MLF Week-1 Open Session Notes

1. (1) What is Machine learning?
(2) Data, Model & ML Tasks
(3) Supervised learning: Regression
(4) ———————— : Classification
(5) Unsupervised learning: dimensionality Reduction
(6) ———————— : density Estimation

---

three main parts:
↳ (1) Linear Algebra
↳ (2) Probability
↳ (3) Optimization

Traditional prog:- Input + Rules → output

2.

ML:- Input + Output → Algorithm finds the rules.

Eg: Netflix/ Youtube Recommendations:

Manual labour:-

Input → Human → output

Prog:-

Human

↓

input → Tool → output

ML:-

3.

Human

data

Tool design

Input → Tool → Output

Alexa →

Increase the..

Lots of images →

ML:-

spam

Non-spam

4. $102 + 205 = 307 \to$ labeled.

$120 + 400 = 520$

exam

how addition works.

we need to have a mental model of addition.

learn a function from input (question) to the output (answer)

real numbers

input $\to$ output

5.

→ suppose you want to predict a student's exam marks. What do you need?

: need a data (hours studied, attendance, past marks)

: You'll need a model

↓

mathematical formula that relates

hours studied → marks.

: Need to define a Task → predict marks

What is data? → train ML models.

Model :- Mathematical simplication of reality.

6.

- Predictive Model ✓
  - ↳ Regression model
  - ↳ Classification model.

- Probabilistic Model

Data: — collection of vectors

	feature1	F2	F3	
M1: –	3	9	1.9	5.0
M2: –	2	7	2.1	3.2
M3: –	4	12	2.8	6.6
M4: –	5	16	0.9	9.8

$$x_1 = [3 \quad 2 \quad 4 \quad 5]' = \begin{bmatrix} 3 \\ 2 \\ 4 \\ 5 \end{bmatrix}$$

$$X_2 = [9 \quad 7 \quad 12 \quad 16]'$$

$$X_3 =$$

Regression Model :-

7. ↳ predicting a continuous numerical value.

has to learn a mapping from input to output.

once this mapping has been learnt, the model can be used to predict the unseen inputs.

Model :

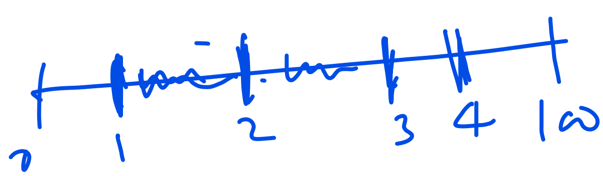
$$f : \mathbb{R}^d \rightarrow \mathbb{R}$$

$$y = f(x)$$

feature vector → model → label

1, 2, 3, 4, 5, ... , ∞

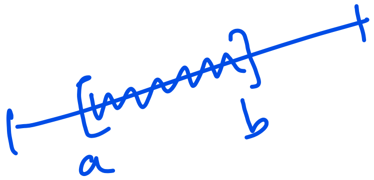
X

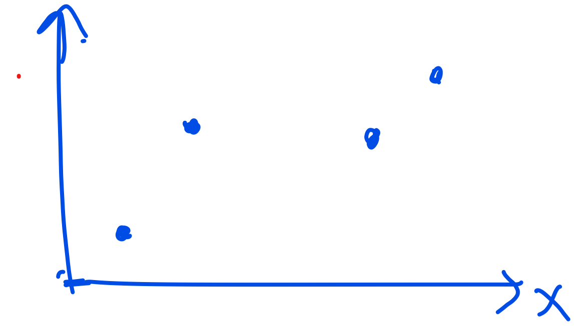


Eg:-

$$\text{marks} = a * \text{hours} + b$$

real number





$x_5 = 6$

weight

height

$$\text{weight} = a * (\text{height})$$

real number.

x_6 = 5 feet 5 inches

Regression model:

$$\boxed{\text{Price}} = 0.5 * \text{Area} - \text{distance}$$

Real numbers

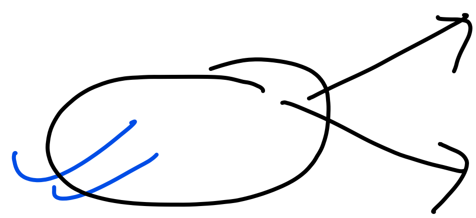
Classification model:

~~9.~~

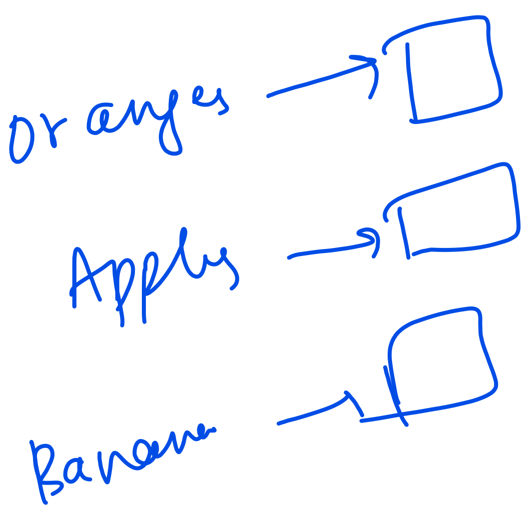
↳ model whether a house is closer than 2kms to a metro ~~based~~ on price & area

$$\text{Answer} = \begin{cases} \text{Close}; & \text{if } \underbrace{2^{a}\ \text{Rooms} - \text{Price}} < 1 \\ \text{Far}; & \text{o.w.} \end{cases}$$

Eg:- ① spam or not spam



② Medical diagnosis: disease = yes/no

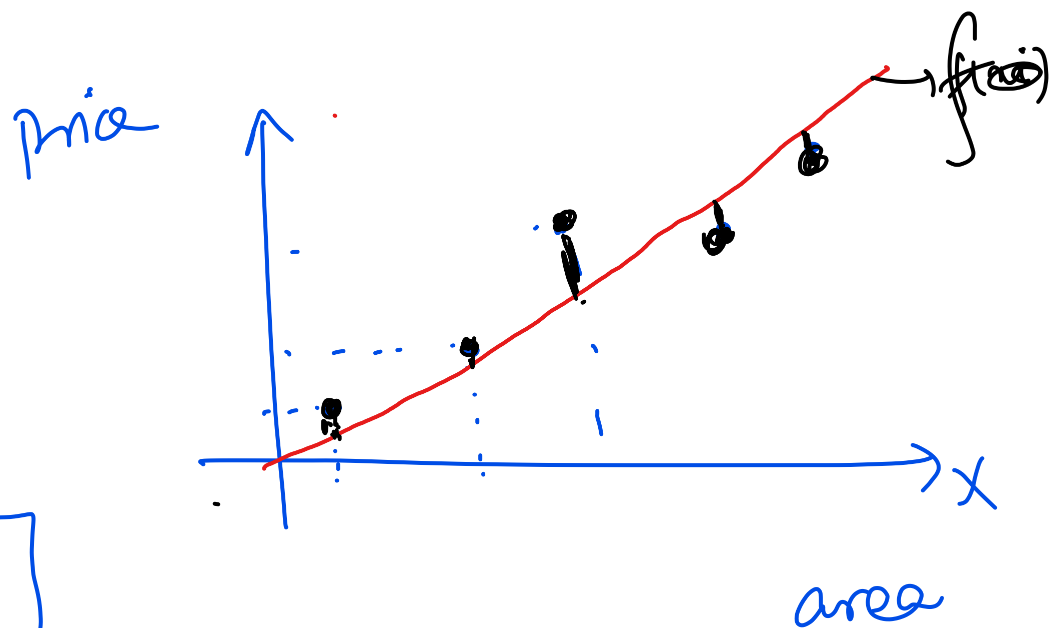


10.

Price = a * (area) + b * (# rooms)

+ c * (distance to metro)

a, b, c → parameters.



Price = a * area

Best → minimum error.

