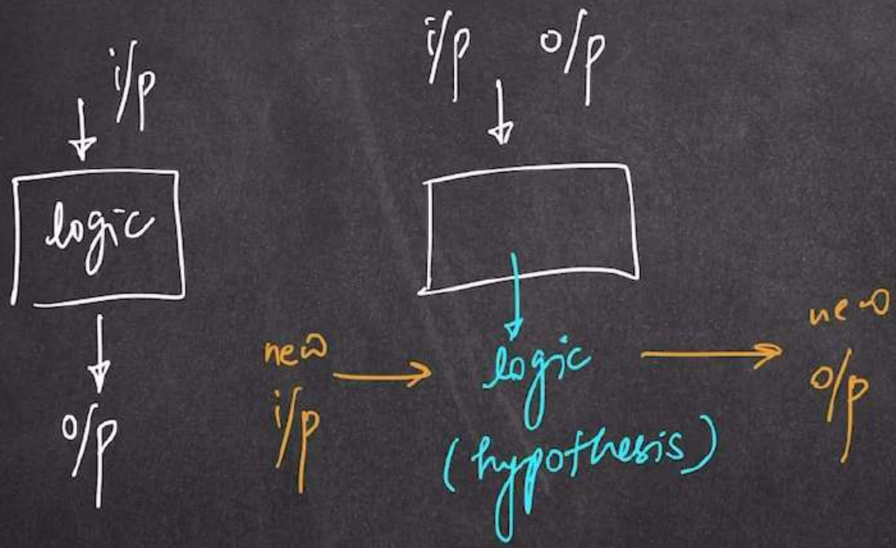
# Machine Learning

ML is a part of computer science where we teach computers to **learn from data**, find patterns & make decision or predictions.




# ML Types

- Supervised Learning → labelled data (i/p → o/p)
- Unsupervised Learning
- Reinforcement Learning

Labelled data is raw data (such as images, text, or audio) that has been paired with a meaningful tag, label, or annotation that informs a machine learning model about the desired outcome



00:36 -04:03

# ML Types

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

labelled data (i/p → o/p)

① classification
(classes / categories)
- spam / not spam
- cats / dogs / rabbit

② Regression
(continuous val)
- age
- stock price
- marks / score

0 – 100
49.5
49.9
49.9

# ML Types

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

labelled data (i/p → o/p)

① classification
(classes / categories)

- spam/not spam
- cats/dogs/rabbit
- grade (A, B, C, D)

② Regression
(continuous val)

- age
- stock price
- marks / score

# ML Types

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

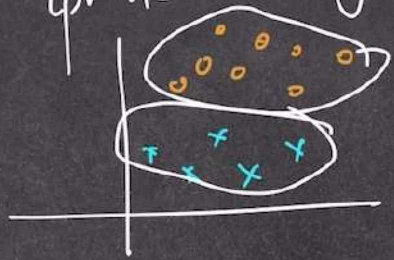
goal-oriented

rewards/penalties

unlabelled data (data)

clustering

iphone + charger



Labelled data (i/p → o/p)

① classification
(classes/categories)
- spam/not spam
- cats/dogs/rabbit
- grade (A, B, C, D)

② Regression
(continuous val)
- age
- stock price
- marks/score

# Supervised Learning

Supervised ML trains algorithms on labeled datasets, meaning each input has a known correct output, to learn patterns and predict outcomes for new data.

i/p o/p

| Email Text | Sender Email | Link | Exclamations | Label |
|---|---|---|---|---|
| "Win a brand new iPhone now! Click the link to claim your prize!" | promo@fakesite.com | yes | 2 | spam |
| "Dear team, please find the attached report for last quarter." | manager@company.in | no | 0 | not spam |
| "Cheap meds available!!! Order today and save big." | sale@pharmacy.in | yes | 3 | spam |

look emails

# Supervised Learning

input (X) output (y)

| Email Text | Sender Email | Link | Exclama ns | Label |
|---|---|---|---|---|
| "Win a brand new iPhone now! Click the link to claim your prize!" | promo@fakesite.com | yes | 2 | spam |
| team, please find the attached report for last quarter." | manager@company.in | no | 0 | not spam |
| "Cheap meds available!!! Order today and save big." | sale@pharmacy.in | yes | 3 | spam |

features

independent (X)

dependent (y)

