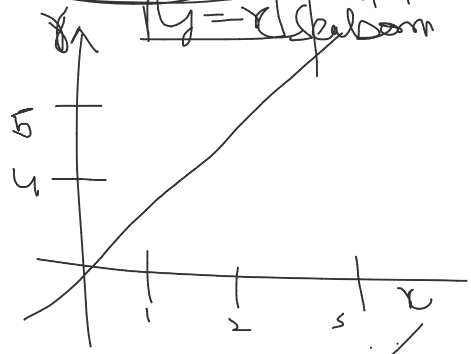
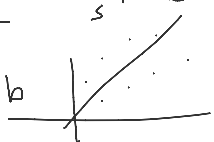
Matplotlib

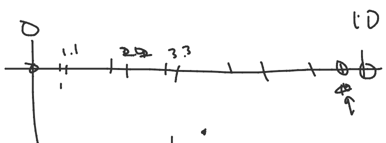
| x | y |
|---|---|
| 1 | 1 |
| 2 | 2 |
| -1 | -1 |

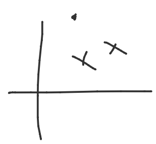
$\boxed{y = x}$ Seaborn




matplotlib 

| Hours | price |
|---|---|
| 1 | 4L |
| 2 | 10L |



ML

Machine learn

Computer program is said to
learn from experience E
with respect to some class
of task T and performance
measure P, if its
performance at task t ,
As measured P , improves
with experience E

E – experience
T – task
P – prob

AI

ML

Ludo
E = 0

Ludo

~~ML~~

E = O

T = Win / Playing

P = Prob. of winning

ML

E T P

LUDO

program Dice

~~loop~~ = ~~tw~~

while(true):

ML

0.1

0.4

how to pick

→ loop (Experience)

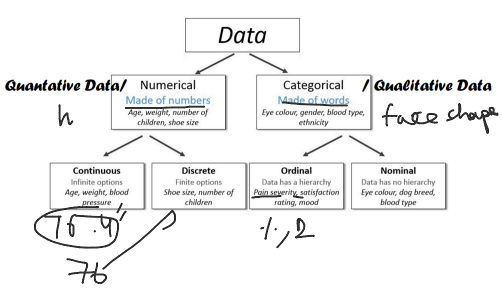
T → Win

Supervised

Unsupervised

Data Type

Numerical categorical




h

face shape

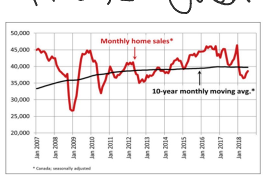
76.4

76

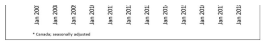
1, 2

time, seqhen

Σ seri




trait Sin



text — Siri, Alexa, Cortana

Supervise | data

| x | 1 | 2 | 3 | 4 | 7 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| y | 2 | 4 | 6 | 3 | 2 | 6 | 4 |

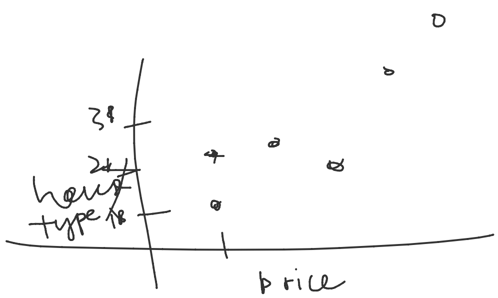
$f(x) = y$

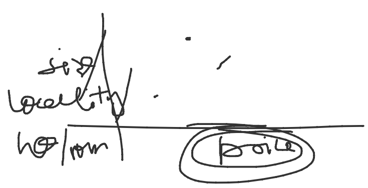
Unsupervised

$f(\bar{x}) = \hat{y}$

50

| | House | price | size | living | rooms |
|---|---|---|---|---|---|
| | 1 | x=101 | | | |
| | 2 | z=60 | | | |
| 3 | 3 | K=20 | | | |




we are given a dataset and already know what our correct output should look like, having the idea that there is a relationship between the input-output

$f(x) \to y$ we know

$x = 10, \ldots, 1000$ certain values

$y$

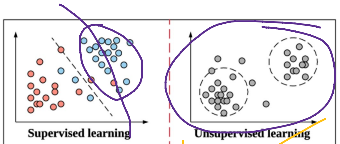
f(IOU)

Hours

Sup → Price → Size
room

Regression    Classification

Unsup



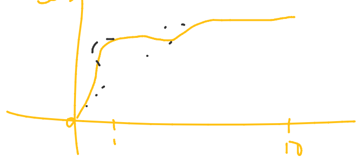

Curves

y = mx + c
$x_1, x^2$
discrete
(not a function)
polygon

Sup

$y = m_1x_1 + m_2x_2 + \dots c$

$x_1 = 1, \dots, c$



LR →

| input | output |
|---|---|
| features | target |
| explanatory variable | response |
| independent | ~~independent~~ dependent var |

$(f(x) = y)$ unique set $\forall x \rightarrow y$

$y = mx + c$

$y = w_0 + w_1x_1 + w_2x_2 \dots w_mx_m$

$y = w_0 + w_1x$

$x = k$   I min [illegible]   how x ___

$y = w_0 + w_1 x$

m = no. of training examp.

