Machine learning intro

UI

Backend

Database

Big data.
- Hadoop
- Spark
- Hive
- Pig
- Cassandra
- Sqoop
- Hbase
- MongoDB

Data analysis.
- PowerBI
- tableau
- SQL
- Python data analysis (statical)

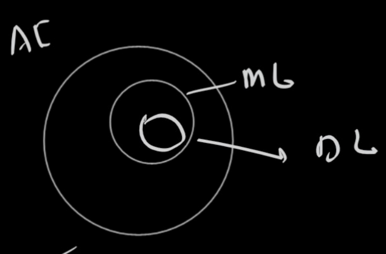
Data science
- ML
- DL
  - CV
  - NLP
- RL

Data
↓
analysis
↓
Predictive model.

AI → Python
Java

- Problem solving
- reasoning
- math.
- ML
- NLP
- Robotics
- Expert system

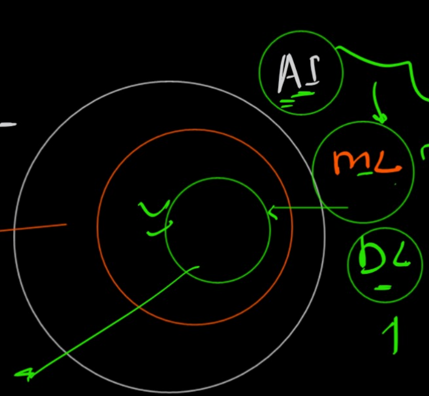
- Embedded System
- Vision based System



Superset ←

Subset ←

Subset of ml.



↑ Datascience

Datascience

Science
- Bio
- $\boxed{\text{math}}$
- Phy.
- Che

Machine learning → (Data → Math)

(Predictive model)

?

analysis

(ML) Structure

(DL) Unstructure

- image
- video
- voice
- text

→ computer vision

→ NLP

ML → (Data)

- Numerical
  - Continuous
  - discrete
- categorical
  - Nominal
  - Ordinal

→ Structure

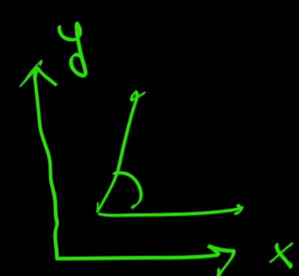
ML → [Data → (math)]

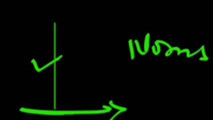
python

Mathematics for machine.

- Linear Algebra
- Probability
- Calculus → (DL)
- Geometry
- Statics

ML





ML → Data → math.

Corr
Cov.

height	weight
165	60kg
170	70kg
180	80kg
185	90kg
190	100kg

Acutal

{62, 71, 80, 89, 102}

= (5)

linear regression

② $(\hat{y} = mx + c)$

① $m = 5$

① $c = 10$

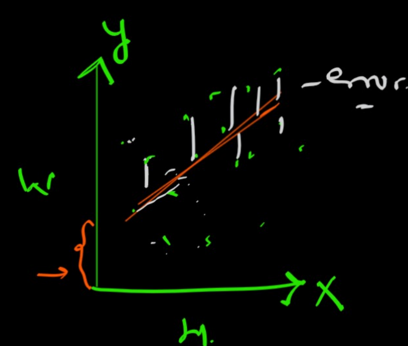
② $= 5 \times 165 + 10$

$= 62$ kg

$= 5 \times 170 + 10$

$= 71$

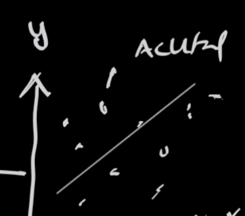
linear



$y = mx + c$

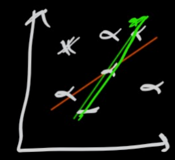
① (Calculation.)