← Student A: 90, 92, 95, 92

Student B: 70, 90, 50, 40

Why?

variability is less

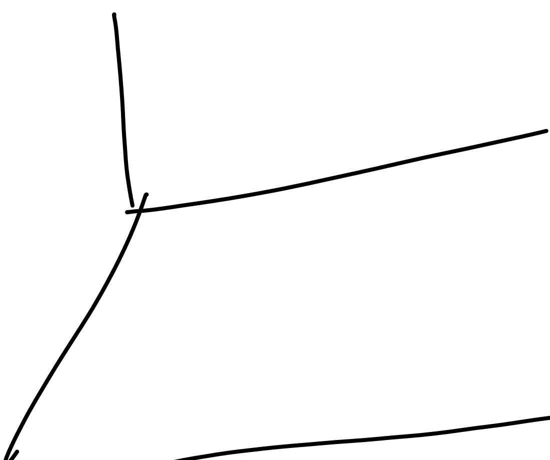
- $\mathbb{R}$: real numbers

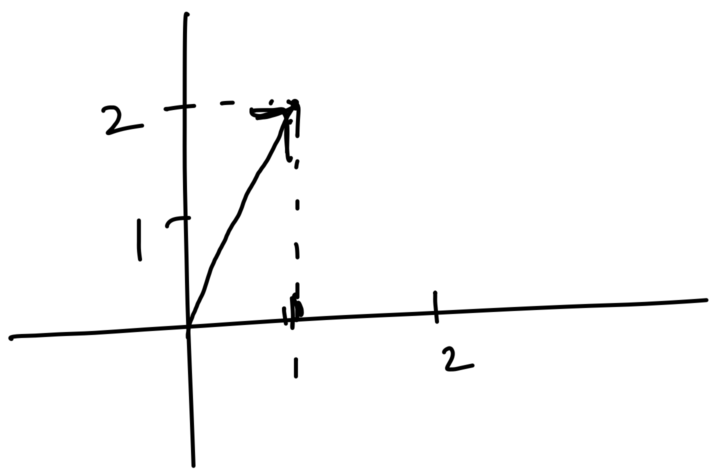
$\begin{bmatrix}1\\2\end{bmatrix} \in \mathbb{R}^2$

- $\mathbb{R}_+$: positive reals
- $\mathbb{R}^d$: d-dimensional vector of reals.
- x: vector.

$\begin{pmatrix}1\\1.2\\-3\end{pmatrix} \in \mathbb{R}^3$

$\begin{bmatrix}1\\2\end{bmatrix}$





$\boxed{x_j : j^{th}\text{-coordinate}:}$

$x = \begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}$

$\|x\|$: length of vector x.

$$\underbrace{\|x\|}^2 = x_1^2 + x_2^2 + x_3^2$$

$$\|x\| = \sqrt{x_1^2 + x_2^2 + x_3^2}$$

$I(2 \text{ is even}) = \underline{1}$, $\quad I(2 \text{ is odd}) = \underline{0}$.

12.

$$\|x\|^2 = x^T x = [x_1 \quad x_2 \quad x_3] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\|x\|^2 = x_1^2 + x_2^2 + x_3^2$$

$$\Rightarrow \quad \|x\| = \sqrt{x_1^2 + x_2^2 + x_3^2}$$

$$x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} : \quad \|x\| = \sqrt{1 + 4 + 9}$$

$$= \sqrt{14}$$

---

x_j^i : jth coordinate of ith vector.

$x^1, x^2, x^3, \ldots, x^n$: collection of n vectors.

x_j^i:

$x_2^1 = 2$

12

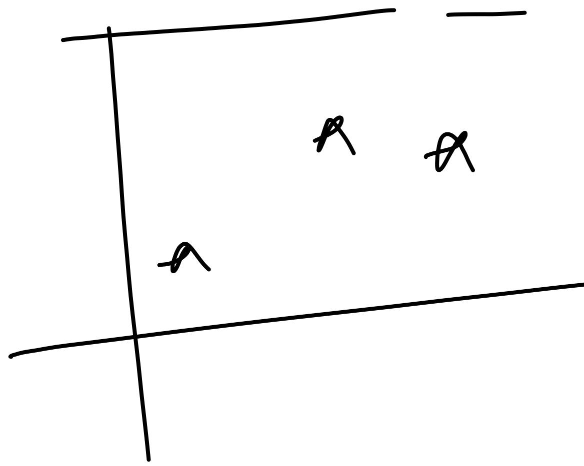
$x^1 = [1, \underline{2}, 3]$

$x^2 = [4, 5, 9]$ ✓

$$x_2^2 = 5$$

Supervised learning

↳ curve-fitting



• Given:

$$\{(x^1, y^1), (x^2, y^2), \dots (x^n, y^n)\} \to \text{Training data.}$$

• Find a model f such that $f(x^i)$ is 'close to' y^i

• Algorithm outputs a model: $f: \mathbb{R}^d \to \mathbb{R}$

$$\boxed{\text{Loss}[f] = \frac{1}{n}\sum_{i=1}^{n} \left(f(x^i) - y^i\right)^2}$$

$$\boxed{f(x) = w^T x + b = \sum_{j=1}^{d} w_j x_j + b}$$

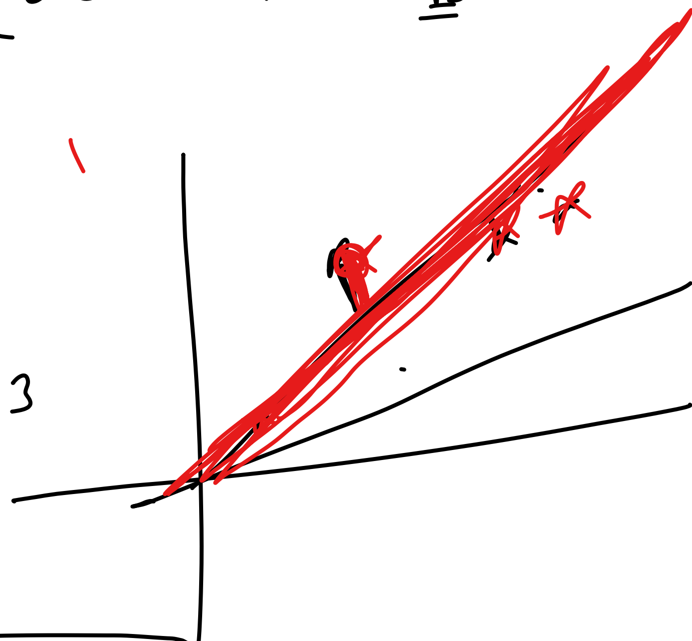
$$= w_1 x_1 + w_2 x_2 + w_3 x_3 + b.$$

$$= w_1 (\# \text{rooms}) + w_2 (\text{area}) + w_3 (\text{distance}) + b$$

$$x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$A_{3\times2}$ $\quad B_{2\times3}$

$$AB = [\quad]_{3\times2} [\quad]_{2\times3}$$



$$\boxed{f(x) = w_1 x_1 + w_2 x_2 + w_3 x_3}$$

$$= [w_1 \quad w_2 \quad w_3]_{1\times3} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{3\times1}$$

$$= w^T x$$

$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

absolute:

15.

1 2 3 4 5

$$\text{Variance} = \frac{1}{n}\sum(x_i-\bar{x})^2$$

$d=1$

x	y	f	g
[1]	2.1	2	5
[2]	3	4	6
[3]	6.5	6	7
[4]	11.5	8	8

$f = 2x$

$g(x) = x+4$

$$\text{Loss}(f) = \frac{1}{n}\sum\left(f(x^i)-y^i\right)^2$$

$$= \frac{1}{4}\left[(2-2.1)^2+(4-3)^2+(6-6.5)^2+(8-11.5)^2\right]$$

$= \_\_\_\_$

$$\text{Loss}[g] = \frac{1}{4}\left[(5-2.1)^2 + (6-3)^2 + (7-6.5)^2 + (8-11.5)^2\right]$$

$= \_\_\_\_$

x_1 Rooms	x_2 Area	x_3 distance	Price (y)	f	g
3	9	1	5.0	5	5
2	7	2.2	3.5	—	—
4	10	3.0	6.5	5	—
5	12	0.8	8.	—	—