$(x_i, y_i)$

↓

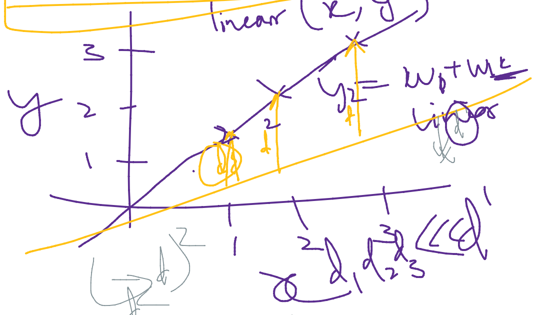
Learning Algo

↓

hypothesis (function)

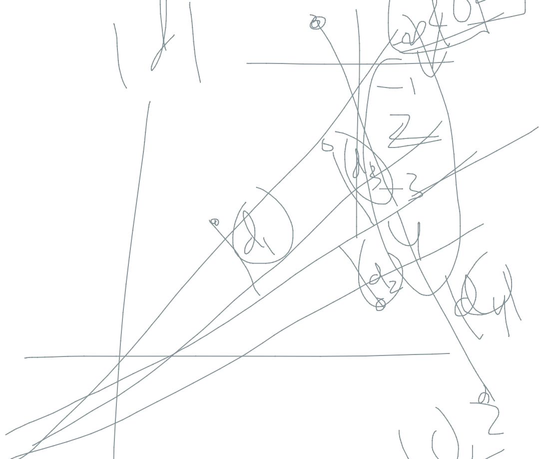
x ← h → y

linear (x, y)



min (d) — ve

$\lvert d \rvert$



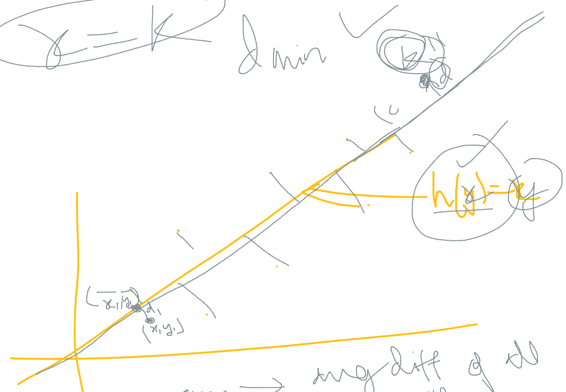
$\min \sum d$

$\lvert d^2 \rvert \geq 0$

$x = k$

d min

$h(x) = y$



aevr → avg diff of all

J

new x

$h(x) = y$

$x_1$ ....... $x_{10}$

$h(x_{(i)}) = y_{(i)}$

x = 10

y = 10

m = 10

h accurat

$h(x^{(i)}) = y^{(i)}$

$J(\theta_1, \theta_2) = \frac{1}{2m} \sum \lvert \bar{y}_i - y_i \rvert$

distance

$y_i = h(x_i)$

no of trainig

$x_1 = 1, y = 2$

$x_2 = 2, y = 4$

$h(x) = y$

$\bar{y}_1 = 5$

$\bar{y}_1 = 2$

$\bar{y}_2 = 4$

$\bar{y}_2 = 1$

mse

mean square error

$J(\theta_1, \theta_2) = \frac{1}{2m} \sum \lvert \bar{y}_i - y_i \rvert$

$J(\theta_1, \theta_2) = \frac{1}{2 \times 2} (\lvert 2-4 \rvert + \lvert 4-2 \rvert)$