data

↓

$\boxed{\text{ML}}$

↓

new data → model → predictions

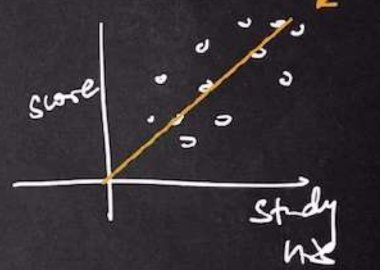
# Supervised Learning

1 Data – preprocess + featuring
(encoding)

2 Select Model

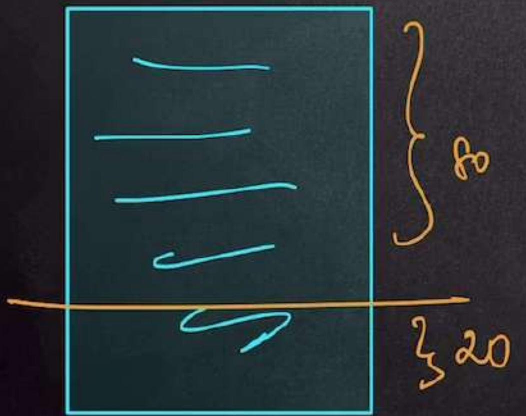
3 Train / Test

4 underfit
overfit




| Email Text | Sender Email | Link | Exclamations | Label |
|---|---|---|---|---|
| "Win a brand new iPhone now! Click the link to claim your prize!" | promo@fakesite.com | ~~yes~~ 1 | 2 | spam |
| "Dear team, please find the attached report for last quarter." | manager@company.in | ~~no~~ 0 | 0 | not spam |
| "Cheap meds available!!! Order today and save big." | sale@pharmacy.in | ~~yes~~ 1 | 3 | spam |

yes (1) no (0)




# **Supervised** Learning

*Types of problems*

1. **Regression** - output is a $\boxed{\text{continuous}}$ numerical value

→ house price
→ weather
→ aqi
→ stock price
→ score
→ car price

2. **Classification** - output is a category or class

| make | Auto/man | milage | kms | year |
|---|---|---|---|---|
| Toyota | 1 | | | 2018 |
| BMW | 0 | | | 2019 |
| M | 0 | | | 2022 |

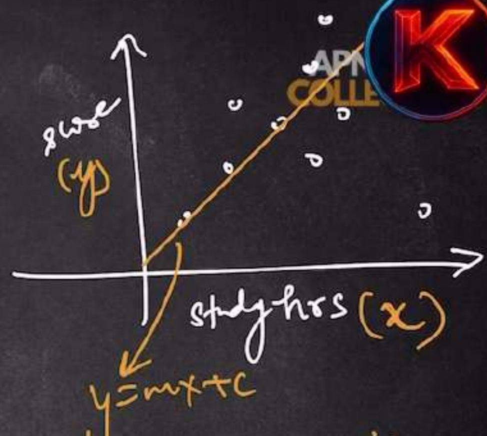
# Supervised Learning

*Types of problems*

1. **Regression** - output is a $\boxed{\text{continuous}}$ numerical value

→ house price
→ weather
→ aqi
→ stock price
→ score
→ car price

2. **Classification** - output is a category or class

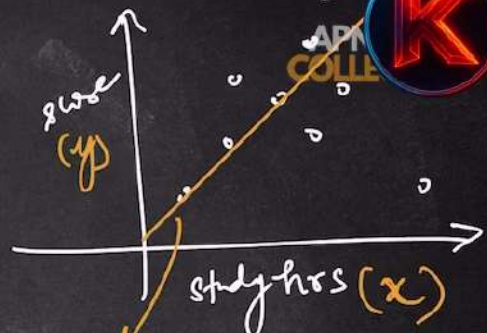



Linear Regression

# Supervised Learning

*Types of problems*

1. **Regression** - output is a $\boxed{\text{continuous}}$ numerical value
   - → house price
   - → weather
   - → aqi
   - → stock price
   - → score
   - → car price




$y = mx + c$

Linear Regression

2. **Classification** - output is a category or class
   - → spam / non-spam
   - → cat / dog / rabbit
   - → benign / malignant
   - → 0–9

   0

   7

(1) Binary Classification

(2) multi-class classification

(3) multi label classification

# Scikit-learn

***sklearn*** is an open source ML library that supports supervised and unsupervised learning.

1. well-documented
2. easy to learn
3. works well with np & pd
4. proffessionals begs.

