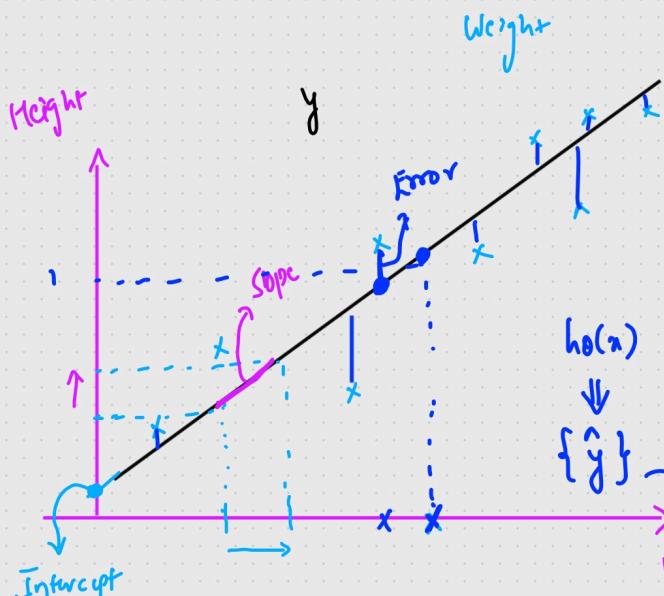
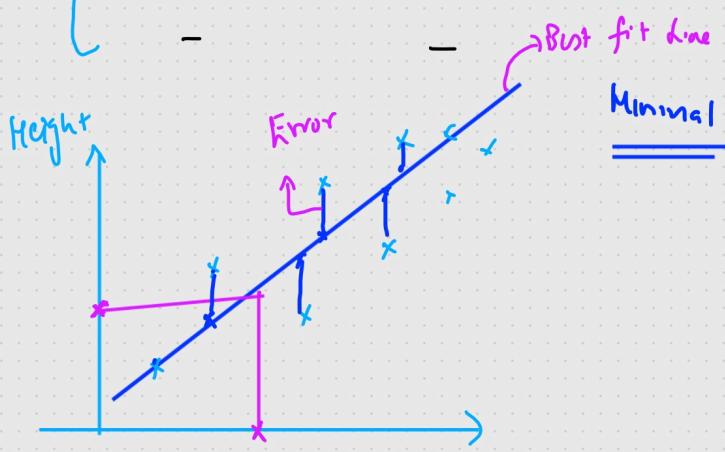


Simple Linear Regression

Supervised ML \rightarrow Regression

<u>Dataset</u>	IIP features
<u>Weight</u>	<u>X</u>
74	170cm
80	180cm
75	175.5cm
-	-



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

$$\theta_0 = \text{Intercept}$$

$$\theta_1 = \text{Slope or Coefficient}$$

$$\text{if } x=0 \quad \underline{\text{Error}} \quad (y - \hat{y})$$

$$h_{\theta}(x) = \theta_0$$



$$y = mx + c$$

$$y = \beta_0 + \beta_1 x$$

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

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots + \theta_n x_n$$

$$\dots + \theta_n x_n$$

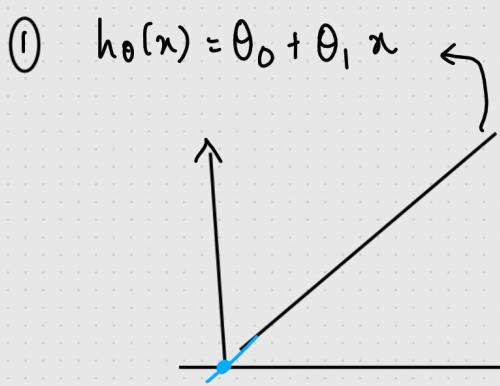
Cost function

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2 \Rightarrow \text{Mean Squared Error}$$

↑ predicted ↑ true O/P
Error

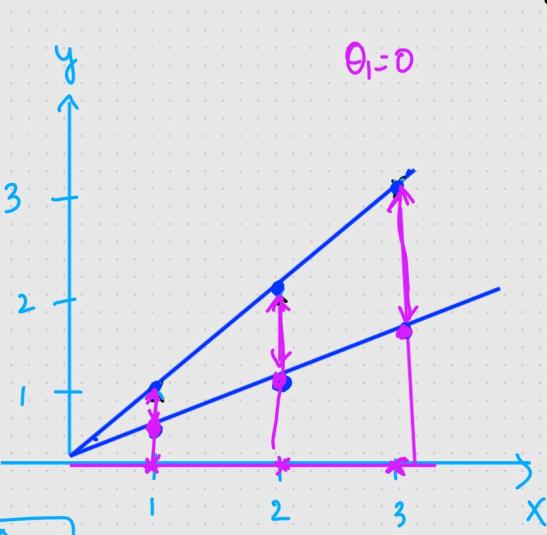
Final Aim: What we need to solve

Minimize $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2$ ↓↓↓
 θ_0, θ_1



$$\boxed{\theta_0 = 0}$$

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



$$h_\theta(x) = \theta_1 x$$

$$\text{det } \theta_1 = 1 \quad \{ \text{slope} \}$$

$$h_\theta(x) = 1 \quad x=1$$

$$h_\theta(x) = 2 \quad x=2$$

$$h_\theta(x) = 3 \quad x=3$$

<u>DATASET</u>	
X	Y
1	1
2	2
3	3

$$h_\theta(x) = \theta_1 x$$

$$\text{det } \theta_1 = 0.5$$

$$h_\theta(x) = 0.5 \quad \text{if } x=1$$

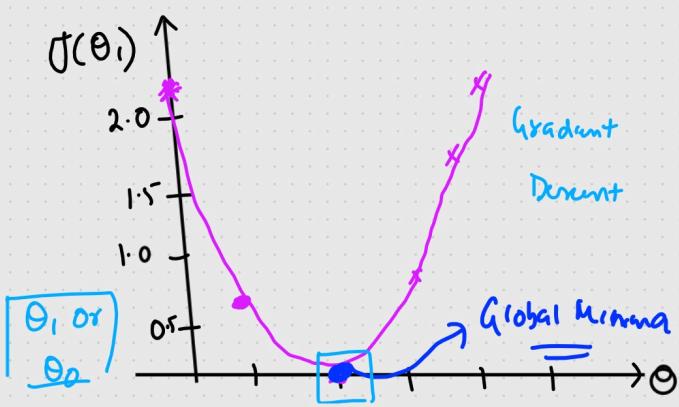
$$h_\theta(x) = 1 \quad \text{if } x=2$$

$$h_\theta(x) = 1.5 \quad \text{if } x=3$$

$$\boxed{\theta_1 = 1}$$

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$= \frac{1}{2 \times 3} \left[(1-1)^2 + (2-2)^2 + (3-3)^2 \right]$$



$$\boxed{\theta_1 \text{ or } \theta_0}$$

$$J(\theta_1) = 0 \leftarrow$$

$$\underline{\underline{\theta_1 = 0.5}}$$

Error has been
minimized

$$J(\theta_1) = \frac{1}{2 \times 3} \left[(0.5 - 1)^2 + (1 - 2)^2 + (1.5 - 3)^2 \right]$$

$$J(\theta_1) \approx 0.58$$

$$\underline{\underline{\text{if } \theta_1 = 0}}$$

$$J(\theta_1) = \frac{1}{2 \times 3} \left[(0 - 1)^2 + (0 - 2)^2 + (0 - 3)^2 \right]$$

$$J(\theta_1) \approx 2.3$$

Convergence Algorithm {Optimize the changes of

θ_1 value}

$$J(\theta)$$

Repeat until convergence

{

θ_1 value much
more efficiently

$$\left\{ \theta_j := \theta_j - d \left[\frac{\partial J(\theta_j)}{\partial \theta_j} \right] \rightarrow -ve \right. =$$

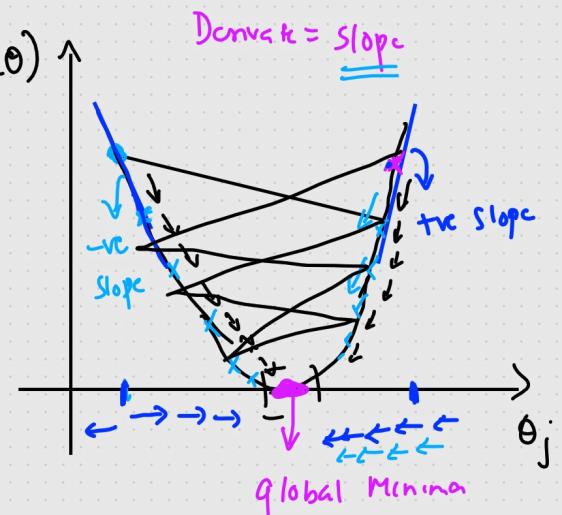
$$\left. \begin{array}{l} \\ \end{array} \right\} \theta_j = \theta_j - d (+ve)$$

$$= \theta_j - (+ve)$$

d = Learning Rate

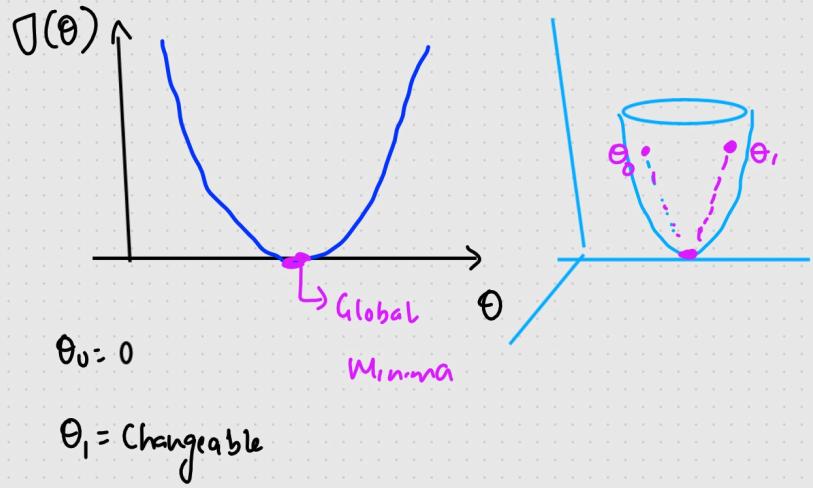
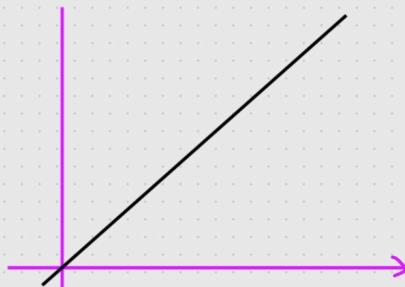
$$\left\{ \begin{array}{l} \theta_j = \theta_j - d (-ve) \\ \theta_j = \theta_j + (+ve) \end{array} \right.$$

$$\boxed{d = 0.001} \leftarrow$$



Final Conclusion

GRADIENT DESCENT



Convergence Algorithm

repeat until convergence

{

$$\theta_j := \theta_j - \alpha \left[\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \right]$$

}

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$j = 0 \text{ and } 1 \quad \frac{\partial}{\partial x} (x)^2 = 2x$$

$$\frac{\partial}{\partial x} x^h = x x^{n-1} \quad \frac{\partial}{\partial x}$$

$$\rightarrow \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2$$

if

$$h_\theta(x) = \theta_0 + \theta_1 x \rightarrow 0$$

$$\begin{aligned} j=0 \Rightarrow \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) &= \frac{\partial}{\partial \theta_0} \frac{1}{2m} \left[\sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2 \right] \\ &= \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) \times 1 \end{aligned}$$

$$j=1 \Rightarrow \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_1} \frac{1}{2m} \left[\sum_{i=1}^m ((\theta_0 + \theta_1 x) - y^{(i)})^2 \right]$$

$$= \frac{1}{m} \sum_{i=1}^m ((\theta_0 + \theta_1 x) - y^{(i)}) x$$

$$\frac{\partial}{\partial \theta_1} [\theta_0 + \theta_1 x] \Rightarrow x =$$

Repeat until Convergence

{

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x)^{(i)} - y^{(i)})$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x)^{(i)} - y^{(i)}) x^{(i)}$$

}

Multiple Linear Regression

Dataset

Weight	Height
-	-
-	-

$$h_0(x) = \theta_0 + \theta_1 x \quad \text{I/P or Independent}$$

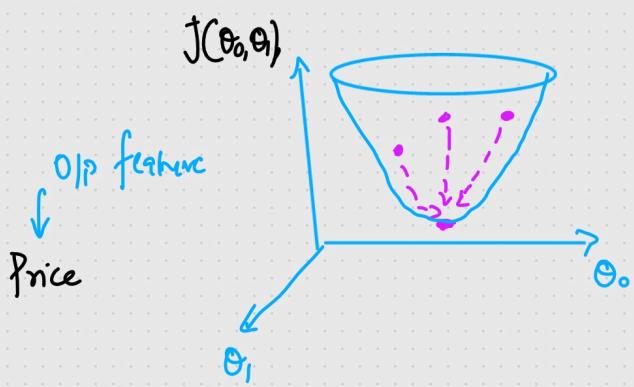
θ_0 = Intercept

θ_1 = Slope

House Pricing Dataset

No. of Rooms	Size of house	Location	Price
--------------	---------------	----------	-------

Independent features



$$h_0(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 \quad \{ \text{Multiple Linear Regression} \}$$

$\theta_1, \theta_2, \theta_3$ = coefficient

θ_0 = Intercept

Performance Metrics Used In Linear Regression

① R squared

③ Adjusted R squared

$$R^2 = 1 - \frac{SS_{Res}}{SS_{Total}}$$

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

$$= 1 - \frac{\text{Small number}}{\text{Big number}}$$

$$= 1 - \text{Small number}$$

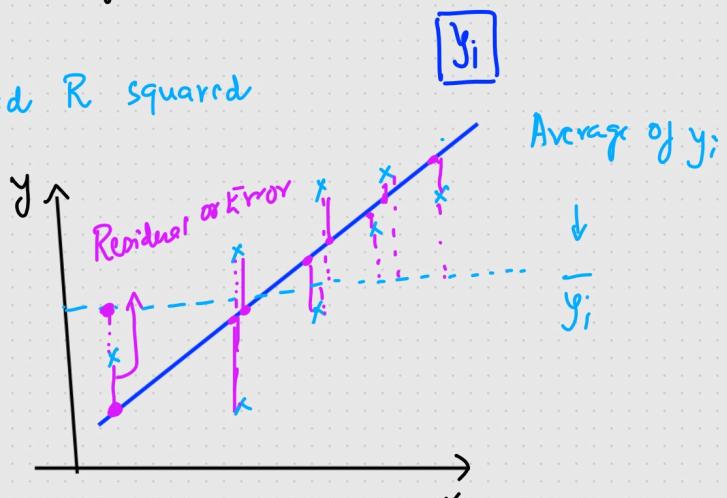
$$\approx 1$$

$$0.70 \Rightarrow 70\%$$

$$0.85 \Rightarrow 85\%$$

$$0.90 \Rightarrow 90\%$$

{ Overfitting, Underfitting }



② Adjusted R squared

Dataset

$\rightarrow (\text{Price})$



Gender

Size of the house No. of bedrooms Location Price



True Correlation

No. of bedrooms ↑ Price ↑

True Correlation

Size of the house ↑ Price ↑

$$R^2 = 75\% \Rightarrow 0.75$$

This is the problem of R squared

$$R^2 \text{ squared} \Rightarrow 80\% \Rightarrow 0.80$$

$$R^2 \text{ squared} \Rightarrow 85\% \Rightarrow 0.85$$

$$R^2 \text{ squared} \Rightarrow 87\% \Rightarrow 0.87$$

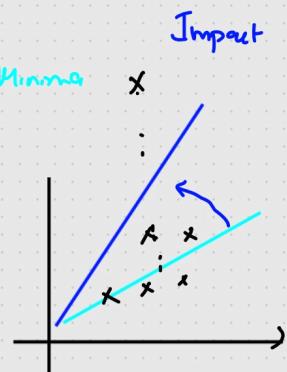
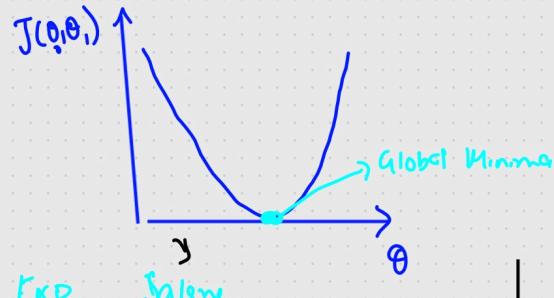
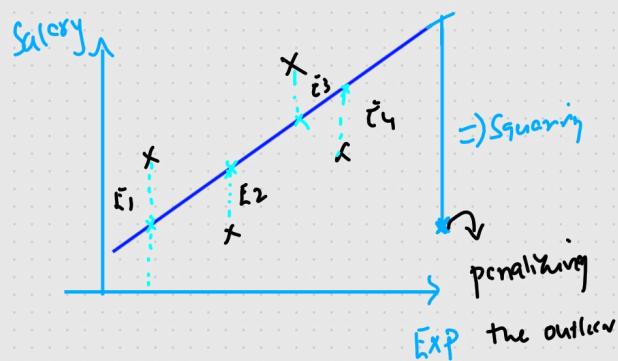
$$\text{Adjusted } R^2 = 1 - \frac{(1-R^2)(N-1)}{N-p-1}$$
$$\left\{ \begin{array}{lll} p=2 & R^2 = 90\% & R^2_{\text{adjusted}} = 86\% \\ p=3 & R^2 = 92\% & R^2_{\text{adjusted}} = 82\% \end{array} \right.$$

N = No. of data points

p = No. of Independent features

MSE, MAE, RMSE [Cost function] → Performance Metrics

R^2 and Adjusted R^2



$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \Rightarrow \text{Cost function } \downarrow \downarrow$$

Quadratic Equation

$aX + bY + c$

$$(a-b)^2 = a^2 - 2ab + b^2$$

Advantage

- ① Differentiable ✓
- ② J has one local and one global Minima

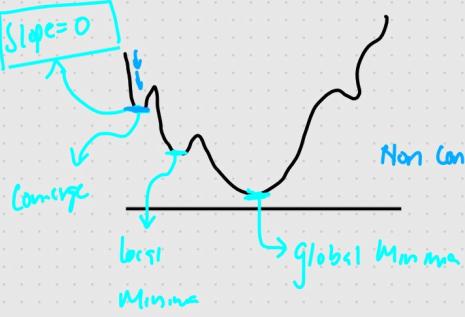
Disadvantage

- ④ Not Robust to outliers
- ④ If is not in same unit.