$w_1 * \text{Rooms} + w_2 * \text{dist}^n$

$$\boxed{f = 2 * \text{Rooms} - \text{distance}}$$

$\rightarrow \text{loss}[f] < \text{loss}[g]$

$g = \text{Rooms} + 2 * \text{distance}$

$$f = 2x_1 - x_2$$

$$\text{loss}[f] = \frac{1}{n}\sum \underbrace{(f(x^i) - y^i)^2} = \frac{1}{4}\left(\qquad \right)$$

17.

$$\text{Loss}[g] = \frac{1}{n}\sum (g(x^i) - y^i)^2 = \frac{1}{4}\left(\qquad \right) L$$

Classification

Regression
$f: \mathbb{R}^d \to \mathbb{R}$

Training data:-

$$\{(x^1, y^1), (x^2, y^2), \ldots, (x^n, y^n)\}$$

spam not spam

$x^i \in \mathbb{R}^d$, $y^i \in \{+1, -1\}$

- Algorithm outputs a model $f: \mathbb{R}^d \to \{+1, -1\}$

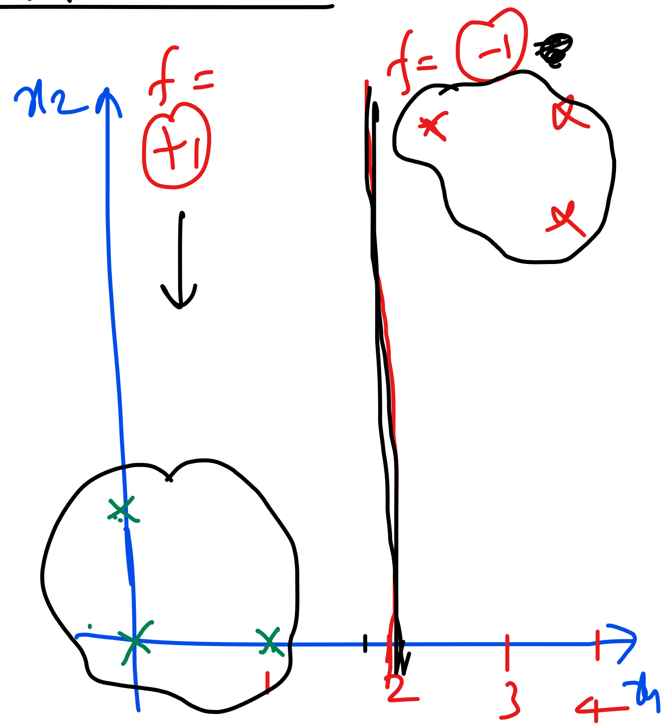
$$\boxed{\text{Loss}[f] = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(f(x^i) \neq y^i)}$$

= fraction of training data classified wrongly by f

- $f(x) = \text{sign}(w^T x + b)$

x	y	f	$g = \text{sign}(x_1 - 2x_2)$
[0,0]	+1	+1	+1
[0,1]	+1	+1	−1
[1,0]	+1	+1	+1
[4,4]	−1	−1	−1
[3,4]	−1	−1	−1
[4,3]	−1	−1	−1

18.



sign(2−0)

$f(x) = \text{sign}(\underbrace{2 - x_1})$ $\quad Loss[f] = \frac{1}{6}(0) = 0$

$g(x) = \text{sign}(x_1 - 2x_2)$ $\quad Loss[g] = \frac{1}{6}(1) = \frac{1}{6}$

100%

80% training data → 20% of train: validation

20%: testing data

Training: practice questions you solve at

19.

Validate: A mock test that helps us check whether my preparation method is working.

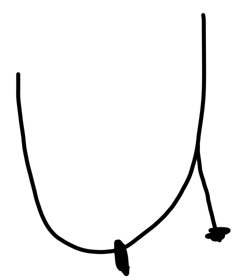
Adjust hyperparameters

Test: the final exam

train → 70%

validate → 15%

test → 15%



$$f(x^{t+1}) = \theta - \eta($$

Unsupervised learning

30 features → 4 features

without losing much information?