② (error - ($y - \hat{y}$)) → (S) ← (m, c)



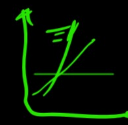
3 Optimization

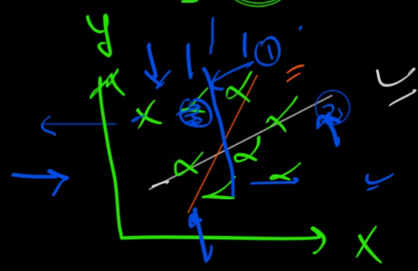
4 7 4
5 10



4 ← 4 8

3 3 7





$y = mx + c$

3 → Tensors

Machine learning

- Supervised
- Unsupervised
- Semisupervised

Independent var

dependent var (target)

Height	Weight	Obese / Not obese
170	65	N
180	70	Y
190	80	N
195	90	Y
200	120	Y

classification

2 → binary

extreme = then

target

- Regression (Numerical) → house price
- Classification (categorical)
  - ↳ Binary classification
  - ↳ multiclass classification

x_1 x_2 x_3 x_4 y house price } Regression

x_1 x_2 x_3 { y. Positive, Neg, Neutral }

more than 3, 4, 5, 6, 7, 8 .. N

Supervised

Unsupervised

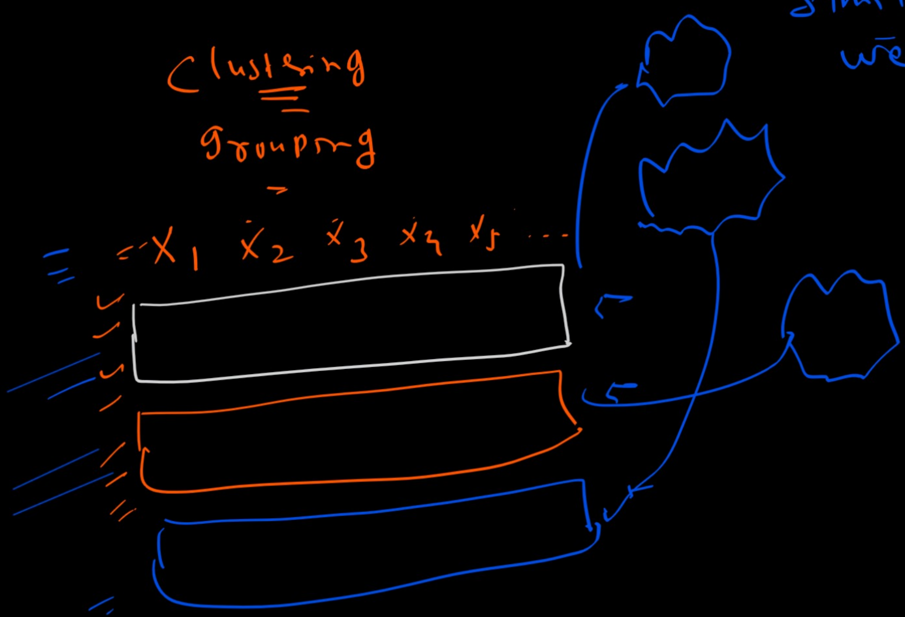
$x_1 \quad x_2 \quad x_3 \quad x_4 \quad \boxed{y}$

Independent
H, W

Dependent
0/NO

Clustering
grouping

Similar a data we are going to group



Semi supervised machine

$\left\{\begin{array}{l}\text{Supervised} \\ + \\ \text{unsupervised}\end{array}\right\}$ → $\boxed{\text{IN. F.}}$ D.F. ← y

→ F. → grouping

Data

Problem

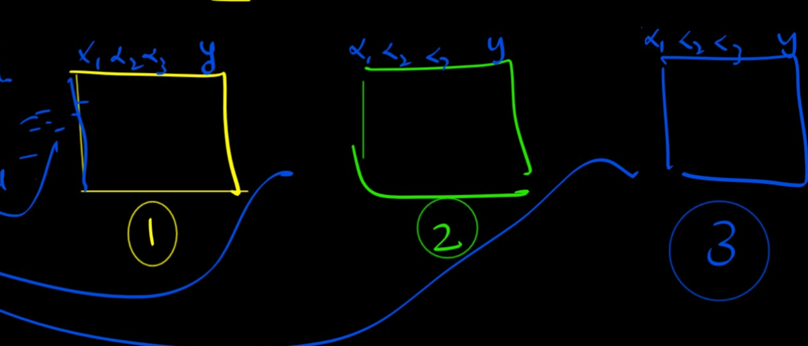
Semi supervised ml
↓
Super
unsuper

	x_1	x_2	x_3	x_4	x_5	y
1						
2						
3						
4						
5						
6						
⋮						
1000						

Grouping & Similar

Unsupervised

Supervised



Supervised :- Linear regression
Logistic regression
SVM.
Decision tree.
Random forest.
XGBoost.
Gradient boost.
Ada boost.
KNN
Naive bayes.

