① Overfitting - high variance

Train — ~90%.
Test — ~70%.
more than 10%.

DT → "RF"

Bagging — LB HV → LB LV

② Underfitting - high bias

Train → less than 70%.
Test → ~50%.

Boosting ⇐ Ada boost
Gradient Boosting
XGBoosting

Regression, SVM, KNN

Variance > Obs

Dataset

(3) Data leakage - Train & Test - Kaggle Project

↓ more accuracy

Merge

Real time Job

very good

UAT

Testing

Production

↓

deploy ⇒ Docker

↳ Kubernetes

Missing | Feature Scaling | Outlier

① Missing data

↓

Train

1
NAN
2
8

Test

3
4
NAN

$\frac{1+2+8}{3} = \frac{11}{3}$ = 3.9

$\frac{3+4}{2} = \frac{7}{3}$ 3.5

Mean

1
NaN
2
3
4
NaN
8

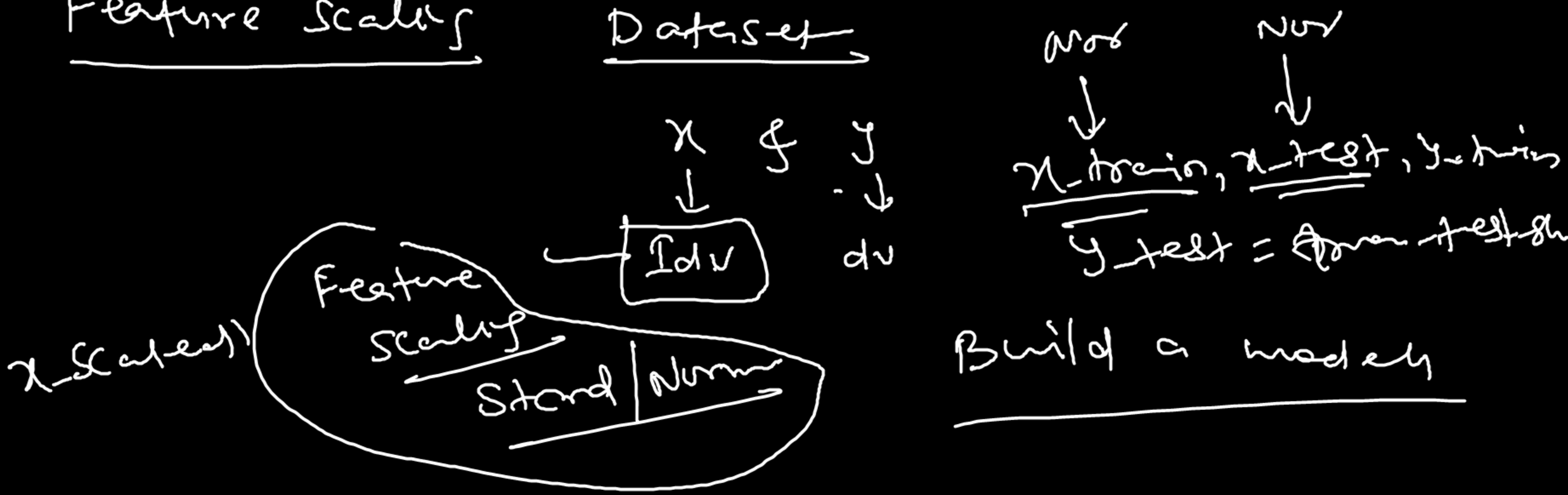
3.6

$\frac{1+2+3+4+8}{5} = \frac{18}{5}$ = 3.6

- Incorrect

Feature Scaling

Dataset