$= \frac{1}{4} \times (2 + 2)$

$= 1 \to 0$

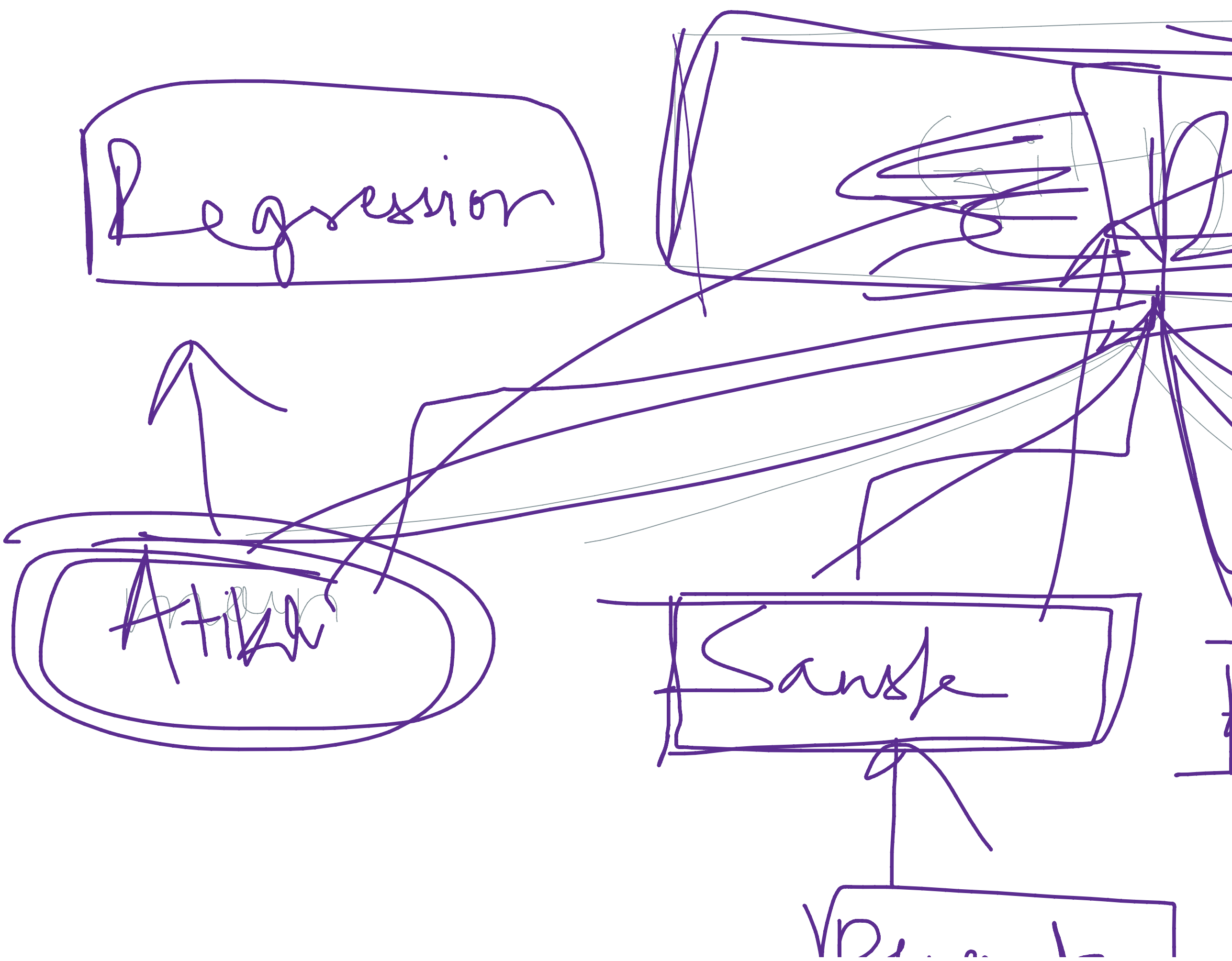
$J(\theta_1, \theta_2) = \frac{1}{2 \times 2} ((2-4)^2 + (4-2)^2)$