10:00

Unsupervised :- K-mean
- DBSCAN
- hierarchical

Software industry (infrastructure)

Data INDUSTRY B.D. D.A. D.S

AI
- ML
- DL — NLP, VISION
- RL

ML → Data - math. (Pattern)

learning :- optimization of params.

Calculation — Super vised, Unsuper (Clustering)

Super vised → Regression, Classification

error — regression: mean square, mean Absolute, R mean sq. err.

Classification: Cross entropy

optimization → Gradient decent approach — BGD, SGD

Regression

Classification

Problem → Regularization

(Optimizer)

Gradient decent. optimize the Parameters

ANN

m, c $y = mx + c$

Actual - Predict.

$(y - \hat{y})$

$m_{new} = m_{old} - \eta \frac{\partial L}{\partial m}$

ML
1. Calculation
2. Loss
3. optimization

$y = mx + c$ $(y - \hat{y})$

$(\hat{y} =)$

$$y = (y - \hat{y}) + |m| \qquad (|m|^2)$$

$(y - \hat{y} + |m|^2)$ $\quad y = mx + c \quad mx + c$

→ $y - \hat{y}$

5	6
10	6
12	7
13	11
14	18

$5 - 6 = (-1)^2$

4

5

$(-3)^2$

$(-4)^2$

error $= y - \hat{y}$

$mx + c - \hat{y}$

optimization: $m_{new} = m_{old} - \eta \frac{\partial L}{\partial m}$

$y - (mx + c)$

$= m_{old} - \eta \frac{\partial (y - \hat{y})}{\partial m}$

MAE

MSE

m_{new}

Variance

MAD

m_l

Work $\quad$ Suppose $\quad$ you

Interest

Practical

Assignment

if Control flow

Condition.

add.

```
a = 10
b = 20
c = a + b
print(c)
```

$10 > 1$

keyword $\quad$ O/P = 30

$20 < 50$

```
if (a>1 and b<50):
```

code:-

```
if <cond 1> :
        T or F
if <cond 2> :

if <cond 3> :
```

(F) $\quad$ terminate

```
if <cond 1>

elif <cond 2>
      T
elif <cond 3>

else
```

Control

if (=)

Program.

else

Cond 1
Cond 2
:

Loop

Syntax.

key. while (condition):

code or logic

Natural :- (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) . . N

while.

10

10

i = 0

while i < 10 :

Print (i)

i = i+1

0 < 10

0

i = i + 0

i = 1

ML → Data → MATH

Structure

unstructure

Numeric

Categorical

Cont.

discrete

nominal

ordinal

1

Supervised

x_1 x_2 x_3 - x_n y

1. LR
2. LogReg
3. DT
4. SVM
5. KNN
6. RF

2

Unsupervised

x_1 x_2 x_3 - - x_n

k-means

DBScan

hirarchical

Semi Supervised.

(1 + 2)

1. grouping (cluster)
2. Supervised

7) XB
8) 4B
9) AB

1) regression.
2) classification

$x_1\ x_2\ x_3\ \dots\ x_n$ → y — Numerical → regression; Categorical → classification

classification → Binary (2) / multiclass (more than 2)

Regression: Linear regression → $y = mx + c$. best fit line.
- Ridge regression
- Lasso regression
- elastic search. (R+L)

Stats (analysis)

Math.
1) Linear Algebra.
2) Calculus.
3) Probability
4) Statics
5) Geometry

- Central tendency, dispersion } descriptive
- Sample and Population, distribution } inferential.

Probability
Probability dist
- Conditional, Bayes theorem } → Naive bayes

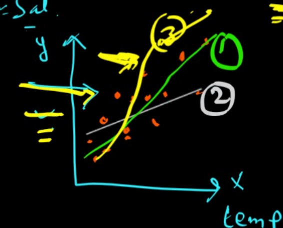
Linear Algebra :-
- linear eq., System of lin. eq., distance and length } Linear reg.
- orthogonality, Angle, eigen value and eign vector → PCA.
- vector → multi, add, sub, dot prod., cross.
- matrix :-

Calculus :- diff., Partial diff, maxima of minima, Integration } → gradient decent, ANN

Linear regression :-

I.V. Temprature	O.V. Sales of ice creme
35°C	5K
40°C	10K
45°C	25K
20°C	2K
15°C	1K
10°C	500
5°C	200

Numerical regression



minimize error → OLS → minimize.