tweets about coca cola → many

1 million tweets → manage

is understanding data

Build models that compress, explain & group data

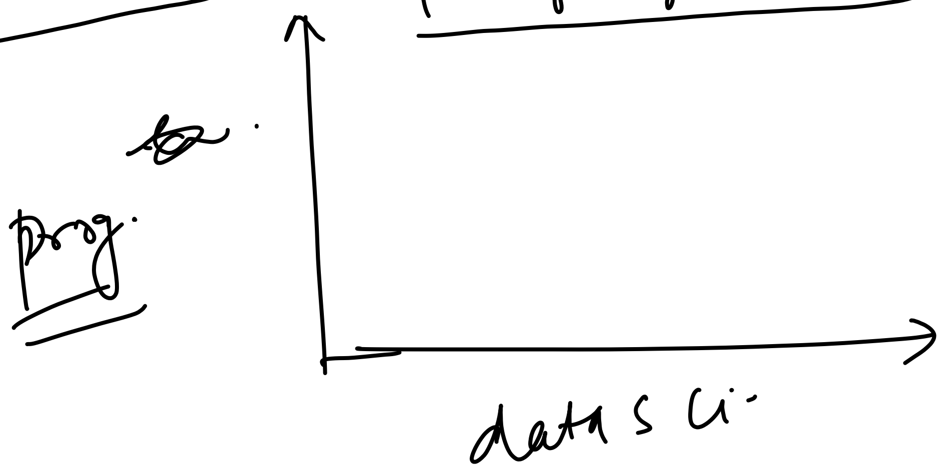
Eg: → I give you exam marks in 10 subjects

21. for each student.

dimensionality reduction:

reduces to 2 components:

math skill vs. language skill.



Diploma level

12 subjects 4000

Too many dimensions.

S1 S2 S3 S4 S5 - - - - - S12

1

22.

2

.
.
.
.

row

image comp = reduce the amount of data needed to present an image while keeping it visually meaning.

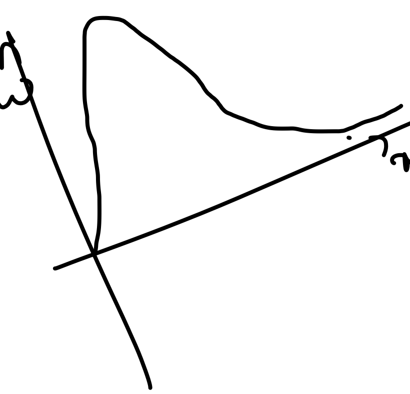
eg: A 1000 X 1000 pixel image = 1,000,000 numbers.

PCA → most imp. features in your image data.

$\boxed{\text{top-}k \text{ principal components}}$

~~$(x^1, y^1), (x^2, y^2)$~~

1000 X 1000 → raw
↓
50 principal components.



mean =
var =

Data: $\{x^1, x^2, \ldots, x^n\}$

23. $x^i \in \mathbb{R}^d$

$d' << d$

- Encoder $f: \mathbb{R}^d \to \mathbb{R}^{d'}$
- decoder $g: \mathbb{R}^{d'} \to \mathbb{R}^d$
- Goal: $\boxed{g(f(x^i)) \approx x^i}$
- loss $= \frac{1}{n} \sum_{i=1}^{n} \| g(f(x^i)) - x^i \|^2$

$d=2$, $d'=1$

	f	g	f	g
[1, 0.8]	0.2	[0.2, 0.2]	0.9	[0.9, 0.9]
[2, 2.2]	−0.2	[−0.2, −0.2]	2.1	[2.1, 2.1]
[3, 3.2]	−0.2	[−0.2, −0.2]	3.1	[3.1, 3.1]
[4, 3.8]	0.2	[0.2, 0.2]	3.9	[3.9, 3.9]

$\left.\begin{array}{l} f(x) = x_1 - x_2 = - \\ g(u) = [u, u] \end{array}\right\}$ models

$\tilde{f}(x) = \frac{x_1 + x_2}{2}$; $g(u) = [u, u]$ } models

24.