$= \frac{1}{4} \times (4 + 4)$

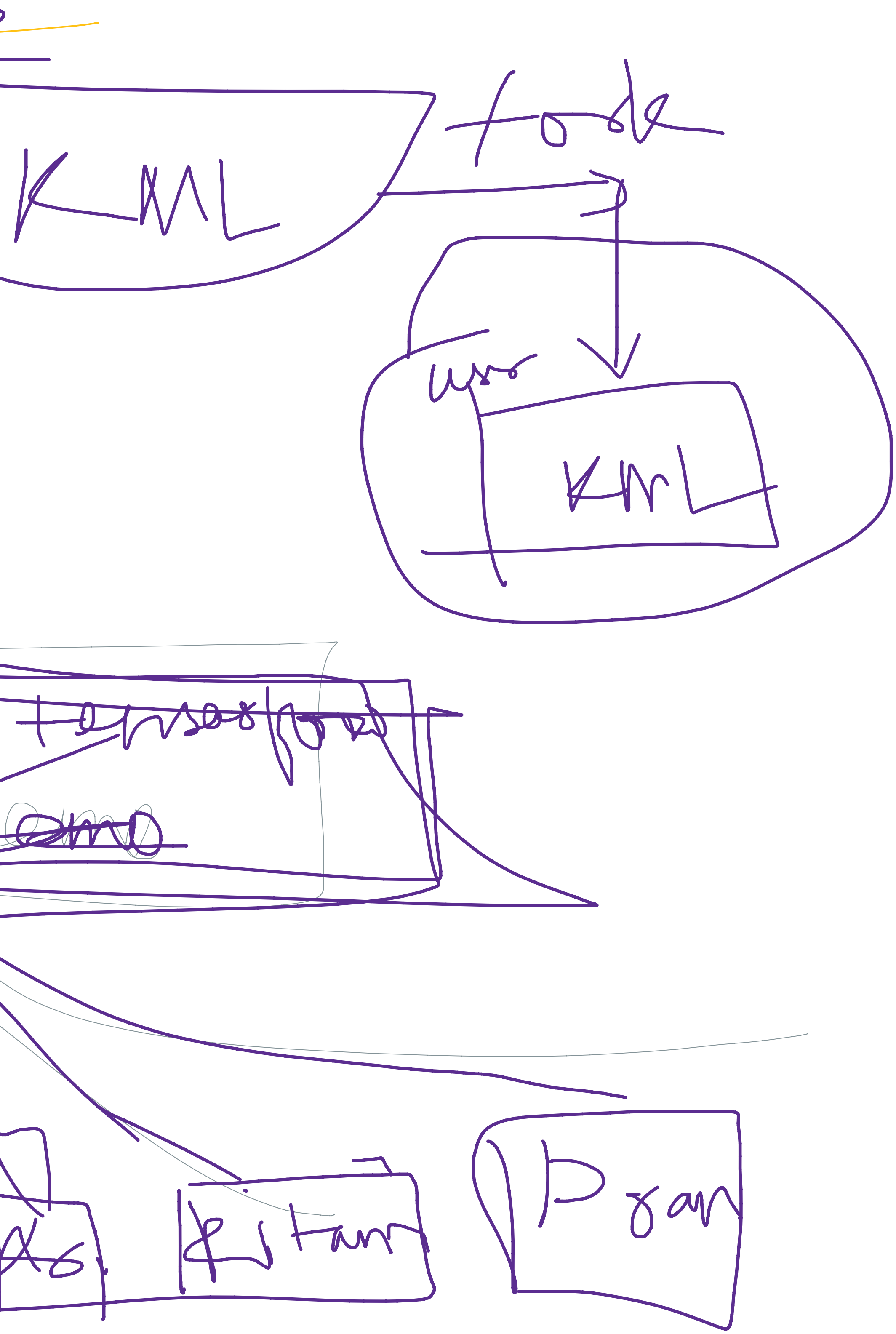
$= 2$

contour plot

$+|d_1|+|d_2|+|d_3|+|d_4|$
local minima

Regression


KML
todo
KML

Pleanh

LR

input, output

prediction

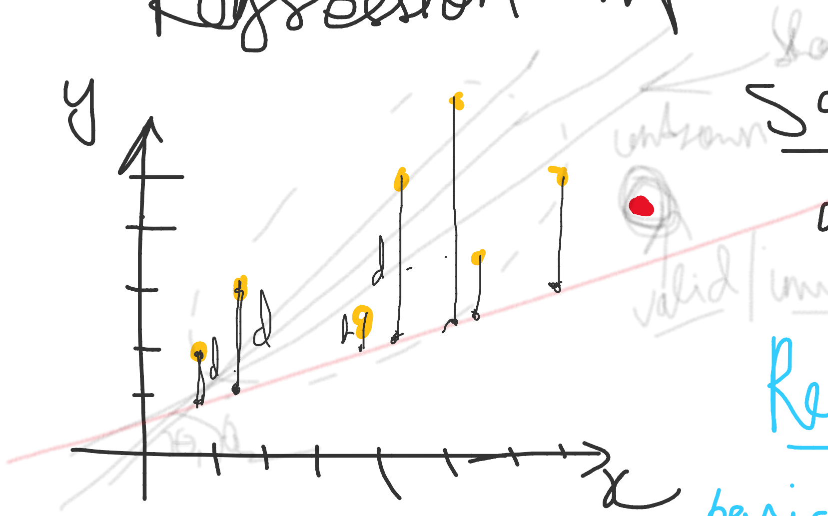
Supervised

quality
- → metrics
- → performance

data collection
( feature / input
output )

numeric data only

Regression improve → E, T, P




Sample set

data sample
(x, y)

