \end{bmatrix}$$

# Matrices and Matrix Vector Product



If  $A \in \mathbb{R}^{m \times n}$  and  $x \in \mathbb{R}^n$ , we can define  $y = Ax$  where  $y \in \mathbb{R}^m$  is a  $m$  dimensional vector.

Matrix vector product is defined as below:

$$Ax = \left\{ \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \right\} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \end{bmatrix}$$

$m \times 1$

$m \times n$

$\uparrow$

$m \times 1$

$\uparrow$

$m \times 1$

# Matrix Vector Product Example

For example, if

$$A = \begin{bmatrix} 1 & -1 & 2 \\ 0 & -3 & 1 \end{bmatrix}$$

and  $\mathbf{x} = (2, 1, 0)$ , then

$$\begin{aligned} A\mathbf{x} &= \begin{bmatrix} 1 & \cancel{-1} & 2 \\ \cancel{0} & \cancel{-3} & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \cancel{2 \cdot 1 - 1 \cdot 1 + 0 \cdot 2} \\ \cancel{2 \cdot 0 - 1 \cdot 3 + 0 \cdot 1} \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ -3 \end{bmatrix}. \end{aligned}$$

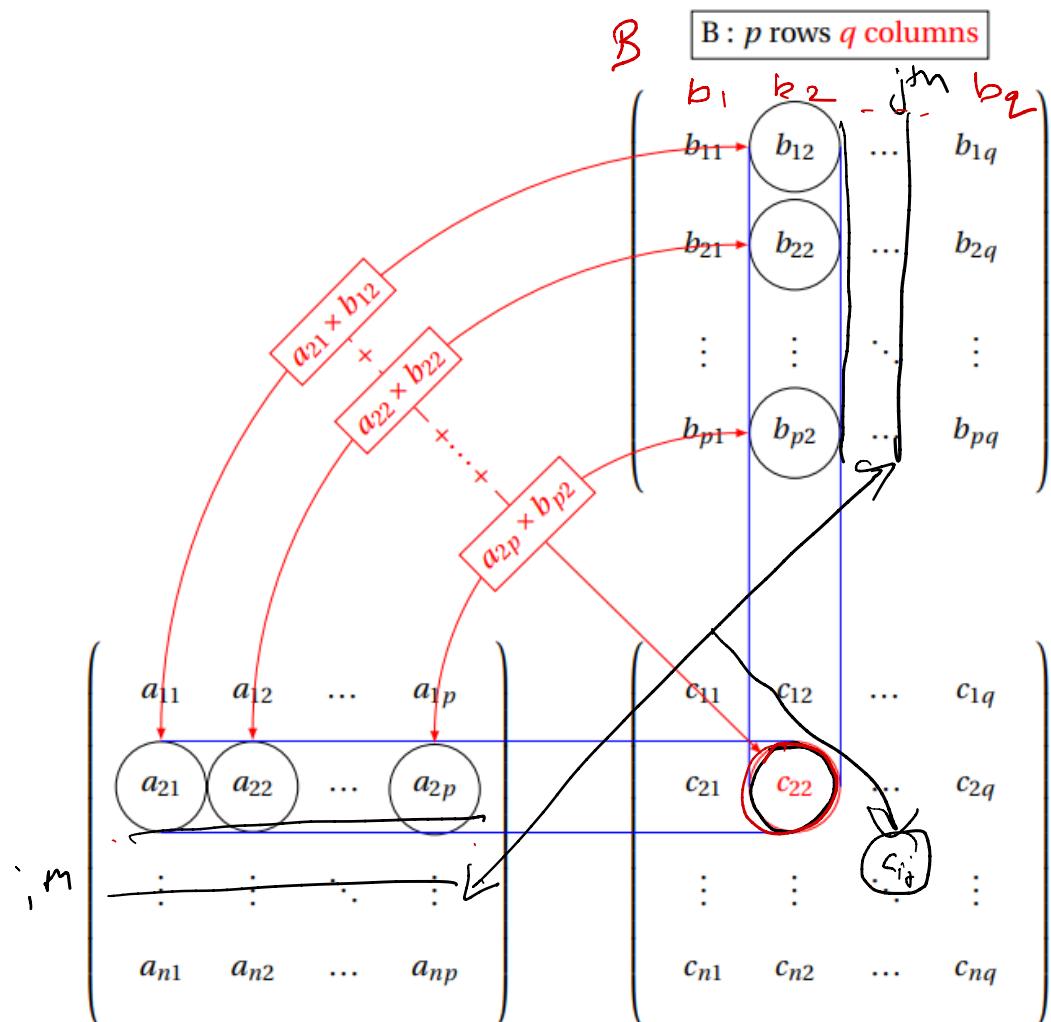
# Matrix Matrix Product



$$C = A \times B$$

$n \times p$      $p \times q$