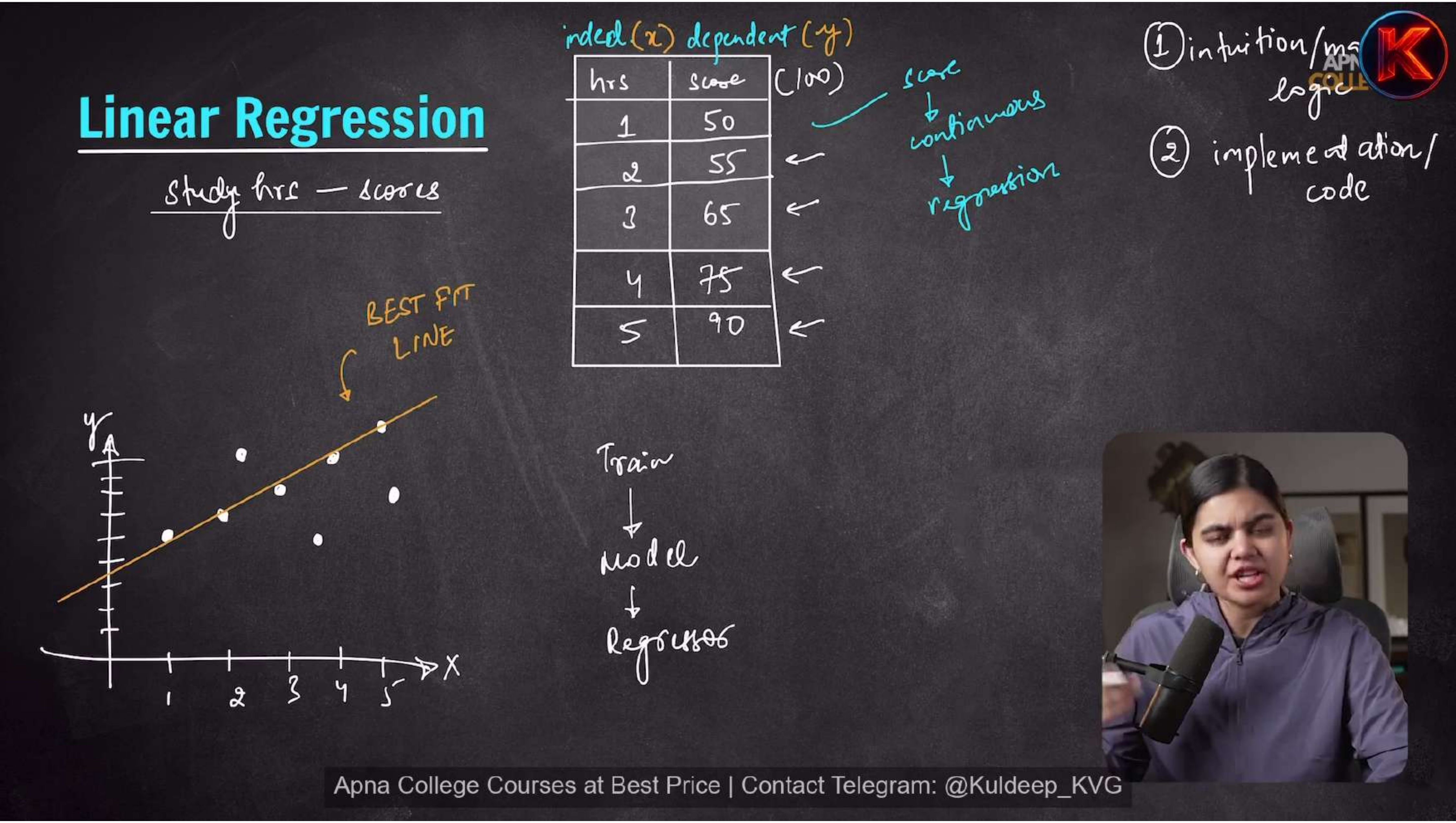
# Algorithms

- Linear Regression
- Logistic Regression
- kNN → k nearest neighbors
- Decision Trees
- Naive Bayes
- SVM etc.

# Algorithms

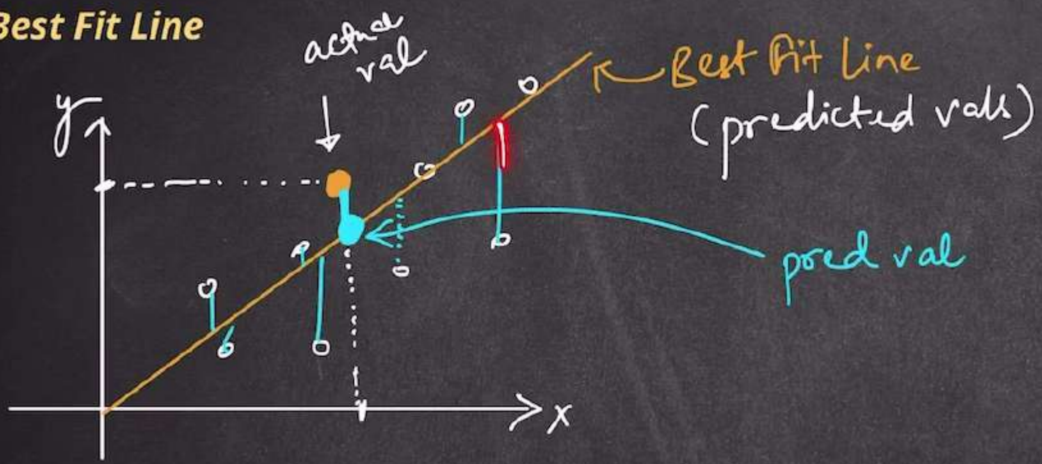
- Linear Regression → regression
- Logistic Regression → classification
- kNN → k nearest neighbors
- Decision Trees → R/C
- Naive Bayes → C
- SVM etc. → R/C

ANN
RNN
CNN


Linear Regression
study hrs — scores
index (x) dependent (y)
hrs score (100)
1 50
2 55
3 65
4 75
5 90
score
continuous
regression
1 intuition/ma
logic
2 implementation/
code
BEST FIT LINE
y
X
1 2 3 4 5
Train
Model
Regressor
Apna College Courses at Best Price | Contact Telegram: @Kuldeep_KVG