Regression

basically helps predict the general curve of sample data

linear reg

predict —— curve

Lad coorinates

1

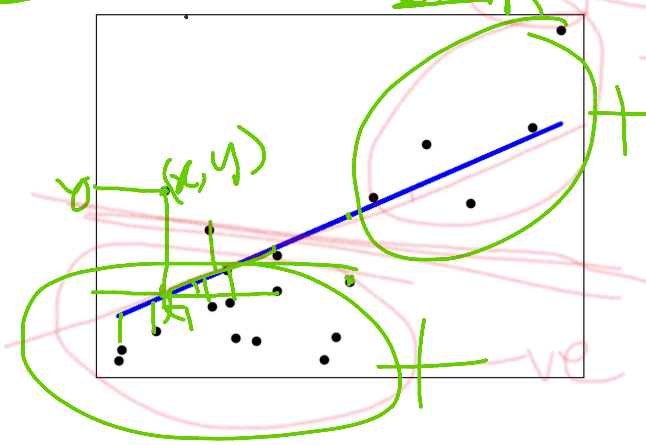
based accurate

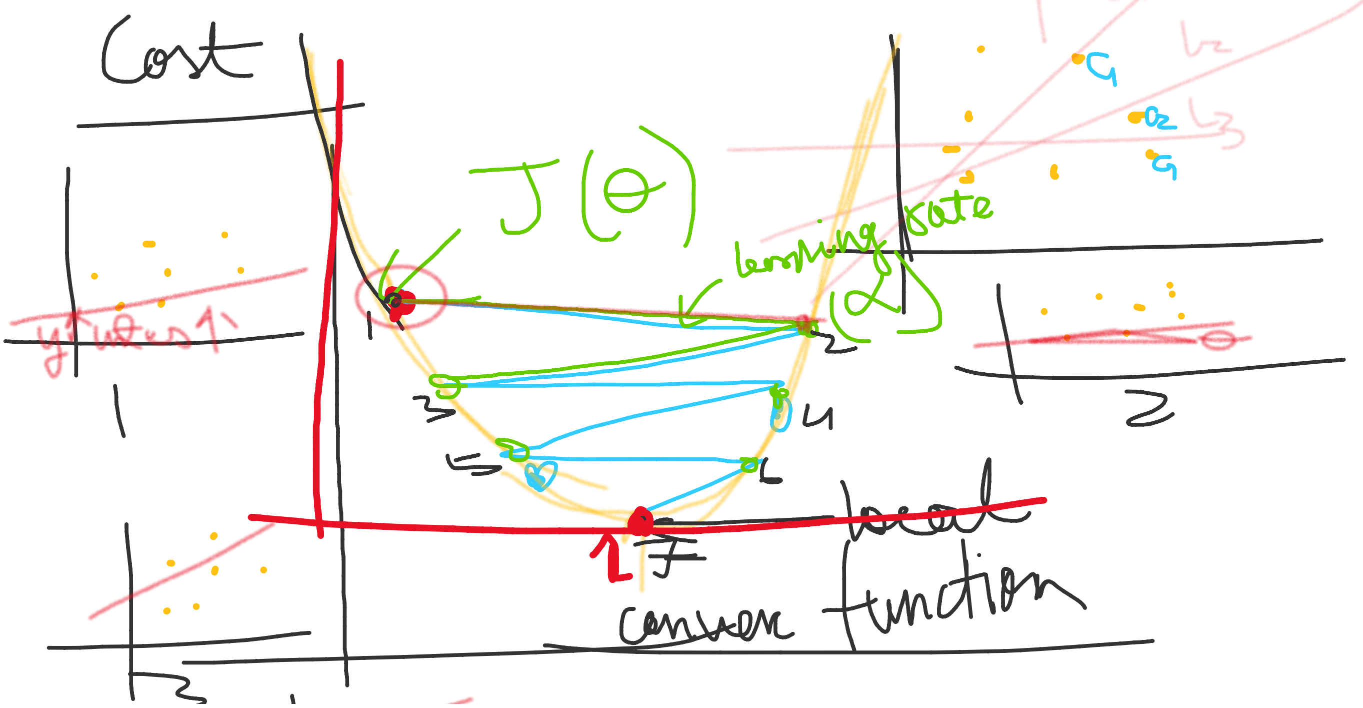
cost function define

$$\frac{\sum (d_i - y_i)^2}{2n}$$ MSE

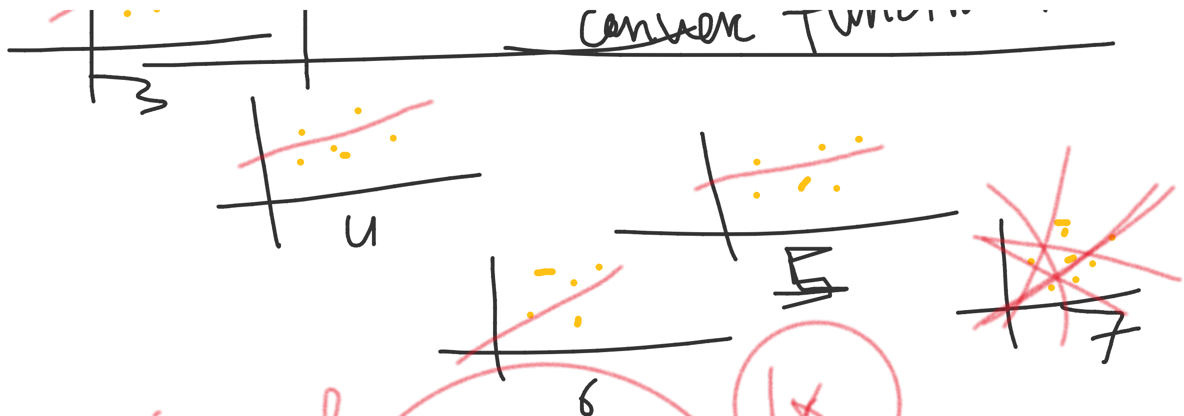
mean = $$\frac{\sum (t_i - y_i)^2}{2n}$$