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

Convex function



$$\begin{matrix} \text{Salary (Lakhs)} \\ X \end{matrix} \quad \begin{matrix} Y \\ \underline{(y_i - \hat{y}_i)^2} \end{matrix} \quad \begin{matrix} (1akh)^2 \\ \text{MSE} \end{matrix}$$

$$\text{Error } c 2.5 \Rightarrow (1akh)^2 \leftarrow$$

② Mean Absolute Error (MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

factors

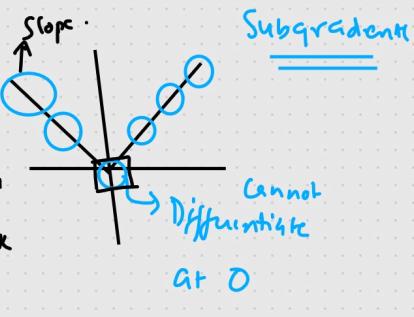
MAE↑↑



Advantage

- ① Robust to outliers ✓
- ② It will be in the same unit
- ③ Convergence usually take more time. Optimization is a complex task
- ④ Time consuming

Disadvantage



Subgradient

③ RMSE {Root Mean Square Error}

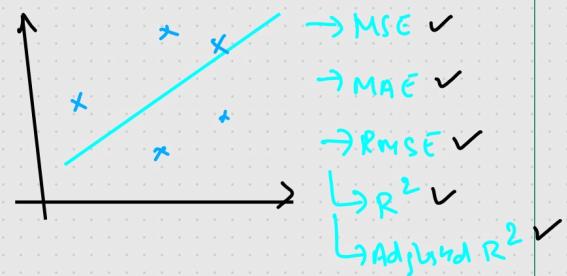
$$\text{RMSE} = \sqrt{\text{MSE}}$$

$$= \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$



MAD, MAE, RMSE

Performance metric ↑



Advantage

- ① Same unit
- ② Differentiable

Disadvantage

- ④ Not Robust to outliers

MSE vs MAE vs RMSE

R² vs Adjusted R²

Overfitting And Underfitting (Bias And Variance)

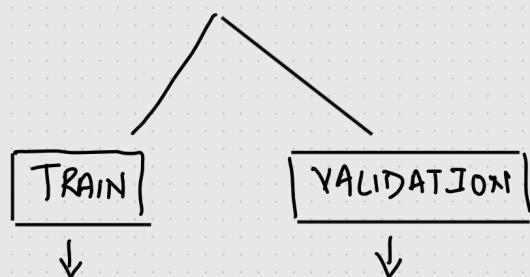
- ① Training dataset
- ② Test dataset
- ③ Validation dataset

70% - 30%



Size of House	No. of bedrooms	Price
-	-	-
-	-	-
-	-	-
-	-	-

TRAINING DATASET



Train the model

Hyperparameters

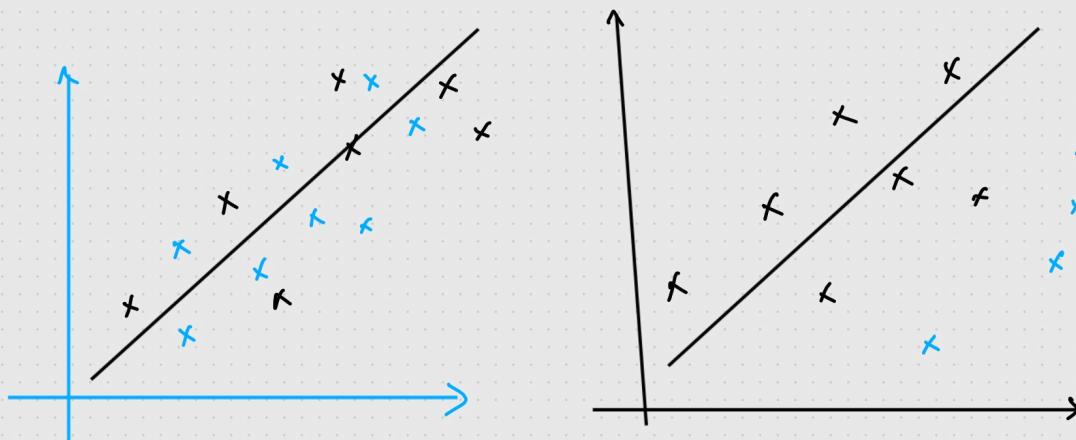
Tunning your
Model

TRAIN	Very Good Accuracy [low Bias]	Very Good Accuracy (90%) [low Bias]
TEST	Very Good Accuracy [low Variance] (85%) ↑	Bad Accuracy (50%) ↓
→ Generalized Model		Model is Overfitting

TRAIN Model Accuracy is low [High Bias]

TEST Model Accuracy is low [High Variance]
↓

Model is Underfitting



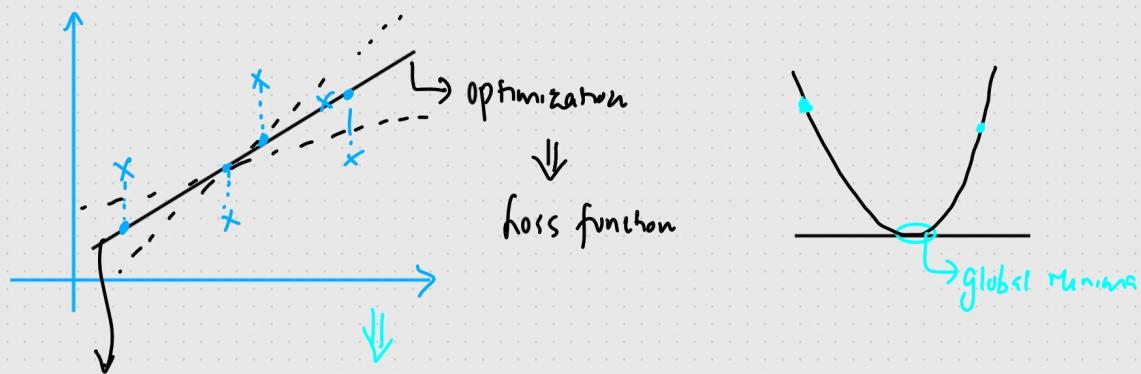
Generalized Model

↓
Low Bias, Low Variance

Overfitting

Low Bias, High Variance

Linear Regression Using OLS {Ordinary Least Square}



$$h(x) = \beta_0 + \beta_1 x, \quad \hat{y} \rightarrow \text{OLS} \rightarrow \text{Formula and Calculate}$$

$$\beta_0 = \quad \& \quad \beta_1 = \quad =$$

Ordinary Least Square

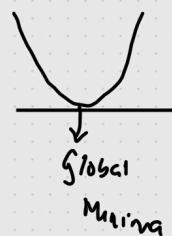
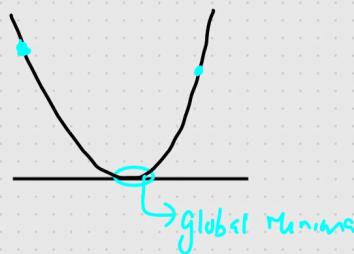
$$S(\beta_0, \beta_1) = \frac{1}{n} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2$$

$$\text{Find } \beta_0, \& \beta_1$$

$$= \quad =$$

$$\begin{aligned} \frac{\partial S}{\partial \beta_0} (\beta_0, \beta_1) &= \frac{2}{n} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i) (0 - 1 - 0) \\ &= -\frac{2}{n} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i) = 0 \longrightarrow ① \end{aligned}$$

$$\begin{aligned} \frac{\partial S}{\partial \beta_1} (\beta_0, \beta_1) &= \frac{2}{n} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i) (-x_i) = 0 \longrightarrow ② \\ &= -\frac{2}{n} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i) (x_i) = 0 \longrightarrow ② \end{aligned}$$



$$y = x^2$$

$$\begin{aligned} \frac{\partial y}{\partial x} &= 2(x)^{2-1} \frac{\partial}{\partial x} (x) \\ &= 2x \end{aligned}$$

Eq → ①

$$-\frac{2}{n} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i) = 0$$

↓

$$-\sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i) = 0$$

$$-\sum_{i=1}^n y_i + n \times \beta_0 + \beta_1 \sum_{i=1}^n x_i = 0$$

$$n\beta_0 + \beta_1 \sum_{i=1}^n x_i = \sum_{i=1}^n y_i$$

$$\beta_0 = \frac{\sum_{i=1}^n y_i}{n} - \beta_1 \frac{\sum_{i=1}^n x_i}{n}$$

Intercept
↓

$$\boxed{\beta_0 = \bar{y} - \beta_1 \bar{x}}$$

Eq 2 ÷

$$-\frac{2}{n} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)(x_i) = 0$$

↓

$$\sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)(x_i) = 0$$

↓

$$\sum_{i=1}^n x_i y_i - \beta_0 \sum_{i=1}^n x_i - \beta_1 \sum_{i=1}^n (x_i)^2 = 0$$

$$\sum_{i=1}^n \left(x_i y_i - \beta_0 x_i - \beta_1 x_i^2 \right) = 0$$

$$\text{Replace } \beta_0 = \bar{y} - \beta_1 \bar{x}$$

$$\sum_{i=1}^n \left(x_i y_i - (\bar{y} - \beta_1 \bar{x}) x_i - \beta_1 x_i^2 \right) = 0$$

$$\sum_{i=1}^n \left(x_i y_i - x_i \bar{y} + \beta_1 \bar{x} x_i - \beta_1 x_i^2 \right) = 0$$

$$\sum_{i=1}^n (y_i - \bar{y} + \beta_1 \bar{x} - \beta_1 x_i) x_i = 0$$

$$\sum_{i=1}^n [(y_i - \bar{y}) + \beta_1 (\bar{x} - x_i)] = 0$$

$$\sum_{i=1}^n (y_i - \bar{y}) + \sum_{i=1}^n \beta_1 (\bar{x} - x_i) = 0$$

$$\begin{aligned} \sum_{i=1}^n \beta_1 (\bar{x} - x_i) &= - \sum_{i=1}^n (y_i - \bar{y}) \\ \beta_1 &= - \frac{\sum_{i=1}^n (y_i - \bar{y})}{\sum_{i=1}^n (\bar{x} - x_i)} \end{aligned}$$

$$\beta_1 = \frac{\sum_{i=1}^n (y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})}$$

Coefficient \Rightarrow

$$\boxed{\beta_1 = \frac{\sum_{i=1}^n (y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})}}$$

$$\boxed{\beta_0 = \bar{y} - \beta_1 \bar{x}}$$

Ohs

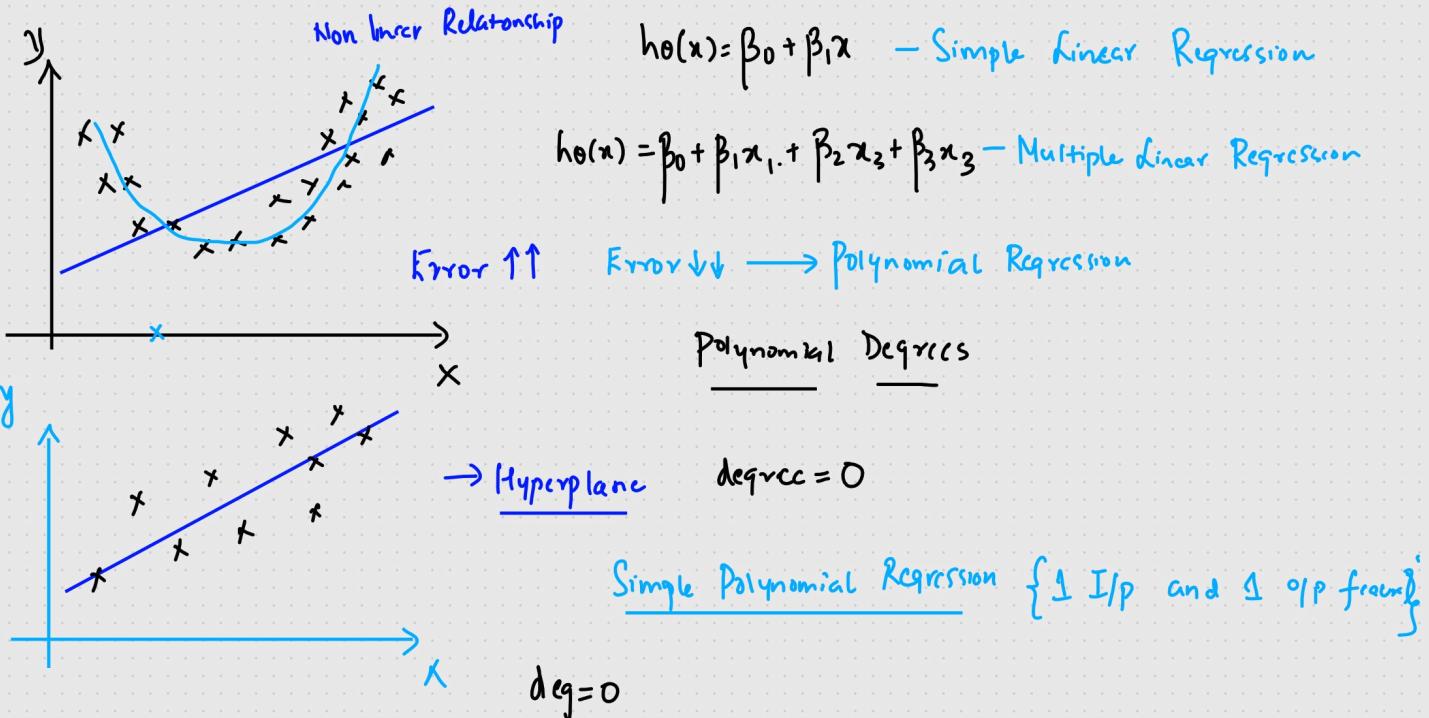
↓ Intercept
↓ Coefficient

$$\begin{array}{ccccccc}
 x & y & (y_i - \bar{y}) & : & (\bar{x}_i - \bar{x}) & \beta_1 & \beta_0 = \bar{y} - \beta_1 \bar{x} \\
 \hline
 - & - & - & & - & & \\
 - & - & - & & - & & \\
 - & - & - & & - & & \\
 - & - & - & & - & & \\
 \hline
 \bar{x} & \bar{y} & & & & &
 \end{array}$$

↓ Coefficient
↓ Intercept

Ohs ≈ linear Regression (sklearn)

Polynomial Regression

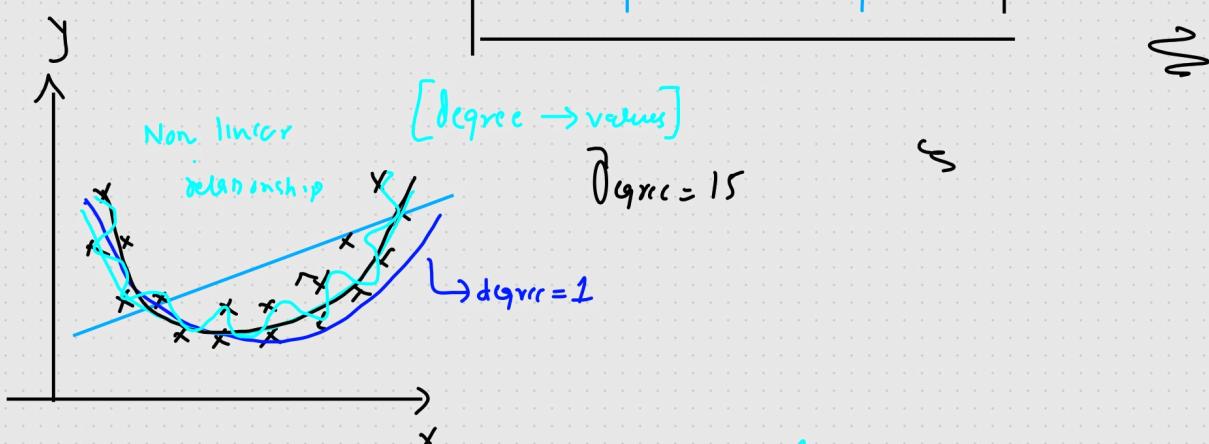


polynomial degree=0 $h_0(x) = \beta_0 \times x^0 \Rightarrow \text{constant value}$

polynomial degree=1
$$h_0(x) = \beta_0 \times x^0 + \beta_1 \times x^{(1)} \rightarrow \text{Simple Linear Regression}$$

polynomial degree=2
$$h_0(x) = \beta_0 \times x^0 + \beta_1 \times x^{(1)} + \beta_2 \times x^{(2)}$$

polynomial degree=n
$$h_0(x) = \beta_0 \times x^0 + \beta_1 \times x^{(1)} + \beta_2 \times x^{(2)} + \beta_3 \times x^{(3)} + \dots + \beta_n \times x^{(n)}$$



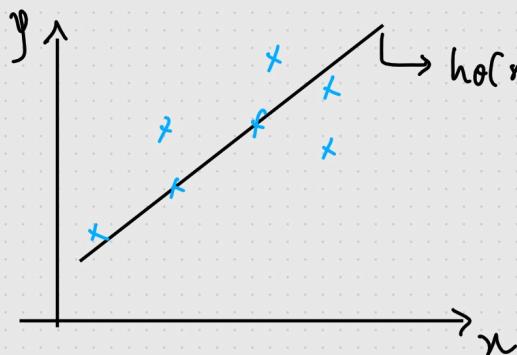
degree=1
$$h_0(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

degree=2
$$h_0(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2$$

{2 independent func}

Ridge Regression, Lasso Regression, Elasticnet Regression

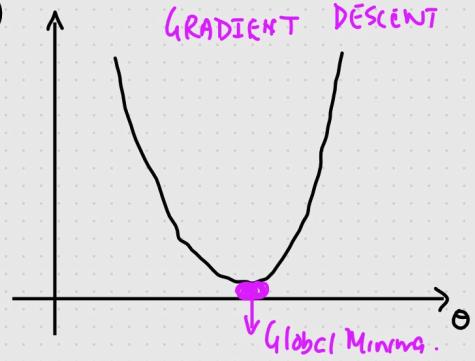
Linear Regression



Independent
↑ features

$$J(\theta)$$

GRADIENT DESCENT

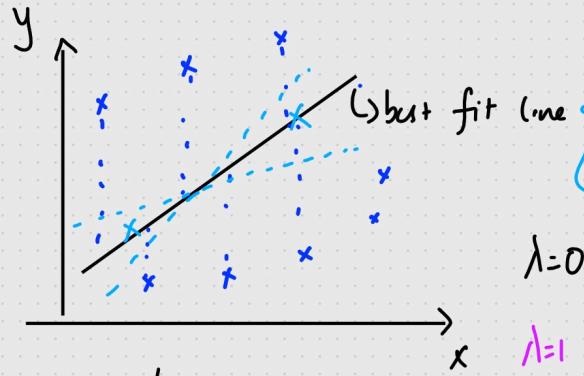


$$\text{Cost fn} = \frac{1}{2m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2$$

Mean Squared Error

① Ridge Regression (L2 Regularization) → Reduce Overfitting

Overfitting



Training data → Acc ↑ → low Bias

Test data → Acc ↓ → High Variance

$\lambda = 30$

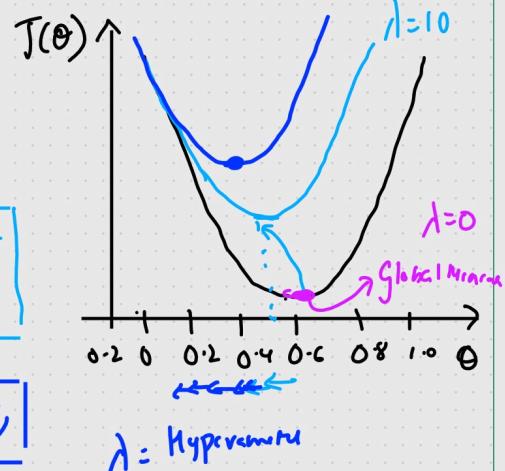
$$h_\theta(x) = \theta_0 + \theta_1 x_1$$

$$\lambda = 0$$

$$\begin{aligned} \text{Cost fn} &= \frac{1}{2m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2 \\ &= 0 + (\lambda) [(\theta_1)^2] \end{aligned}$$

Hyperrparameter

$$\lambda \leq (\text{slope})^2$$



$$\boxed{\lambda = 1}$$

$$> 0$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

$$\overline{\overline{\lambda}}$$

$$= 0.34 + \underline{0.52} x_1 + \underline{0.48} x_2 + \underline{0.24} x_3$$



$$= 0.34 + 0.40x_1 + 0.38x_2 + \boxed{0.14x_3}$$

② Lasso Regression (λ_1 Regularization) \rightarrow Feature Selection

$$\text{Cost fn} : \frac{1}{2m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2 + \boxed{\lambda \sum_{i=1}^n |\text{slope}|}$$

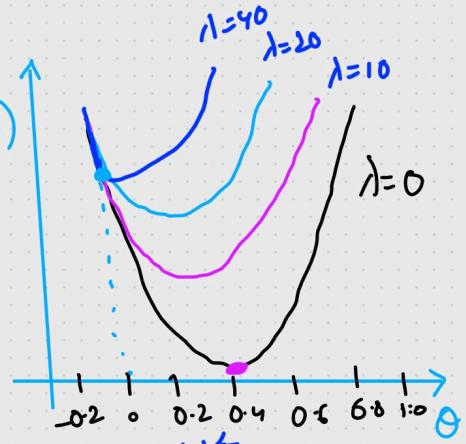
$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$$

$$h_\theta(x) = 0.52 + 0.65x_1 + 0.72x_2 + 0.34x_3 + \boxed{0.12x_4}$$

\Downarrow
Lasso Regression

\Downarrow

$$= 0.52 + 0.51x_1 + 0.60x_2 + 0.14x_3 + \boxed{0 \times x_4}$$



③ ElasticNet Regression \rightarrow ① Reduce Overfitting

\rightarrow ② Feature Selection

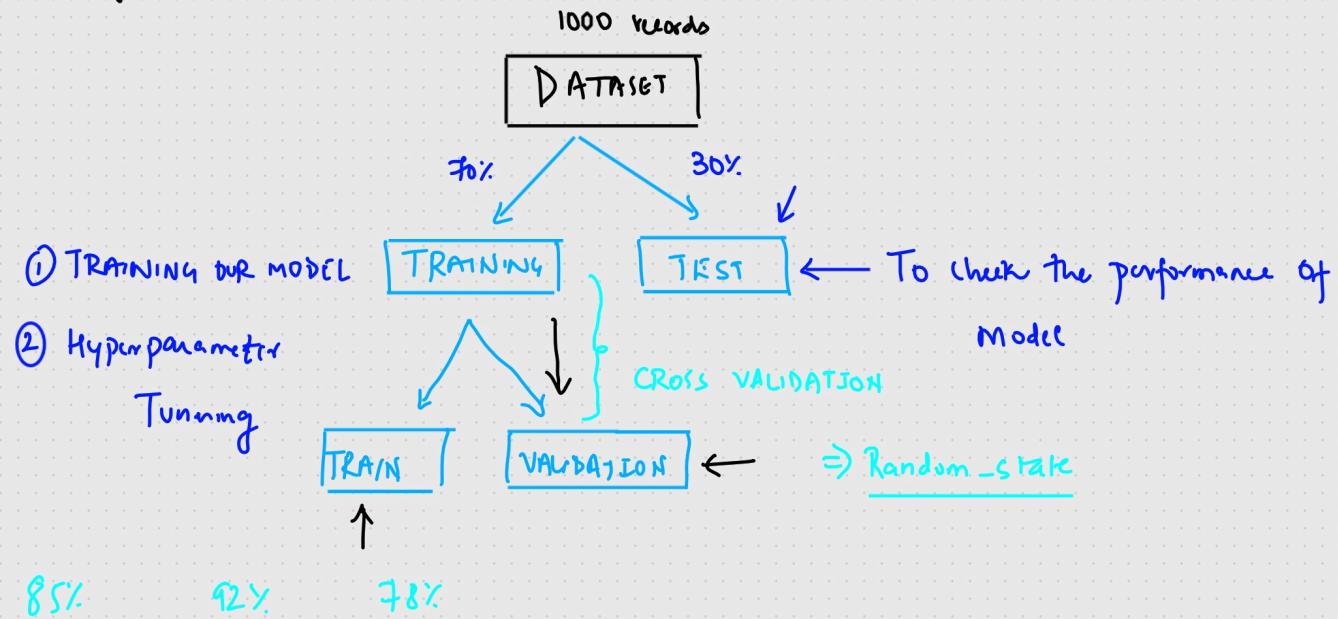
$$\text{Cost fn} = \frac{1}{2m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2 + \boxed{\lambda_1 \sum_{i=1}^m (\text{slope})^2} + \boxed{\lambda_2 \sum_{i=1}^m |\text{slope}|}$$