$\tilde{g}(u) = [u, u]$ $\quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ $\quad \| g(f(x^{(1)})) - x^{(1)} \|^2$

$x \quad g(f(x))$

$\|x\|^2 = x_1^2 + x_2^2 + x_3^2$

$[0.2, 0.2]$

$[1, 0.8]$

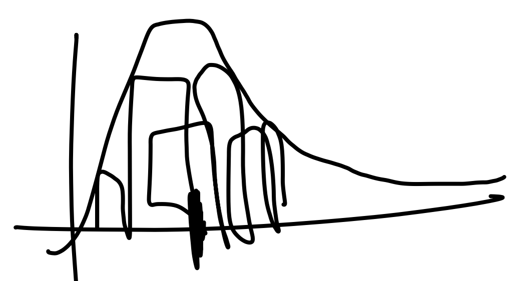
$$\frac{1}{4}\left[(0.2 - 1)^2 + (0.2 - 0.8)^2 + \quad \right]$$

Density Estimation

→ Process of estimating the prob. distn of data.

→ Helps us understand how data is spread.

↳ detects rare/unusual patterns (outliers)



Fraud detection:-

Normal transcations = high-density areas

Fraudulent = low-density area

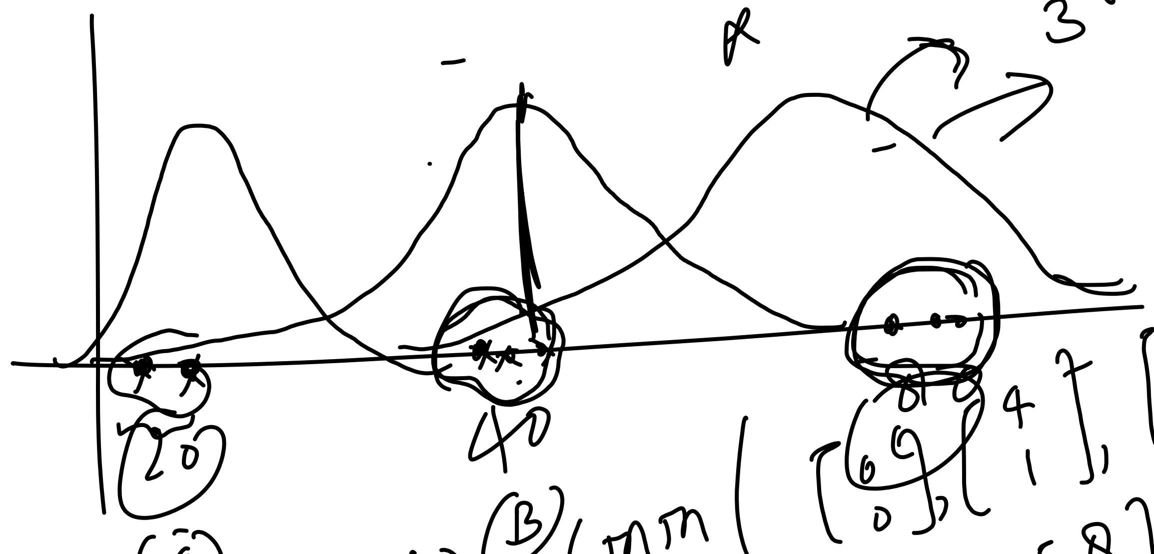
Data: $\{\underbrace{x^1, x^2, \ldots, x^n}\}$ $\qquad \sum p_i = 1$

$x^i \in \mathbb{R}^d$

Prob. mapping $P : \mathbb{R}^d \to \mathbb{R}_+$, that 'sums' to one

Goal: $p(x)$ is large if $x \in$ data & low otherwise

loss:



0]

15]

A $\qquad$ B

$p(1) = GMM\left(\begin{bmatrix}6\\0\end{bmatrix}, \begin{bmatrix}4\\1\end{bmatrix}, \begin{bmatrix}7\\3\end{bmatrix}\right)$

$p(2) = GMM\left(\begin{bmatrix}5\\5\end{bmatrix}, \begin{bmatrix}8\\-9\end{bmatrix}, \begin{bmatrix}-1\\-2\end{bmatrix}\right)$ $\quad$ 3,8]

1.9		~1/5	0
4.3	1/10	~1/5	1/5
4.8	1/10	-1/5	1/5

$loss[p(1)] = \left(\frac{1}{10}\right)\log\left(\frac{1}{10}\right)$

$loss[p(2)] =$