$n \times q$

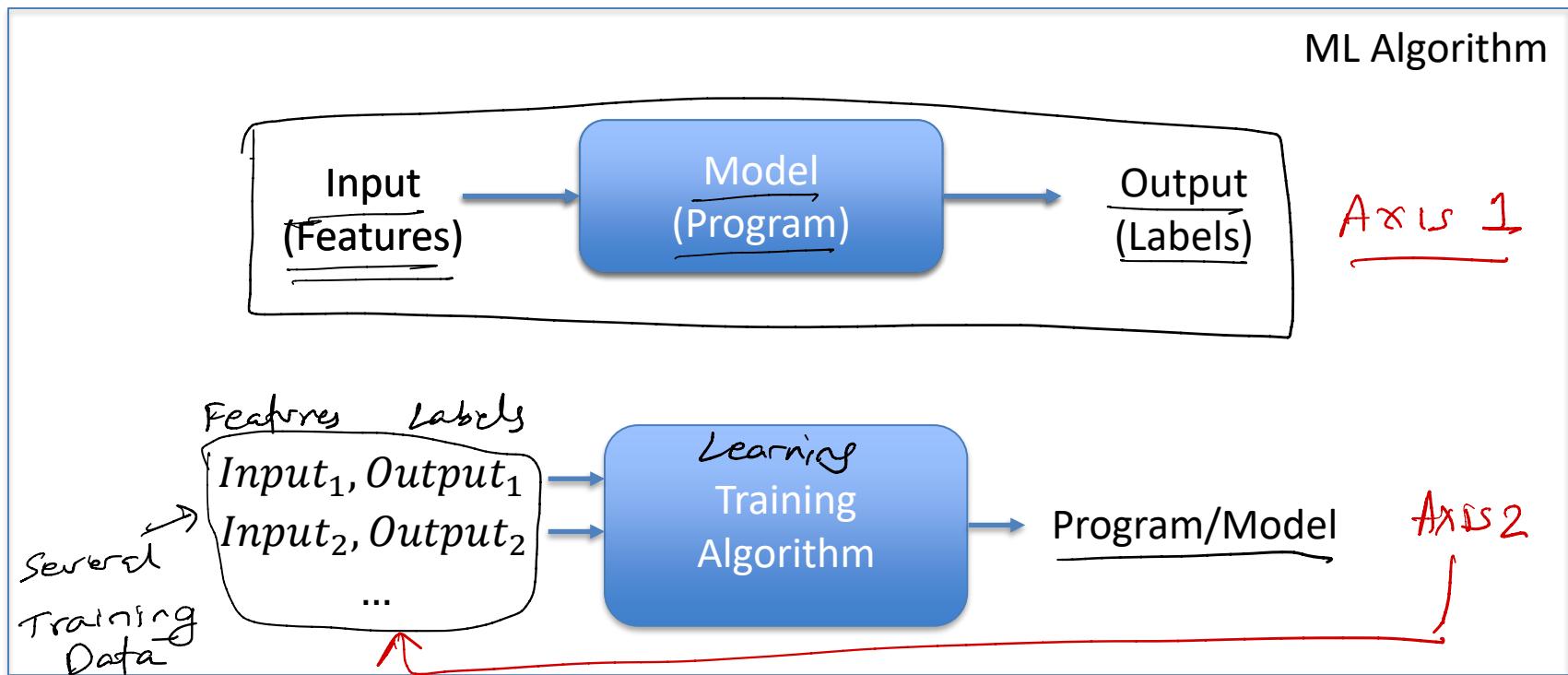


$\mathcal{A}$

$A : n \text{ rows } p \text{ columns}$

$C = A \times B : n \text{ rows } q \text{ columns}$

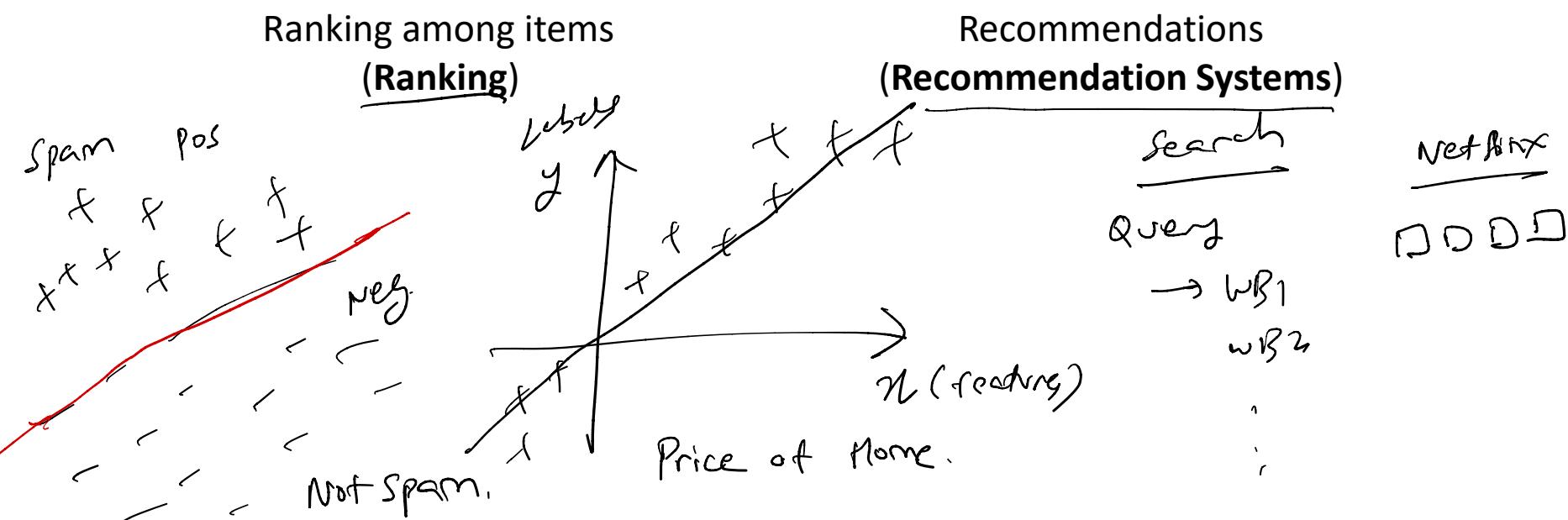
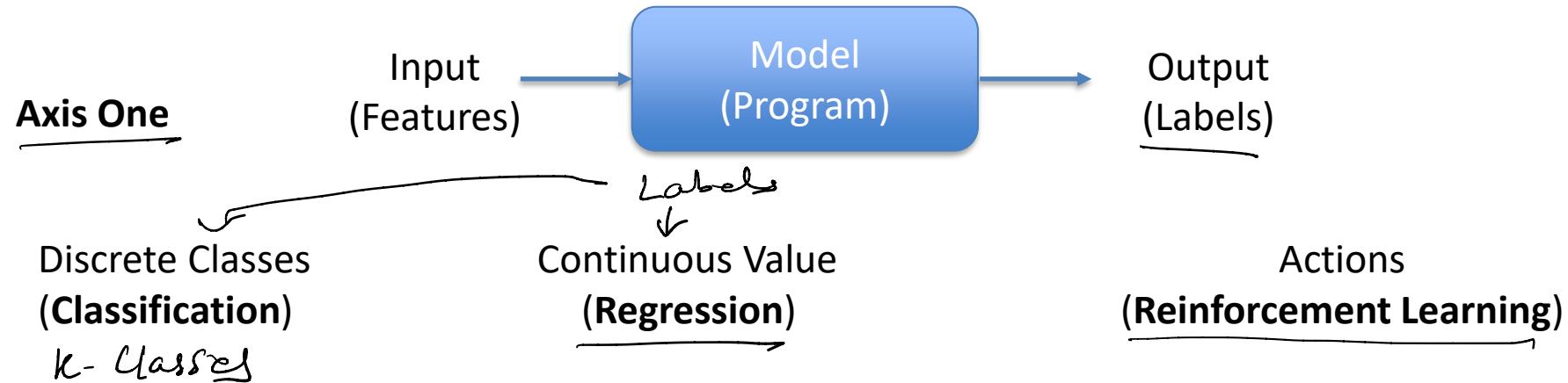
# Types of Machine Learning



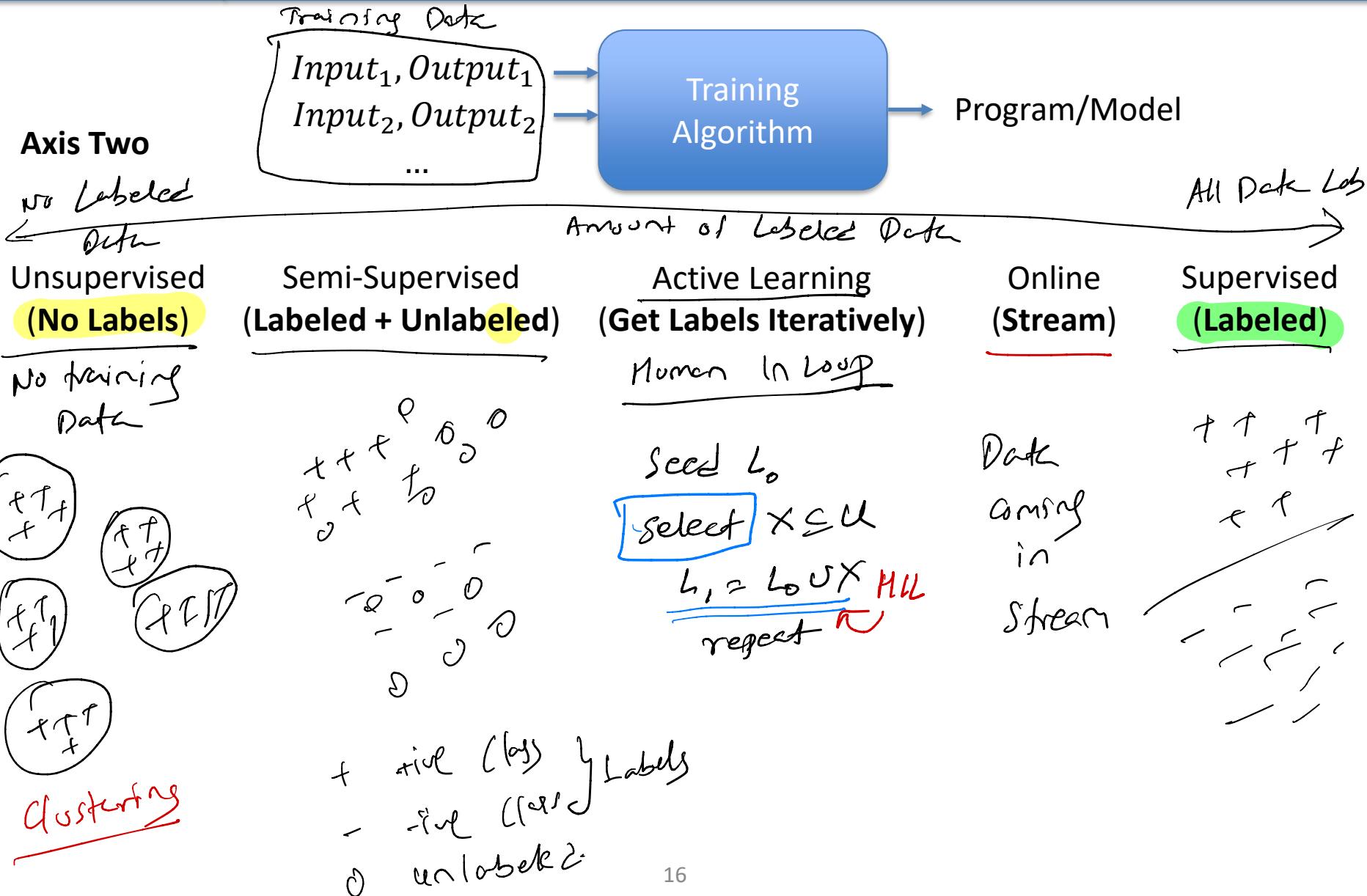
**Axis One:** What is the Output? (Labels)

**Axis Two:** Amount of Labeled Data for training and how is it available to us

# Types of Machine Learning



# Types of Machine Learning

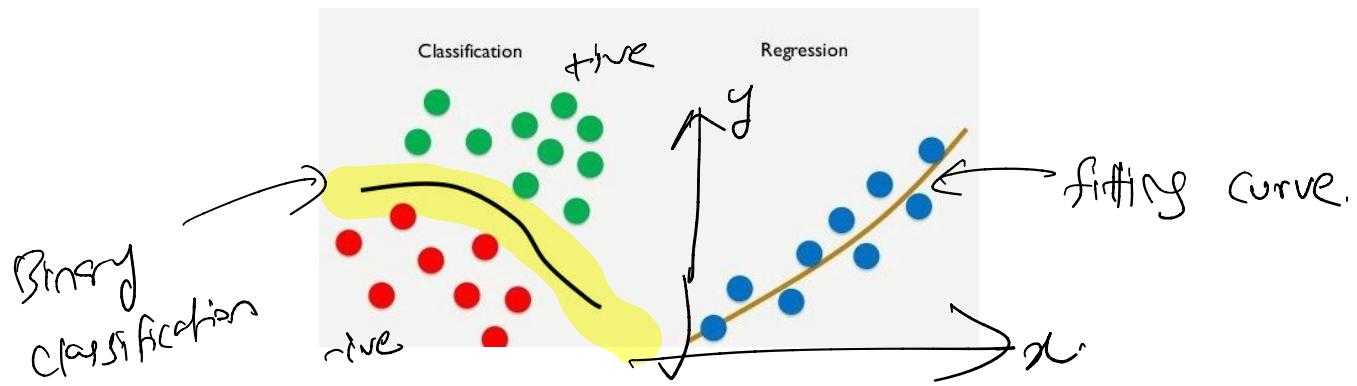


Examples of  
Machine Learning in  
Real-world Applications

# Supervised Learning



- $M$  training data points  
All labeled
- **Input:**  $(x^{(1)}, y^{(1)}), \dots, (x^{(M)}, y^{(M)})$
  - $x^{(m)}$  is the  $m^{th}$  data item and  $y^{(m)}$  is the  $m^{th}$  **label**  
Features                                      Labels
  - **Goal:** find a function  $f$  such that  $f(x^{(m)})$  is a “good approximation” to  $y^{(m)}$
  - Can use it to predict  $y$  values for previously unseen  $x$  values