\Downarrow
Reduce
Overfitting

\Downarrow
Feature
Selection

{ Hyperparameter Tuning the }
Linear Regression }

Types of Cross Validation

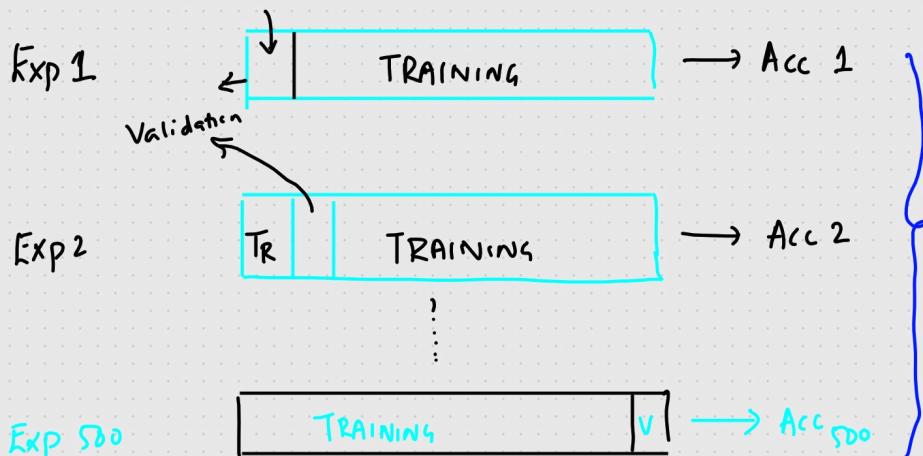


① leave One Out CV (k=0cv)

② leave p out CV

$p=10 \quad p=20 \quad p=30$

TRAINING → 500 Records ↑ Complexity of Training Model



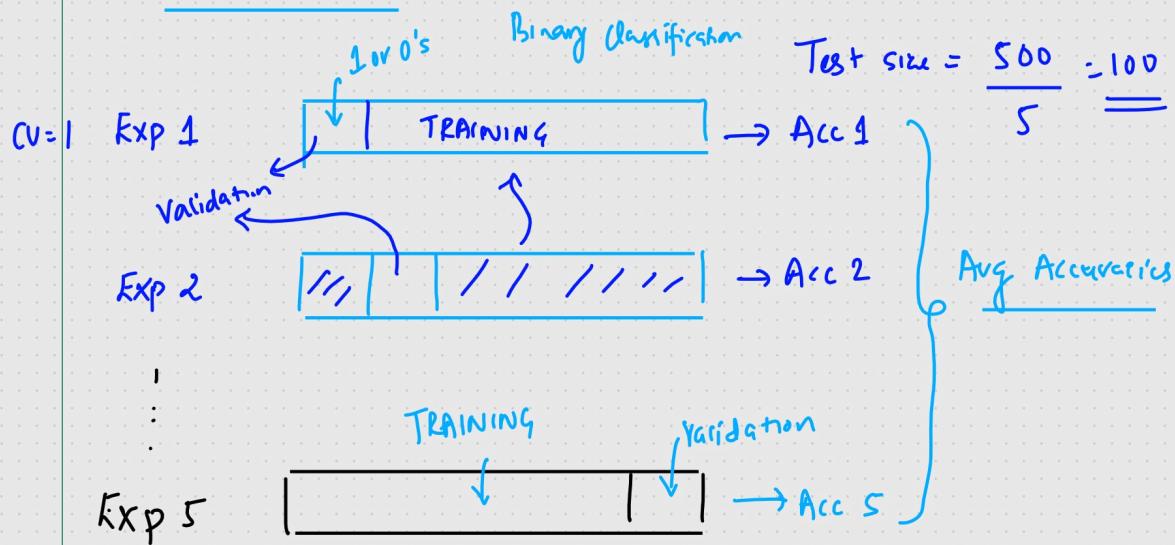
① Overfitting → TRAINING \uparrow Acc → New Test → Acc \downarrow
 Validation Acc \downarrow

Data:

③ K Fold CV

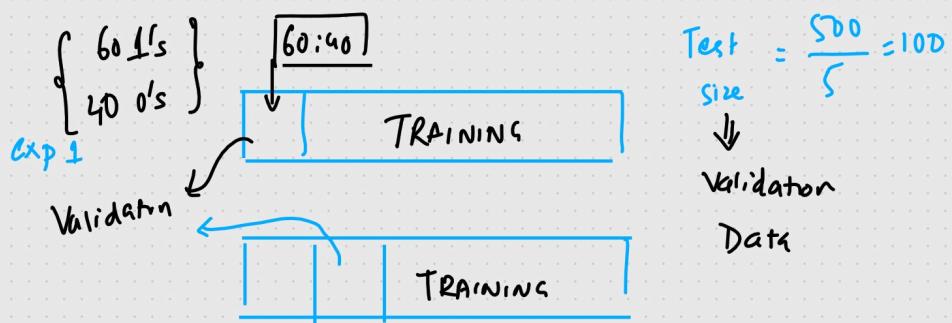
$K=5$

$n=500$



④ Stratified K Fold CV

$K=5$



⑤ Time Series CV

Reviews
Product Sentiment Analysis

Time

JAN → DEC

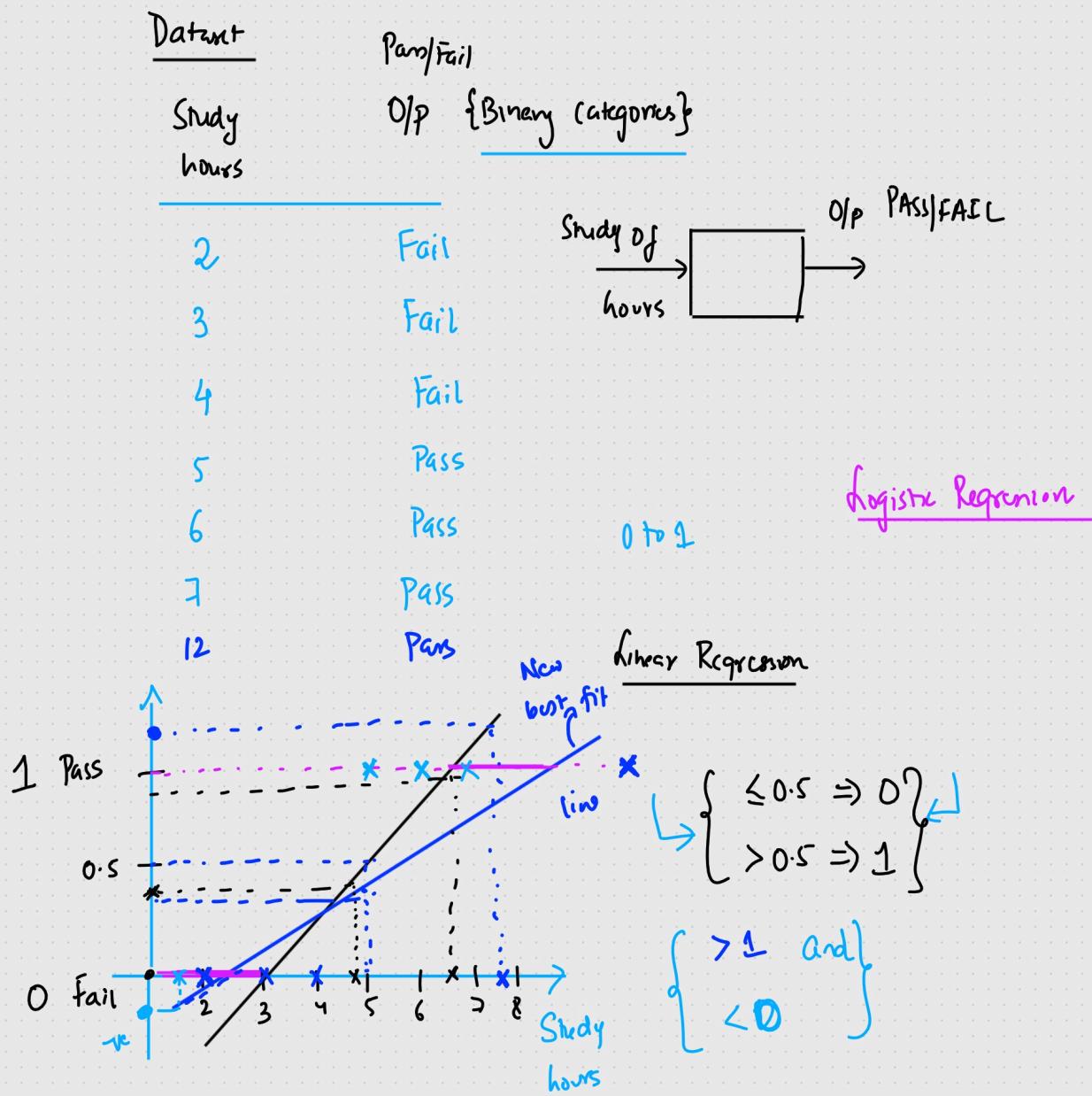
TRAINING

Validation

DAY 1 DAY 2 DAY 3 DAY 4 · - - - DAY N

Time Series Application

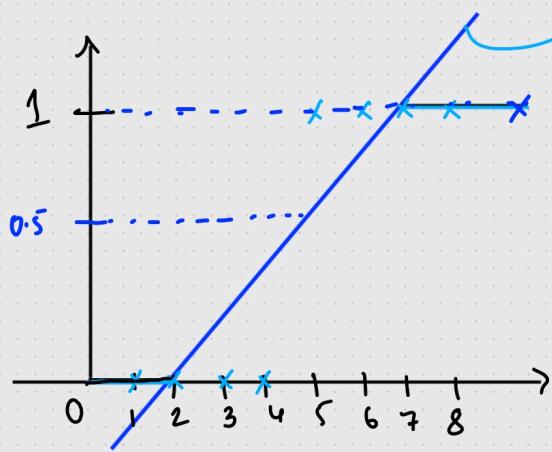
Logistic Regression (Binary classification) ←



Why we cannot use Linear Regression for Classification?

- ① Outlier {Best fit line change}
- ② > 1 and < 0 {Squash line}

How Logistic Regression Solves Classification Problem

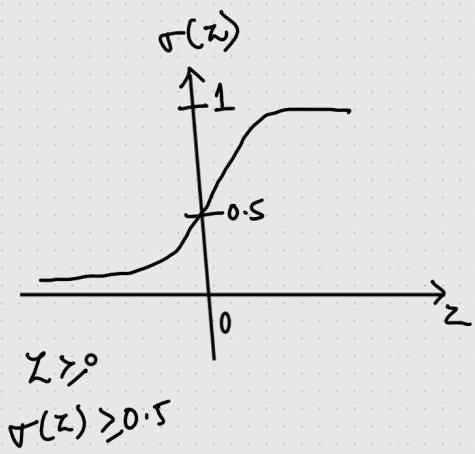


$$h_{\theta}(x) = \theta_0 + \theta_1 x_1$$

But fit line

Sigmoid Activation

$$\frac{1}{1+e^{-z}}$$



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

\uparrow
 $z = \theta_0 + \theta_1 x_1$

$$f = \frac{1}{1+e^{-z}}$$

$$= \sigma(z)$$

Logistic Regression
hypothesis

$$h_{\theta}(x) = \frac{1}{1+e^{-z}}$$

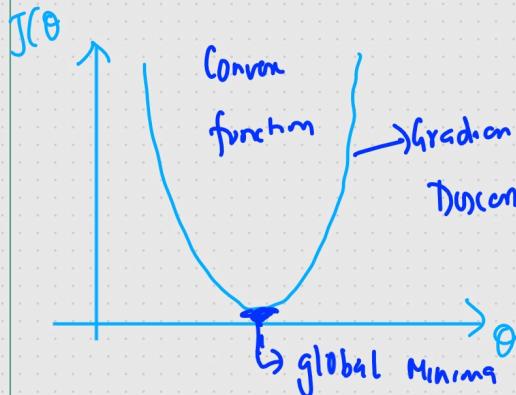
$$z = \theta_0 + \theta_1 x_1$$

Linear Regression Cost function

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

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

\downarrow
Convex function



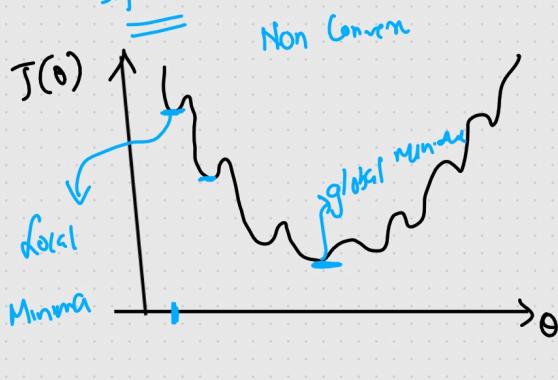
Logistic Regression Cost function

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

$$h_{\theta}(x) = \frac{1}{1+e^{-z}}$$

\uparrow
Sigmoid

Non Convex
function



$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m \underbrace{(h_\theta(x)^{(i)} - y^{(i)})^2}_{\text{fisB Denote } \text{Cost}(h_\theta(x)^{(i)}, y^{(i)})}$$

$$h_\theta(x)^{(i)} = \frac{1}{1+c^{-2}} z = \theta_0 + \theta_1 x_i$$

\Downarrow

{log loss}

$$\text{Cost}(h_\theta(x)^{(i)}, y^{(i)}) = \begin{cases} -\log(h_\theta(x)) & \text{if } y=1 \\ -\log(1-h_\theta(x)) & \text{if } y=0 \end{cases}$$

\Downarrow convex function

$$\text{Cost}(h_\theta(x)^{(i)}, y^{(i)}) = -y \log(h_\theta(x)) - (1-y) \log(1-h_\theta(x))$$

$$J(\theta_0, \theta_1) = -\frac{1}{2m} \sum_{i=1}^m \left(y^{(i)} \log(h_\theta(x)^{(i)}) - (1-y^{(i)}) \log(1-h_\theta(x)^{(i)}) \right)$$

Mimimize Cost function $J(\theta_0, \theta_1)$ by changing

θ_0 & θ_1

Convergence Algorithm

Repeat

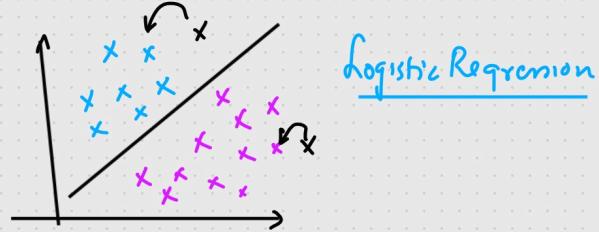
$$\left\{ \begin{array}{l} j=0 \text{ and } 1 \\ \theta_j := \theta_j - \alpha \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_j} \end{array} \right.$$

$$\left\{ \begin{array}{l} j=0 \text{ and } 1 \\ \theta_j := \theta_j - \alpha \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_j} \end{array} \right.$$

Performance Metrics, Accuracy, Precision, Recall And F-Beta

Topics to be covered

- ① Confusion matrix
- ② Accuracy
- ③ Precision
- ④ Recall
- ⑤ F-Beta Score



R squared
Adjusted R squared

Dataset		O/P		pred by model
x_1	x_2	y	\hat{y}	
-	-	0	1	
-	-	1	1	
-	-	0	0	
-	-	1	1	
-	-	0	1	
-	-	1	0	

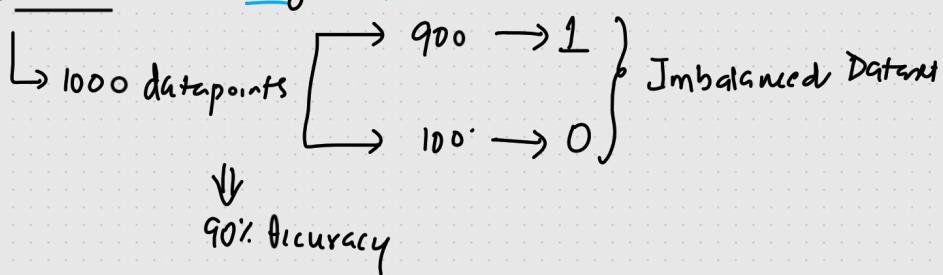
① Confusion Matrix

		Actual values	
		1	0
1	3	2	
0	1	1	

Predicted values		Actual	
1	0	1	0
1	TP	FP	
0	FN	TN	

$$\begin{aligned}
 \text{Accuracy} &= \frac{TP+TN}{TP+FP+FN+TN} \\
 &= \frac{3+1}{3+2+1+1} \\
 &= \frac{4}{7}.
 \end{aligned}$$

② Dataset Binary classification



$$\textcircled{1} \quad \text{Precision} = \frac{TP}{TP+FP}$$

Out of all the actual value
how many are correctly predicted

	1	0	Actual
1	TP	FP	
0	FN	TN	

Predicted

$$\textcircled{2} \quad \text{Recall} = \frac{TP}{TP+FN}$$

Out of all the predicted value
how many are correctly predicted

Usecase 1

Spam classification

		Spam	Not Spam	
		1	0	Actual
Spam	1	TP	FP	
	0	FN	TN	
Not Spam				

Mail → Spam }
Model → Spam } Good

Mail → Not Spam }
Model → Spam } Blunder

$$\text{Precision} = \frac{TP}{TP+FP}$$

Usecase 2

To predict whether person has diabetes or not

✓ Truth → diabetes
✓ Model → Doesn't diabetes } Blunder

Diab No Diab

TP	FP
FN	TN

$$\text{Recall} = \frac{TP}{TP+FN}$$

Truth → diabetes
Model → " } good

usecase of disease

Truth \rightarrow Not diabetes }
 Model \rightarrow Diabetes } \Rightarrow 2nd opinion
 ↓
 check

Assignment

④ Tomorrow the stock market will crash or not

Reducing $FP \downarrow$ or $FN \downarrow$

$$\text{F-Beta Score} := (1 + \beta^2) \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

① If FP & FN are both important

$$\beta = 1$$

$$F1 \text{ Score} = 2 \times \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