# Linear Regression

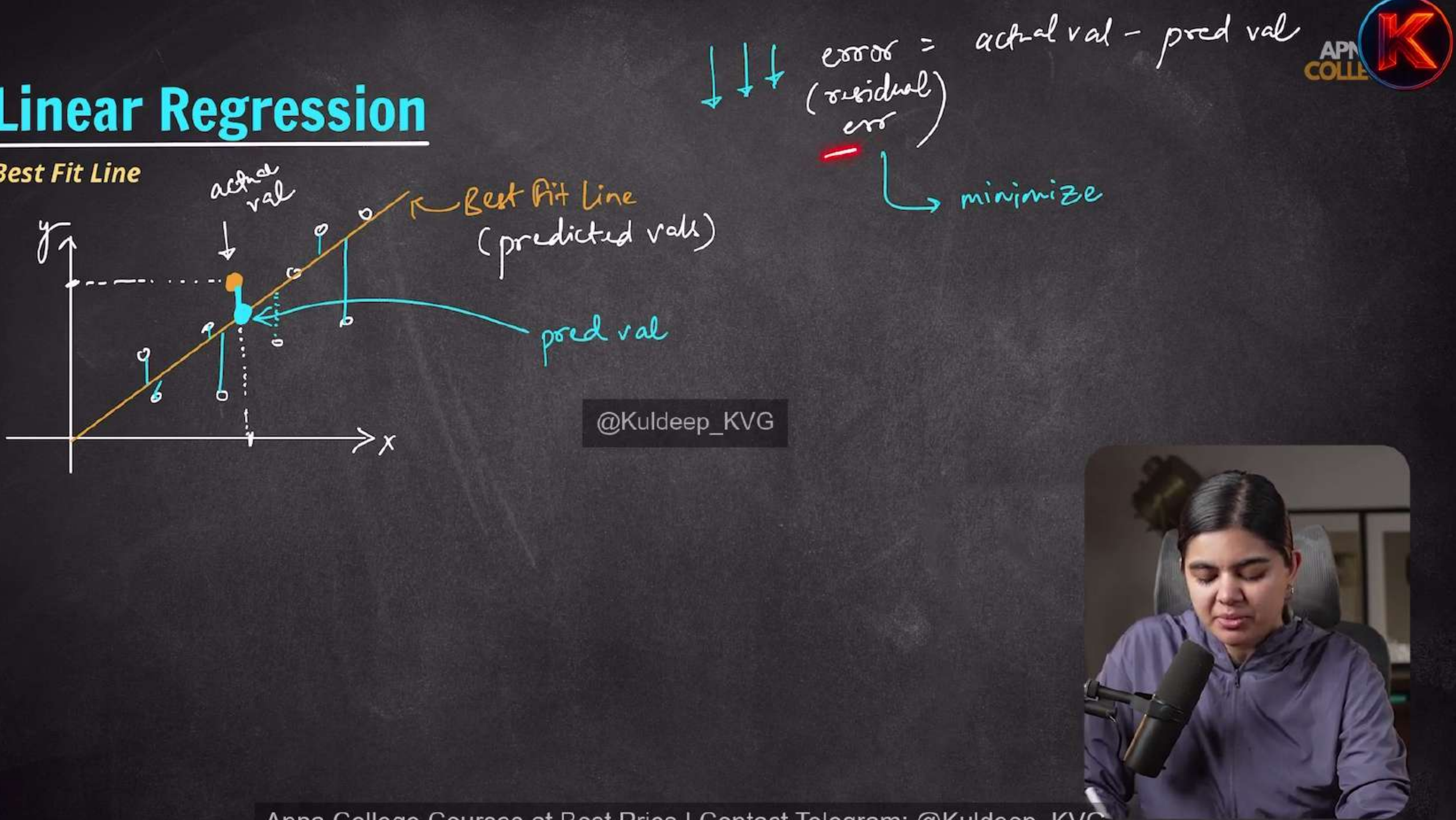
***Best Fit Line***




error = actual val - pred val
(residual err)

# Linear Regression

***Best Fit Line***



# Linear Regression

**Best Fit Line**

$y = c + mx$

$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1}$$

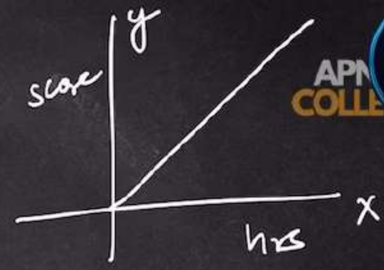
1st feature

hypothesis fnx

bias/ intercept

1st param/ 1st coeff

① 2D




② 3D

③ 4D → hyperplane

5D, 6D .. —

# Linear Regression

**Best Fit Line**

$y = c + mx$

$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1}$$

$x_1$ → 1st feature

$h_\theta(x)$ → hypothesis fnx

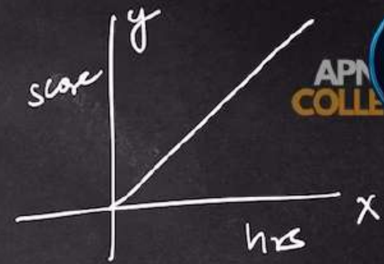
$\theta_0$ → bias/ intercept

$\theta_1$ → 1st param/ 1st coeff

$x_1, x_2, x_3 \ldots \ldots x_n$ → n i/p features

$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \ldots\ldots \theta_n x_n}$$

