Machine Learning : Lecture -1

Introduction

Topics (Part -1)

1. Linear modelling:

Least square (LS)
Non-linear response from linear models
Generalization versus overfitting
Regularized LS: L2, L1 regularization (LASSO)

2. Maximum likelihood (ML) approach

3. Classifiers:
Probabilistic Classifier: Bayes classifier, Logistic regression
Non-Probabilistic classifier: K-nearest neighbours

4. Decision Trees

5. Random Forests

6. Gradient Boosting

Tutorials: (Mon 5.00 -6.30 PM)

R-102 – Batch 1
R-104 – Batch 2
R-105 – Batch 3
R-110 – Batch 4

Text Books:
1 , 2, 3: A first Course on Machine Learning by Simon Rogers
4,5, 6: To be decided later

Reference:
Machine Learning: Foundations, Methodologies, and Applications by Alexander Jung

Assessment Plan: Part 1 (50 Marks)

Class Quiz (After Aug 31) : 10 Marks

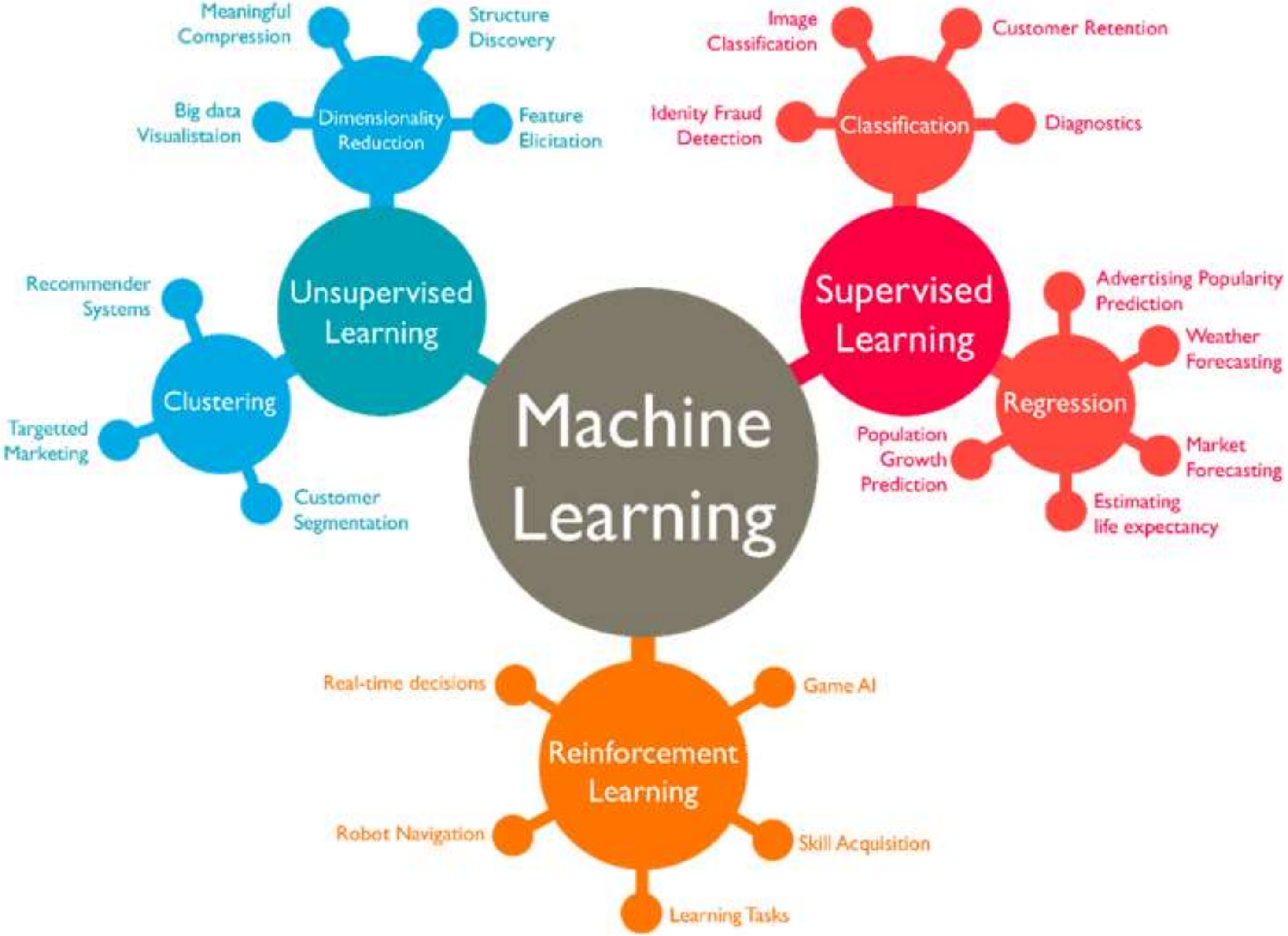
Assignment – 1 (After Sep 15) : 10 Marks