\Rightarrow Harmonic Mean

② If FP is more important than FN

$$\beta = 0.5$$

$$F_{0.5} \text{ Score} = (1 + 0.25) \frac{P * R}{P + R}$$

③ If $FN > FP$

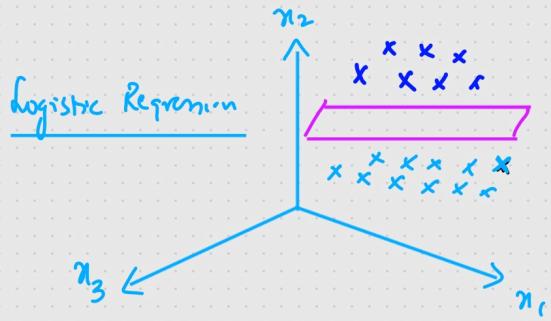
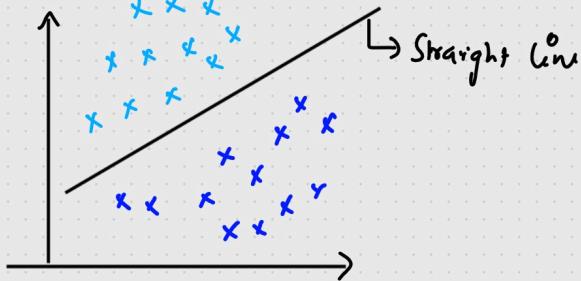
$$\beta = 2$$

$$F_2 \text{ Score} = (1 + 4) \frac{P * R}{P + R}$$

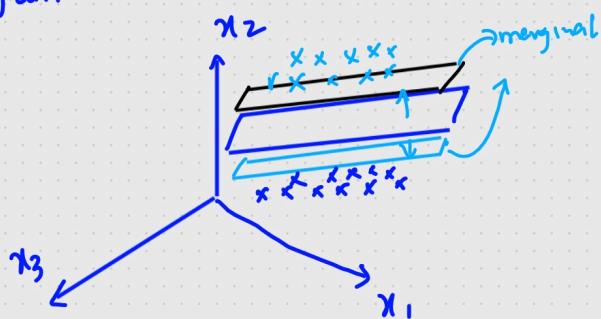
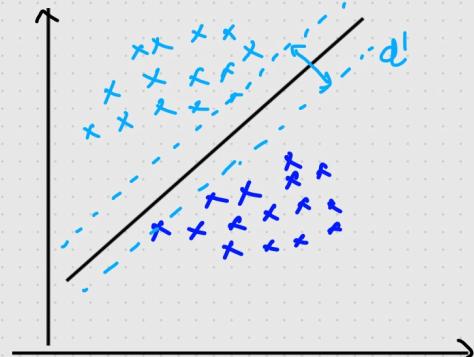
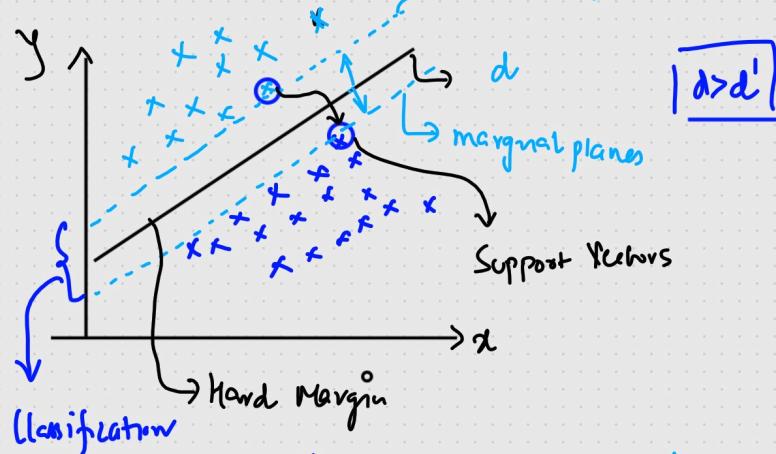
Support Vector Machines ML Algorithms.

① SVC (Support Vector Classifier)

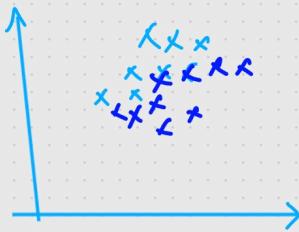
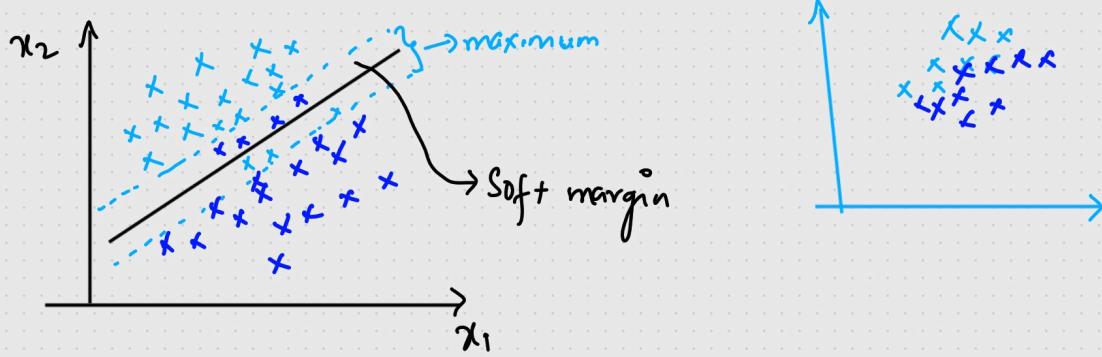
② SVR (Support Vector Regressor)



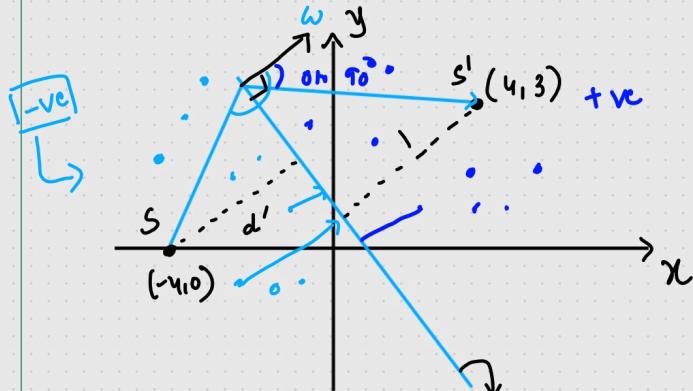
① Support Vector Machine (SVC)



Soft Margin And Hard Margin In SVM



① Support Vector Machines (SVC) Maths Intuition



$$ax + by + c = 0$$

$$\downarrow$$

$$w_1x_1 + w_2x_2 + b = 0$$

$$\boxed{w^T x + b = 0}$$

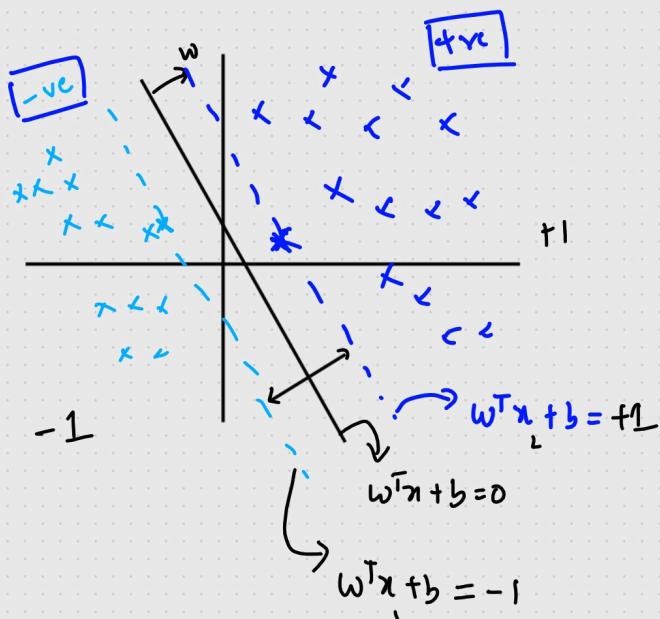
$$\downarrow$$

$$b=0$$

$$\boxed{w^T x = 0}$$

$d = -ve$ below plane

$d = +ve$ above plane



$$w^T x_1 + b = 1$$

$$w^T x_2 + b = -1$$

(-) (+)

$$\frac{w^T(x_1 - x_2)}{\|w\|} = \frac{+2}{\|w\|}$$

Unit vector {Magnitude of the vector is 1}

Cost function

Maximize $\frac{2}{\|w\|} \Rightarrow$ Distance between Marginal plane
 w, b ↳ correctly classified point

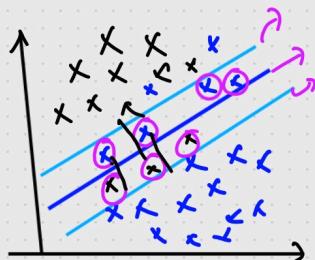
Constraint such that $y_i \begin{cases} +1 & w^T x + b \geq 1 \\ -1 & w^T x + b \leq -1 \end{cases}$

For all correct points

Constraint $\rightarrow \boxed{y_i \cdot (w^T x + b) \geq 1}$

Maximize $\frac{2}{\|w\|} \Rightarrow \boxed{\min_{(w,b)} \frac{\|w\|}{2}}$

$C_i=6$ ✓



Cost function of SVM (SVC)

$\min_{w,b} \frac{\|w\|}{2} + \boxed{\sum_{i=1}^n \max\{0, 1 - y_i(w^T x_i + b)\}}$ Hinge loss

↓ Summation of the

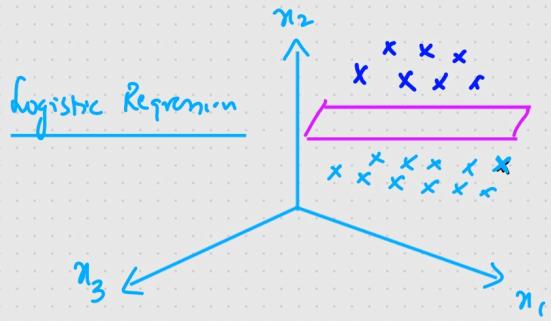
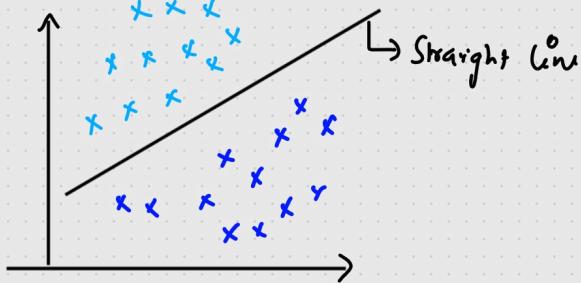
{ How many distance of the
 points we want incorrect data points
 to avoid misclassification from the marginal
 } plane }

Soft Margin

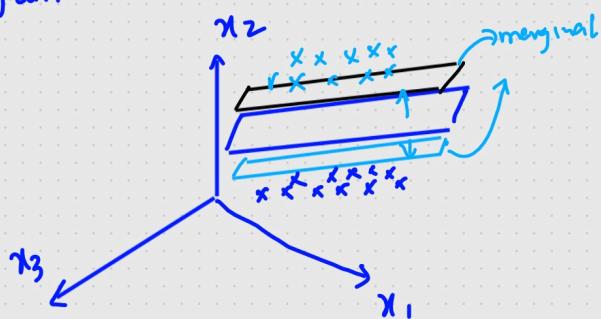
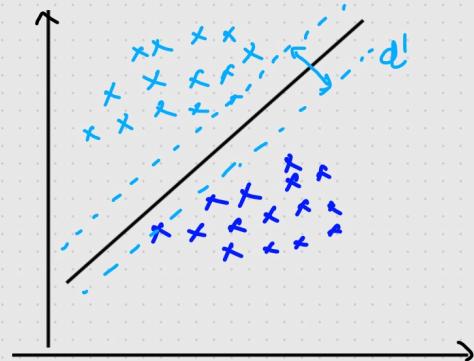
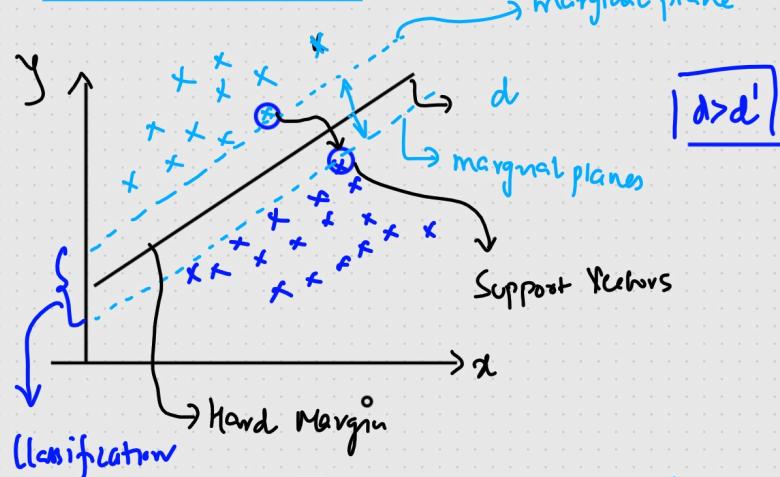
Support Vector Machines ML Algorithms.

① SVC (Support Vector Classifier)

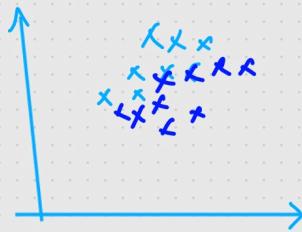
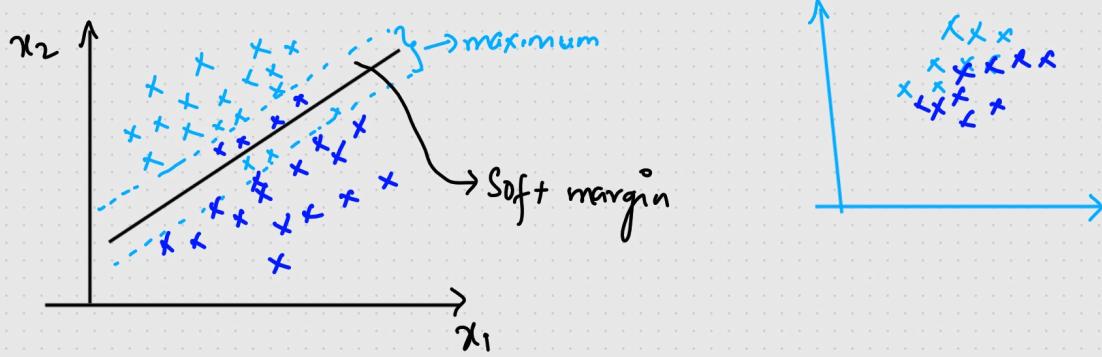
② SVR (Support Vector Regressor)



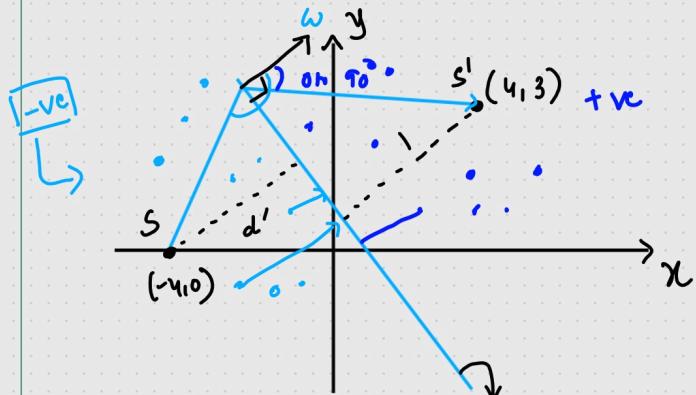
① Support Vector Machine (SVC)



Soft Margin And Hard Margin In SVM



① Support Vector Machines (SVC) Maths Intuition



$d = -ve$ below plane

$d = +ve$ above plane

$$ax + by + c = 0$$

\Downarrow

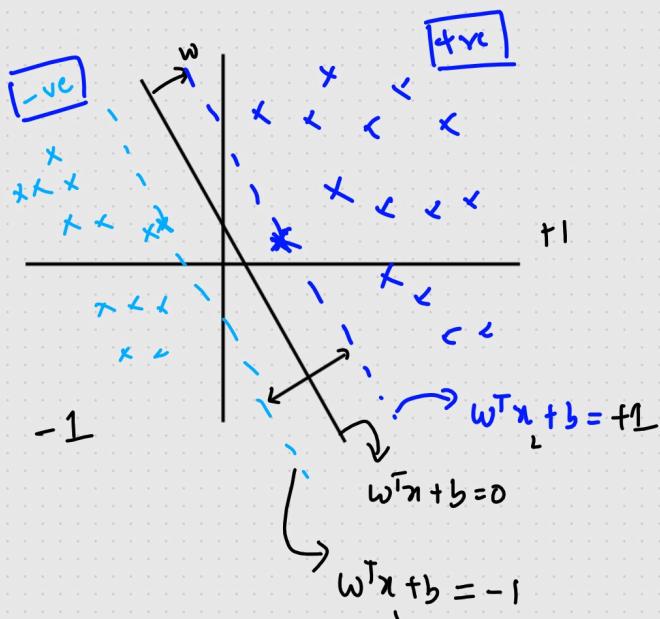
$$w_1x_1 + w_2x_2 + b = 0$$

$$\boxed{w^T x + b = 0}$$

\Downarrow

$$b = 0$$

$$\boxed{w^T x = 0}$$



$$w^T x_1 + b = 1$$

$$w^T x_2 + b = -1$$

$$\frac{w^T(x_1 - x_2)}{\|w\|} = \frac{+2}{\|w\|}$$

Unit vector {Magnitude of the vector is 1}

Cost function

Maximize $\frac{2}{\|w\|} \Rightarrow$ Distance between Marginal plane
 w, b ↳ correctly classified point

Constraint such that

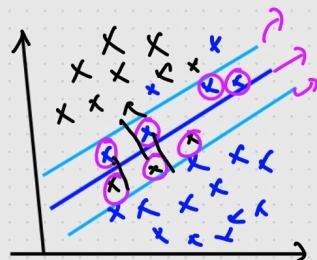
$$y_i \begin{cases} +1 & w^T x + b \geq 1 \\ -1 & w^T x + b \leq -1 \end{cases}$$

For all correct points ↑ predicted points

Constraint $\rightarrow \boxed{y_i \cdot (w^T x + b) \geq 1}$

Maximize $\frac{2}{\|w\|} \Rightarrow \boxed{\min_{(w,b)} \frac{\|w\|}{2}}$

$C_i=6$ ✓



Cost function of SVM (SVC)

$\min_{w,b} \frac{\|w\|}{2} + \boxed{\sum_{i=1}^n \max\{0, 1 - y_i(w^T x_i + b)\}}$ Hinge loss

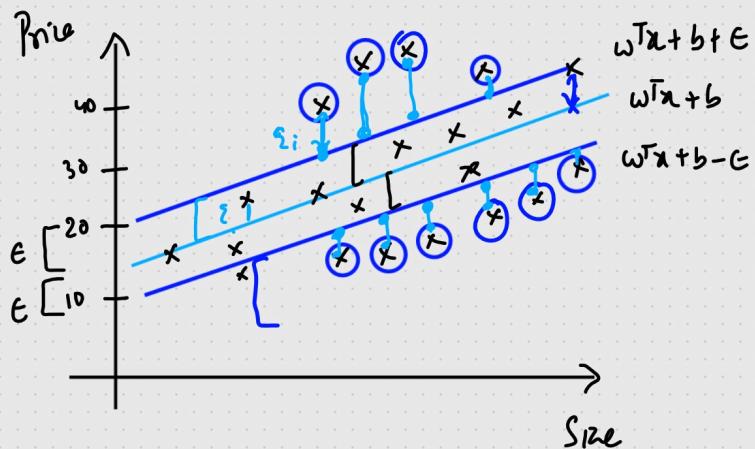
↓ summation of the

{ How many distance of the
 points we want incorrect data points
 to avoid misclassification from the marginal
 } plane }

Soft Margin

Support Vector Regression

ϵ : Marginal Error



Cost function

$$\text{Min}_{w,b} \frac{\|w\|}{2} + C \sum_{i=1}^n \xi_i$$

Hyperparameter

Margin Loss

Constraint =

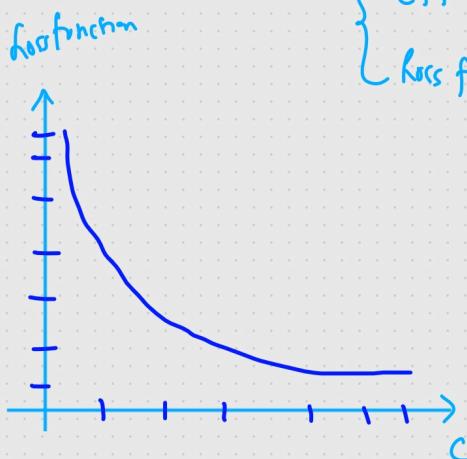
$$|y_i - w_i x_i| \leq \epsilon + \xi_i$$