① 2D



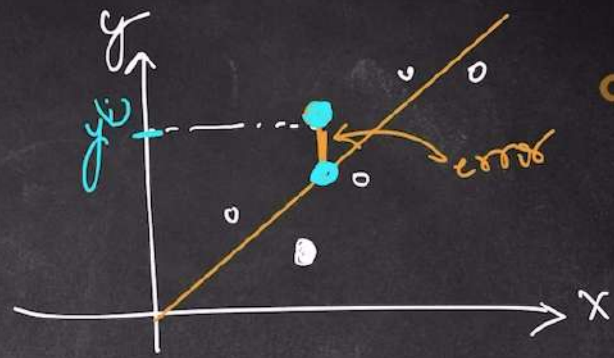

② 3D

③ 4D → hyperplane

5D, 6D .. —

# Linear Regression

$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1}$$




### *Cost Function*

Measures how well the model's predictions match the actual data

pred val - actual val

↓

$$\sum_{i=1}^{m} h_\theta(x^{(i)}) - y^{(i)}$$

squared err

log of err

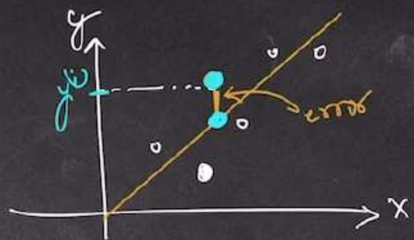
absolute err

⋮

Mean Squared Error (MSE)

# Linear Regression

$$h_\theta(x) = \theta_0 + \theta_1 x_1 \rightarrow 1$$




$\theta_0, \theta_1$ – i

↓

$J(\theta)\downarrow$

↓

$\theta_0, \theta_1$ – update

↓

$J(\theta)\downarrow$

↓

$\theta_0, \theta_1$ – upd

⋮

### Cost Function

Measures how well the model's predictions match the actual data

pred val - actual val

↓

$$\sum_{i=1}^{m} h_\theta(x^{(i)}) - y^{(i)}$$

squared err

log of err

absolute err

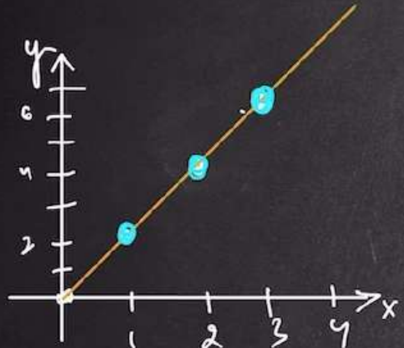
⋮

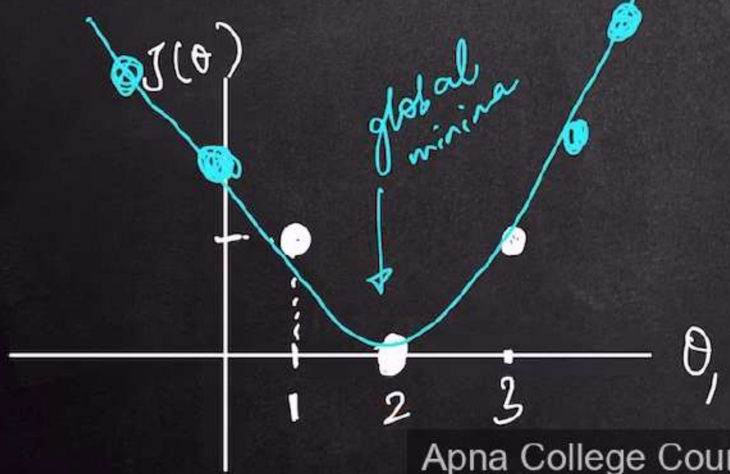
Mean Squared Error (MSE)

$$\boxed{J(\theta) = \frac{1}{2.m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2} \rightarrow 2$$

# Linear Regression

data points = (1, 2) (2, 4) (3, 6)




case 1: $\theta_0 = 0, \theta_1 = 2$ $\quad h_\theta(x) = 2x$

$$J(\theta) = \frac{1}{6}(0^2 + 0^2 + 0^2) = 0$$

$\theta_1 = 2, J(\theta) = 0$

case2: $\theta_0 = 0, \theta_1 = 1$ $\quad h_\theta(x) = x$

$$J(\theta) = \frac{1}{6}(1^2 + 2^2 + 3^2) = \frac{14}{6} = 2.34$$

$\theta_1 = 1, J(\theta) = 2.34$

case 3: $\theta_0 = 0 \quad \theta_1 = 3$ $\quad h_\theta(x) = 3x$

$$J(\theta) = \frac{1}{6}(1^2 + 2^2 + 3^2)$$

$J(\theta) = 2.34, \theta_1 = 3$

$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1}$$

$$\boxed{J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2}$$

**Gradient Descent**

$$\frac{\Delta J(\theta)}{\Delta(\theta)} = \frac{\partial (J(\theta))}{\partial \theta}$$