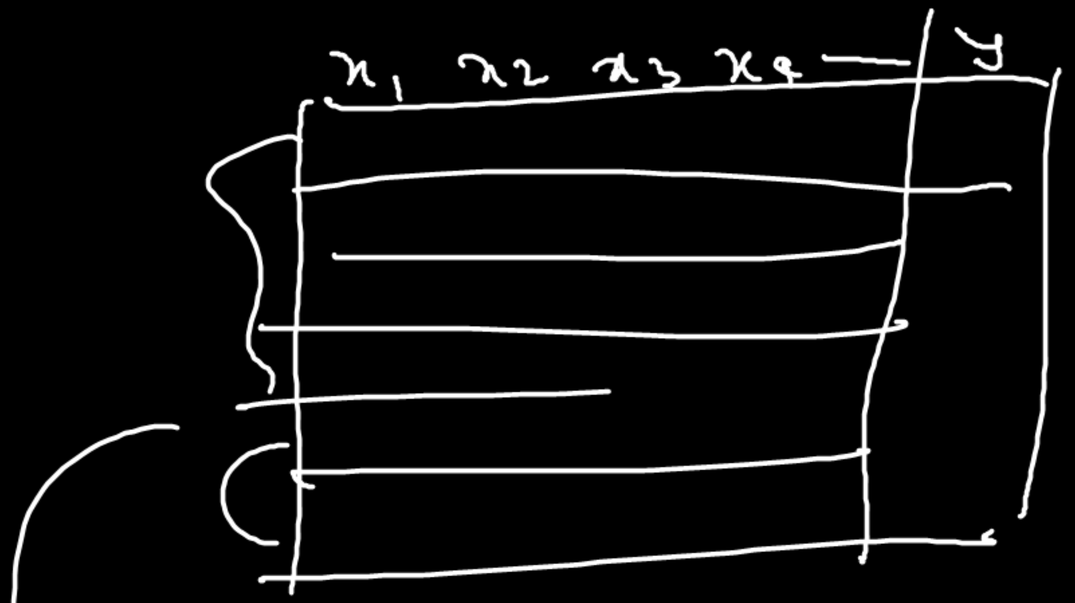
Build a model

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

Feature Scale — X_train

Model — without Scal

```
Pred(X_test)
```



$x = x_1 \quad x_2 \quad x_3 \quad x_4 \; —$

$y = y$

Feature Scaling = x X

Train_Test_Split =

x_train , x_test —

R1 R3 R5

R2 R4

FS

-3 +3

	Age	Sal	DV=y
-3 R1	20	50K	
R2	23	70K	
R3	30	100K	
R4	45	150K	
R5 +3	60	200K	

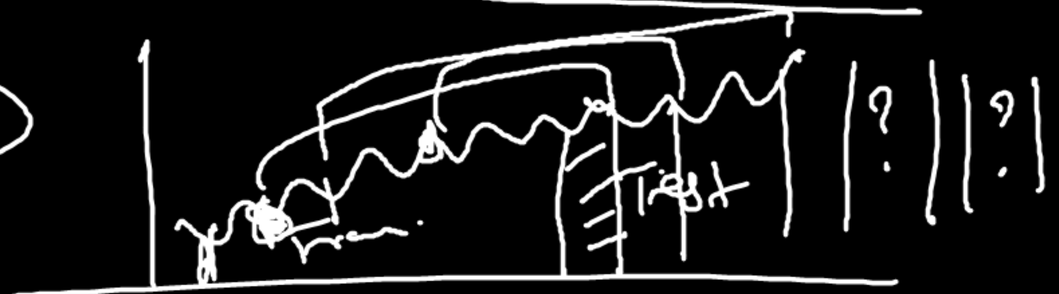
TSF → Sequence of

Time Series forecast →

↳ Future Predict-

Past data for a sequence order

- Seasonality
- Trend
- Cyclicity
- Random / Noisy Error



1- 100 from 101

2 - 101 — 102

3 - 102 — 103

Low variance
high bias

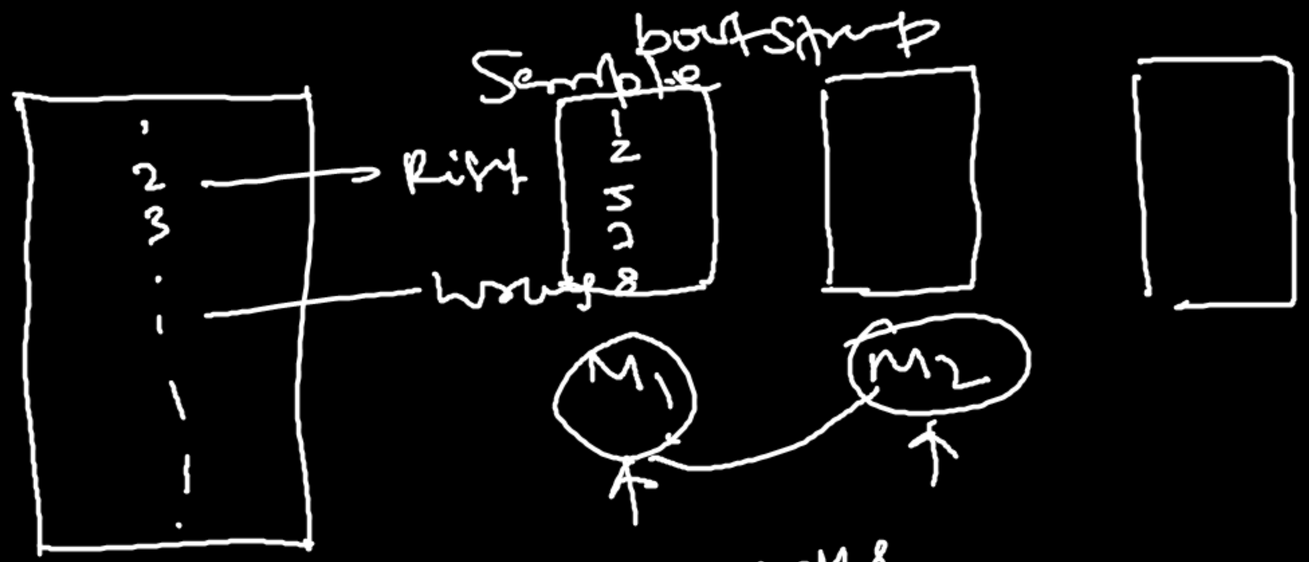
Boosting → Sequence
- underfitting | high bias