\Downarrow
Loss function

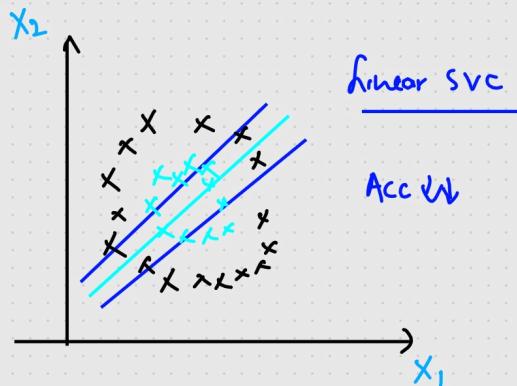
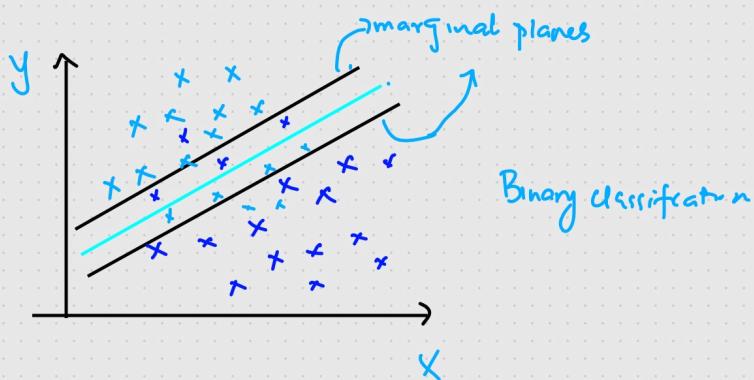
Relationship
 { C ↑
 Loss function } ↓

ϵ → margin error

ξ_i → Error above the margin



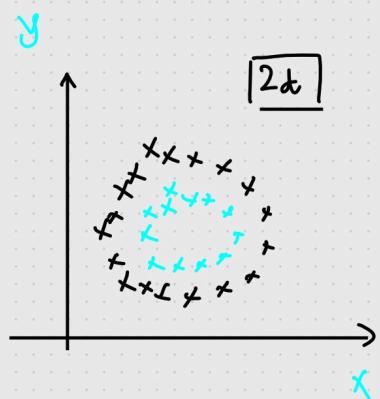
SVM KERNELS



linear SVC



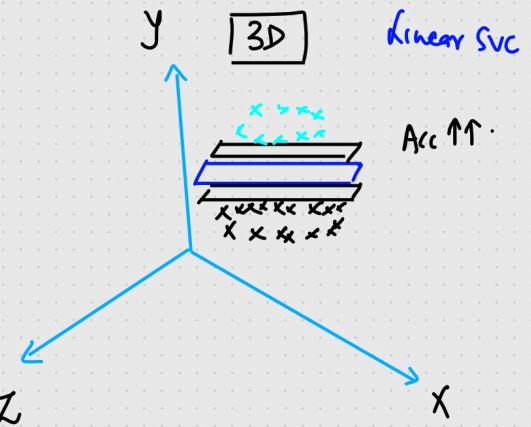
SVM Kernels



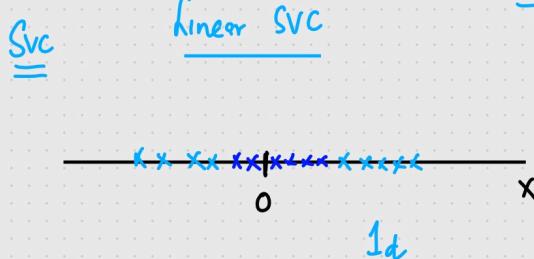
⇒ Transformations



Mathematical
formula



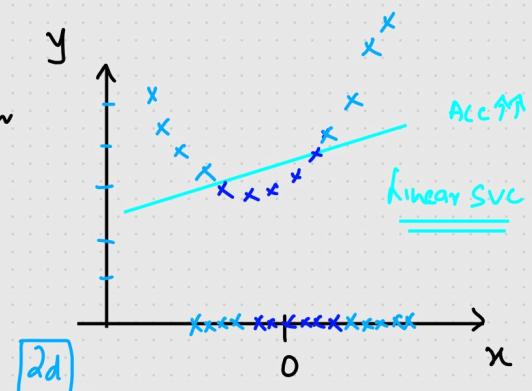
Dataset : 1d



SVM Kernel

⇒ Transformation

$$y = n^2$$



① Polynomial Kernel

② RBF Kernel

③ Sigmoid Kernel

Naive Baye's Algorithm (Classification)

- ① Probability
- ② Baye's Theorem

Independent Events

Rolling a Dice $\{1, 2, 3, 4, 5, 6\}$

$$Pr(1) = \frac{1}{6} \quad Pr(2) = \frac{1}{6} \quad Pr(3) = \frac{1}{6}.$$

Dependent Events

① What is the probability of removing a orange marble and then a yellow marble

0	0
0	0

$$\hookrightarrow P(o) = \frac{3}{5} \rightarrow 1^{\text{st}} \text{ Event} \rightarrow \text{Orange marble has been removed}$$

0	0
0	0

$$\rightarrow P(y/o) = \frac{2}{4} = \frac{1}{2} \Rightarrow 2^{\text{nd}} \text{ Event} \Rightarrow \text{Removed the Yellow Marble}$$

$$P(o \text{ and } y) = P(o) * \boxed{P(y/o)} \rightarrow \text{Conditional Probability}$$

$$= \frac{3}{5} * \frac{1}{2} = \boxed{\frac{3}{10}}$$

$$\boxed{P(A \text{ and } B) = P(A) * P(B/A)}$$

Bayes Theorem

$$P(A \text{ and } B) = P(B \text{ and } A)$$

$$P(A) * P(B/A) = P(B) * P(A/B)$$

$$\boxed{P(A/B) = \frac{P(A) * P(B/A)}{P(B)}} \Rightarrow \text{Bayes Theorem.}$$

$P(A|B)$ = Probability of Event A given B has occurred

$P(A)$ = Probability of Event A

$P(B)$ = Probability of Event B

$P(B/A)$ = Probability of Event B given A has occurred.

$$\boxed{P(A/B) = \frac{P(A) * P(B/A)}{P(B)}} \Rightarrow \text{Bayes Theorem.}$$

J/P features			↓ Dependent
x_1	x_2	x_3	y
-	-	-	Yes
-	-	-	No
-	-	-	Yes
-	-	-	No

$$P(y/(x_1, x_2, x_3)) = \frac{P(y) * P(x_1, x_2, x_3)/y}{P(x_1, x_2, x_3)}$$

$$P(y/(x_1, x_2, x_3)) = \frac{P(y) * P(x_1, x_2, x_3)/y}{P(x_1, x_2, x_3)}$$

$$= \frac{P(y) * P(x_1/y) * P(x_2/y) * P(x_3/y)}{P(x_1) * P(x_2) * P(x_3)}$$

I/P features			↓ Dependent
x_1	x_2	x_3	y
-	-	-	Yes
-	-	-	No
-	-	-	Yes
-	-	-	No

New test data

$$Pr(y_{us}/(x_1, x_2, x_3)) = \frac{P(y_{us}) * P(x_1/y_{us}) * P(x_2/y_{us}) * P(x_3/y_{us})}{P(x_1) * P(x_2) * P(x_3)} \Rightarrow \text{constant}$$

$= 0.60$

$$Pr(No/(x_1, x_2, x_3)) = \frac{P(No) * Pr(x_1/No) * Pr(x_2/No) * Pr(x_3/No)}{P(x_1) * P(x_2) * P(x_3)} = 0.40$$

$\Rightarrow \text{constant}$

Let's Solve this Problem

Outlook

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Sunny

Yes

No

$P(E|Y_{13})$

$P(E|N_{13})$

2

3

2/9

3/5

Overcast

4

0

4/9

6/5

Rain

3

2

3/9

2/5

Temperature

$\rightarrow \text{Test } (\text{Sunny}, \text{Hot}) \rightarrow O/P$

PLAY (Y_{13})

	Yes	No	$P(E Y_{13})$	$P(E N_{13})$		$P(Y_{13})$	$P(N_{13})$
Hot	2	2	2/9	2/5	Yes	9	9/14
Mild	4	2	4/9	4/5	No	5	5/14
Cool	3	1	3/9	3/5			

$$P(Y_{13} | (\text{Sunny}, \text{Hot})) = \frac{P(Y_{13}) * P(\text{Sunny}|Y_{13}) * P(\text{Hot}|Y_{13})}{P(\text{Sunny}) + P(\text{Hot})}$$

$$= \frac{1}{14} * \frac{2}{9} * \frac{2}{5}$$

$$= \frac{2}{63} = \underline{\underline{0.031}}$$

$$P(N_{13} | (\text{Sunny}, \text{Hot})) = P(N_{13}) * P(\text{Sunny}|N_{13}) * P(\text{Hot}|N_{13})$$

$$= \frac{8}{14} * \frac{3}{5} * \frac{2}{5}$$

$$= \frac{3}{35} = \underline{\underline{0.085}}$$

$$Pr(Y_{13} | (\text{Sunny}, \text{Hot})) = \frac{0.031}{(0.031) + (0.085)} = 0.27 = 27\%$$

$$\Pr(No \mid \text{Gunny, hot}) = \frac{0.085}{(0.031 + 0.085)} = 0.73 = 73\%$$

Now
 Test \Rightarrow Outlook Temperature O/P
 Sunny Hot $73\% \Rightarrow$ They will not play
 Tennis

$27\% \Rightarrow$ They will play
 Tennis.



$0 \Rightarrow$ Person is ^{going to} not playing
 Tennis

Variants of Naive Bayes

- ① Bernoulli Naive Bayes
- ② Multinomial Naive Bayes
- ③ Gaussian Naive Bayes

① Bernoulli Naive Bayes

Whenever your features are following a Bernoulli Distribution, then we need to use Bernoulli Naive Bayes Algorithm.

Dataset

Bernoulli $\rightarrow 0,1$

f1	f2	f3	O/P
Yes	Pass	Male	Yes
Yes	Fail	Female	No
No	Pass	Male	Yes
Yes	Fail	Female	No

② Multinomial Naive Bayes $\Rightarrow I/p = \text{Text}$

Dataset : Spam Classification

O/P

I/P \rightarrow Email Body O/P
feature Spam/Not Spam

You have Million \$ lottery Spam

KRISH YOU HAVE DONE
GOOD JOB MAM

↓

Numerical Values \Rightarrow Natural Language Processing

↓
vectors

- ① BOW
- ② Tf-IDf
- ③ Word2Vec

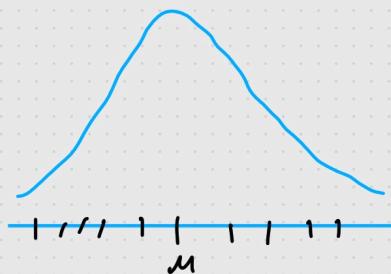
③ Gaussian Naive Bayes

If the features are following Gaussian Distribution, then we use

Gaussian Naive Bayes

DATASET \rightarrow CONTINUOUS

[IRIS Dataset]



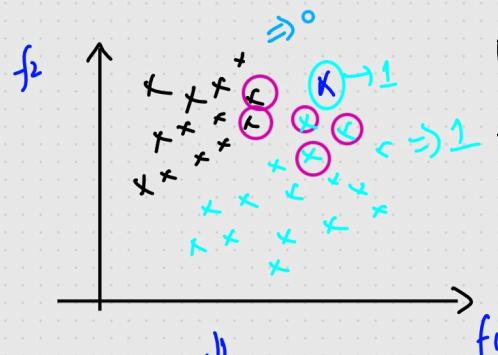
Age	Height	Weight	Yes/No
25	170	78	
38	160	75	
22	150	60	
29	170	35-	

K Nearest Neighbour (KNN)

① Classification

② Regression

① Classification



$K=5$

0's → 2

1's → 3

f_1 f_2

— —

0

1

y [Binary Category]

① We have to initialize the K value

$K=5$

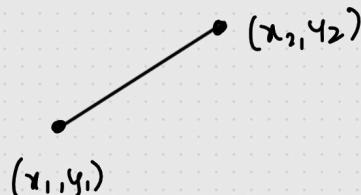
$K > 0 \dots \infty$

$K=1, 2, 3, 4, 5, \dots \Rightarrow$ Hyperparameter

② Find the K Nearest Neighbour for
The Test Data.

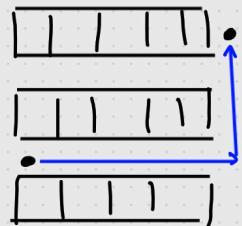
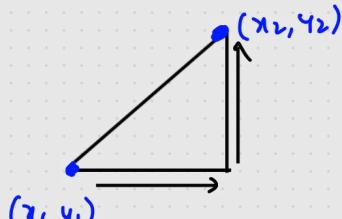
③ From those $K=5$ how many
neighbour belong to 0 category
and 1 category

① Euclidean Distance

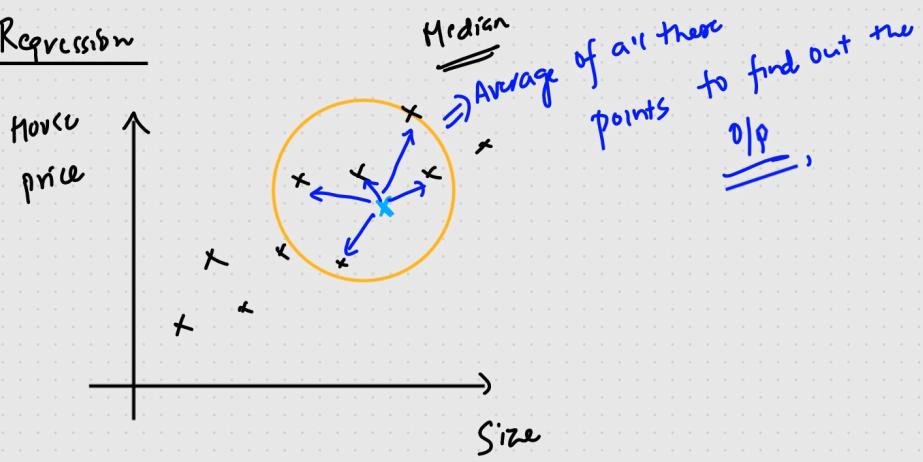


$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

② Manhattan Distance



② Regression



Time complexity

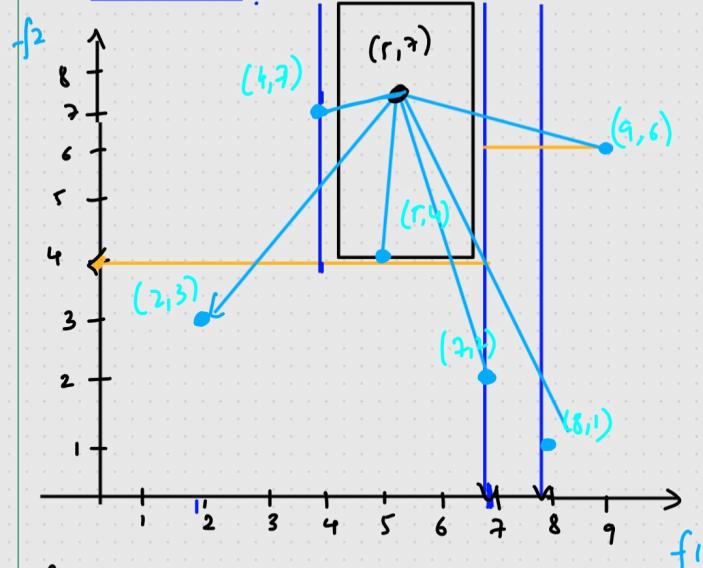
$O(n)$ \Rightarrow Million of data point

{ ① KD Tree ② Ball Tree } \Rightarrow Optimize

Binary Tree

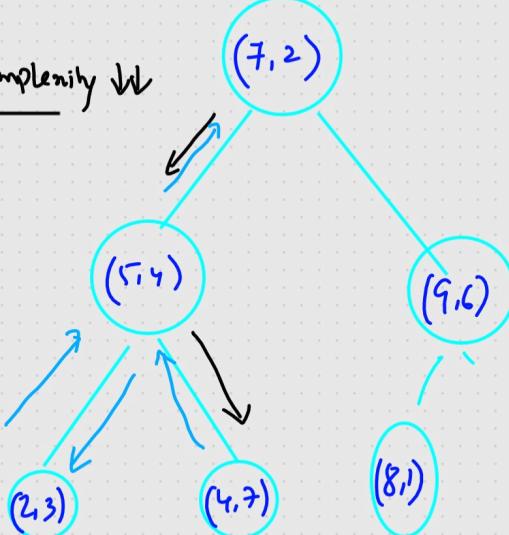
Variants of KNN

KD Tree.



Binary Tree [KD Tree]

Time Complexity $\downarrow\downarrow$



$$2, 4, \boxed{5, 7} 8, 9$$

$$\frac{5+7}{2} = \frac{12}{2} = 6.5$$

f_2

$$1, 2, \boxed{3, 4}, 6, 7$$

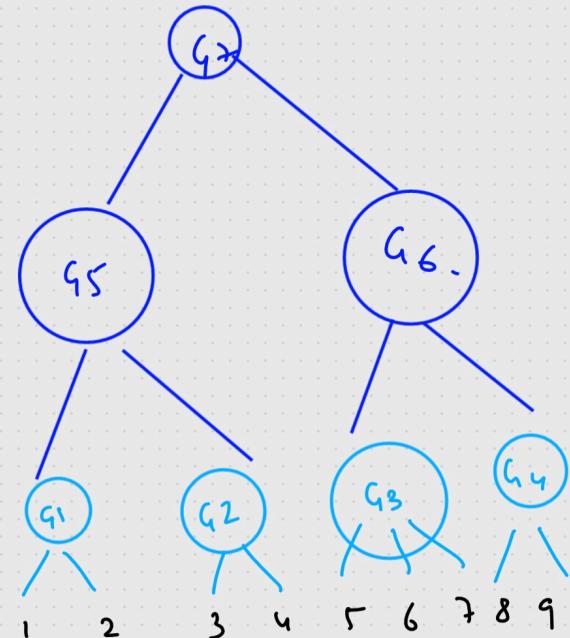
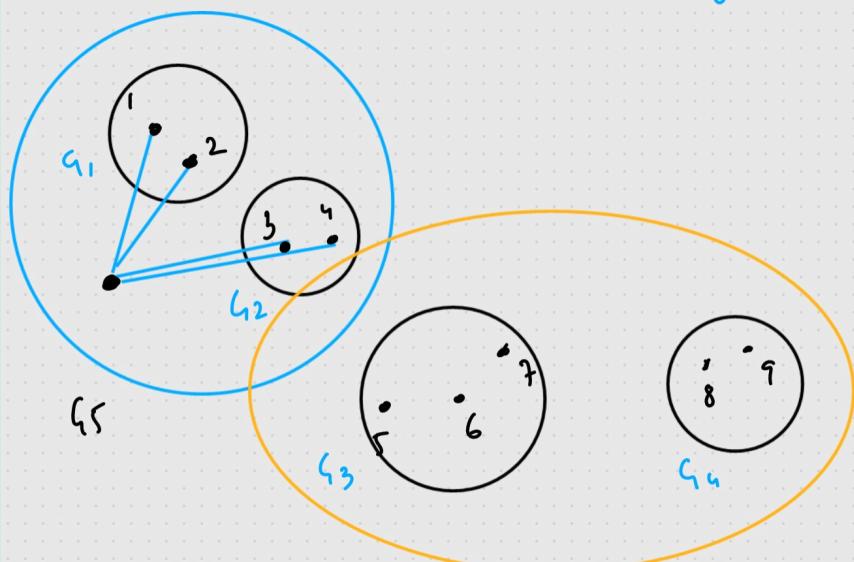
f_1	f_2
7	2
5	4
9	6
2	3
4	7
8	1

Median Median

Back Tracking

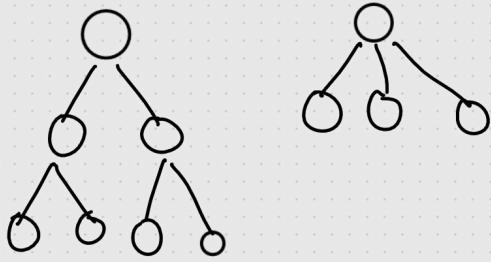
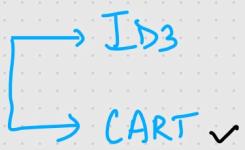
② Ball Tree

Time Complexity $\downarrow\downarrow$



Decision Tree Classifier

Decision Tree Classifier



a) Entropy and Gini Index → Purity Split

b) Information Gain → features to select for

DT construction

age = 14

if ($age \leq 15$):