Mid-term: 20 Marks

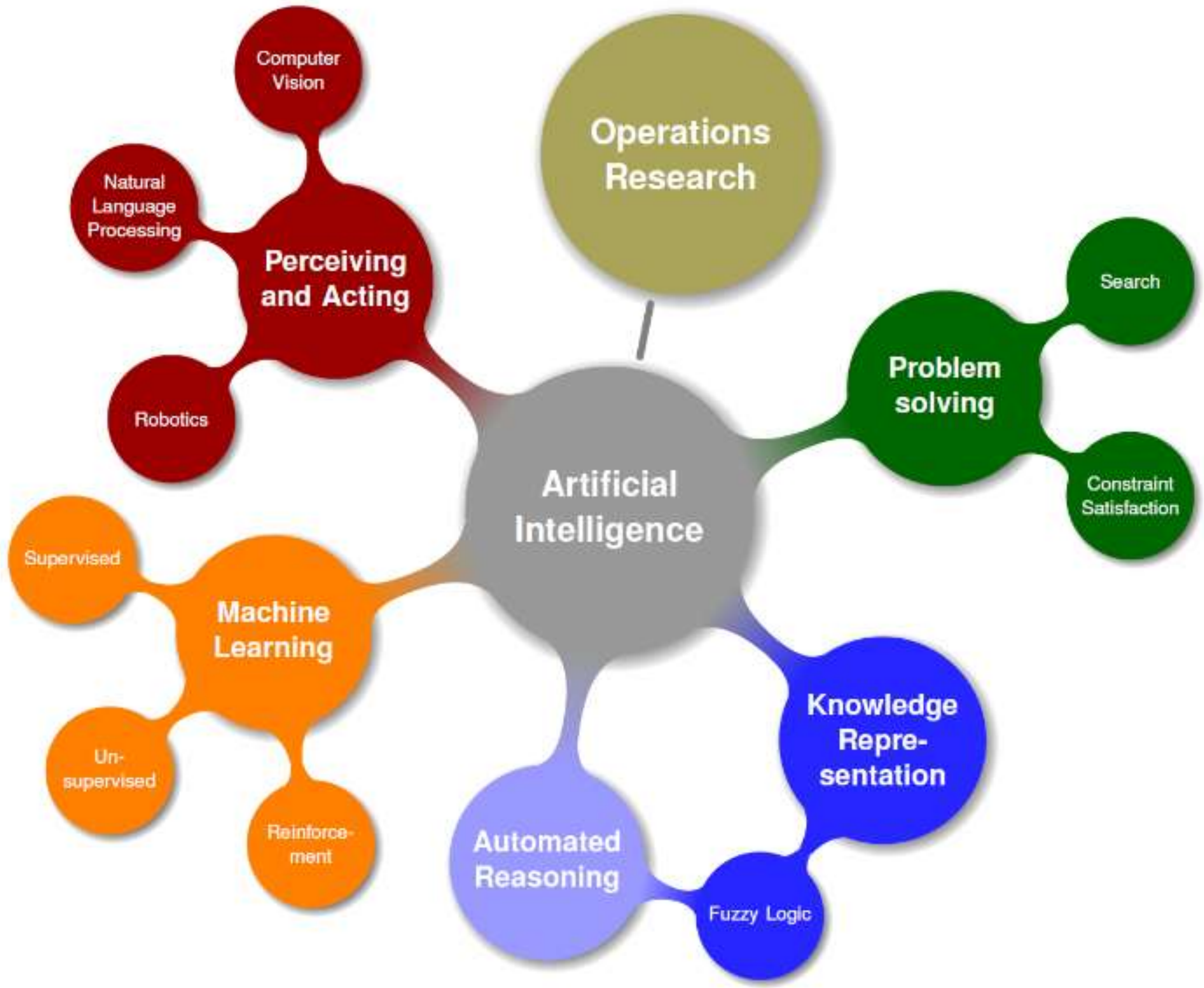
Tutorial Assessment (Multiple assessments spread across random tutorial days): 5 Marks

Class Participation: 5 Marks

Machine Learning



Machine Learning and Artificial Intelligence



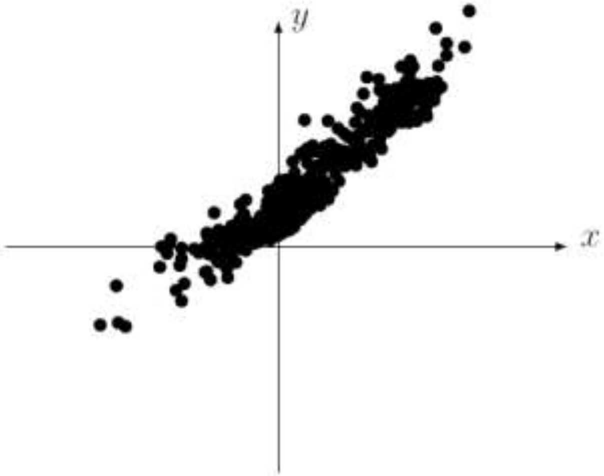
A Simple Learning Example

Objective: Predict the maximum day temperature after observing the temperature at 7 AM.