↙

general ensemble method that aims to creat a strong classifier or regressor by combining the Predictor of several weaker model.

The idea is to build the Strong model incrementally, by sequentially adding weak models that are trained to correct the mistake made by the existing ensemble.

Sequential model



1. sequentially trains
2. Sample data - with replacement
3. Build model

Accuracy - Poor

marks

Boosting - Powerful

- Adaboost
- Gradient Boosting
- XGBoost ✓

Cat boosting
Light GBM

HBLV

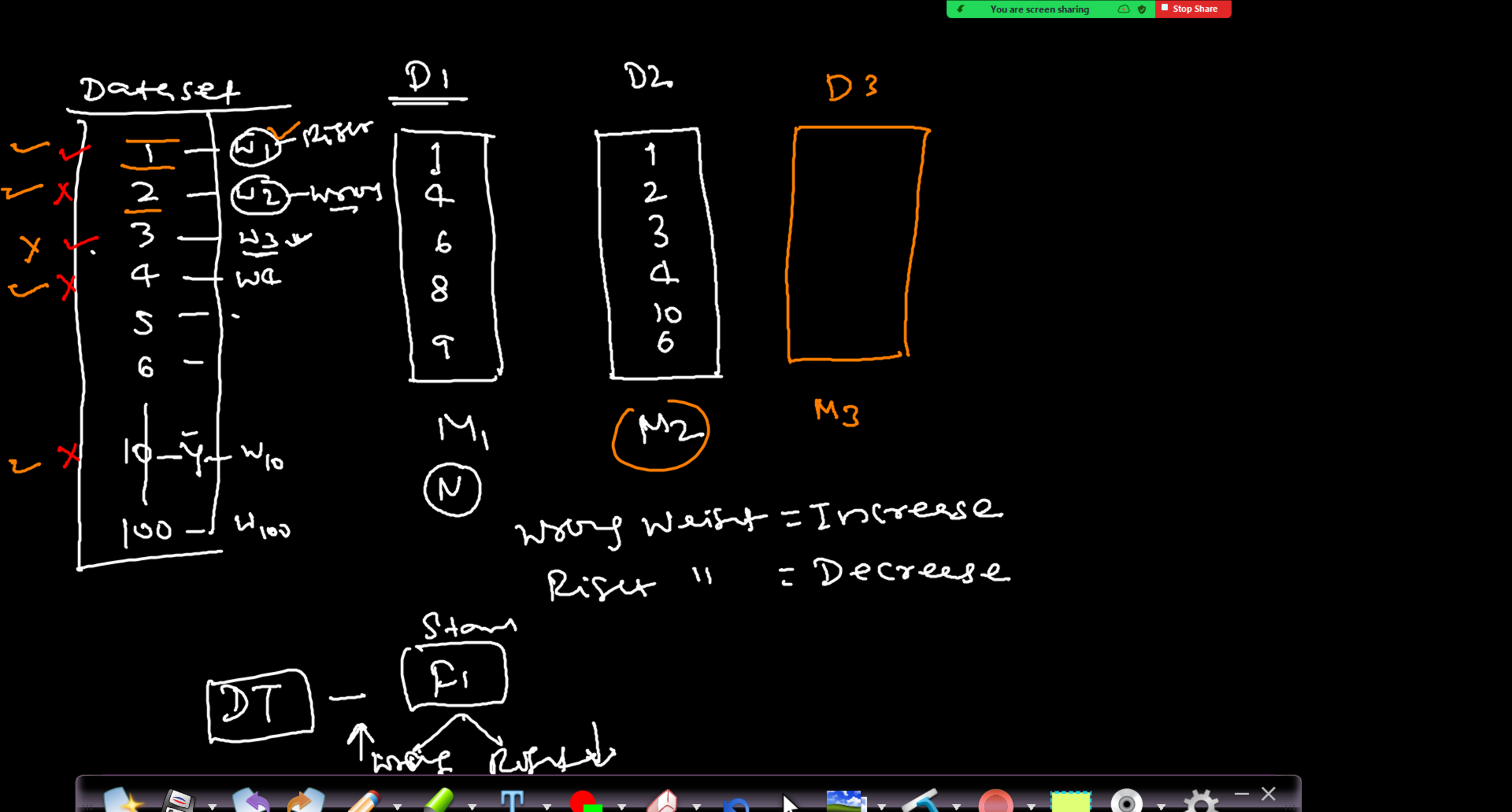
LB LV - object

Ada boost

1. Weak learner → Min ip 5 1.1.
2. Decision Stumps
3. +1 & −1
   Positive Negative

Deep Learning - MLP

Stage wise - Additive method



F1	F2	F3	— —	Fm	o/p	Sample weight
					Y	1/5
					N	1/5
					Y	1/5
					Y	1/5
					N	1/5
					1	1