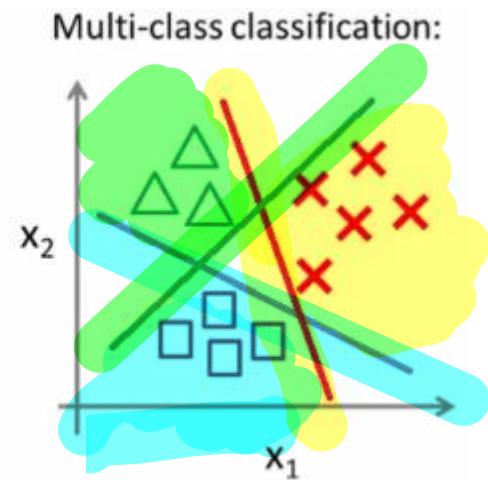
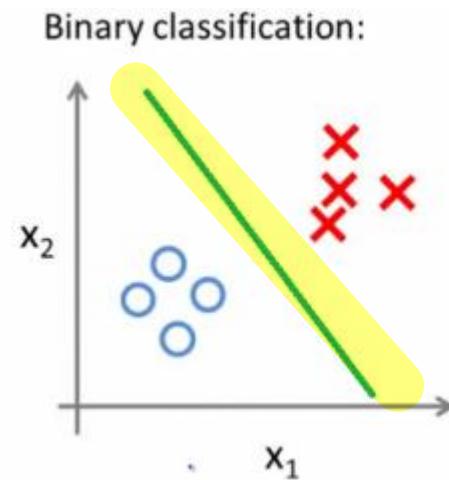
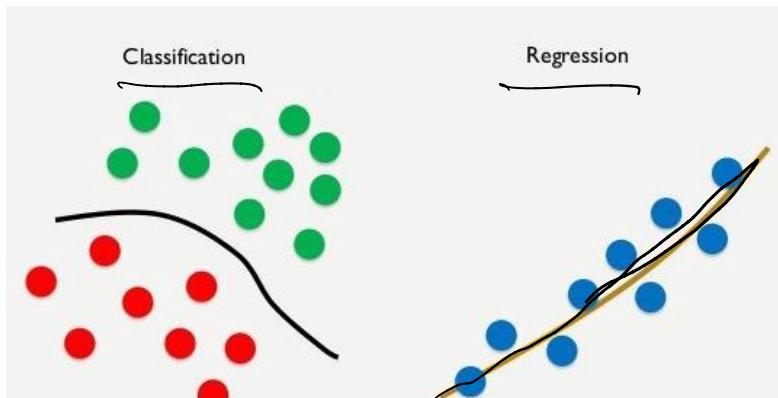


# Supervised Learning

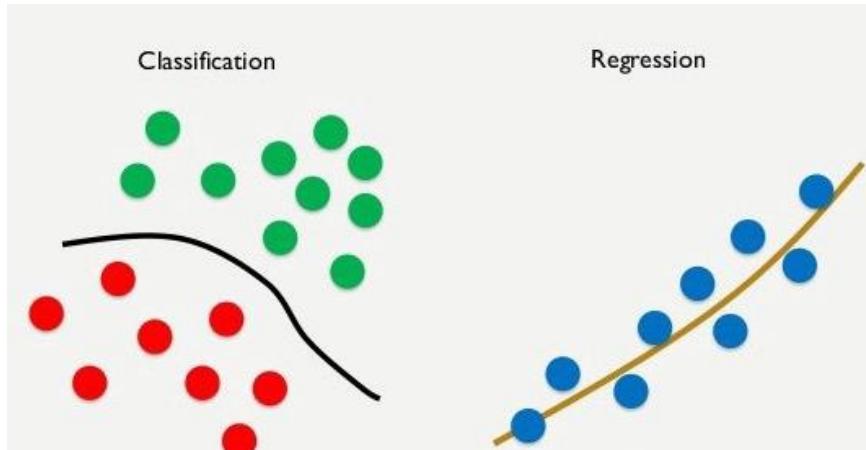


## Classification vs Regression

- Input: pairs of points  $(x^{(1)}, y^{(1)}), \dots, (x^{(M)}, y^{(M)})$  with  $x^{(m)} \in \mathbb{R}^d$
- Regression case:  $y^{(m)} \in \mathbb{R}$  (*Cont*)
- Classification case:  $y^{(m)} \in [0, k - 1]$  [*k-class classification*]
- If  $k = 2$ , we get *Binary classification* (*Discrete*)



# Examples of Supervised Learning



## Classification

- Spam email detection
- Handwritten digit recognition
- Medical Diagnosis
- Fraud Detection
- Face Recognition

## Regression

- Housing Price Prediction
- Stock Market Prediction
- Weather Prediction
- Market Analysis and Business Trends

# Classification – Medical Diagnosis



## Do Not Have Diabetes

blood glucose = 30

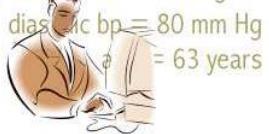
body mass index = 120 kg/m<sup>2</sup>

diastolic bp = 79 mm Hg

age = 32 years



blood glucose = 22  
body mass index = 160 kg/m<sup>2</sup>



blood glucose = 22  
body mass index = 160 kg/m<sup>2</sup>  
bp = 80 mm Hg  
age = 18 years



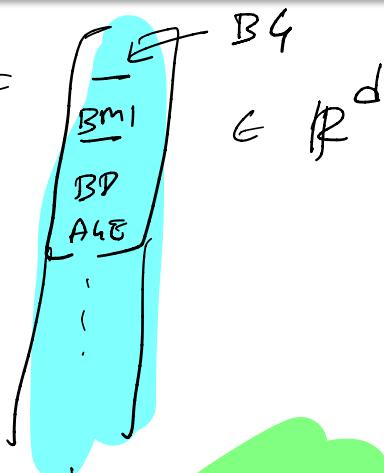
blood glucose = ?  
body mass index = ? kg/m<sup>2</sup>  
diastolic bp = 73 mm Hg  
age = 27 years

① Feature Eng

② Collected Lab Data

③ Train a ML Model.

person =



blood glucose = 40  
body mass index = 150 kg/m<sup>2</sup>  
diastolic bp = 110 mm Hg  
age = 63 years



blood glucose = 45  
body mass index = 180 kg/m<sup>2</sup>  
bp = 95 mm Hg  
age = 49 years

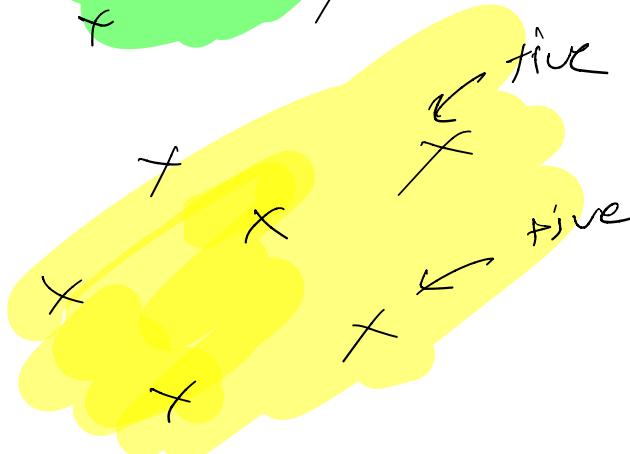


blood glucose = 21  
body mass index = 140 kg/m<sup>2</sup>  
diastolic bp = 99 mm Hg  
age = 37 years

## Have Diabetes

0 → Not Diab

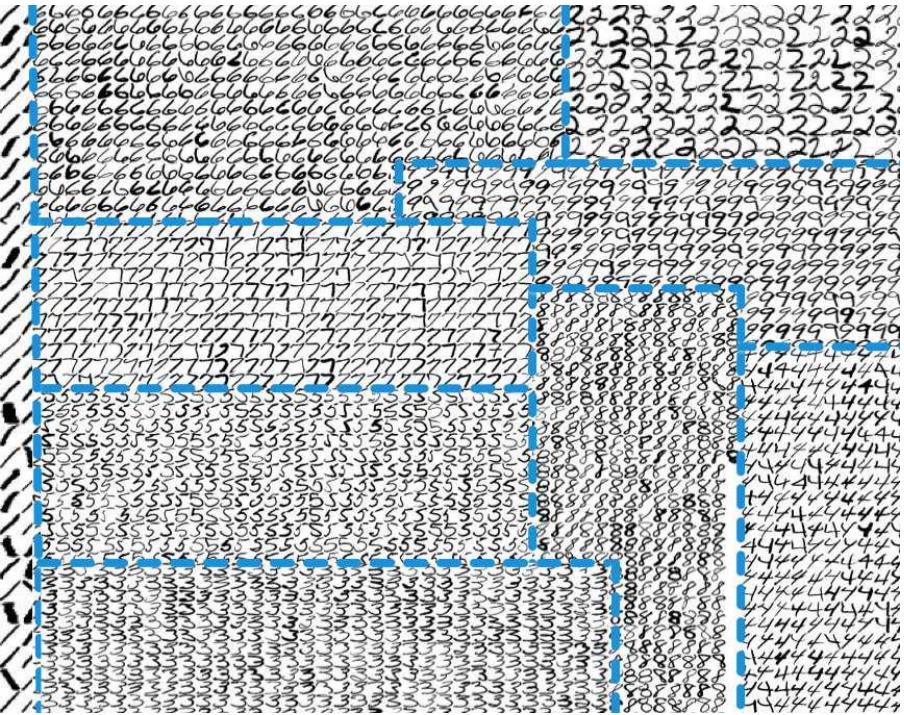
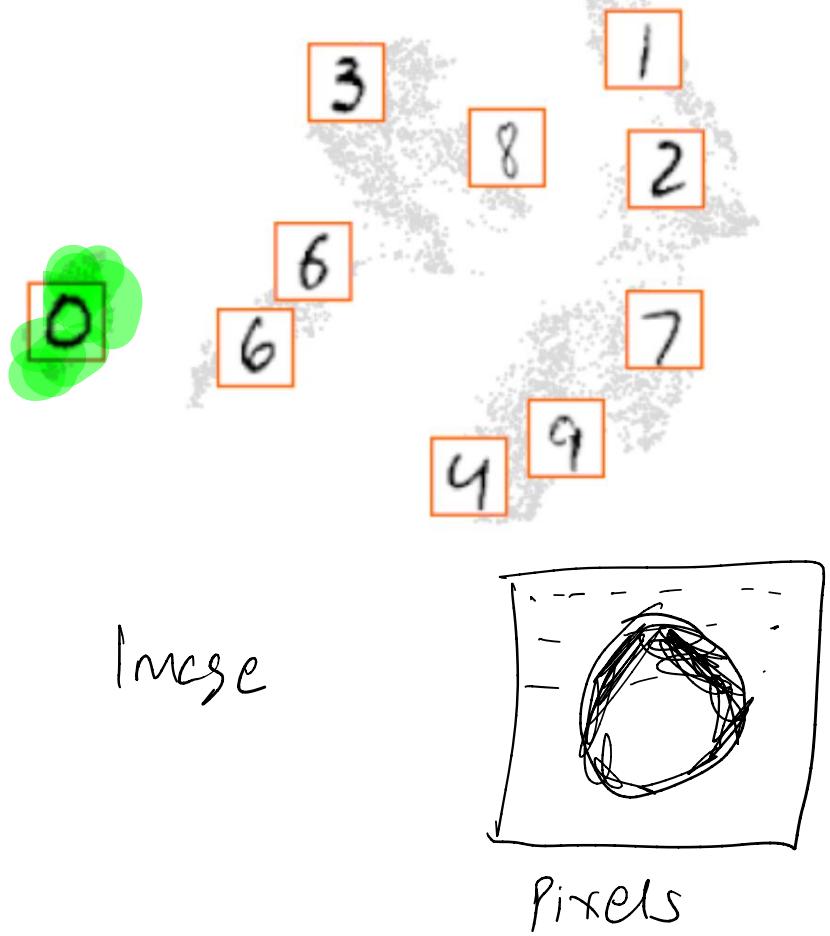
1 → Diab