→ Calculation Base

L.R → $y = mx + c$ → Line } end goal

trg Line, best fit line

y actual y predict

1 Calculation → result (Predict) − Actual

$= y = mx + c$ Line.

RMSE, MSE, MAE

2 error → minimize error → best fit line

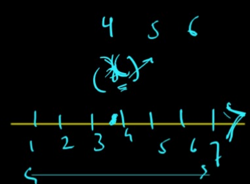
3 optimize — Gradient → Calculus → differentiation, maxima and minima } → ANN

Batch, Sochtic, Line.

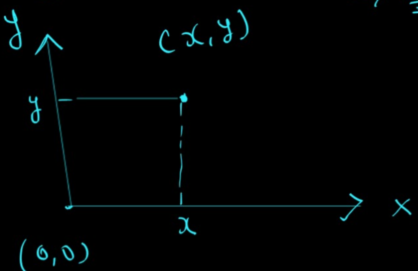
Anatomy → $y = mx + c$

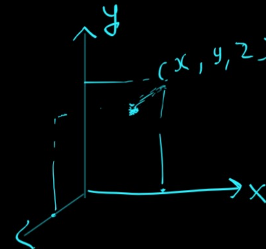
(Point slop)

1D.



→ 2D.



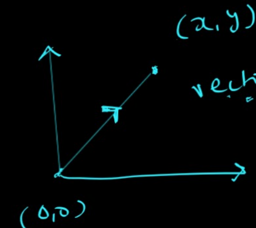


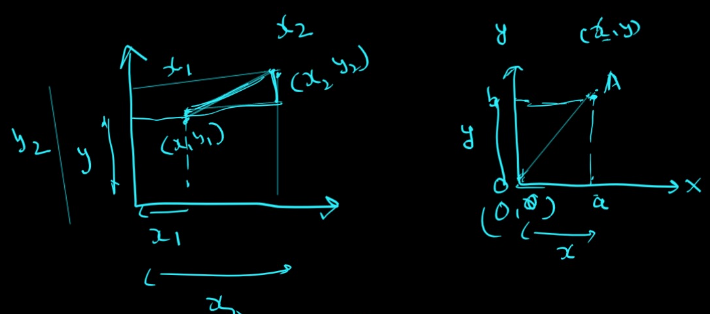
(x, y) vector $= x\hat{i} + y\hat{j}$

$\rightarrow [x, y]$

$[x_1 x_2 x_3 \ldots x_n]_{1 \times n}$ row.

$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}_{n \times 1}$





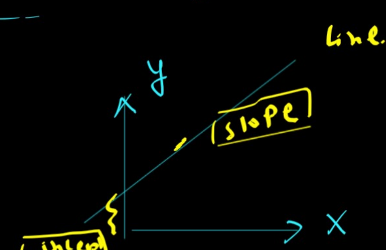
$H^2 = B^2 + L^2$

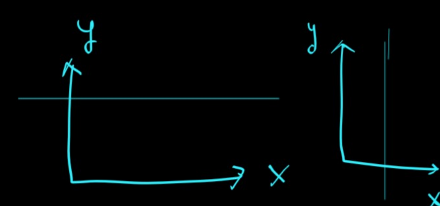
$OA^2 = Oa^2 + Ob^2$

$OA = \sqrt{Oa^2 + Ob^2}$

$\boxed{OA = \sqrt{x^2 + y^2}}$ eculidian

kmean → $x_1 x_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$



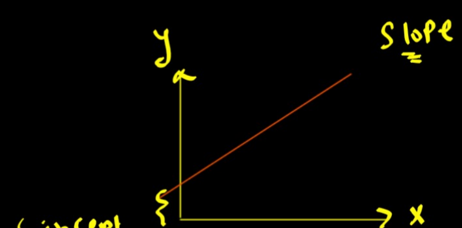


→ general :- $ax + by + c = 0$

Point Solpe :- $y = mx + c$

horizontal $y = x$

vertical : $x = y$



son = x

fath = $2x$

mse ase

father = 2 × son

$m = \dfrac{(y_2 - y_1)}{(x_2 - x_1)}$ = ratio $\dfrac{2}{1} = y$

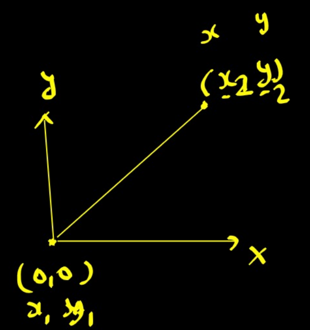
slope.

diff.