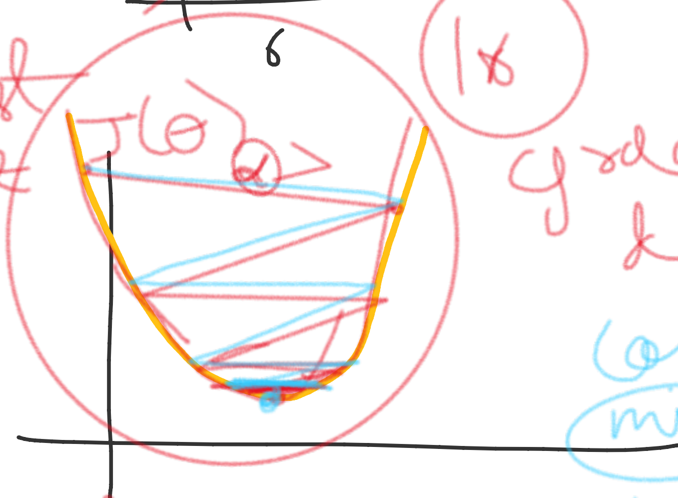
cost sensitive opt.



convex function





cost funct

$J(\theta)$ $\alpha$

gradient descent

cost min

contour plot

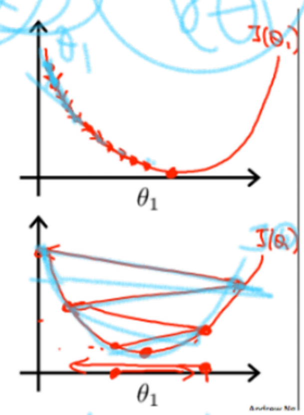
diff deriv

$J(\theta_1)$

$$\boxed{\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)}$$

If α is too small, gradient descent can be slow.

If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.




0.001

$\alpha$ — lr

$\alpha \uparrow \rightarrow 0.01$


function (GD)

$J(\theta)$

$\theta_1 = \theta_1 - \alpha \left( \frac{\partial J(\theta_1)}{\partial \theta_1} \right)$

LR

the curve

LR

~~the~~ lower

$J(\theta_1)$

first

lr

$\theta_1$

lr

ni

$\theta_1$ — . . .