# Classification – Digit Recognition

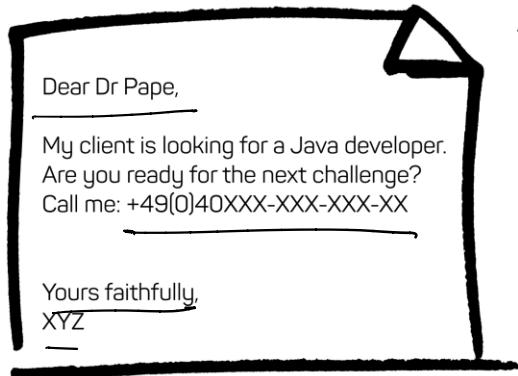


MNIST



$0, 1, \dots, 10$   
 $\underbrace{\hspace{10em}}$   
 $(0 \sim C)_{\text{class}}$

# Classification – Spam

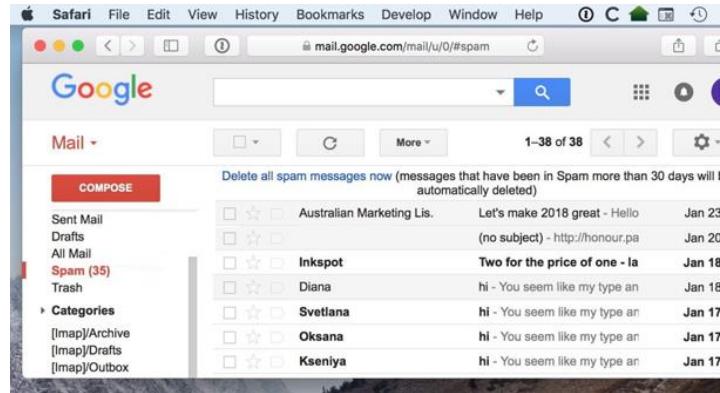


SPAM

vs.

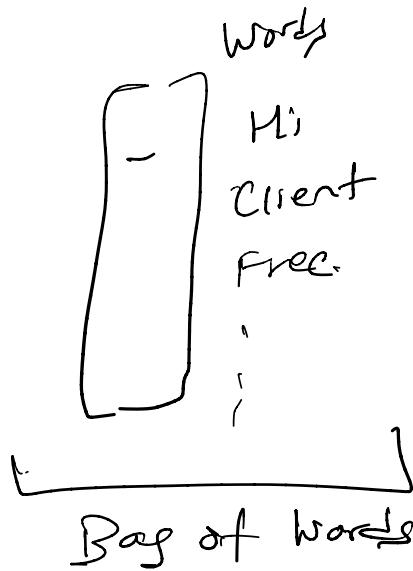
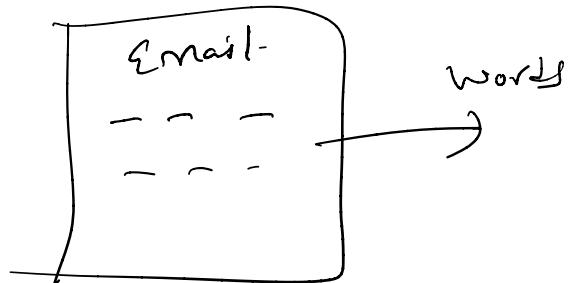


HAM

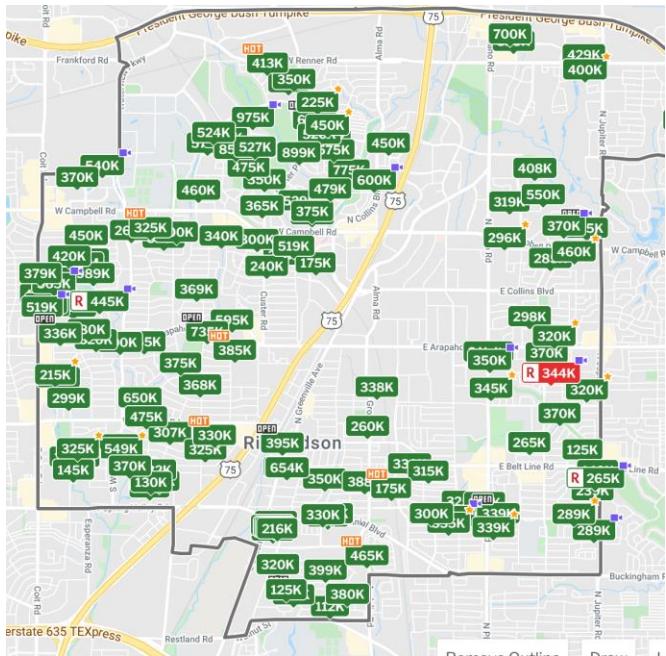


Hi!!  
FREE!!

Label: spam (+1)  
not spam (-1)



# Regression – Housing Price Prediction



$\sim 411000 \$$

Status: Active

Redfin Estimate: \$411,577 On Redfin: 2 days

[Overview](#)
[Property Details](#)
[Property History](#)
[Schools](#)
[Tour Insights](#)
[Public Facts](#)
[Redfin](#)



**NEW 2 DAYS AGO HOT HOME**

**Home Facts**

Status	Active	Time on Redfin	2 days
Property Type	Residential, Single Family	HOA Dues	\$4/month
Year Built	1969	Style	Single Detached, Mid-Century Modern, Ranch, Traditional
Community	Canyon Creek Country Club 9	Lot Size	10,019 Sq. Ft.
MLS#	14375892		

$\begin{cases} SF \\ TH \end{cases}$

Home price  $\approx y$

Features =  $x$

$y = f(x)$

size :  $\frac{1974}{\text{sq. ft.}} (\text{VUIMP})$   
No Beds / No Baths

# Ranking – Search Engines

ranking machine learning



All News Images Videos Shopping More

Settings Tools

About 134,000,000 results (0.77 seconds)

## Scholarly articles for ranking machine learning

Beyond PageRank: machine learning for static ranking - Richardson - Cited by 239

... structures for drug discovery: a new machine learning ... - Agarwal - Cited by 114

... learning and ranking by pairwise comparison - Fürnkranz - Cited by 598

A 5 Ways to make a million dollars without working

8: 10 Reasons you Should Drink Milk Every Morning (you won't believe number 7!)

**Learning to Rank**

Learning to Rank is a machine learning technique used in information retrieval systems to rank documents based on their relevance to a query. It involves training a model to learn the ranking function from labeled training data, where each document is assigned a label indicating its relevance to the query.

The process typically involves the following steps:

- Documents are indexed and stored in an index.
- A user query is input into the system.
- The query is processed by an indexer to find relevant documents.
- The retrieved documents are passed through a ranking model.
- The ranking model uses a learning algorithm to refine its ranking based on feedback.
- The final ranked list of documents is presented to the user.

Learning to rank or machine-learned ranking (MLR) is the application of machine learning, typically supervised, semi-supervised or reinforcement learning, in the construction of ranking models for information retrieval systems.

en.wikipedia.org/wiki/Learning\_to\_rank

## Learning to rank - Wikipedia

About Featured Snippets Feedback

cs.nyu.edu/~mohri/mls/ml\_ranking PDF

## Foundations of Machine Learning Ranking - NYU Computer ...

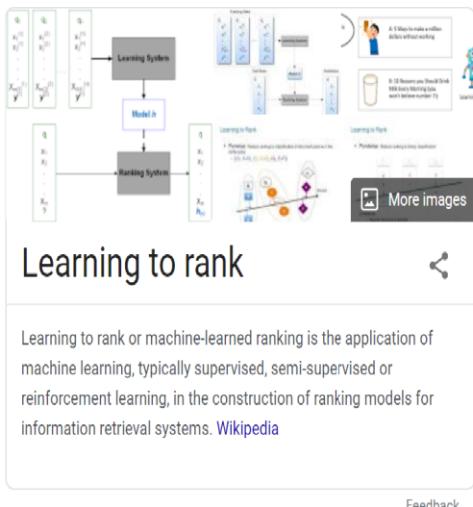
Mehryar Mohri - Foundations of Machine Learning. Motivation. Very large data sets: • too large to display or process. • limited resources, need priorities. • ranking ...