Observations:

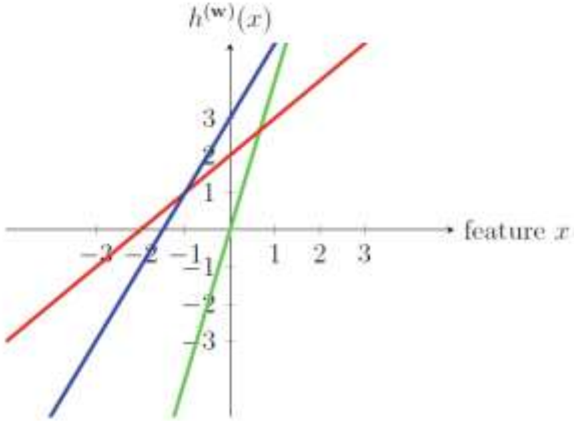
x : Temperature at 7 AM

y : Maximum Day Temperature



How can we find suitable w_1 and w_0?

Sample hypotheses: $h(x) := w_1 x + w_0 \quad w_1 \in \mathbb{R}_+, w_0 \in \mathbb{R}.$

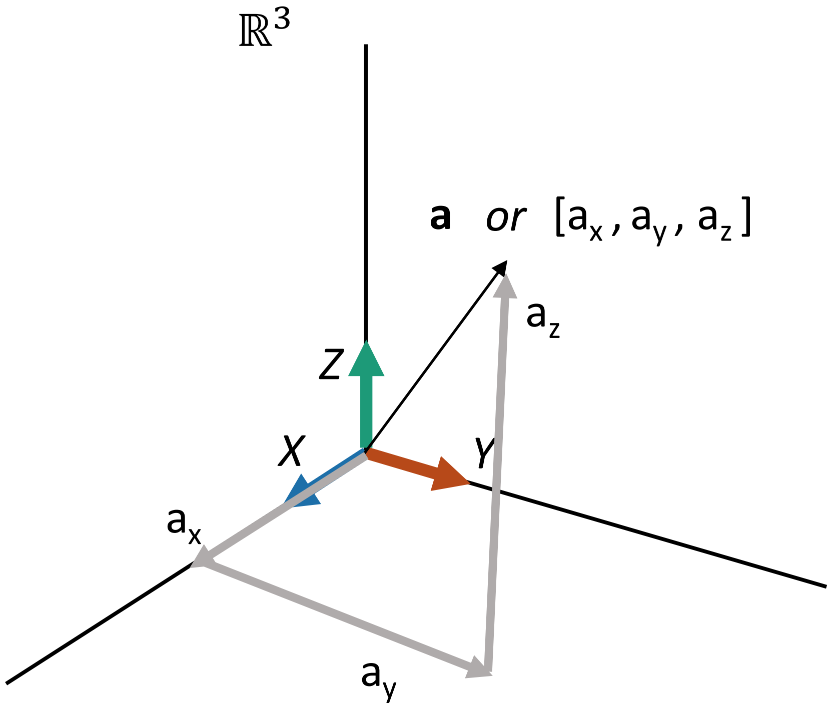


Model:

$$y \approx w_1 x + w_0$$

ML Objective: Find the “best” hypothesis from the set of feasible hypotheses

[ML & Linear Algebra] : Vector Representation



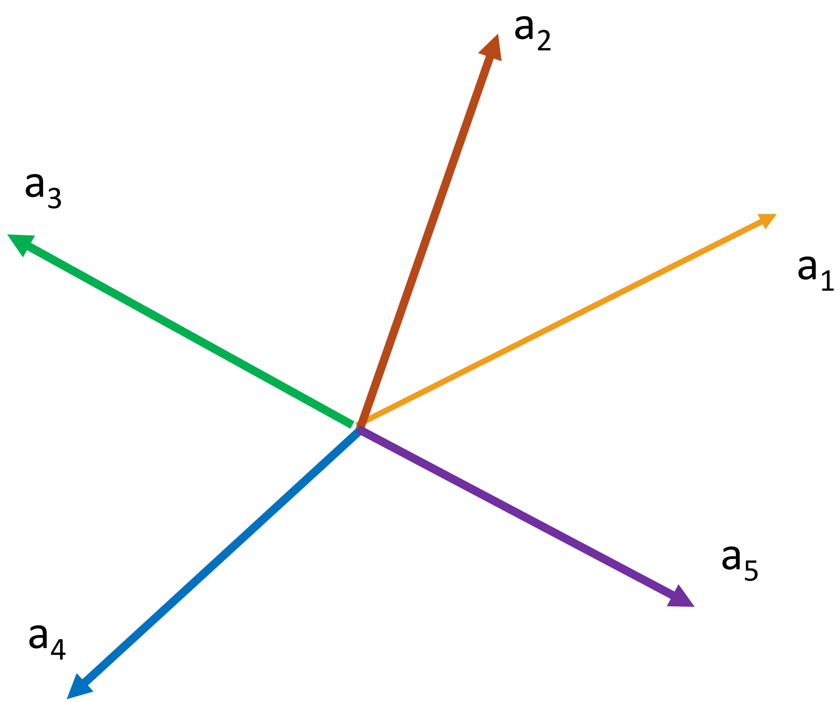
$a \in \mathbb{R}^3$

$[a_x, a_y, a_z]$ ➡ Point in 3- dimensional Space / 3-dimensional Vector / Data

[ML & Linear Algebra] : Data in Multi-Dimensional Space

5 –Dimensional Space ($\mathbb{R}^5$)