$m = \dfrac{y}{x} =$

$y = mx + c$

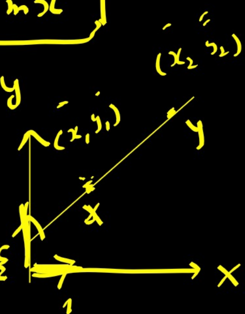
(2) (1)



$m = \frac{(y_2 - y_1)}{(x_2 - x_1)}$

$m = \frac{y - 0}{x - 0}$

$m = \frac{y}{x} \Rightarrow \boxed{y = mx}$



$m = \frac{(y_2 - y_1)}{(x_2 - x_1)}$

$m(x_2 - x_1) = y_2 - y_1$

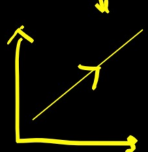
$mx_2 - mx_1 = y_2 - y_1$

$y_2 = mx_2 - mx_1 + y_1$

$y_2 = m(x_2 - x_1) + y_1$

$\boxed{y_2 = mx + y_1}$

$\boxed{y_2 + y_1 + mX = 0}$



bestfit

gradient

derivation

$m = \left\{ \frac{dx}{dy} \right\}$ Linear regression

X	y (Actual)
25	2k
30	3k
35	4k
40	5k
45	6k
50	7k

$y = mx + c$

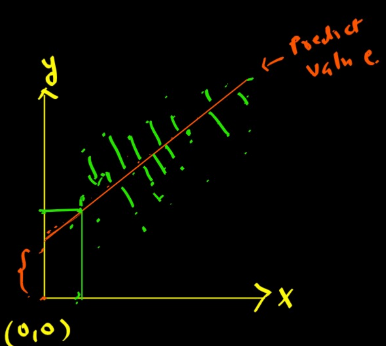
system

$4x_1 + 3y_1 + 4 = 0$

$4x_2 + 3y_2 + 4 = 0$

x_1, y_1

x_2, y_2



y Actual: 80, $\hat{y}$ Predicted: 82 $= \boxed{2}$

Reg Classification

(2) binary → (2)

X (weight)	y (Obese / Not obese)
50	No
60	No
70	No
80	No
90	0
100	0

1000

900 No

100 0

1 →

1000

900

{600, 400}

$\boxed{\text{gradient}}$

$y = mx + c$

$\boxed{m_{new} = m_{old} - \eta \frac{\partial L}{\partial m}}$

$y = mx + c$

→ Linear reg

error $(y - \hat{y})$

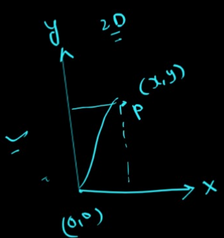
diff

= Ridge

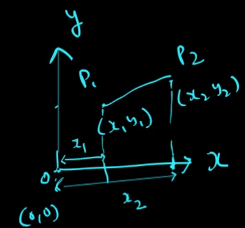
→ Lasso

→ Elastic (R+L)

Visualization →



$= \sqrt{x^2+y^2}$



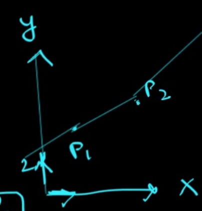
$(x_2 - x_1)$

$(y_2 - y_1)$

$= \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

Line eq. :- $y = mx + c$ → Point slop formula

$ax + by + c = 0$ → general eq.