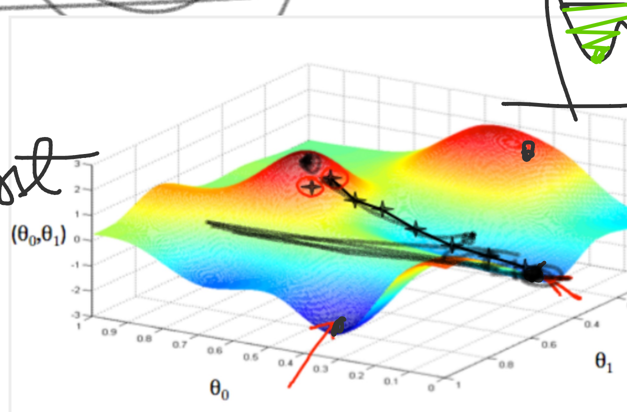
0 — 2

1

0.01

local minima not achieved

GD



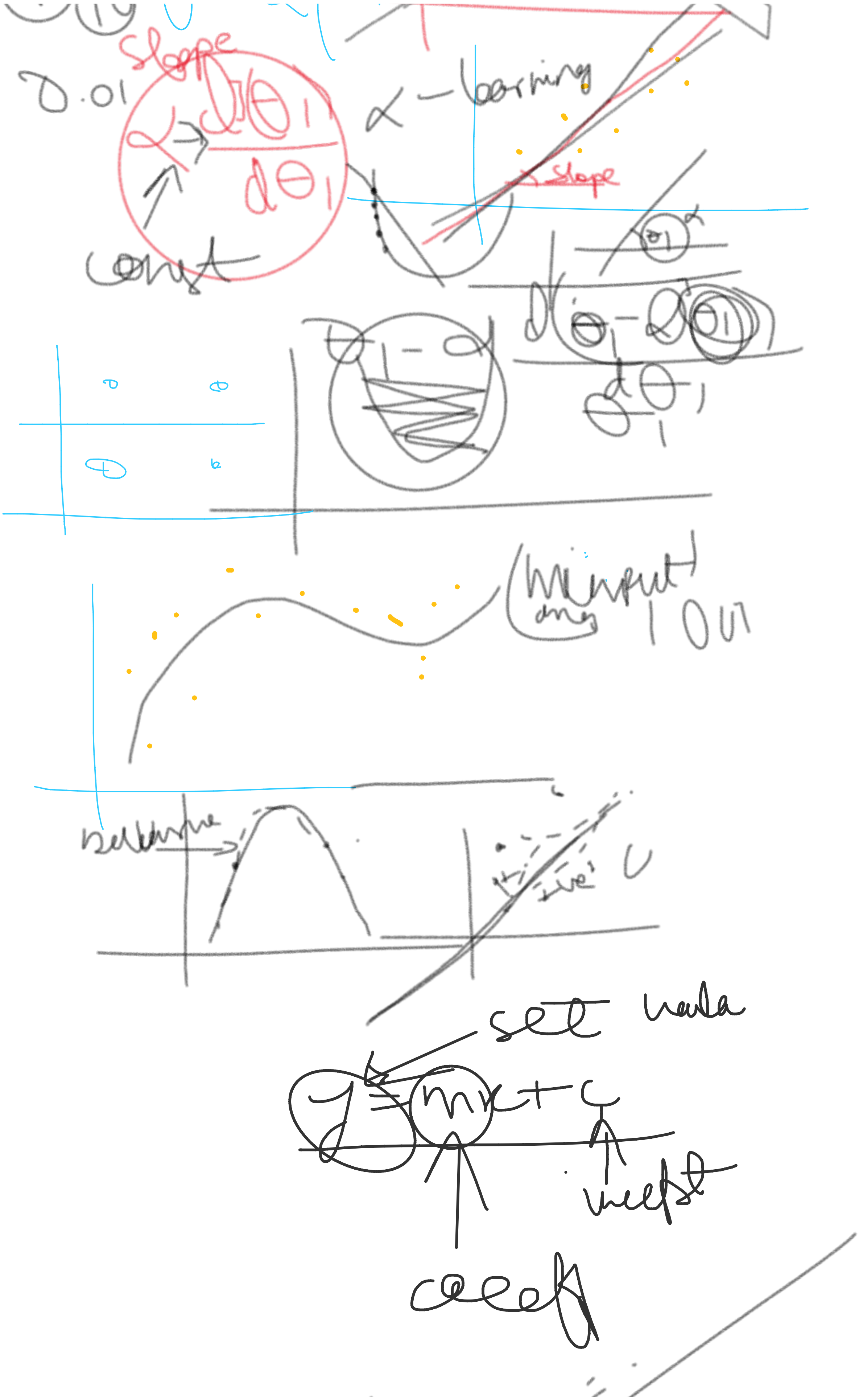

random first line

lr α

$\theta - \alpha \left( d \right)$

10

slope



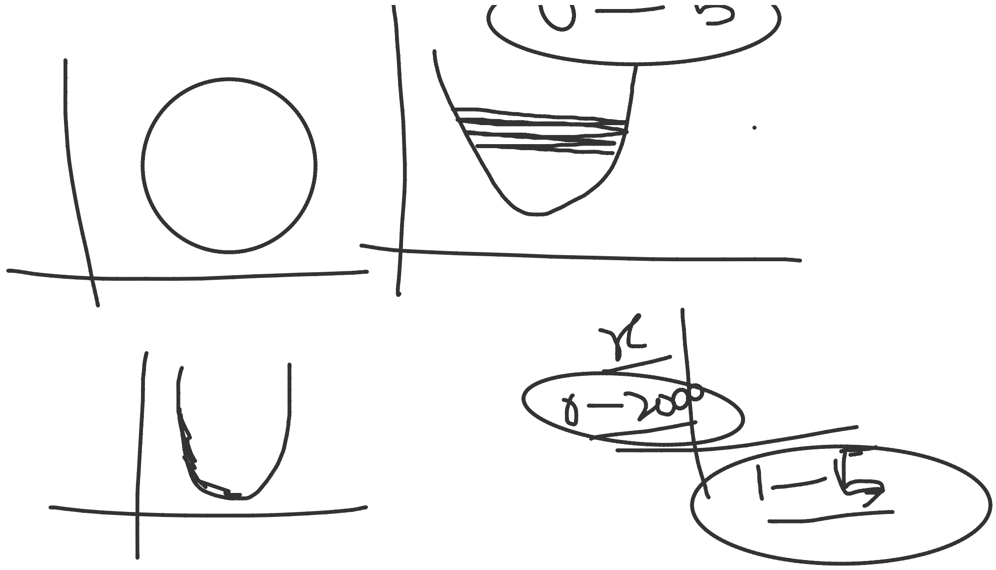
0.01
slope
$\alpha \to \frac{dJ(\theta_1)}{d\theta_1}$
const
$\alpha$ — learning
slope
$\theta_1 - \alpha \frac{dJ(\theta_1)}{d\theta_1}$
input
OUT
+ve
set
$y = mx + c$
coeff

$$\theta_1 - \alpha \left( \frac{dJ(\theta_1)}{d\theta_1} \right) x_i$$

~~Feature~~

FS (x,y)