URL

Title

Text

feature



Label =

P,

P<sub>2</sub>,

P<sub>10</sub>

Rank 1

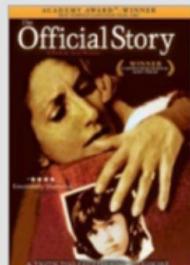
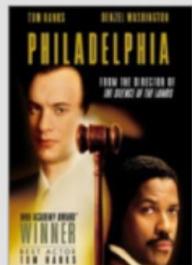
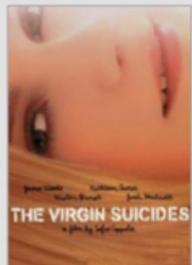
Rank 2<sub>24</sub>

# Recommendation – Movie Recommendations



## Friends' Favorites

Based on these friends:



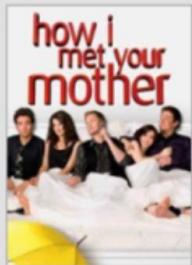
## Watched by your friends

Daniel Jacobson

John Ciancutti

Mark White

mike Kail

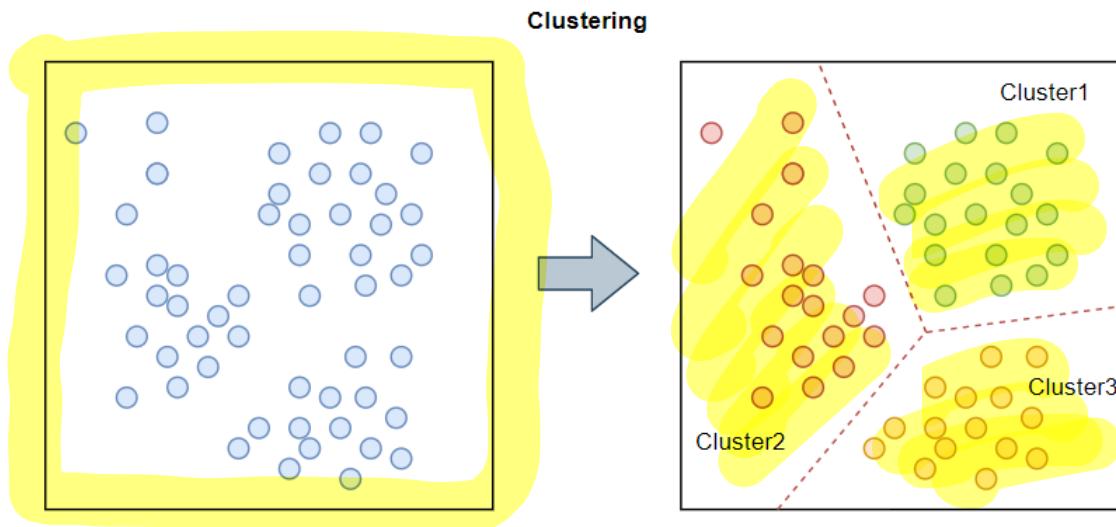


① My Past (History)      ② Friends (similar Movies)      {  
Recommendations}

# Unsupervised Learning



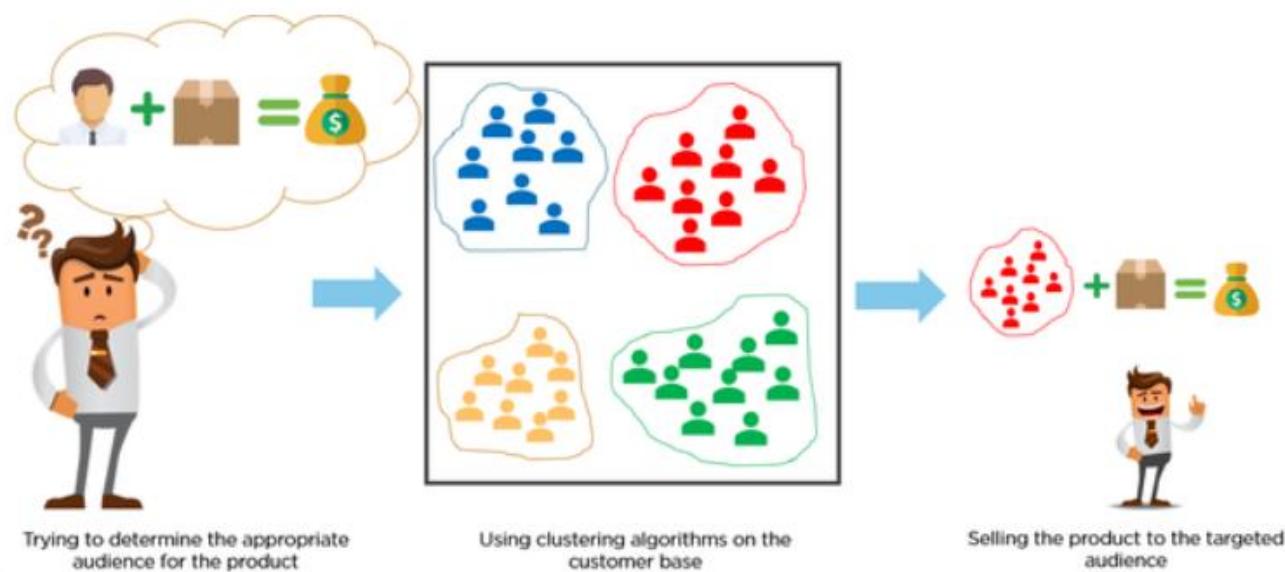
- **Input:**  $\underbrace{x^{(1)}, \dots, x^{(M)}}_{}$ 
  - $x^{(m)}$  is the  $m^{th}$  data item
  - **No Label!**
- **Goal:** find a clustering/grouping of data points into  $k$  clusters so that each cluster consists of similar points