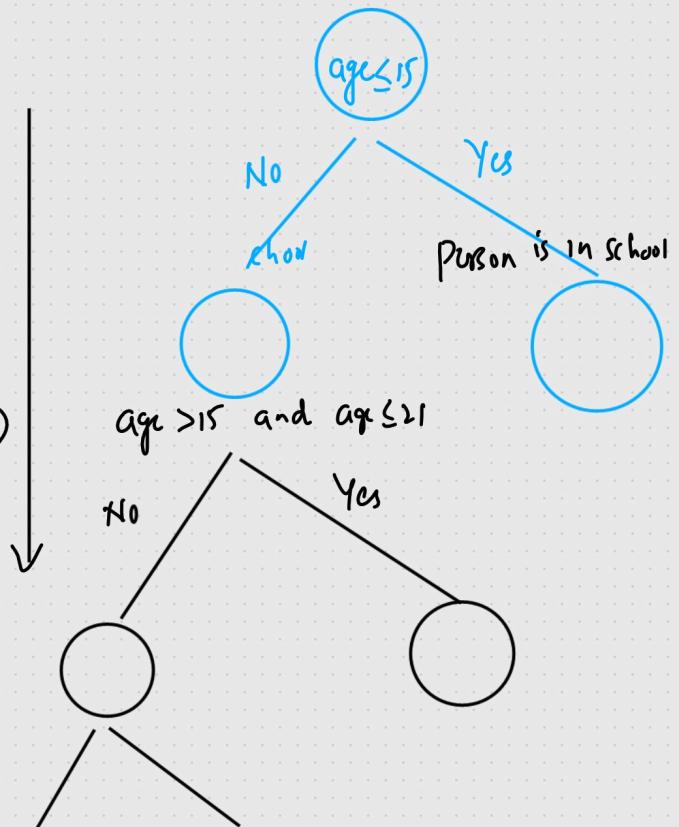
 Print ("The person is in School")

elif ($age > 15$ and $age \leq 21$):

 Print ("The person may be college")

else:

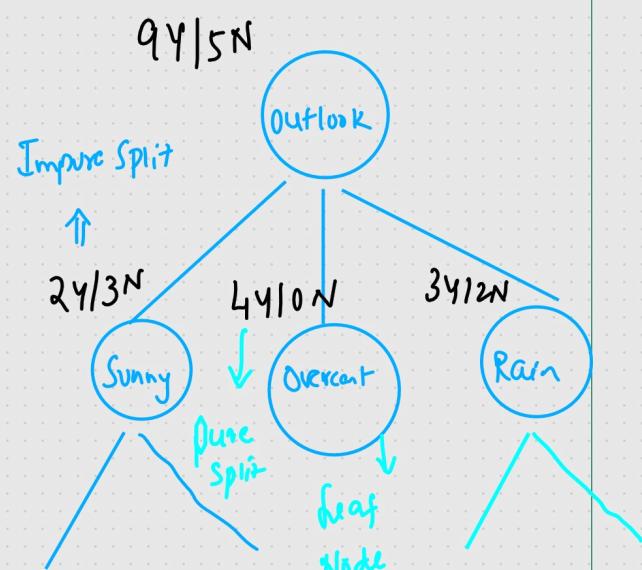
 Print ("The person has passed")



Data set

Binary classification

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No



① Purity \rightarrow Pure or Impure Split

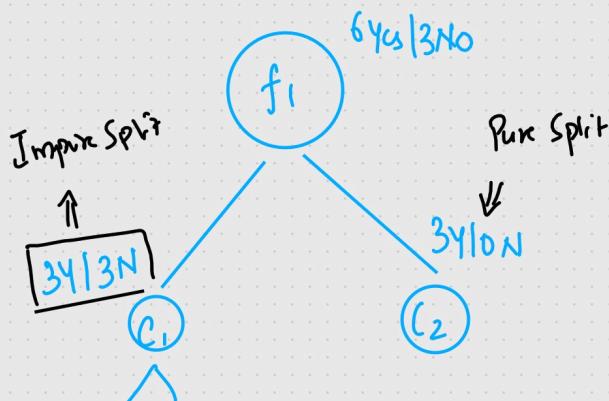
$\begin{cases} \text{Entropy} \\ \text{Gini Impurity} \end{cases}$

② What feature you need Select for
Splitting \rightarrow Information Gain }

1
0
{Binary classification}

1) Entropy

$$H(S) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$

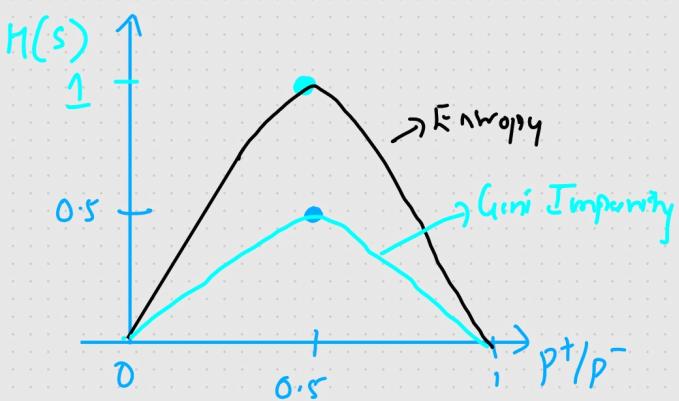


$$H(C_1) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$

$$= -\frac{3}{6} \log_2 \frac{3}{6} - \frac{3}{6} \log_2 \frac{3}{6}$$

2) Gini Impurity

$$G.I. = 1 - \sum_{i=1}^n (P_i)^2$$



= 1 \Rightarrow Impure Split

$$H(C_2) = -\frac{3}{3} \log_2 \frac{3}{3} - 0 \log_2 0$$

= $-1 \log_2 1 \Rightarrow 0 \Rightarrow$ Pure Split

(2) Gini Impurity

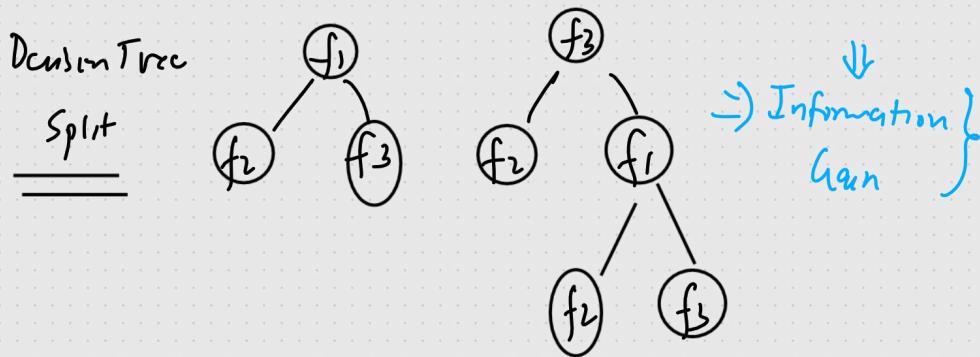
$$\begin{aligned} G \cdot I &= 1 - \sum_{i=1}^n (p_i)^2 \\ &= 1 - ((p_+)^2 + (p_-)^2) \\ &= 1 - \left(\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right) \\ &= \underline{\underline{0}} \Rightarrow \text{Pure Split} \\ &= \underline{\underline{0.5}} \Rightarrow \text{Impure Split} \end{aligned}$$

3y/10N

$$\begin{aligned} &= 1 - \left(\left(\frac{3}{3}\right)^2\right) \\ &= 1 - 1 \end{aligned}$$

$\underline{\underline{0}} \Rightarrow \text{Pure Split}$

f₁ f₂ f₃



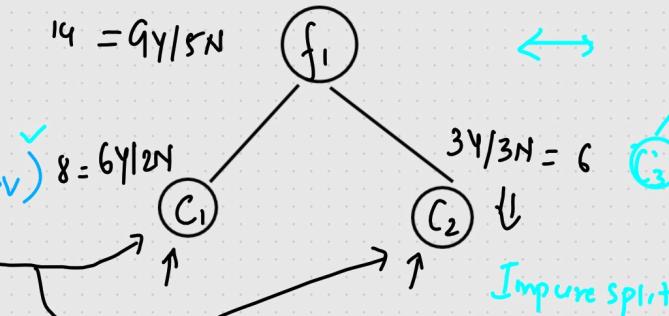
Information Gain ↴

f₁ f₂ f₃ O/P

$$\text{Gain}(S, f_1) = H(S) - \sum_{v \in \text{eval}} \frac{|S_v|}{|S|} H(S_v)$$

↑ Entropy of the root node
↑ Root Node

$$H(S) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$



$$= -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} \quad H(C_1) = -\frac{6}{8} \log_2 \frac{6}{8} - \frac{2}{8} \log_2 \frac{2}{8}$$

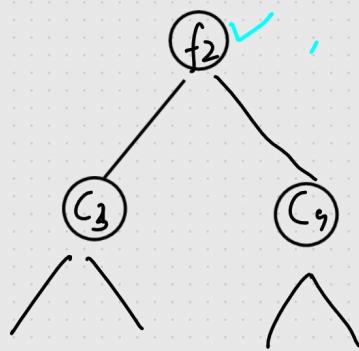
≈ 0.94

$$H(C_1) \approx 0.81$$

$$H(C_2) = 1$$

$$\text{Gain}(S, f_1) = 0.94 - \left[\frac{8}{14} \times 0.81 + \frac{6}{14} \times 1 \right]$$

$$\text{Gain}(S, f_1) = 0.049$$



$$\text{Gain}(S, f_2) = 0.051 > \text{Gain}(S, f_1) = 0.049$$

Information is Basically calculated.

Entropy Vs Gini Impurity

$$H(S) = -P_+ \log_2 P_+ - P_- \log_2 P_- \quad G.I. = 1 - \sum_{i=1}^n (P_i)^2 \Rightarrow$$

O/P = 3 categories

$$H(S) = -P_{C_1} \log_2 P_{C_1} - P_{C_2} \log_2 P_{C_2} - P_{C_3} \log_2 P_{C_3}$$

Whence dataset is small \rightarrow Entropy
large \rightarrow Gini Impurity

Decision Tree Split for Numerical Feature.

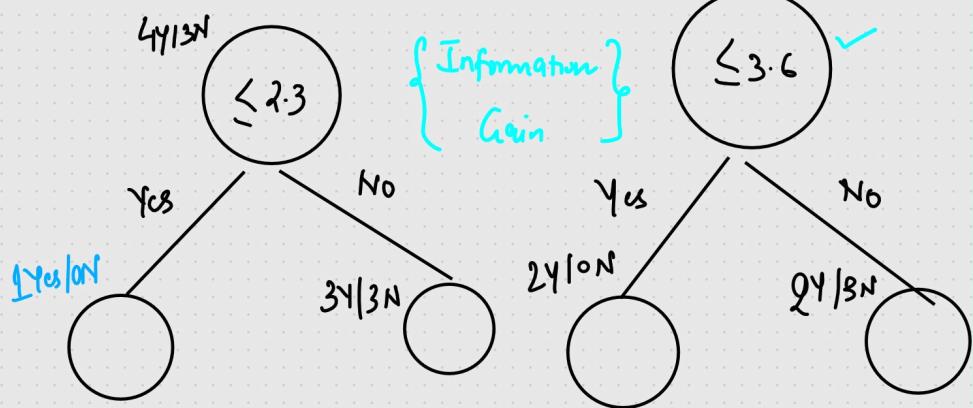
Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

① Sort the feature value

f_1	O/P
2.3	Yes
3.6	Yes
4	No
5.2	No
6.7	Yes
8.9	No
10.5	Yes

① Threshold = 2.3

② Threshold = 3.6



Millions of records

(Time Complexity ↑↑)

Decision Tree Regression

	Dataset		<u>qp</u>
	Career	Gap	Salary
→ 1	Exp	Yes	40K
→ 2	2	Yes	42K
→ 3	2.5	No	52K
→ 4	3	No	60K
→ 5	4.5	Yes	56K

$[40K, 42K, 52K, 60K, 56K]$

≤ 2

60.8

Yes

No

$40K$

$42, 42, 60, 50.$

$V(C_1) = 100$

$V(C_2) = 51$

Variance Reduction (Regression Problem)

$$\text{① Variance} = \frac{1}{n} \sum_{i=1}^n (y - \bar{y})^2$$

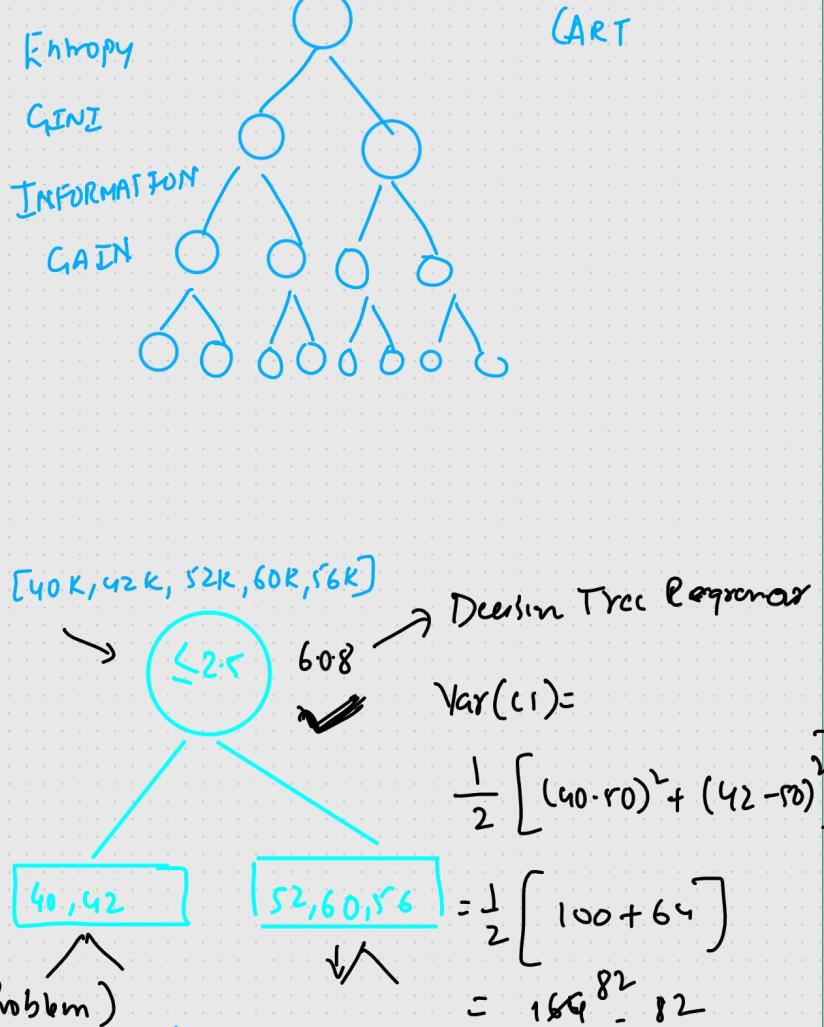
Mean Square Error
Average

$$\text{Variance(Root)} = \frac{1}{5} \left[(40-50)^2 + (42-50)^2 + (52-50)^2 + (60-50)^2 + (56-50)^2 \right]$$

$$= \frac{1}{5} \left[100 + 64 + 4 + 100 + 36 \right]$$

$$= 60.8$$

$$\text{Variance}(C_1) = \frac{1}{n} \sum_{i=1}^n (y - \bar{y})^2$$



$$= \frac{1}{1} (40 - 50)^2 \\ = 100$$

$$\text{Variance } (c2) = \frac{1}{n} \sum_{i=1}^n (y - \bar{y})^2 \\ = \frac{1}{4} \left[(42 - 50)^2 + (52 - 50)^2 + (60 - 50)^2 + (56 - 50)^2 \right] \\ = \frac{1}{4} [64 + 4 + 100 + 36] \\ = 51$$

Variance Reduction \downarrow

$$= \text{Var}(\text{Root}) - \sum w_i \text{Var}(\text{child}) \\ = 60.8 - \left[\frac{1}{8} \times 20 + \frac{4}{5} \times 51 \right] \\ = 60.8 - 20 - 40.8$$

Variance Reduction = 0

0 0.004
 $\text{Variance Reduction}(\text{Left Split}) < \text{VR}(\text{Right Split})$

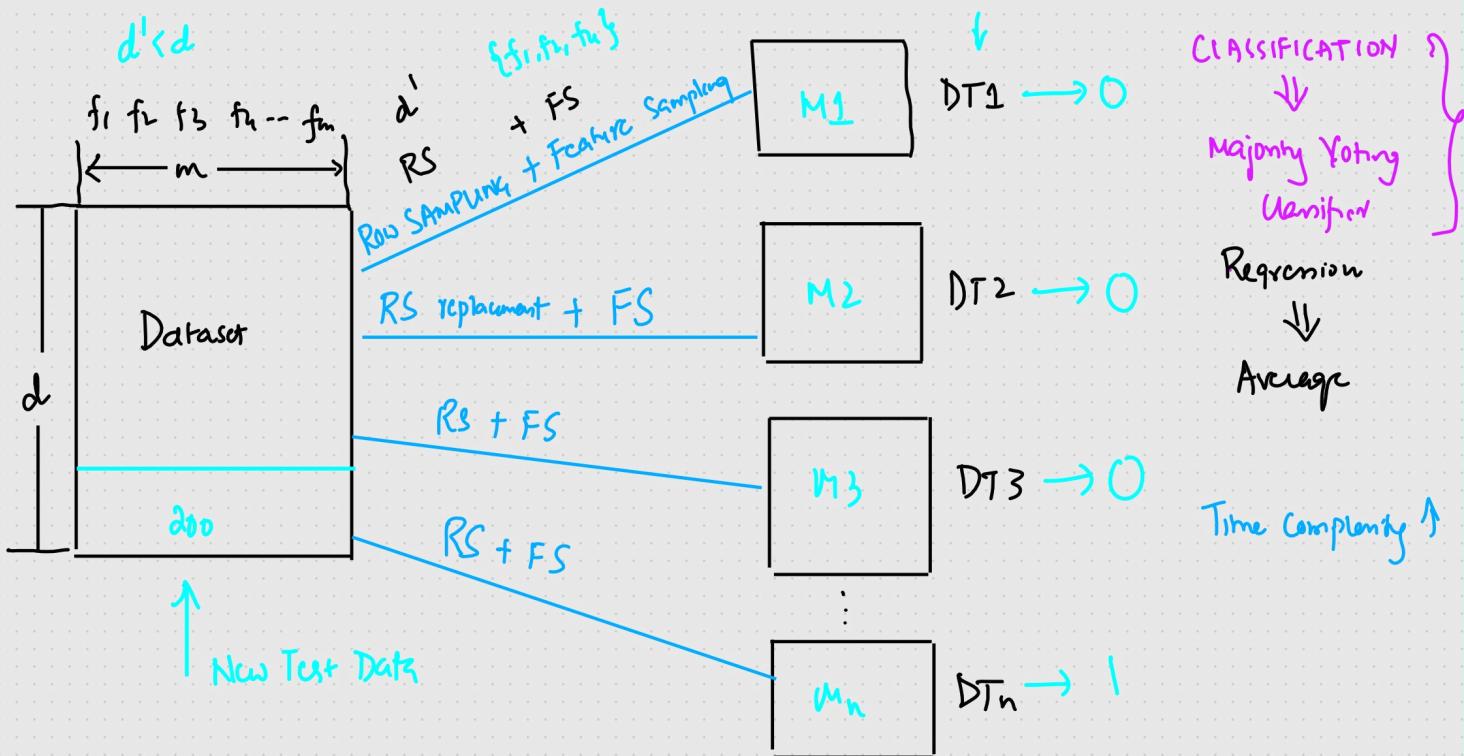


O/P = $\left[\frac{40 + 42}{2} \right]$

= 41

Decision Tree
O/P Reg.
 $\frac{52 + 60 + 56}{3} = \underline{\underline{56}}$

Random Forest CLASSIFICATION AND REGRESSIONS



CLASSIFICATION - MAJORITY VOTING CLASSIFIER

REGRESSION - Average O/P of the Models.

Why should we use Random Forest instead of DT?