Gradient → slope

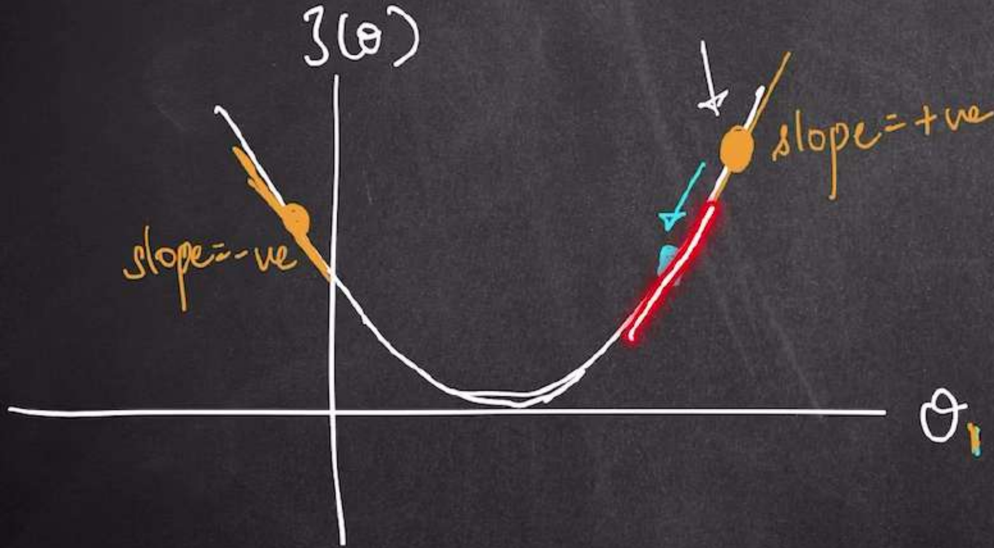
Descent → moving downwards

$\alpha \rightarrow$ lear[...]

# Linear Regression

### *Gradient Descent*

Gradient Descent is an iterative optimization algorithm used to minimize a cost function by adjusting model parameters in the direction of the steepest descent of the function's gradient.




convergence theorem

$$\boxed{\theta_k = \theta_k - \alpha \cdot \frac{\partial (J(\theta))}{\partial \theta_k}}$$

CT: $\theta_1 = \theta_1 - \underbrace{\alpha(\text{slope})}_{+ve}$

# Linear Regression

$$\frac{\Delta J(\theta)}{\Delta(\theta)} = \frac{\partial(J(\theta))}{\partial\theta}$$

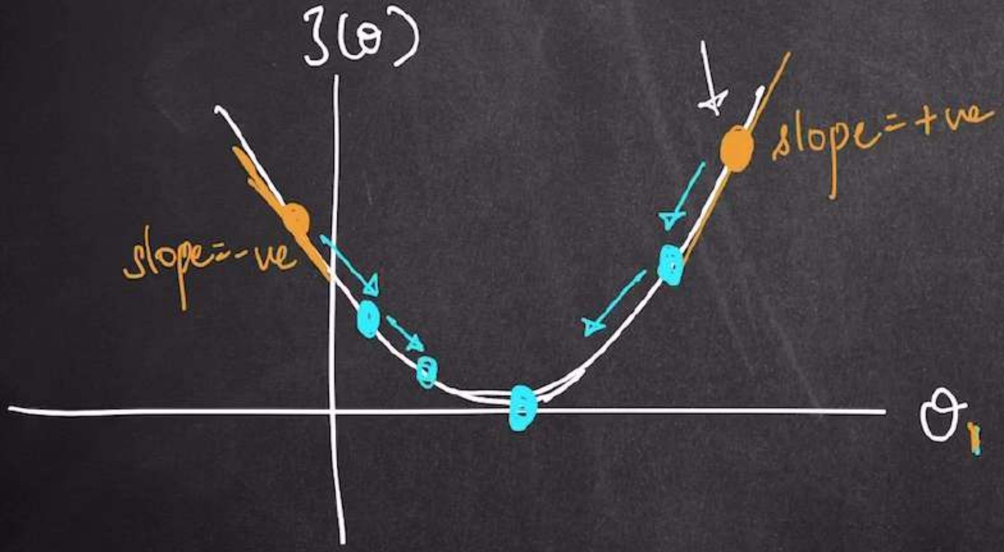
Gradient $\downarrow$ slope

Descent $\downarrow$ moving downwards

$\alpha \rightarrow$ lear

## Gradient Descent

Gradient Descent is an iterative optimization algorithm used to minimize a cost function by adjusting model parameters in the direction of the steepest descent of the function's gradient.




convergence theorem

$$\boxed{\theta_k = \theta_k - \alpha \cdot \frac{\partial(J(\theta))}{\partial\theta_k}}$$

CT: $\theta_1 = \theta_1 \underbrace{- \alpha(\underbrace{\text{slope}}_{-ve})}_{+ve}$

$$\frac{\Delta J(\theta)}{\Delta(\theta)} = \frac{\partial(J(\theta))}{\partial\theta}$$

Gradient → slope

Descent → moving downwards

$\alpha \rightarrow$ lear

# Linear Regression

### Gradient Descent

Gradient Descent is an iterative optimization algorithm used to minimize a cost function by adjusting model parameters in the direction of the steepest descent of the function's gradient.