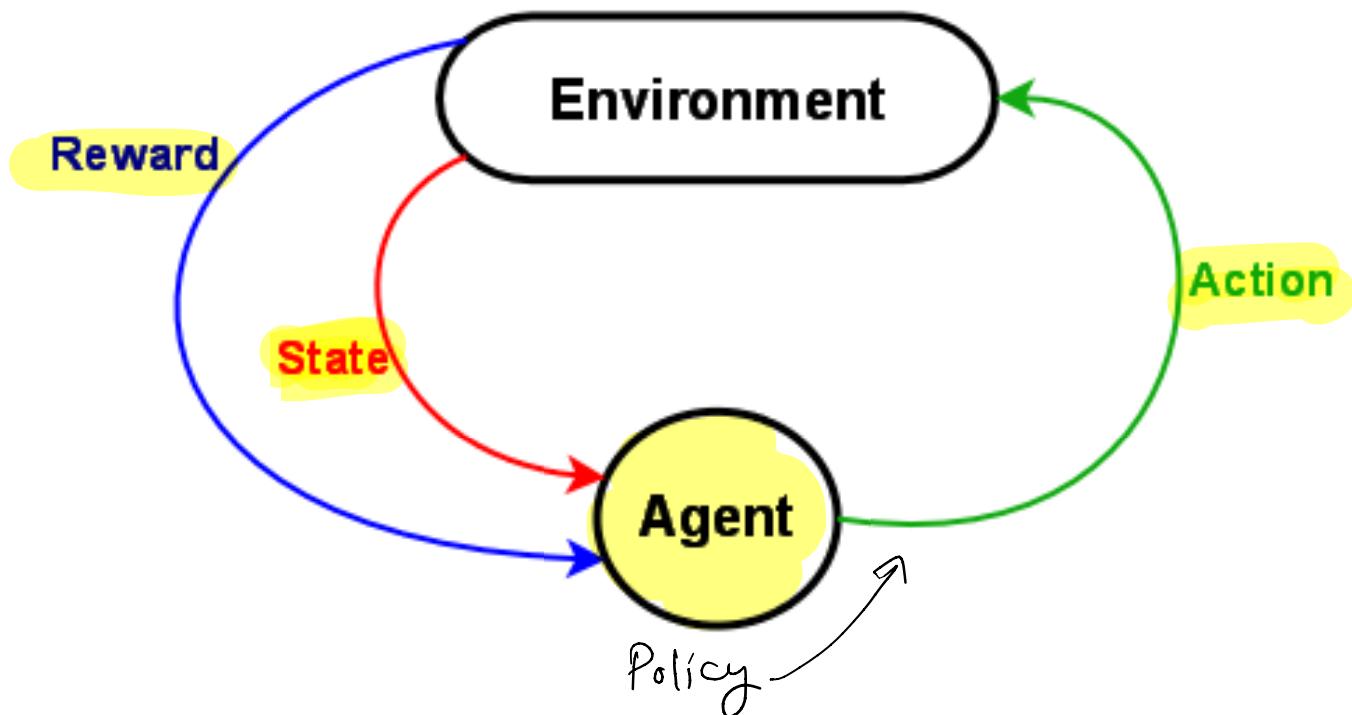
# Applications of Unsupervised Learning



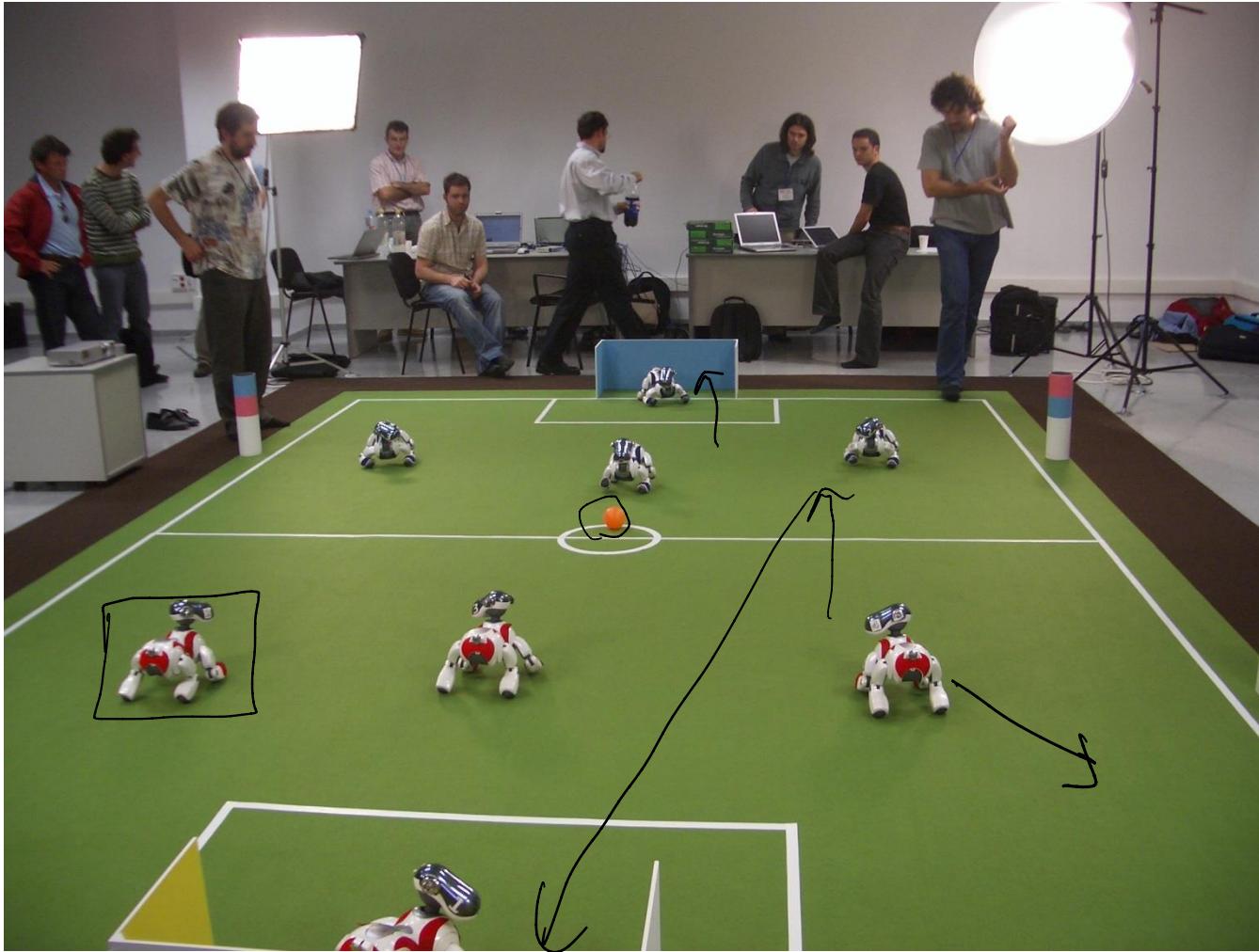
- Item Categorization
- Clustering Customers
- Similar Item Recommendation
- Outlier Detection



# Reinforcement Learning



# Reinforcement Learning – Robocup Soccer



# Other Types of Learning

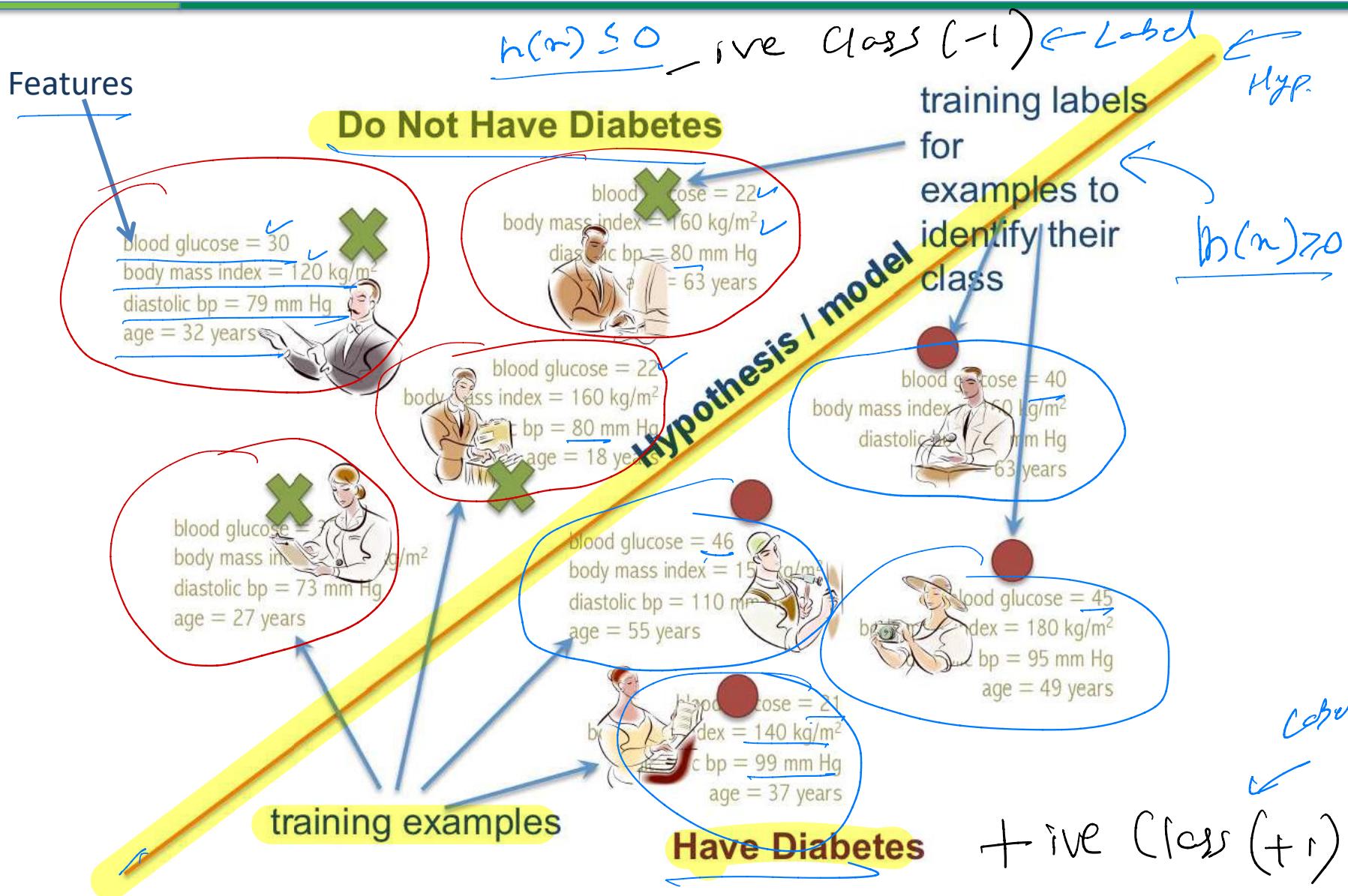
---

- Semi-supervised
  - Training Labeled + Unlabeled Data Jointly
- Active learning
  - Semi-supervised learning where the algorithm can ask for the correct outputs for specifically chosen data points
- Online Learning
  - Data and Labels coming in a stream
- Reinforcement learning
  - The learner interacts with the world via allowable actions which change the state of the world and result in rewards
  - The learner attempts to maximize rewards through trial and error

## Part IV

Terminology, Loss Functions  
& Evaluation Criteria.

# Terminology



# Terminology



$$y_i = f^*(n_i)$$

$$\{(n_1, y_1), \dots, (n_m, y_m)\}$$

- **Training Example:  $\langle x, y \rangle$**

- **x:** feature vector (describes the attributes of something) ↗ Training Set
- **y:** label (continuous values for regression problems:  $[1, 2, \dots, k]$  for classification problems)

- **Training set** A set of training examples drawn randomly from  $P(x, y)$

- **Key Assumption:** Independent and identically distributed. i.e., all the examples are drawn from the same distribution but are drawn independent of one another

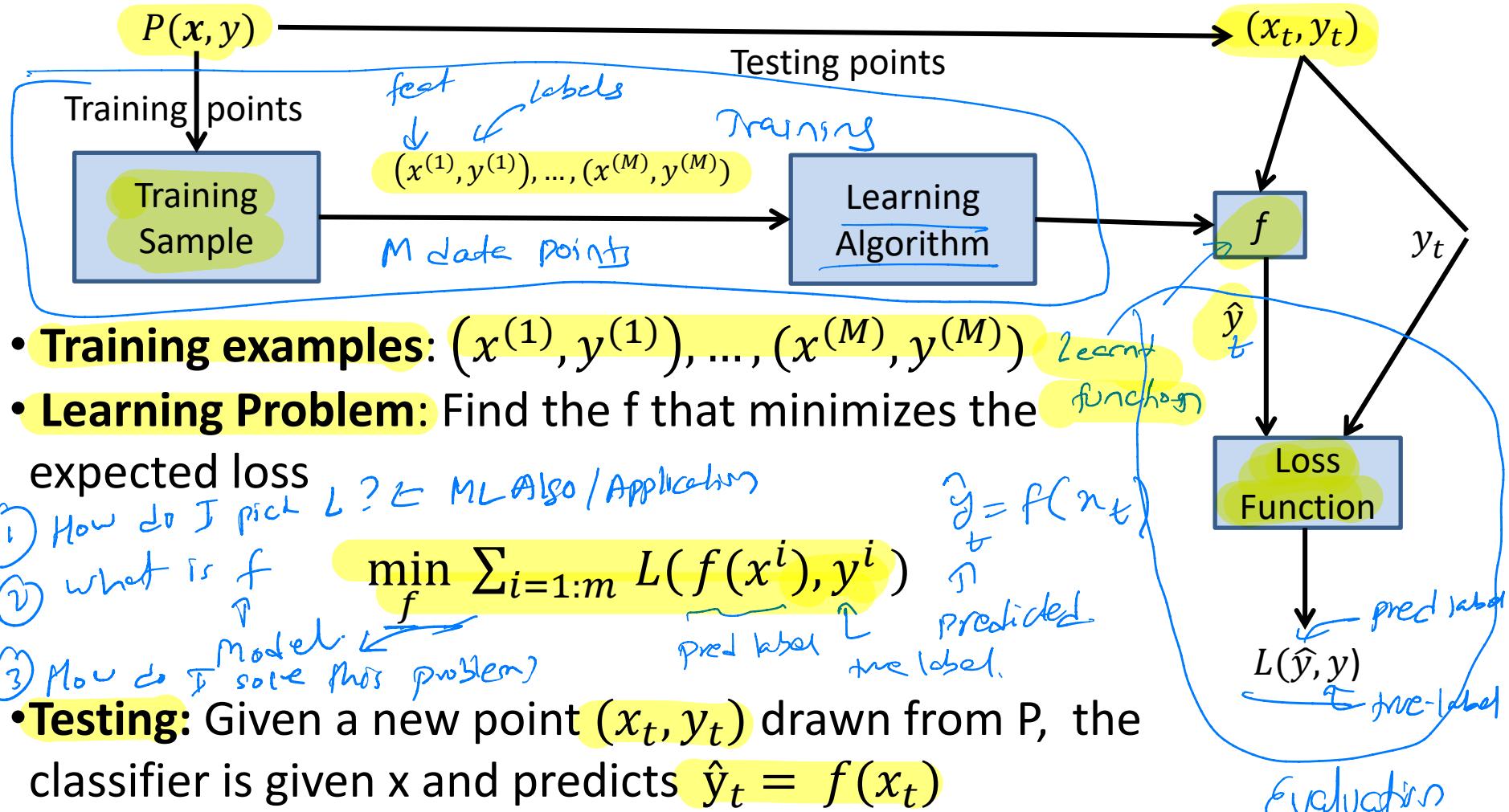
- **Target function** True mapping from  $x$  to  $y$   $f^*(n) = y$