Decision Tree

Generalized Model → Low Bias
→ Low Bias

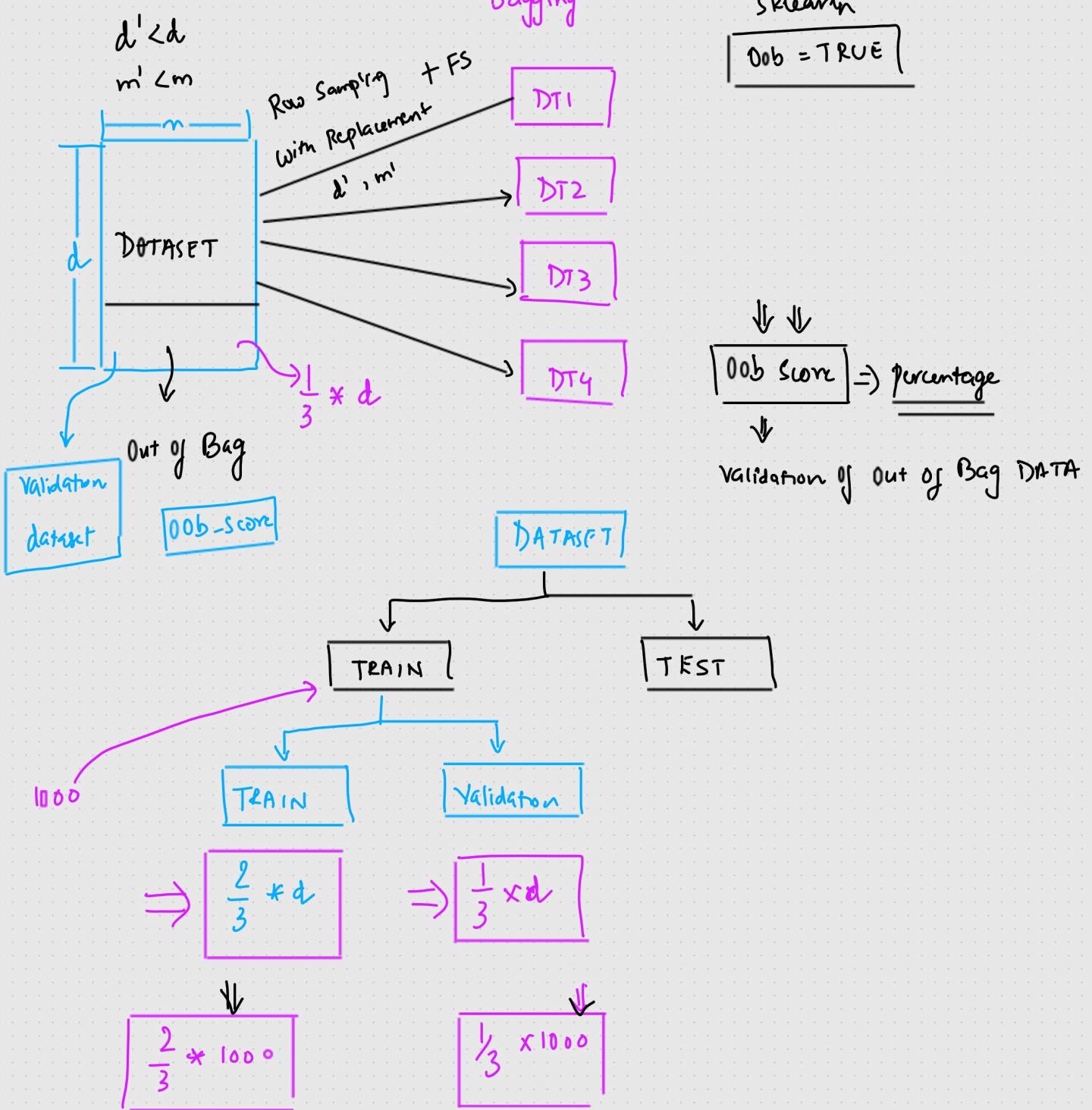
Overshooting

Train Acc \uparrow → Low Bias → Low Bias

Test Acc \downarrow → High Variance → Low Variance

Random Forest

Out of Bag Evaluation (Random Forest) Out of Bag Score?



AdaBoost Machine Learning Algorithms

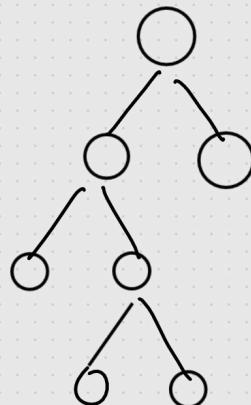
Decision Trees

- {
 ① Random Forest Classifier
 ② Random Forest Regressor
 }
 {
 Base Learners
 Bagging
 }

Decision Tree :

Overfitting : Train Acc \uparrow
Test Acc \downarrow

Low Bias
High Variance

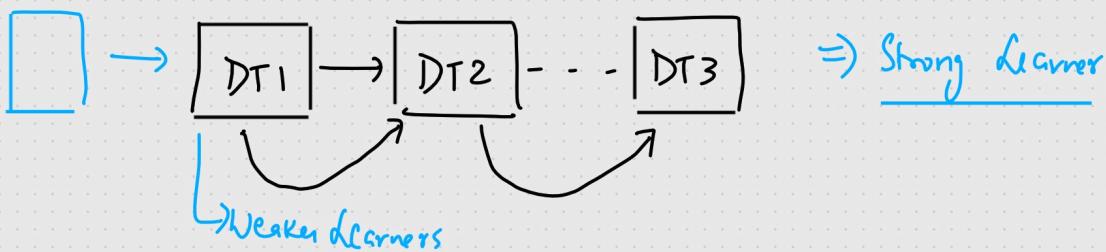


Random Forest { Bagging }

{
 Low Bias
 Low Variance
}

Boosting

{ Sequentially connected }



Weak Learner \rightarrow Haven't learnt much from the
Training Dataset

Random Forest \rightarrow Majority Voting classifier
Average of (O/P)

{ weak learners }

Boosting

- {
 ① AdaBoost
 ② Gradient Boosting
 ③ Xgboost
}

AdaBoost → Assignment weights to the weak learner

$M_1 \dots M_n \rightarrow \underline{\text{Decision Tree Stumps}}$

$$f = \alpha_1(M_1) + \alpha_2(M_2) + \alpha_3(M_3) + \dots + \alpha_n(M_n)$$

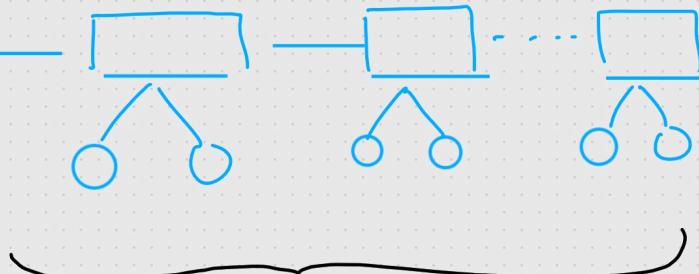
$\{\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n\} \Rightarrow \text{weights}$

CLASSIFICATION

REGRESSION

Decision Stump

Depth = 1



Weak learner?

Underfitting

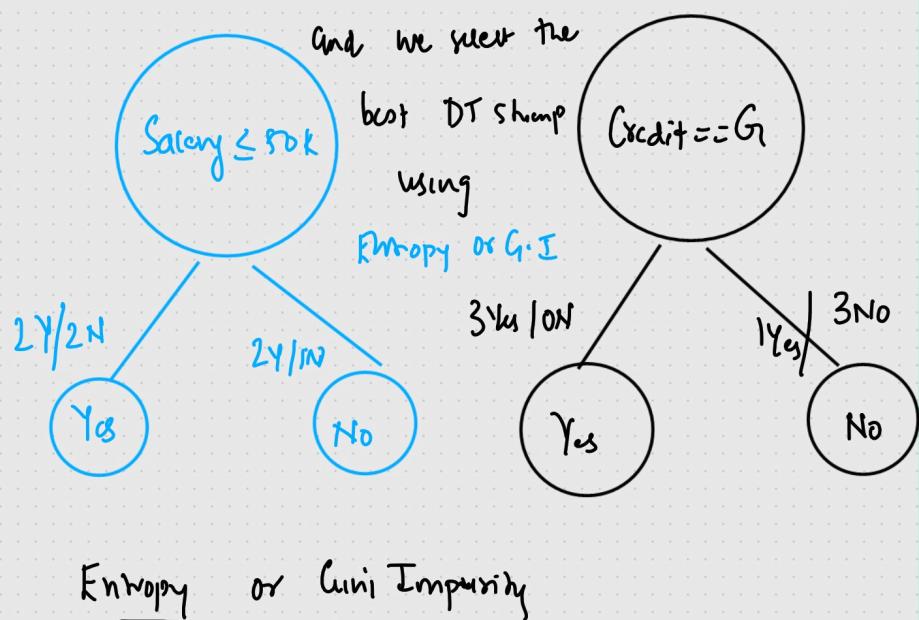
Train Acc ↓ 40%
Test Acc ↑ 45%

Decision Tree Stump

$\left\{ \begin{array}{l} \text{High Bias} \\ \text{Low Variance} \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \text{Low Bias} \\ \text{High Variance} \end{array} \right\}$

AdaBoost Classifier Maths Indepth Intuition ① We create Decision Tree Stump

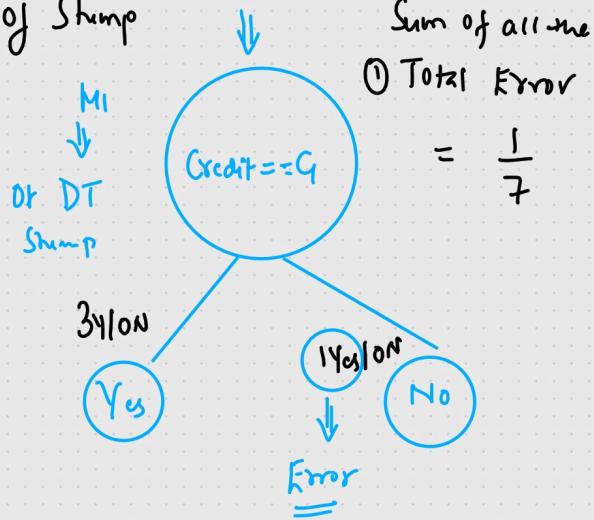
Salary	Credit	Approval
$\leq 50K$	B	No
$\leq 50K$	G	Yes
$\leq 50K$	G	Yes
$> 50K$	B	No
$> 50K$	G	Yes
$> 50K$	N	Yes
$\leq 50K$	N	No



$$H(S) = -P + \log_2 P + -P - \log_2 P -$$

② Sum of the Total Errors And Performance of Stump

Salary	Credit	Approval	Sample Weights
$\leq 50K$	B	No.	$\frac{1}{7}$
$\leq 50K$	G	Yes	$\frac{1}{7}$
$\leq 50K$	G	Yes	$\frac{1}{7}$
$> 50K$	B	No	$\frac{1}{7}$
$> 50K$	G	Yes	$\frac{1}{7}$
$ > 50K$	N	Yes	$\frac{1}{7}$
$\leq 50K$	N	No.	$\frac{1}{7}$



$$\textcircled{2} \text{ Performance of Stump} = \frac{1}{2} \ln \left[\frac{1 - TE}{TE} \right]$$

$$= \frac{1}{2} \ln \left[\frac{1 - \frac{1}{7}}{\frac{1}{7}} \right]$$

$$= \frac{1}{2} \ln [6] \approx \underline{\underline{0.896}}$$

Performance of Stump ≈ 0.896

$$f = d_1(M_1) + d_2(M_2) + \dots + d_n(M_n)$$

$$d_1 = 0.896 \Rightarrow \text{weight}$$

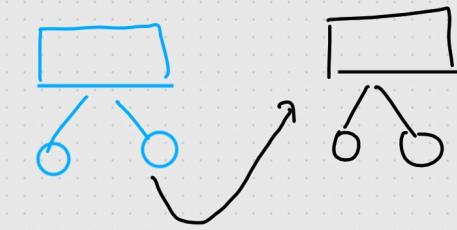
③ Update the weights for correctly and Incorrectly classified points

Salary	Credit	Approval	Sample Weights	Update wts	For correct classified points
$\leq 50K$	B	No.	$\frac{1}{7} \downarrow$	0.058	$= \text{weight} * e^{-\text{Performance of Stump}}$
$\leq 50K$	G	Yes	$\frac{1}{7} \downarrow$	0.058	$= \frac{1}{7} * e^{-(0.896)}$
$\leq 50K$	G	Yes	$\frac{1}{7} \downarrow$	0.058	$= 0.058$
$> 50K$	B	No	$\frac{1}{7} \downarrow$	0.058	
$> 50K$	G	Yes	$\frac{1}{7} \downarrow$	0.058	
$ > 50K$	N	Yes	$\frac{1}{7} \uparrow$	0.349	For Incorrect classified points
$\leq 50K$	N	No.	$\frac{1}{7} \downarrow$	0.058	$= \text{weight} * e^{\text{Performance of Stump}}$
					$= \frac{1}{7} * e^{(0.896)}$

$$= 0.349$$

④ Normalized Weights Computation And Assigning Bins

Salary	Credit	Approval	Update wts	Normalized Weights	Bins Assignment
$\leq 50K$	B	No	0.058	0.08	$0 - 0.08$
$\leq 50K$	G	Yes	0.058	0.08	$0.08 - 0.16$
$\leq 50K$	G	Yes	0.058	0.08	$0.16 - 0.24$
$> 50K$	B	No	0.058	0.08	$0.24 - 0.32$
$> 50K$	G	Yes	0.058	0.08	$0.32 - 0.40$
$> 50K$	N	Yes	0.349	0.50	$0.40 - 0.70$
$\leq 50K$	N	No	0.058	0.08	$0.50 - 0.58$
				<u>0.697</u>	<u>≈ 1</u>



$f_1 = 0.896$ Prepare
datapoints

⑤ Select data points to send to Next Step

① Iteration process selecting random value between 0 and 1

Salary	Credit	Approval	Bins Assignment
$\leq 50K$	B	No	$0 - 0.08$
$\leq 50K$	G	Yes	$0.08 - 0.16$
$\leq 50K$	G	Yes	$0.16 - 0.24$
$> 50K$	B	No	$0.24 - 0.32$
$> 50K$	G	Yes	$0.32 - 0.40$
$> 50K$	N	Yes	$0.40 - 0.70$
$\leq 50K$	N	No	$0.50 - 0.58$

S	Credit	Approval	Random
$> 50K$	N	Yes	0.50
$\leq 50K$	G	Yes	0.10
$> 50K$	N	Yes	0.60
$> 50K$	N	Yes	0.75
$\leq 50K$	G	Yes	0.24
$> 50K$	B	No	0.32
$> 50K$	N	Yes	0.87

⑥ These records will be sent to next DT stump.

S	Credit	Approve	Sample weight	
>50K	N	Yes	$\frac{1}{6}$	TE
<=50K	G	Yes	$\frac{1}{6}$	Performance Stump $\Rightarrow 0.65$
>50K	N	Yes	$\frac{1}{6}$	
>50K	N	Yes	$\frac{1}{6}$	
<=50K	G	Yes	$\frac{1}{6}$	
>50K	B	No	$\frac{1}{6}$	
>50K	N	Yes	$\frac{1}{6}$	

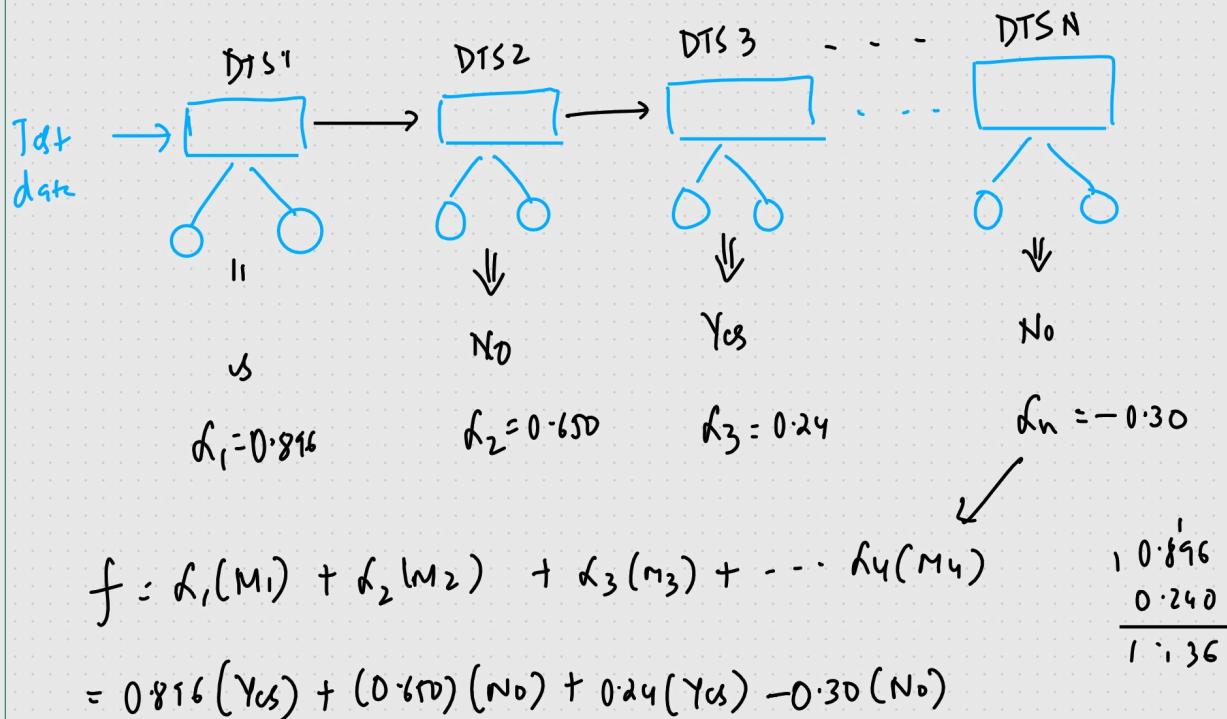
$f_1 = \alpha_1(M_1) + \alpha_2(M_2)$

$\alpha_1 = 0.896$ $\alpha_2 = 0.65$

————— —————

⑦ Final Prediction

Test data ($\leq 50K, G$)



$$= \boxed{1.136} (\gamma_u) + 0.350 (N_0) \Rightarrow \underline{\underline{O/P}} : \underline{\underline{\gamma_{us}}}$$

Performance of say (γ_u) = 1.136 >

Performance of say (N_0) = 0.350

GRADIENT BOOSTING ALGORITHM

① Regression

② CLASSIFICATION

Regression Data

		y		\hat{y}	$(y - \hat{y})$		\hat{y}	R_1	R_2	\hat{y}	R_3	R_4
Exp	Degree	Salary		\hat{y}	R_1	R_2	\hat{y}			\hat{y}	R_3	R_4
→ 2	B.E	50K		75K	-25K	-23	72.7			72.7	-22.7	-
3	Masters	70K		75K	-5K	-3	74.7			74.7	-4.7	-
5	Masters	80K		75K	5K	3	-			-	-	-
6	PhD	100K		75K	25K	20	-			-	-	-
				75K								

Steps

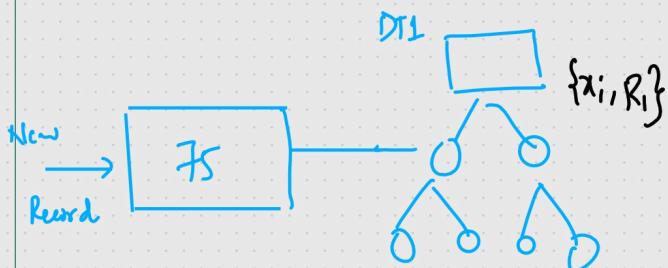
① Create a Base Model

75 Average = $\frac{50 + 70 + 80 + 100}{4} = 75$

② Compute Residuals, Error

③ Construct a Decision Tree

Consider inputs x_i and O/p R_i



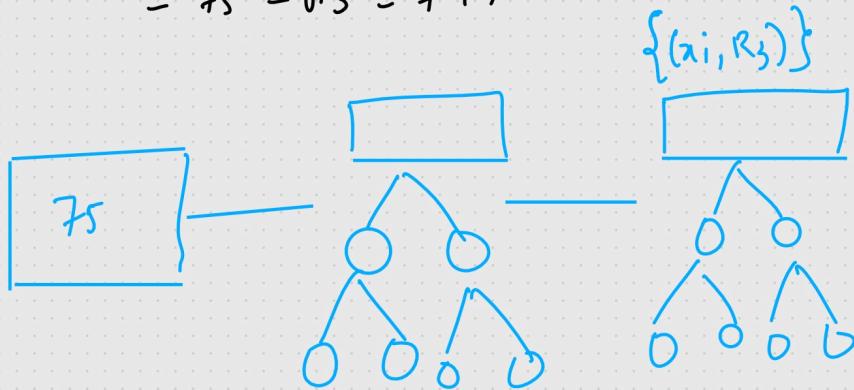
Predicted o/p $\Rightarrow 75 + (-23) = 75 - 23 = 52$ Overfitting

$$\text{Predicted O/P} \Rightarrow 75 + d_1(D_1) = 75 + (0.1)(-23)$$

$$\begin{aligned} d &= \text{Learning Rate} \\ d &= 0.1 \end{aligned}$$

$$\begin{aligned} &= 75 - 2.3 \\ &= 72.7 \end{aligned}$$

$$\begin{aligned} &\Rightarrow 75 + 0.1(-3) \\ &= 75 - 0.3 = 74.7 \end{aligned}$$