(Do not bother to Imagine !
Look only Algebraic way)



$$a_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad a_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad a_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

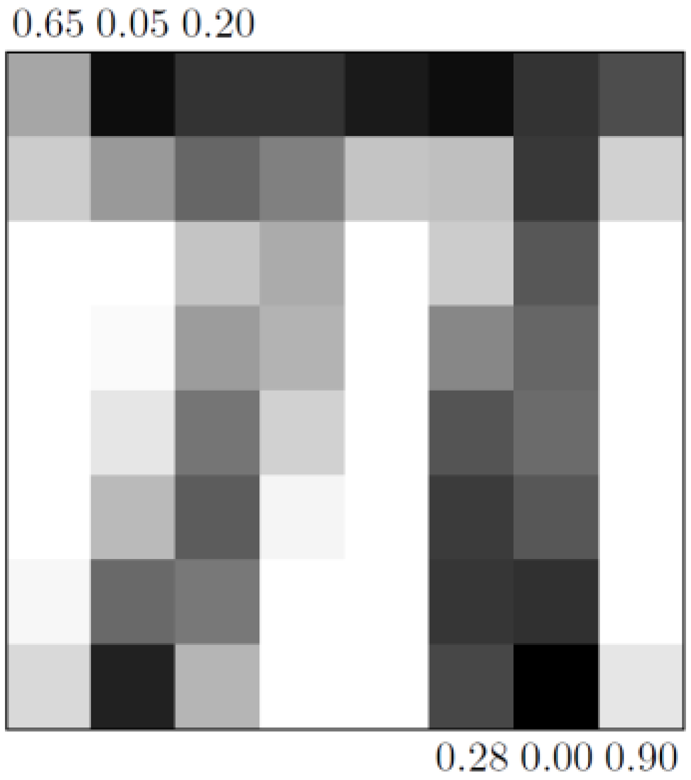
$$a_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad a_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Dimension	Meaning	Value
a_1	Height (ft)	6
a_2	Age	30
a_3	Weight (kg)	70
a_4	Waist-Size(in)	32
a_5	Gender	1

5- Dimensional Data

$\mathbf{a}$ = [6 , 30 , 70 , 32, 1] $\Rightarrow$ $\mathbf{a} = 6\mathbf{a_1} + 30\mathbf{a_2} + 70\mathbf{a_3} + 32\mathbf{a_4} + \mathbf{a_5}$

[ML & Linear Algebra] : Image Representation



8×8 image

8×8 image can be represented as 64 dimensional vector

$x = [0.65 \;\; 0.05 \; 0.20 \;\; \ldots \ldots \ldots . \, 0.28 \; 0.00 \; 0.90 \,]$

x is a point in 64-dimension space, $x \in \mathbb{R}^{64}$

Videos:

k frames in a video of resolution $m \times n$ can be represented as as $m \times n \times k$ vector

$x \in \mathbb{R}^{m\times n \times k}$

Poll : $x \in \mathbb{R}^{m\times n \times k}$ & $x \in \mathbb{R}^{m\times k \times n}$ represents the same video ?

[ML & Linear Algebra] : Document Representation

Documents:

D1 : I like deep learning

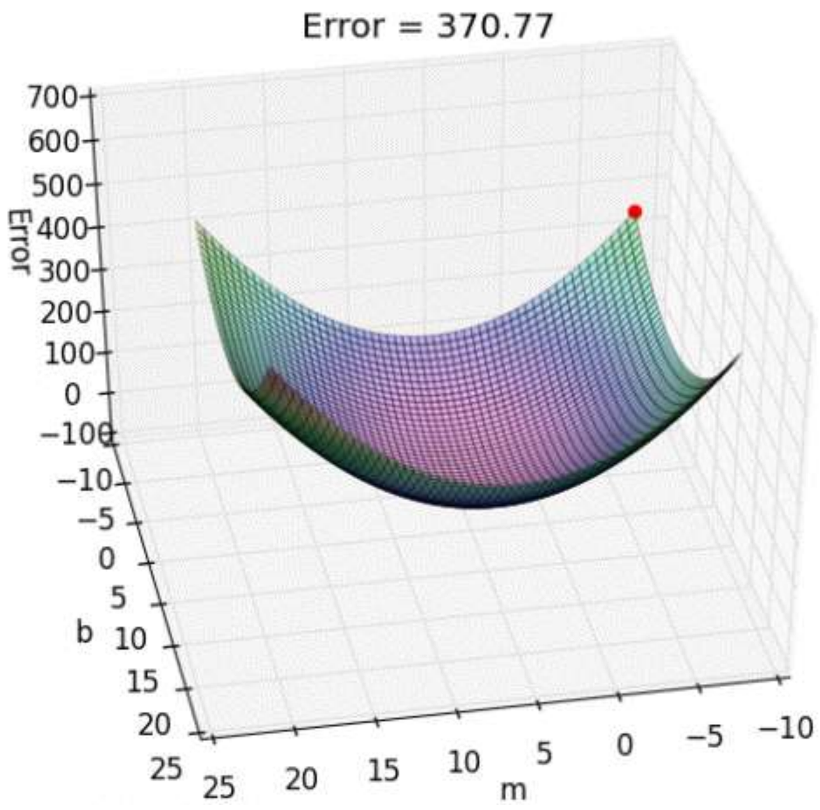
D2 : I like NLP

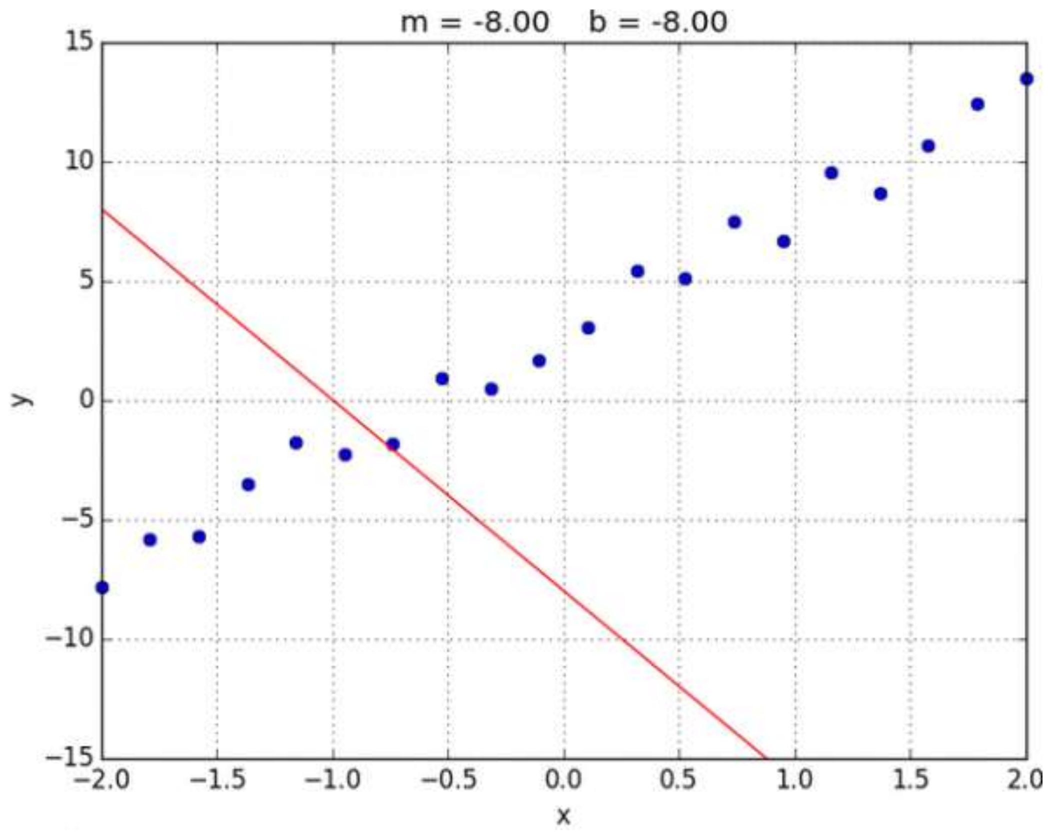
D3 : I enjoy flying

Word Count Histogram:

counts	D1	D2	D3
I	1	1	1
like	1	1	0
enjoy	0	0	1
deep	1	0	0
learning	1	0	0
NLP	0	1	0
flying	0	0	1
.	1	1	1

[ML & Optimization] Cost Function Minimization





Line Fitting: Loss function minimization with two parameters

[ML & Probability]

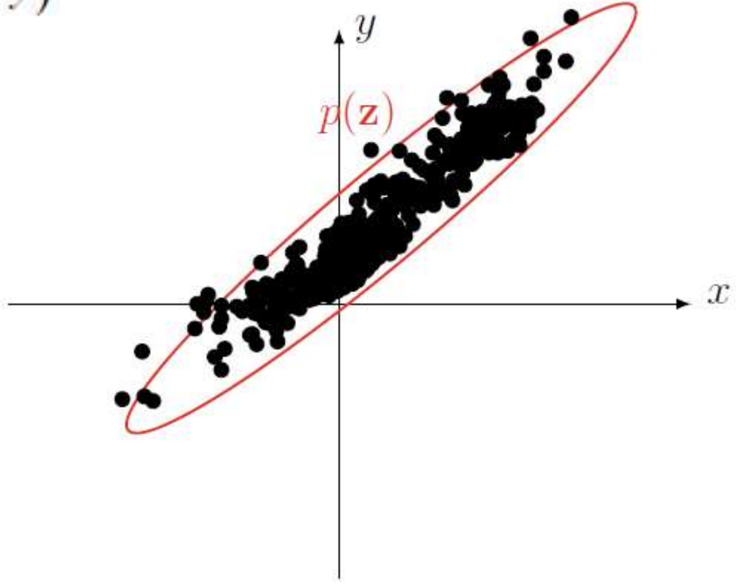
What if we can model the joint probability distribution of data and labels ?

What if we can additionally sample new data from this joint distribution ?