derivation

Slope $= m = \dfrac{(y_2 - y_1)}{(x_2 - x_1)}$ = ratio $\boxed{\dfrac{2}{1}}$

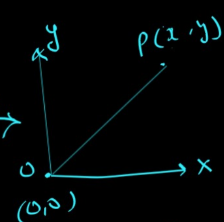
$(y_2 - y_1) = (x_2 - x_1)\, m$

$y = mx$

$(y_2 - y_1) = m(x_2 - x_1)$

$\boxed{y_2 = m(x_2 - x_1) + y_1}$ → Point slop

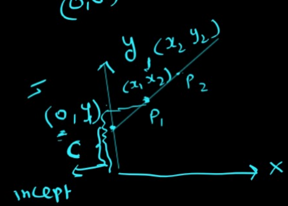
$= \boxed{y = mx + c}$



$m = \dfrac{y_2 - y_1}{x_2 - x_1}$

$m = \dfrac{y - 0}{x - 0}$

$\boxed{y = mx}$



$m = \dfrac{y_2 - y_1}{x_2 - x_1}$ ⇒ $y = mx + c$

$y = mx + c$

Actual

X Height	y weight	$\hat{y}$ (Predict)	
5.5	50		
5.7	60		
6.0	62 → 199		= 90
6.2	63 →		20
6.3	64 →		30
6.8	65 →		40
7.1	70 →		50
7.2	80 →		

$y_1 = mx_1 + c$

$y_2 = mx_2 + c$

$50 = m \times 5.5 + c$ — (1)

$60 = m \times 5.7 + c$ — (2)

$m = ?$

$c = ?$

$\dfrac{(90+20+30+40+50)}{5}$

= error

Error

1. Data should be a linear
2. error value tends to be Zero
3. there should not be any autocorrelation b/w error
4. there should not be multicolinearity b/w col.
5. there shou be homoscadrity

$50 = m\,5.5 + c$ $\quad c = (50 - 5.5m)$ (1) eq1.

$60 = 5.7m + c$

(2) error

(3) optimization

$60 = 5.7m + (50 - 5.5m)$

$60 = 50 + 5.7m - 5.5m$

$60 = 50 + 0.2m$

$60 - 50 = 0.2m$

$m = \frac{100}{0.2}$ $\quad \frac{100}{22} =$

slope. $\boxed{m = 50}$ $\quad$ 2.3.4

$\boxed{c = 75}$

System of Linear eq.

2 eq.

2 var.

Ridge lasso

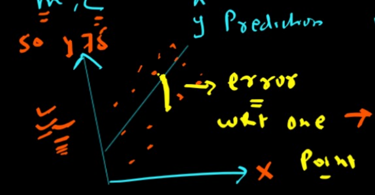
$m = ?$ (slope) $= 5$ $\quad m = \frac{y}{x} = \frac{5}{1} =$

$c = ?$ (intercept)

very first

m, c

$\hat{y}$ Prediction



error

wrt one point

wrt all point $= \boxed{\sum_{i=1}^{n} (y - \hat{y})}$ error

$y = mx + c$

$y = 50x + 75 \rightarrow$ eq

$= 50 \times 1.2 + 75 =$

$= 62 \times 5 + 75$

$= 124 + 75 = 199$

$(y - y_{mean})^2$ = variation

act.

ma $(+, -)$

(1) Calculation

(2) error $\rightarrow$ MSE $\rightarrow \left(\frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2\right) \rightarrow$ error

MAE

RMSE

$(y - \hat{y}) \rightarrow$ stats var. std

$\boxed{m} = 50$

$\boxed{c} = 75$

Act. Predict

Calculation : $y = mx + c$

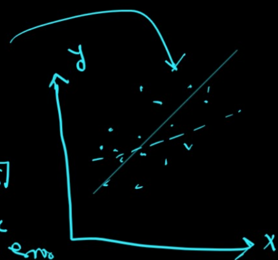
error mse : $\frac{1}{n}\sum_{i=1}^{n}(y - \hat{y}) = \boxed{125}$