- **Hypothesis:** A function  $h$  considered by the learning algorithm to be similar to the target function  $h$   $st$   $f^*(n) = h(n)$

- **Test set:** A set of examples drawn from  $P(x, y)$  to evaluate the "goodness of  $h$ " ↗ Unseen  $= y$ .

- **Hypothesis Space:** The space of all hypotheses that can in principle be considered and returned by the learning algorithm Space of  $h$

# Supervised Learning Workflow



- **Training examples:**  $(x^{(1)}, y^{(1)}), \dots, (x^{(M)}, y^{(M)})$
  - **Learning Problem:** Find the  $f$  that minimizes the expected loss
- ① How do I pick  $L$ ?  $\in$  ML Algo / Application
- ② what is  $f$
- ③ How do I solve this problem?  $\rightarrow$  Model
- $$\min_f \sum_{i=1:m} L(f(x^i), y^i)$$
- $\hat{y} = f(x_t)$
- ↑  
predicted  
the label.
- **Testing:** Given a new point  $(x_t, y_t)$  drawn from  $P$ , the classifier is given  $x$  and predicts  $\hat{y}_t = f(x_t)$
- **Evaluation:** Measure the error  $Err(\hat{y}_t, y_t)$  – often same as  $L$

# Linear Regression



## • Simple linear regression

$$\hat{a}^T \mathbf{x} = a \cdot \mathbf{x}$$
$$f(\mathbf{x}) = a \cdot \mathbf{x} + b$$

- Input: pairs of points  $(x^{(1)}, y^{(1)}), \dots, (x^{(M)}, y^{(M)})$  with  $x^{(m)} \in \mathbb{R}^d$  and  $y^{(m)} \in \mathbb{R}$  (Regression)
- Hypothesis space: set of linear functions  $f(x) = a^T x + b$  with  $a \in \mathbb{R}^d, b \in \mathbb{R}$  Parameterized
- Error metric and Loss Function: squared difference between the predicted value and the actual value

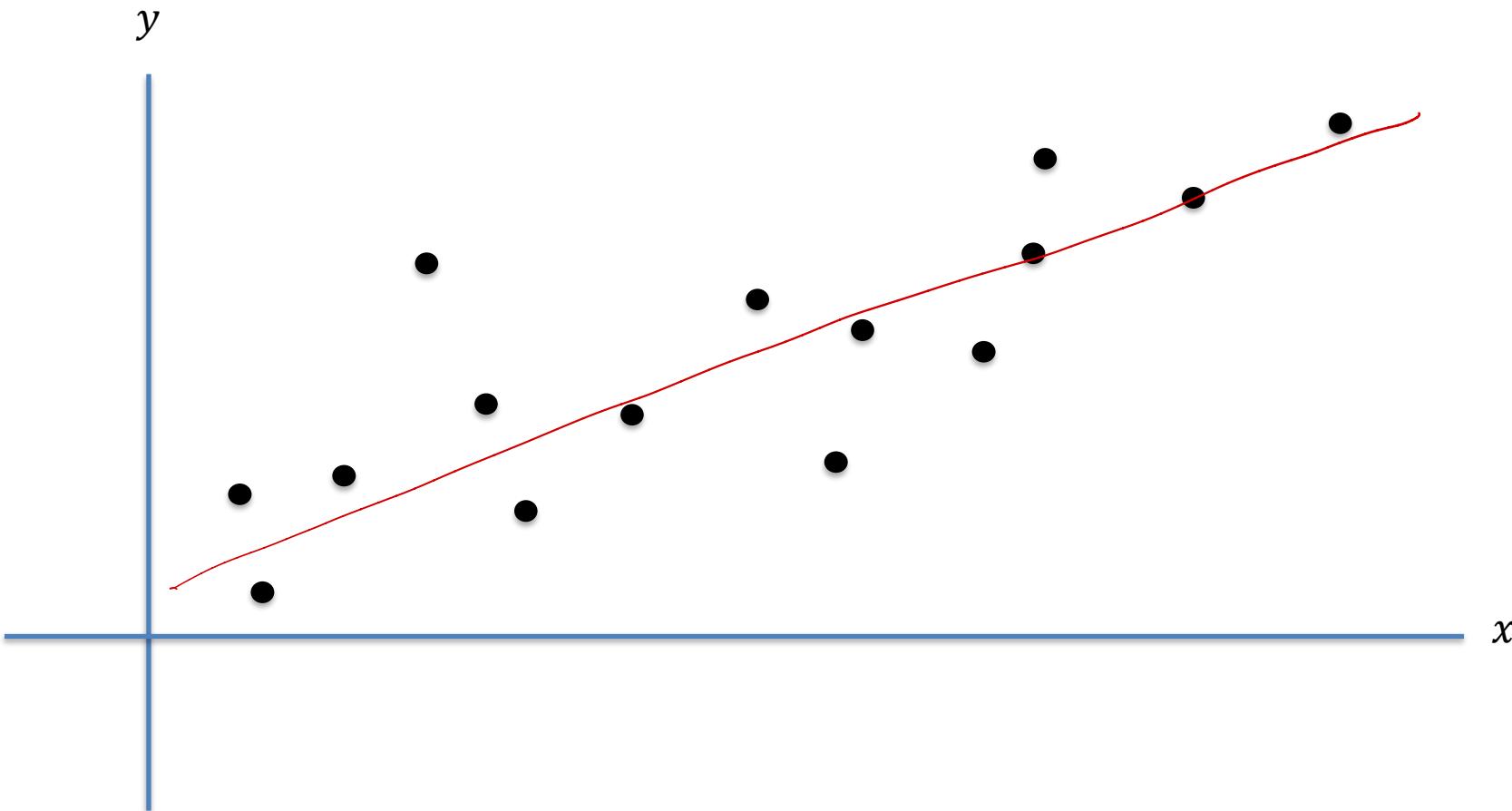
$$f(\mathbf{x}) = a \cdot \mathbf{x} + b, \quad y$$