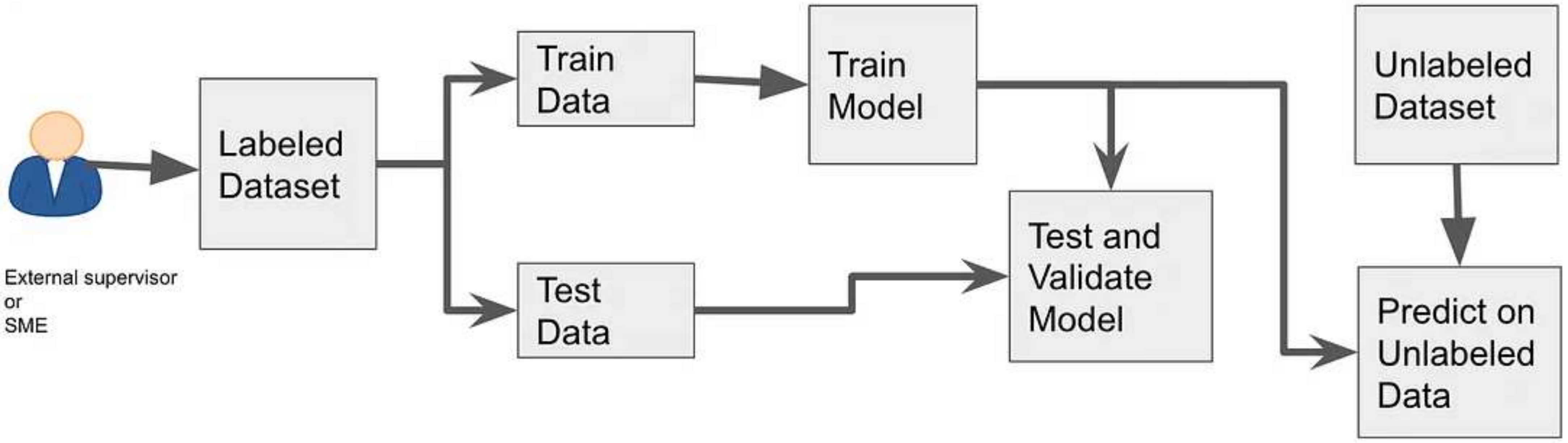
Generative Models

7 AM Temperature vs. Max Day temperature:

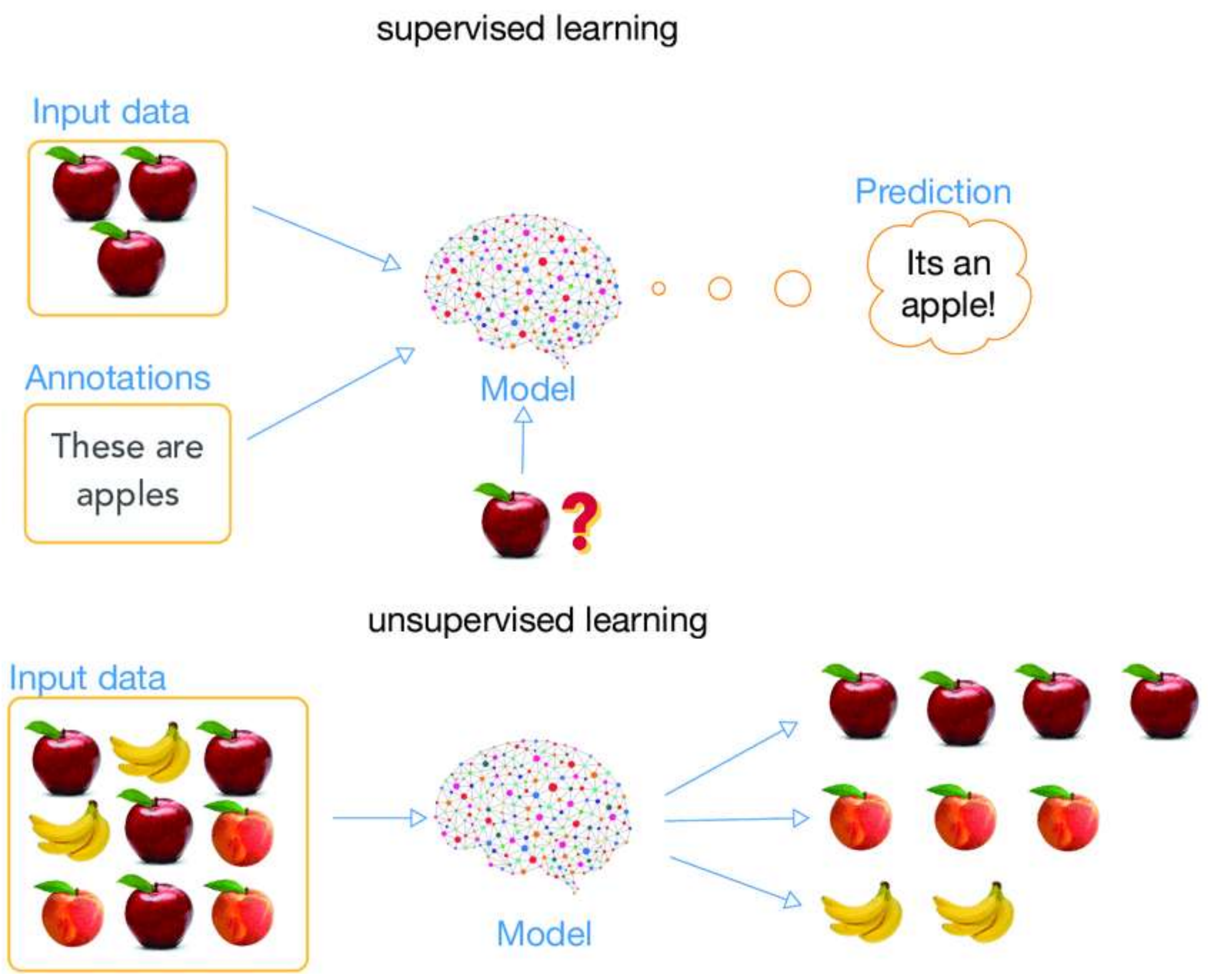
$\mathbf{z} = (x, y)$

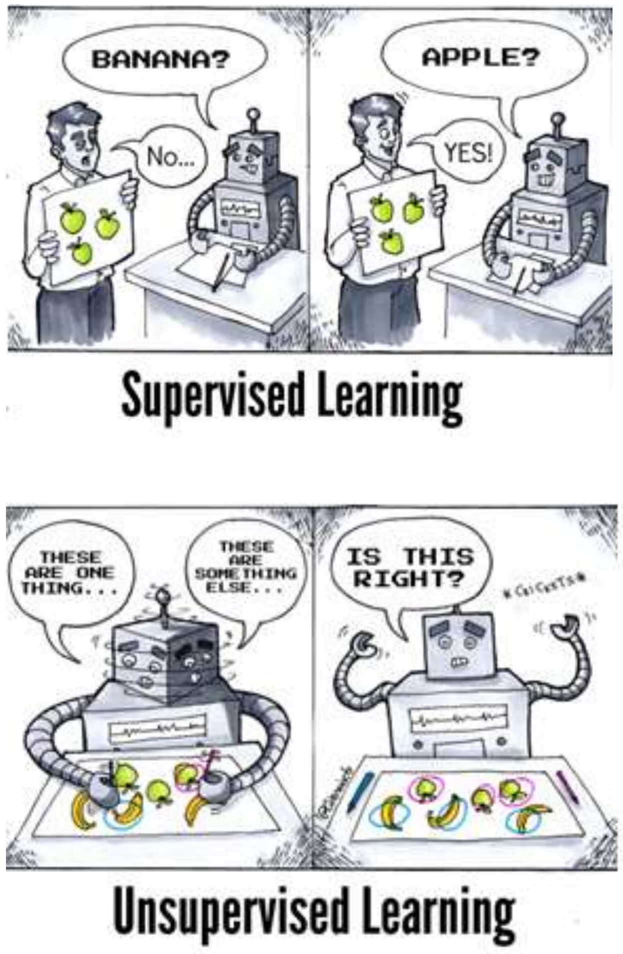


[ML Flavours]: Supervised Learning

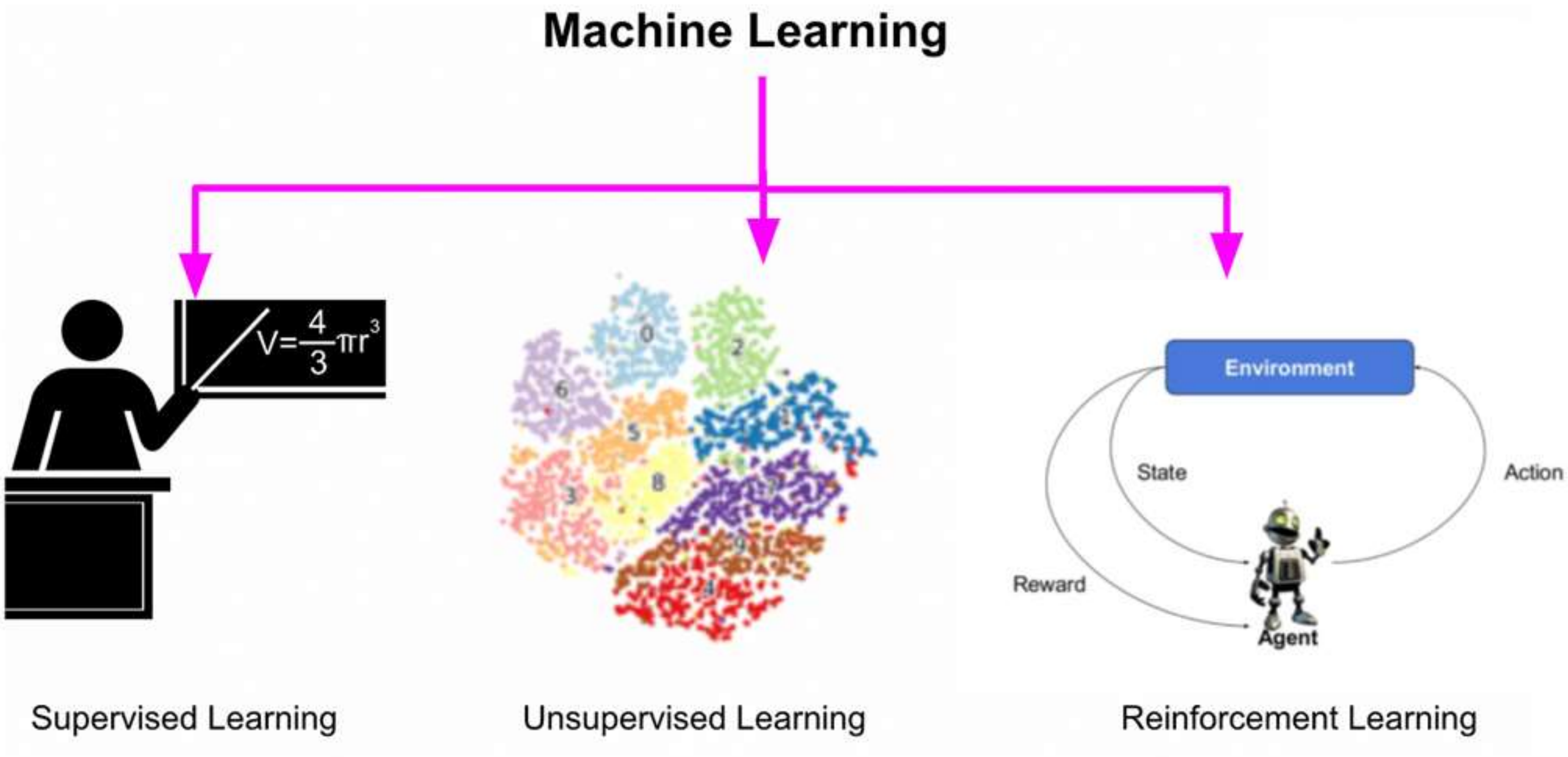