minimum

reduce this error

120

115

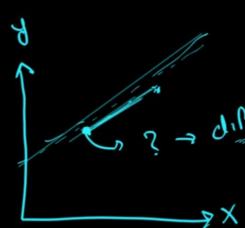


optimization gradient decent

derivative

Calculus

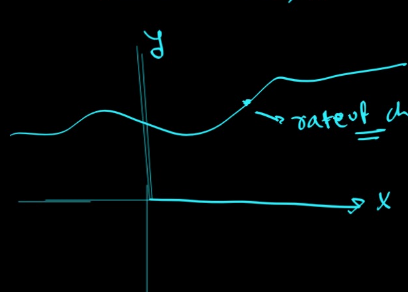
$\left(\frac{dy}{dx}\right)$



? $\rightarrow$ difference

m = rate of changes $= \frac{y_2 - y_1}{x_2 - x_1} = \frac{\Delta y}{\Delta x}$

rate of change



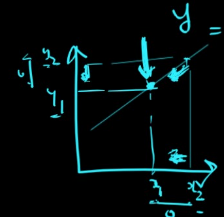
$\rightarrow f(x) = y$

rate of change

$y = mx + c$

$= f(x) = mx + c$

$y = mx + 6$



$f(x) = mx + c$

$m = \frac{\Delta y}{\Delta x}$

$\Delta x \rightarrow 0$

$\Delta y \rightarrow 0$

$\boxed{\frac{dy}{dx}} = \boxed{\frac{\Delta y}{\Delta x}} \lim_{\substack{x \to 0 \\ y \to 0}}$

$= \frac{f(\Delta x + x) + x}{\Delta x}$

Calculus

$x^2 \Rightarrow 2x$

$x^n \Rightarrow n x^{n-1}$

① ② ⓓ

Optimization formula:-
(gradient decent)

slope: $m_{new} = m_{old} - \eta \frac{d(error)}{dm}$

intercept: $c_{new} = c_{old} - \eta \frac{d(error)}{dc}$

error or loss $\rightarrow \boxed{\frac{\partial L}{\partial m}}$

L.R $\rightarrow$ MSE

$m_{new} = m_{old} - \eta \frac{dL}{dm}$

(MSE) $L = \frac{1}{n}\sum_{i=1}^{n}(y-\hat{y})^2$

$= (y-\hat{y})^2$

$= (y-(mx+c))^2 \Rightarrow (a-b)^2 \Rightarrow (a-b)(a-b)$ Partial ANN Chainrule

a^2+b^2-2ab

$= y^2+(mx+c)^2 - 2 \times y^2 \times (mx+c)$

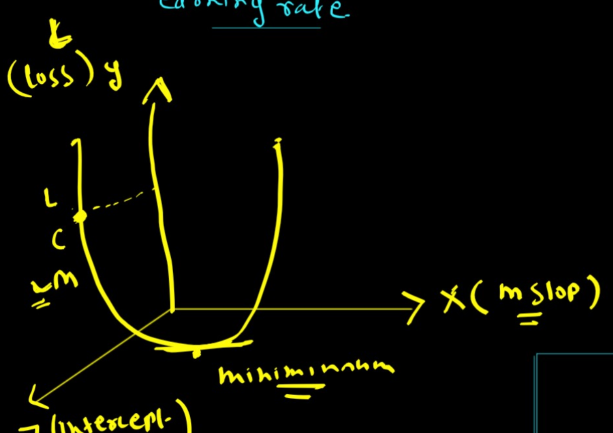
$= y^2+(mx+c)^2 - 2ymx+c$

$L = y^2+(mx)^2+c^2+2mx+c-2y(mx+c)$

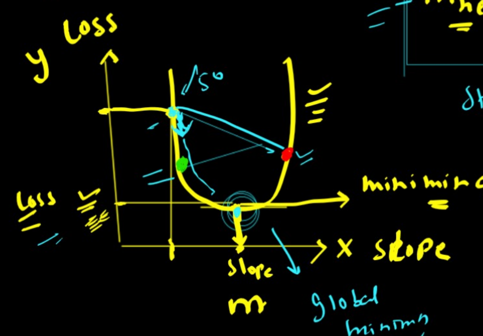
$m_{new} = m_{old} - \eta\frac{dL}{dm} \Rightarrow m_{old} - \eta \frac{1}{n}\sum_{i=1}^{n}\frac{d}{dm}(y-\hat{y})^2$

$\boxed{m_{new}} = m_{old} - \eta \frac{1}{n}\sum_{i=1}^{n}\frac{d}{dm}\left(y^2+(mx)^2+c^2+2mx+c-2y(mx+c)\right)$

optimize learning rate



50 70
LR
$m_{new} = m_{old} - \eta \frac{dL}{dm}$



step by step L.R

0.0001

$\times 0.0001$

$= 50 - 70$

$= 50 - 0.007$

=

LR $y = mx+c$

Data.