$$L(f(\mathbf{x}), y) = [f(\mathbf{x}) - y]^2$$

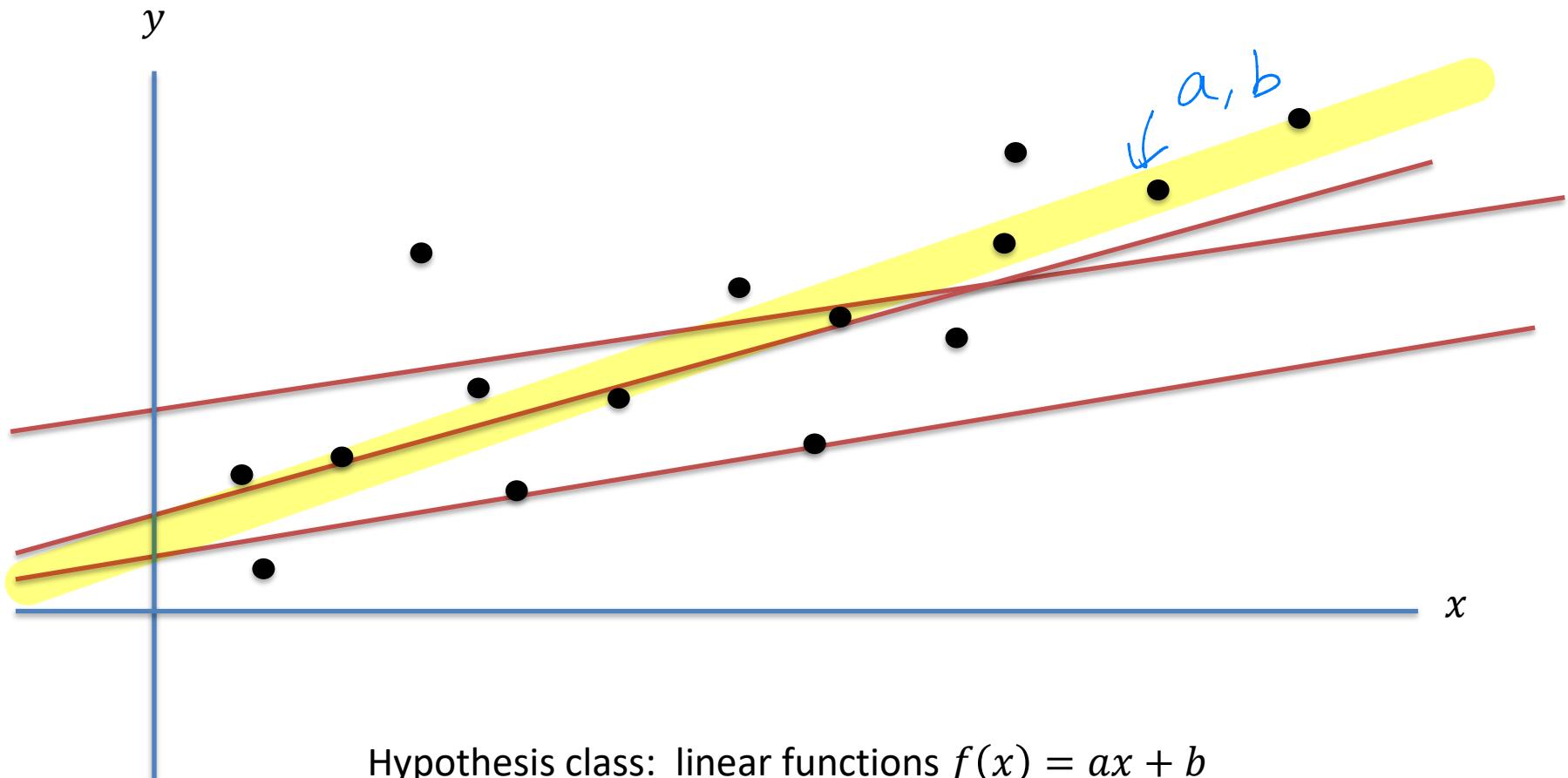
↑  
23      ↑  
25      4

$$\min_{a, b} \sum_{i=1}^m [a \cdot \mathbf{x}_i + b - y_i]^2$$
$$a \in \mathbb{R}^d, b \in \mathbb{R}$$

# Regression



# Regression



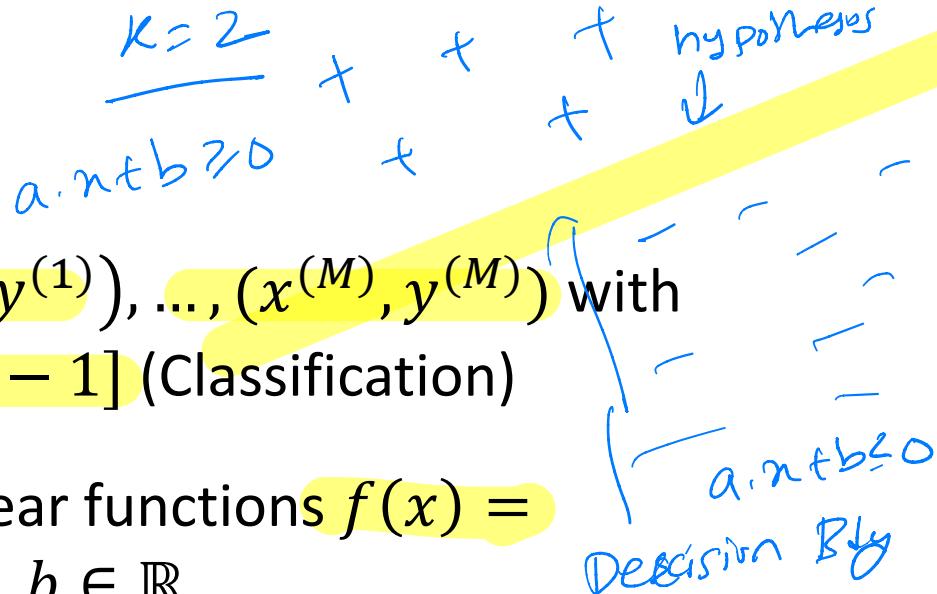
How do we compute the error of a specific hypothesis?

# Linear Classification



- Simple linear classification

- Input: pairs of points  $(x^{(1)}, y^{(1)}), \dots, (x^{(M)}, y^{(M)})$  with  $x^{(m)} \in \mathbb{R}^d$  and  $y^{(m)} \in [0, k - 1]$  (Classification)
- Hypothesis space: set of linear functions  $f(x) = \text{sign}(a^T x + b)$  with  $a \in \mathbb{R}^d, b \in \mathbb{R}$
- Error metric: Accuracy (or more complex like AUC, ...)
- Loss Function: Log Loss, Hinge Loss, Perceptron Loss...



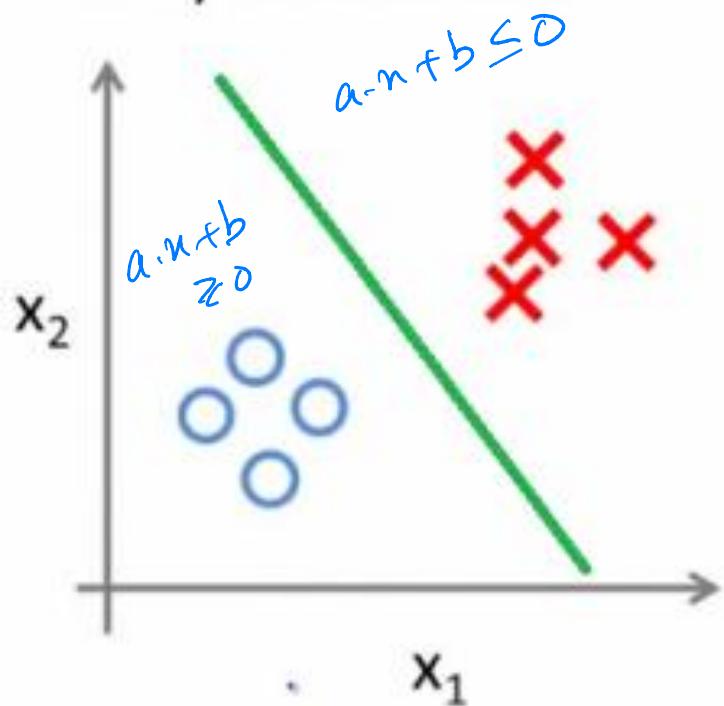
$$\begin{aligned} f(n) &= +1 && \text{if } a \cdot n + b \geq 0 \\ &= -1 && \text{if } a \cdot n + b \leq 0 \end{aligned}$$

↑ Relaxations of Accuracy

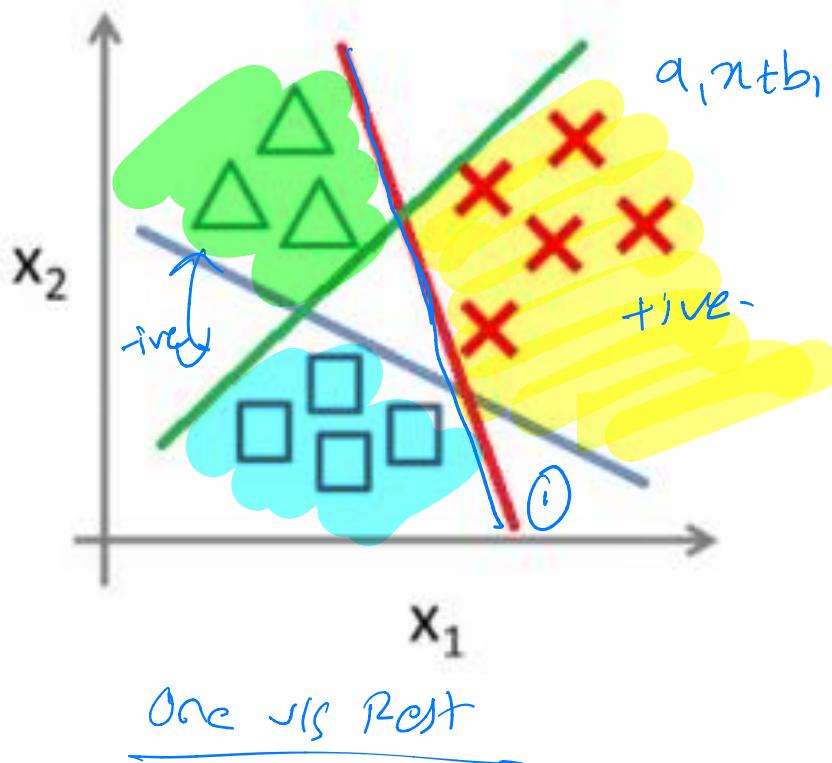
# Linear Classification



Binary classification:



Multi-class classification:



# Binary Classification

- Regression operates over a continuous set of outcomes
- Suppose that we want to learn a function  $f: X \rightarrow \{0,1\}$
- As an example:

*Feats*                    *label*

	$x_1$	$x_2$	$x_3$	$y$
1	0	0	1	0 -1
2	0	1	0	1 +1
3	1	1	0	1 +1
4	1	1	1	0 -1

How many functions with three binary inputs and one binary output are there?

# Binary Classification



	$x_1$	$x_2$	$x_3$	label
	0	0	0	?
1	0	0	1	0
2	0	1	0	1
	0	1	1	?
	1	0	0	?
	1	0	1	?
3	1	1	0	1
4	1	1	1	0

4 data points

	$x_1$	$x_2$	$x_3$	label
	0	0	0	?
1	0	0	1	0
2	0	1	0	1
	0	1	1	?
	1	0	0	?
	1	0	1	?
3	1	1	0	1
4	1	1	1	0

$2^8$  possible fs

$2^8$  possible functions

$2^4$  are consistent with the observations

How do we choose the best one?

What if the observations are noisy?

# Challenges in ML

- How to choose the right hypothesis space?
  - Number of factors influence this decision: difficulty of learning over the chosen space, how expressive the space is,  
...  
*Simpler*
- How to evaluate the quality of our learned hypothesis?
  - Prefer “simpler” hypotheses (to prevent overfitting)
  - Want the outcome of learning to **generalize** to unseen data
- Computational Tractability
- Can we trust the results? Explainability!

# Challenges in ML



- How do we find the best hypothesis?  
• This can be an NP-hard problem!
  - Need fast, scalable algorithms if they are to be applicable to real-world scenarios
- How easy is it to optimize over hypotheses?
- f
- ① Relaxations → Computationally tractable
- ② Scalability