[ML Flavours]: Supervised vs Un-supervised Learning

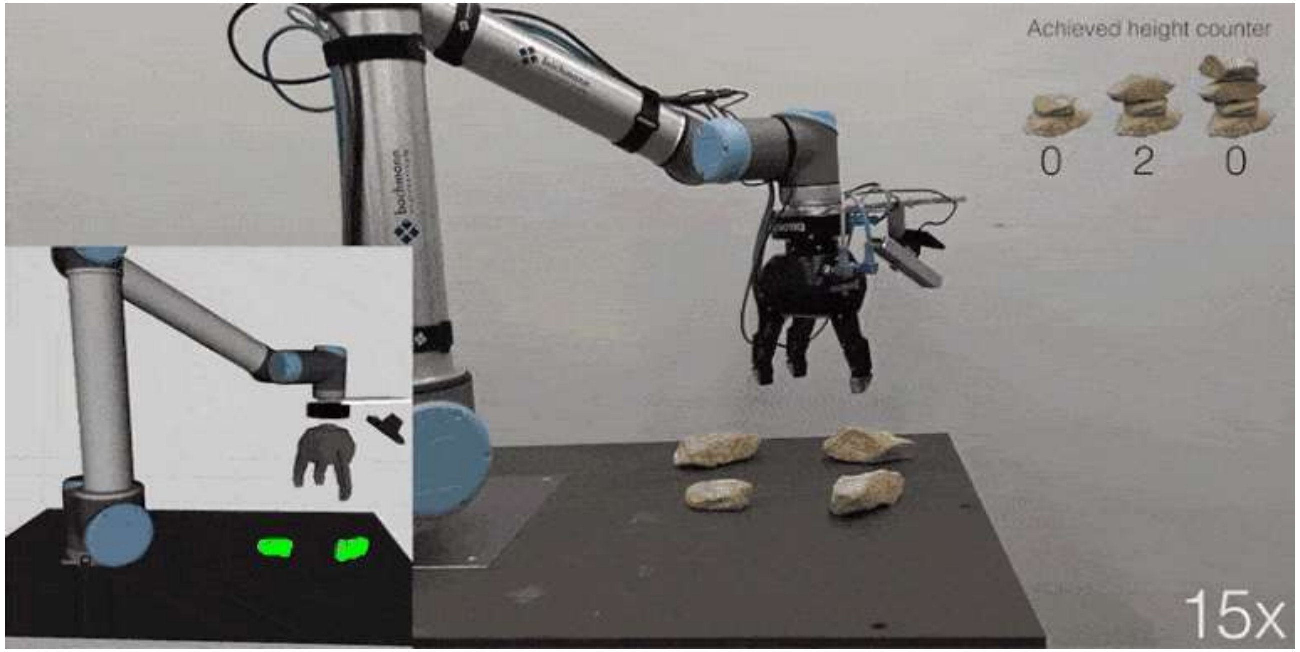




[ML Flavours]: Supervised vs Un-supervised Learning vs Reinforcement Learning



[ML Flavours]: Reinforcement Learning Example



Learning to assemble using robotic arms

Example:

Warehouse automation : picking objects of different size and shape

States:
-*Robot State*: joint angles, joint velocities, the end-effector pose
-*Ref Object State:* object positional information

Actions:
-*Torque on various joints*

Reward:
-1 for success
-0 for failure