	pred
Y	Yes
N	No
Y	No
N	N
Y	Y.

Decr.

Increse

How to increase weight

Performance of Stump $= \frac{1}{2}\log_e\left(\frac{1-TE}{TE}\right)$

Total Error $= \frac{1}{5}$

$= \frac{1}{2}\log_e\left(\frac{1-\frac{1}{5}}{\frac{1}{5}}\right)$

$= \frac{1}{2}\log_e\left(\frac{4/5}{1/5}\right)$

$= \frac{1}{2}\log_e(4) = 0.3$

NOTE: We have to increase the weight which is wrongly classified and decrease the weight for rightly classified.

* Incorrectly classified — Increase the weight

$$(\text{New Sample})_{\text{weight}} = (\text{Old Sample})_{\text{weight}} * e^{\text{Per. of Stump}}$$

$$= \left(\frac{1}{5}\right) * e^{0.3} \quad (e^{0.3} \approx 2.718) \quad = \frac{1}{5} * (2.718)^{0.3} = 0.7$$

* Rightly Classified Weight — Decrease the weight

$$(\text{New sample})_{\text{weight}} = (\text{Old sample})_{\text{weight}} * e^{-\text{P.Stump}}$$

$$= \frac{1}{5} * e^{-0.3} = \frac{1}{5} * (2.718)^{-0.3} = \frac{1}{5} * \frac{1}{(2.718)^{0.3}} = 0.74$$

Table

F_1	F_2	F_3	——	F_{10}	O/P	Old weight	New weight	Normalize 0 - 1
					Y →	1/5 = 0.	0.1	
					Y	1/5	0.1	
					N	1/5	0.35	
					Y	1/5	0.1	
					N	1/5	0.1	

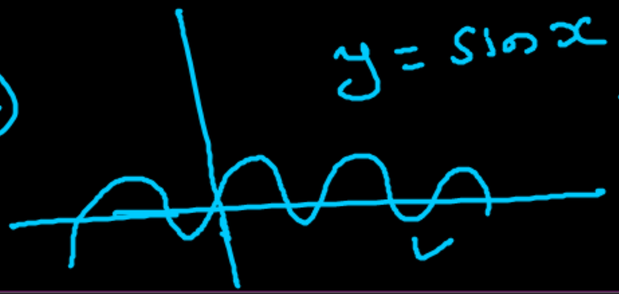
Σ = 1

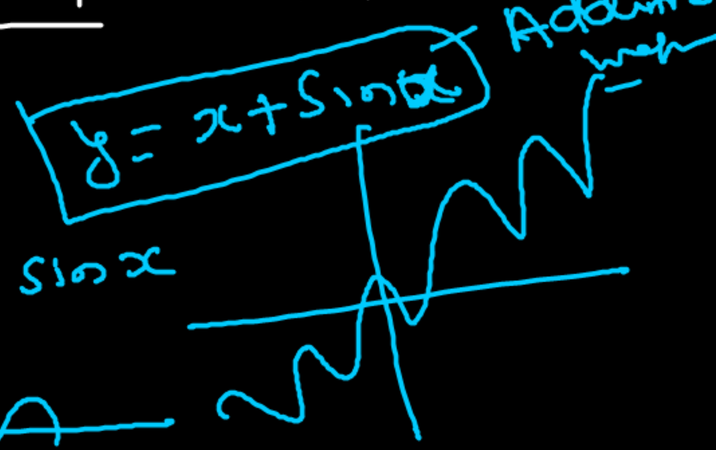
Gradient Boosting :- Using the concept - Additive modelling

↳ typically decision tree. The method works by iteratively adding model to the ensemble, with each new model trained to correct the mistakes made by the combined ensemble of existing model.

Loss function, Residual, Gradient

Additive model





Sequential —

M_1 M_2 M_3

$f_1(x)$ $f_2(x)$ $f_3(x)$

$$f(x) = f_1(x) + f_2(x) + f_3(x)$$

↳ Additive model

Additive model

x	y

$y = f(x)$

M_1 $f_1(x)$ $f_2(x)$ $f_3(x)$ — →