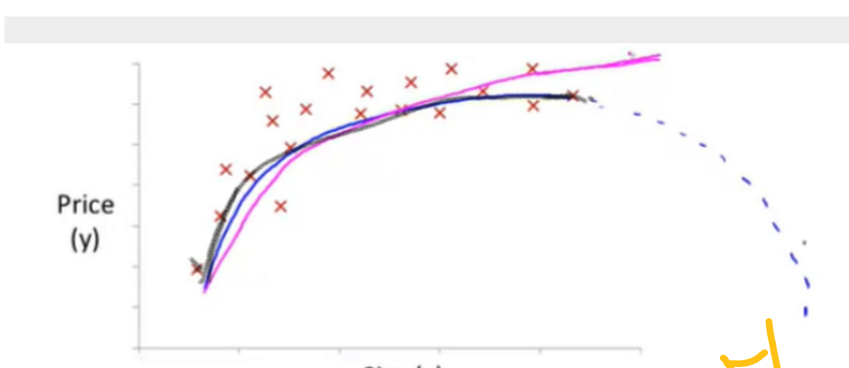
0 — 5




## Poly Reg

$y = mx + c$

$y = m_1x_1 + m_2x_2 + c \ldots + c$


$$J(\theta) = (X^T X)^{-1} X^T y$$

deter

inverse

inverse of mat

Examples: $m = 4.$

| $x_0$ | Size (feet²) $x_1$ | Number of bedrooms $x_2$ | Number of floors $x_3$ | Age of home (years) $x_4$ | Price ($1000) $y$ |
|---|---|---|---|---|---|
| 1 | 2104 | 5 | 1 | 45 | 460 |
| 1 | 1416 | 3 | 2 | 40 | 232 |
| 1 | 1534 | 3 | 2 | 30 | 315 |
| 1 | 852 | 2 | 1 | 36 | 178 |

$$X = \underbrace{\begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}}_{m \times (n+1)} \qquad y = \underbrace{\begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}}_{m\text{-dimensional vector}}$$

$$\boxed{\theta = (X^T X)^{-1} X^T y}$$

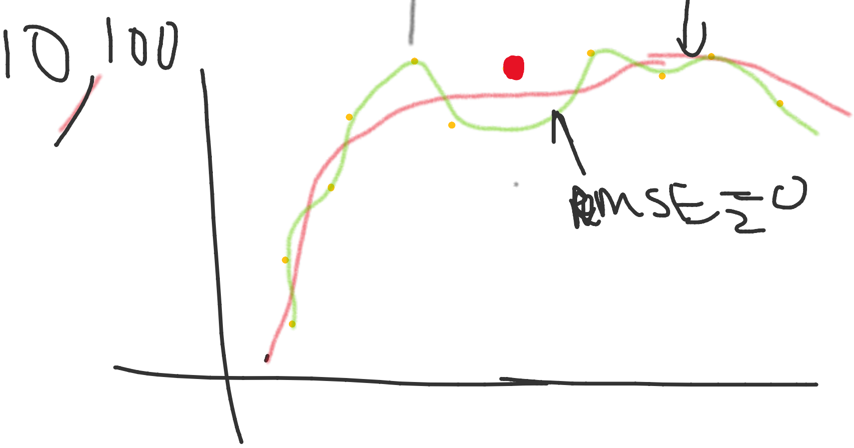
$[(X)\ y\ (1 \times 1)]$

$[X\ y]$

Poly Regress

$x,\ y\ \ \Xi$

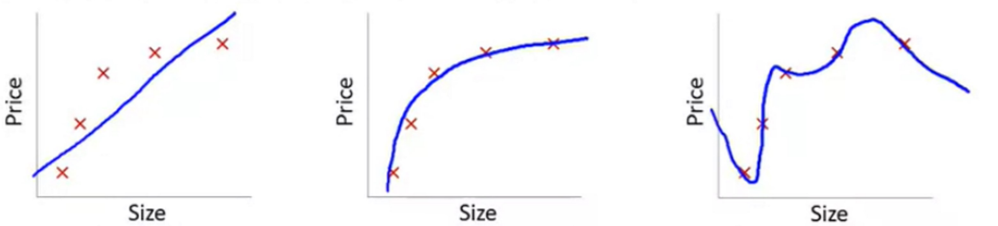
feature output




Overfitting

xample: Linear regression (housing prices)

xample: Linear regression (housing prices)




$\rightarrow \theta_0 + \theta_1 x$ $\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2$ $\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta$

"Underfit" "High bias" "Just right" "Overfit" "High varian

f

$x_1 - - . x_{12}^{10}$

10

100