① Calculation

$m = ?$

$c = ?$

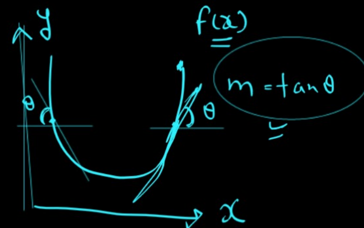
② error Actual Predict

$\rightarrow MSE = \frac{1}{n}\sum_{i=1}^{n}(y-\hat{y})^2 =$

Calculus

③ Optimization → G.D

m ✓
C ✓



Gradient descent.

$$m_{new} = m_{old} - \eta \frac{dL}{dm}$$

↑ eta

$$\Rightarrow m_{old} - \eta \frac{d \frac{1}{n} \sum_{i=1}^{n} (y-\hat{y})^2}{dm}$$

$$(\underset{A}{y} - (\underset{P}{mx+c}))^2$$

$$\Rightarrow m_{old} - \eta \frac{1}{n} \sum_{i=1}^{n} \frac{d(y-\hat{y})^2}{dm}$$

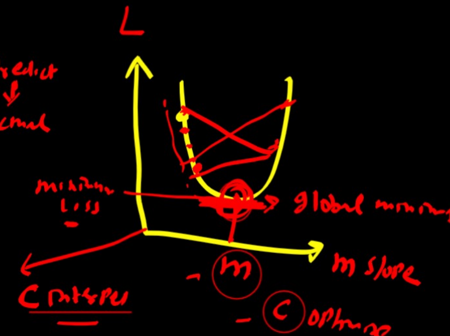
$$\Rightarrow m_{old} - \eta \frac{1}{n} \sum_{i=1}^{n}$$

m_{new} ✓

Learning (very very small)

→ 0.0001 0.001 0.1

– Predict ↓ Actual



– m

– C optimize

$$M_{new} = M_{old} - \eta \frac{\partial L}{\partial m} (-ve)$$

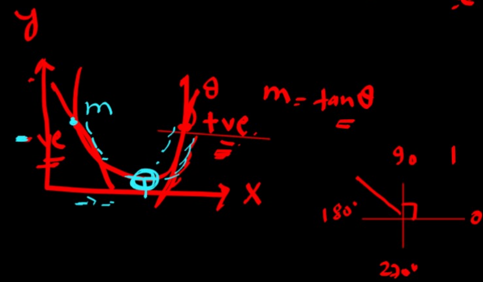
↓ +ve

$$= M_{old} - \eta(+ve)$$

$$= m_{old} - ✓$$

$$= m_{old} - \eta(-ve)$$

$$= m_{old} + ✓$$



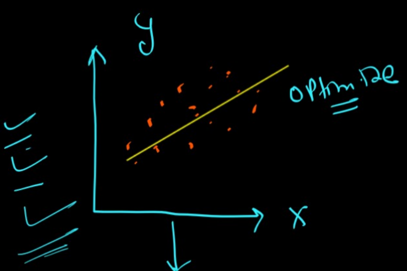
0-∞

flavour :- Ridge, Lasso (extend)

regularization L_1 & L_2

Linear { eq. error g.D } next Optimize or minimum loss

1. Collect.
2. EDA
3. Preprocessing
4. Algo. (model)
5. evaluation



Linear Regression

R^2 and adj. R^2 = good or Bad

one term

evaluate

$$R^2 = 0 - 1$$

{ 0.8 good, 0.9 better, 0.2 → not good }

X weight — y height

X — y

$y = mx + c$

X_1	X_2	X_3	X_4	X_5	y

$$\boxed{y = m_1x_1 + m_2x_2 + m_3x_3 + m_4x_4 + m_5x_5 + C}$$

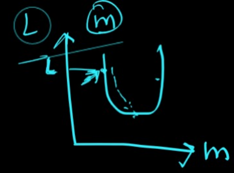
Code :- Actual = = Iteration

$\circlearrowleft$ - change

Script

python

= stop



$$m_{new} = m_{old} - \eta \frac{\partial L}{\partial m} \leftarrow (+)$$

$$\boxed{m_{new} \cdot m_{old}} - \eta (S.V)$$

0

optimal

no need need

J+V

$$\boxed{m_{new} = m_{old}} - \eta \frac{\partial L}{\partial m} (-ve)$$

$$= m_{old} - 0$$

m_{new} $m_{new} = m_{old} +$