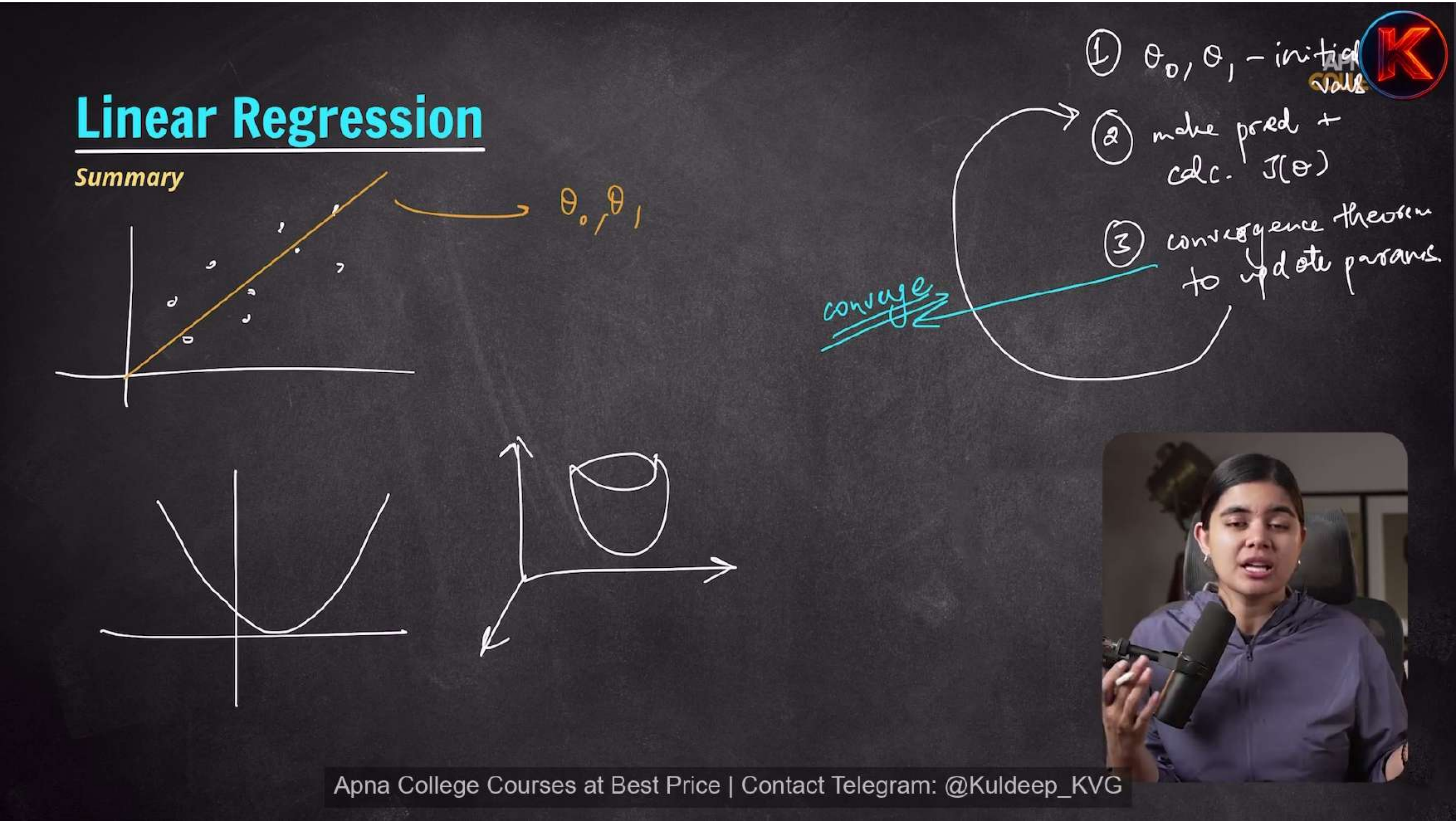

convergence theorem

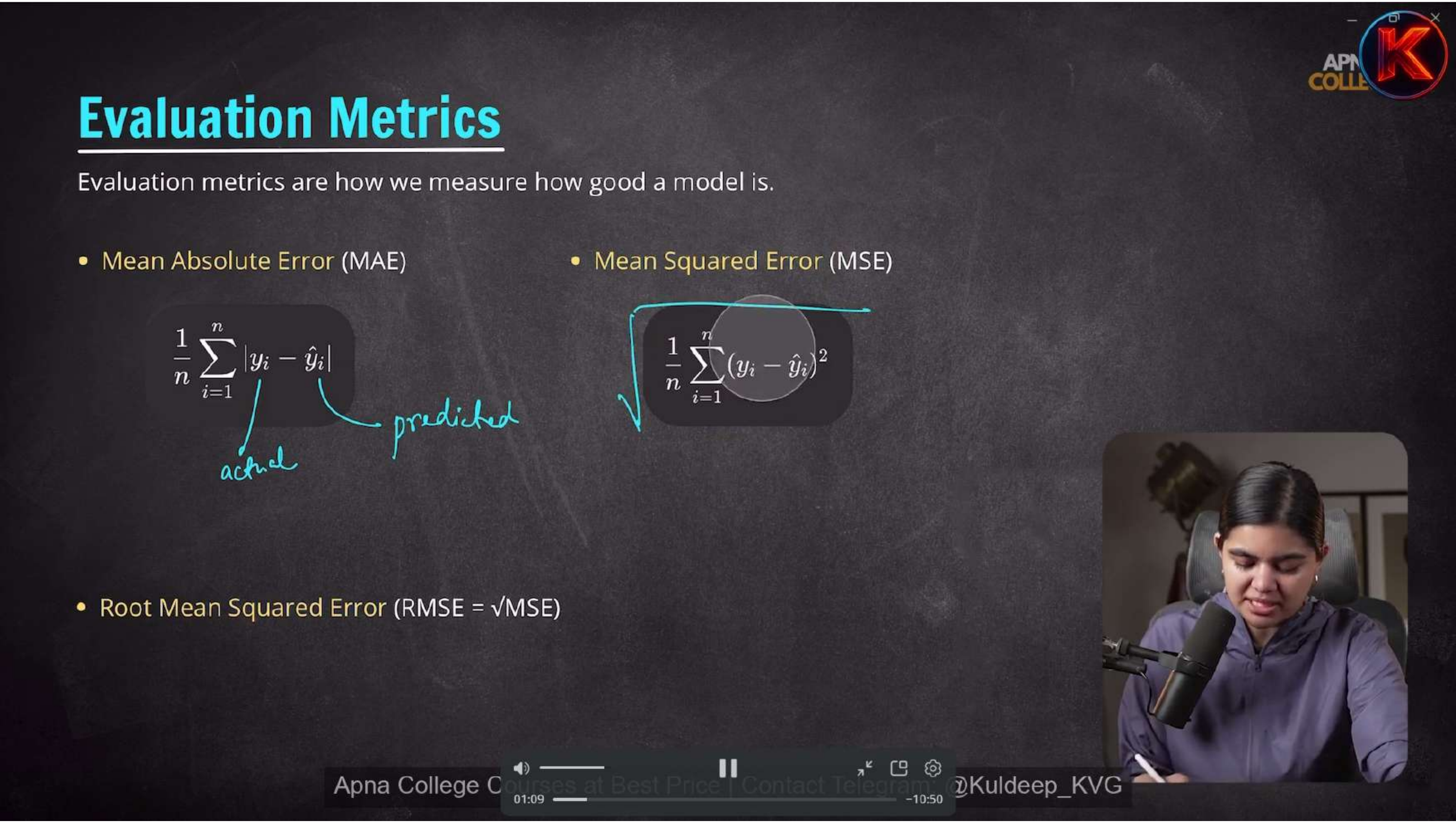
$$\boxed{\theta_k = \theta_k - \alpha \cdot \frac{\partial(J(\theta))}{\partial\theta_k}}$$

$\alpha \rightarrow$ appropriate

$\alpha \rightarrow$ too small $\longrightarrow$ slow training

$\alpha \rightarrow$ too large $\longrightarrow$ divergence / miss minima


Linear Regression
Summary
θ0, θ1
1 θ0, θ1 - initial vals
2 make pred + calc. J(θ)
3 convergence theorem to update params
converge
Apna College Courses at Best Price | Contact Telegram: @Kuldeep_KVG


Evaluation Metrics
Evaluation metrics are how we measure how good a model is.
• Mean Absolute Error (MAE)
$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
actual
predicted
• Mean Squared Error (MSE)
$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
• Root Mean Squared Error (RMSE = √MSE)
Apna College Courses at Best Price | Contact Telegram: @Kuldeep_KVG
01:09
-10:50

# Evaluation Metrics

Evaluation metrics are how we measure how good a model is.

variance

0 - 1

X it explains nothing

model is perfect fit

0.7-0.8

0.8-0.9

0.9+

- R-squared ($R^2$)

$(\text{residual error})^2$

$$1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

- Adjusted R-squared

$$1 - \frac{(1 - R^2)(n-1)}{n - p - 1}$$

$X = [\text{study hrs} \rightarrow \text{scores } (y)] \rightarrow R^2$

for color

$R^2 = 0.85 \uparrow$

adj $R^2 = 0.75 \downarrow$

# Evaluation Metrics

Evaluation metrics are how we measure how good a model is.

- R-squared ($R^2$)

(residual error)$^2$

$$1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

$0.78 \rightarrow 78\%$ strong

- Adjusted R-squared

$$1 - \frac{(1 - R^2)(n-1)}{n-p-1}$$

$n$ = no. of rows

$p$ = no. of features

$R^2$ = R. squared



09:03 -02:56