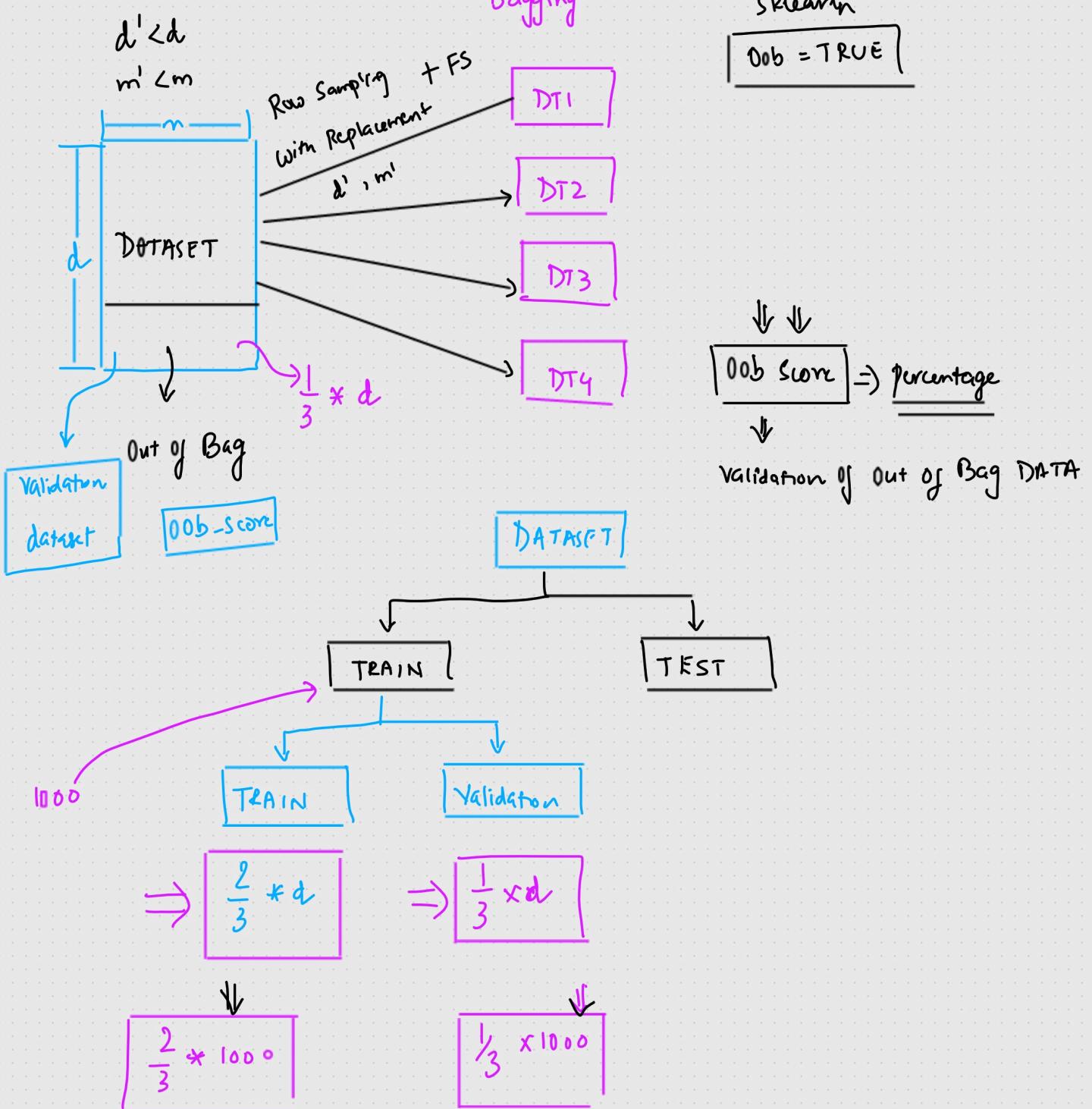
$$F(x) = L_0 h_0(x) + L_1 h_1(x) + L_2 h_2(x) + L_3 h_3(x) + \dots + L_n h_n(x)$$

Learning Rate $d = 0.1$

$$F(x) = \sum_{i=0}^n d_i h_i(x)$$

⇒ Final Function
of Gradient
Boosting

Out of Bag Evaluation (Random Forest) Out of Bag Score?



Xgboost ML Algorithm (Classification)

$$\hat{y} = 0.5$$

Dataset

	Salary	Credit	Approval	\hat{y}	R1	\hat{y}	R2
→	<=50K	B	O	-0.5	0.52	-0.48	
	<=50K	G	I	0.5	0.58	0.42	
	>50K	G	I	0.5	—	—	
	>50K	B	O	-0.5	—	—	
	>50K	G	I	0.5	—	—	
	>50K	N	I	0.5	—	—	
	<=50K	N	O	-0.5	—	—	

Steps

① Construct a base Model

② Construct a Decision Tree with root.

③ Calculate Similarity Weight

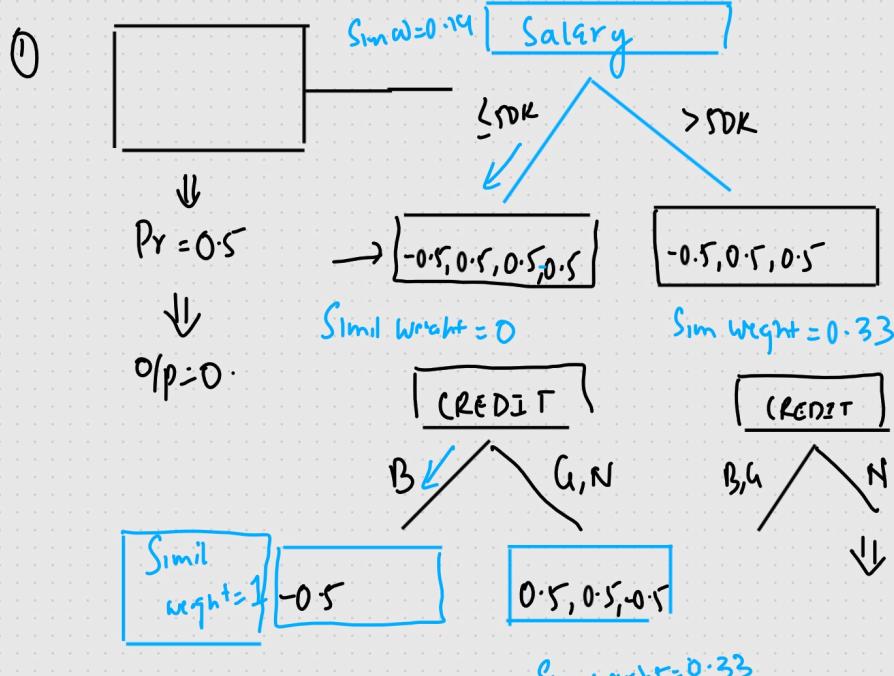
$$= \frac{\sum (\text{Residual})^2}{\sum P_r(1-P_r) + \lambda}$$

(Core value $\leftarrow \boxed{\sum P_r(1-P_r) + \lambda}$)

④ Calculate Gain

Hyperparameter

Test data



$$\log(\text{odds}) = \log \left(\frac{P}{1-P} \right)$$

$$\log(\text{odds}) = \log \left(\frac{0.5}{0.5} \right) \\ = 0$$

$$\text{Similarity}(dC) = \frac{0.25}{0.25} = 1$$

$$\text{Similarity}(R_C) = \frac{0.25}{0.75} = 0.33$$

$$\text{Gain} = 1 + 0.33 - 0 = \boxed{1.33}$$

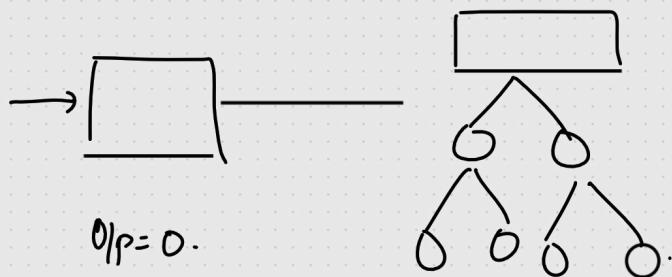
$$\text{Similarity weight } (\delta C) = \frac{\sum (\text{Residual})^2}{\sum P_r(1-P_r)}$$

$$= \left[(-0.5 + 0.5 + 1.5 - 0.5)^2 \right] = 0$$

$$1 \in [0.5(1-0.5) + 0.5(1-0.5) + 0.5(1-0.5) + 0.5(1-0.5)]$$

$$\text{Gain} = 0 + 0.33 - 0.14 = 0.21$$

Final O/p



$$\xrightarrow{\substack{\text{Now} \\ D_{gtz}}} = \sigma \left(0 + \delta(1) \right) \quad \delta = 0.1$$

Logistic & Sigmoid Activation function

$$= \sigma \left(0 + (0.1)(1) \right)$$

$$= \frac{1}{1 + e^{-0.1}} = 0.52$$

Second Record

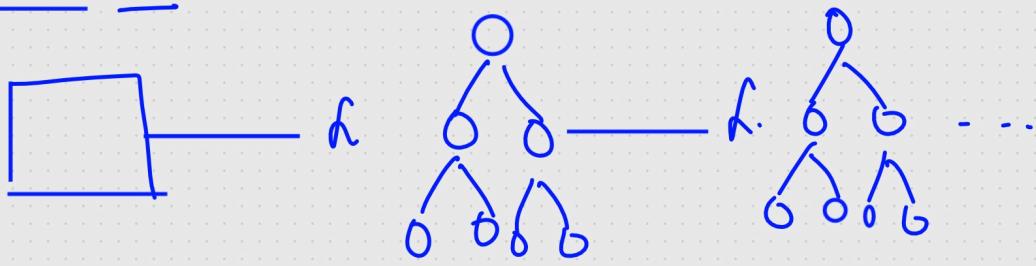
$$O/p = \sigma \left(0 + \delta(0.33) \right)$$

$$= \sigma \left(0 + 0.1(0.33) \right)$$

$$\text{Similarity weight } (R_C) = \frac{(-0.5 + 0.5 + 1.5 - 0.5)^2}{0.75} = \frac{0.25}{0.75} = \frac{1}{3}$$

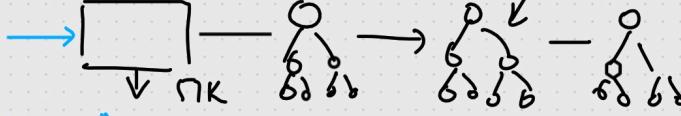
$$= \frac{1}{1+e^{-0.033}} = 0.508$$

Xg boost classifier



$$\text{O/p} = \sigma \left(\text{Basefimer} + f_1(\text{DT1}) + f_2(\text{DT2}) + f_3(\text{DT3}) \right).$$

Xgboost Regressor Mh Algorithm

Dataset {Regression} → 

<u>Exp</u>	<u>Gap</u>	Salary	<u>R₁</u>	<u>\hat{y}</u>	<u>R₂</u>
→ 2	Yes	40K	-11	49.9	-9.9
→ 2.5	Yes	42K	-9	49.9	-7.9
→ 3	No	52K	1	51.5	0.5
4	No	60K	9	51.5	8.5
4.5	Yes	62K	11	52.1	9.9
$\approx 51K$					

$[51 + (0.1)(-10)] = 51 - 0.1 = 49.9$

$[51 + (0.1)(5^-)] = 51.5$

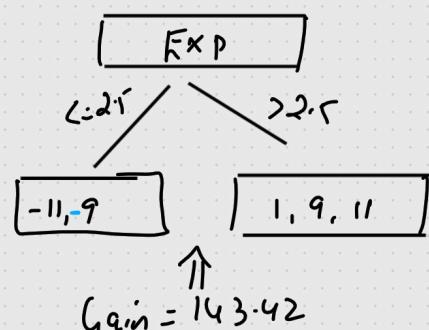
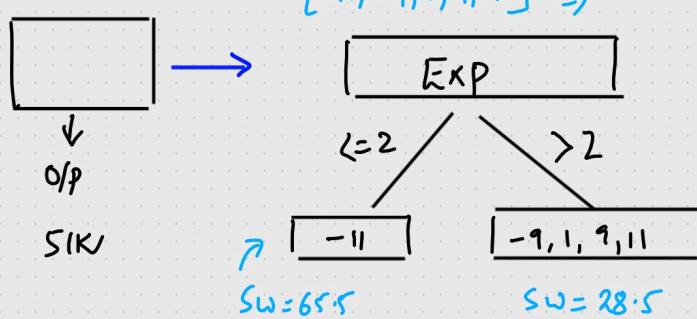
$[51 + 0.1(11)] = 51 + 1.1 = 52.1$

$$\text{Similarity weight} = \frac{\sum (\text{Residual})^2}{\sum p_r(1-p_r)}$$

Gain

Steps

- ① Create a Base Model
- ② Residual Computation
- ③ Construct DT1 using $\{x_i, R_i\}$



$$\text{Similarity weight} = \frac{\sum (\text{Residual})^2}{\text{No. of Residuals}}$$

$\lambda = 1$ → Hyperparameter

$$SW(\text{regt. resid.}) = \frac{121}{1+1} = 121/2 = 65.5 \text{ II.}$$

$\lambda \uparrow SW \downarrow$

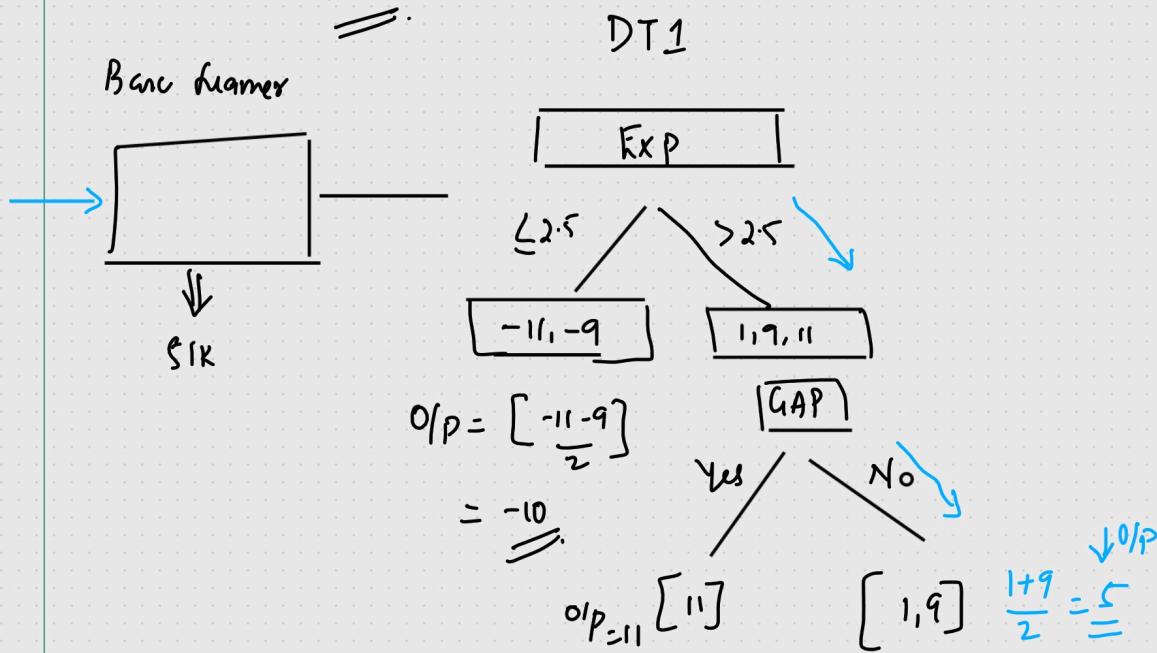
$$SW(\text{Right child}) = \frac{(-9+1+9+11)^2}{4+1} \\ = \frac{144}{5} = 28.5$$

③ Calculate Gain

$$\text{Gain} = 65.5 + 28.5 - 0.16 \\ = 98.34$$

\equiv

Bare Frame



α : Learning Rate $\alpha = 0.1 \Rightarrow$ Hyperparameter

$$XGB \text{ Boost Classifier} = \text{Bare Frame} + \alpha_1(DT_1) + \alpha_2(DT_2) + \dots + \alpha_n(DT_n)$$

$$XGB \text{ Boost Regressor} = SIK + 0.1(5) \\ = 51 + 0.5 \\ = \underline{\underline{51.5}}$$

$$\text{Similarity Weight} = \frac{\sum (\text{Residual})^2}{\text{No. of Residuals} + \lambda}$$

{ Regression }

$$\text{Similarity weight} = \frac{\sum (\text{Residual})^2}{\sum P_Y(1-P_Y) + \boxed{\lambda}}$$

Unsupervised Machine Learning

Supervised ML

→ Regression
→ Classification



Unsupervised ML → Clustering { group your data into
similar clusters }

Data

<u>Age</u>	<u>Experience</u>	<u>Salary</u>	<u>No O/p</u>
------------	-------------------	---------------	---------------

{ — — —

⇒ Unsupervised ML



{ — — —

Group or cluster
this data

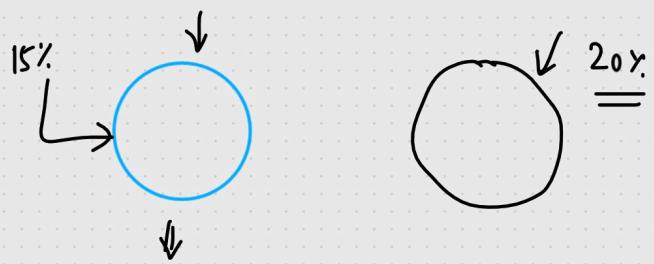
{ — — —

Eg: Customer Segmentation

Apple Product = Data Salary Spending-Score

↳ New Product ← Discount

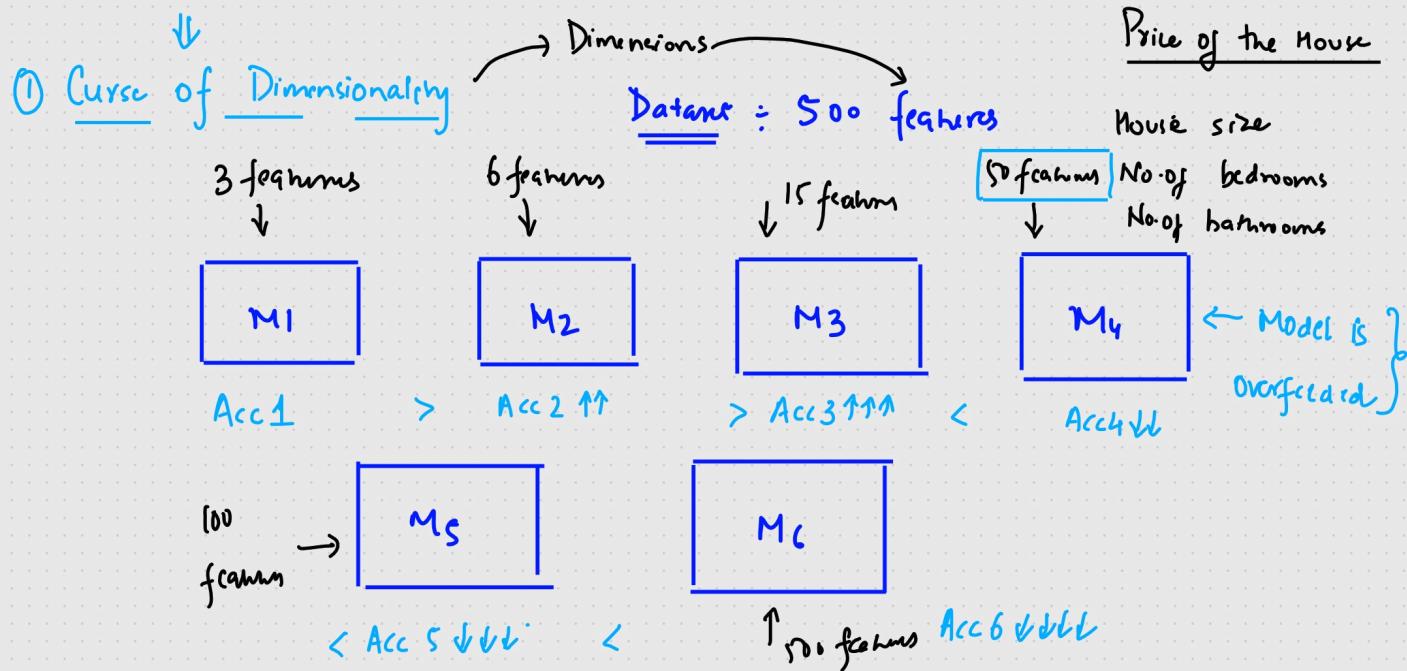




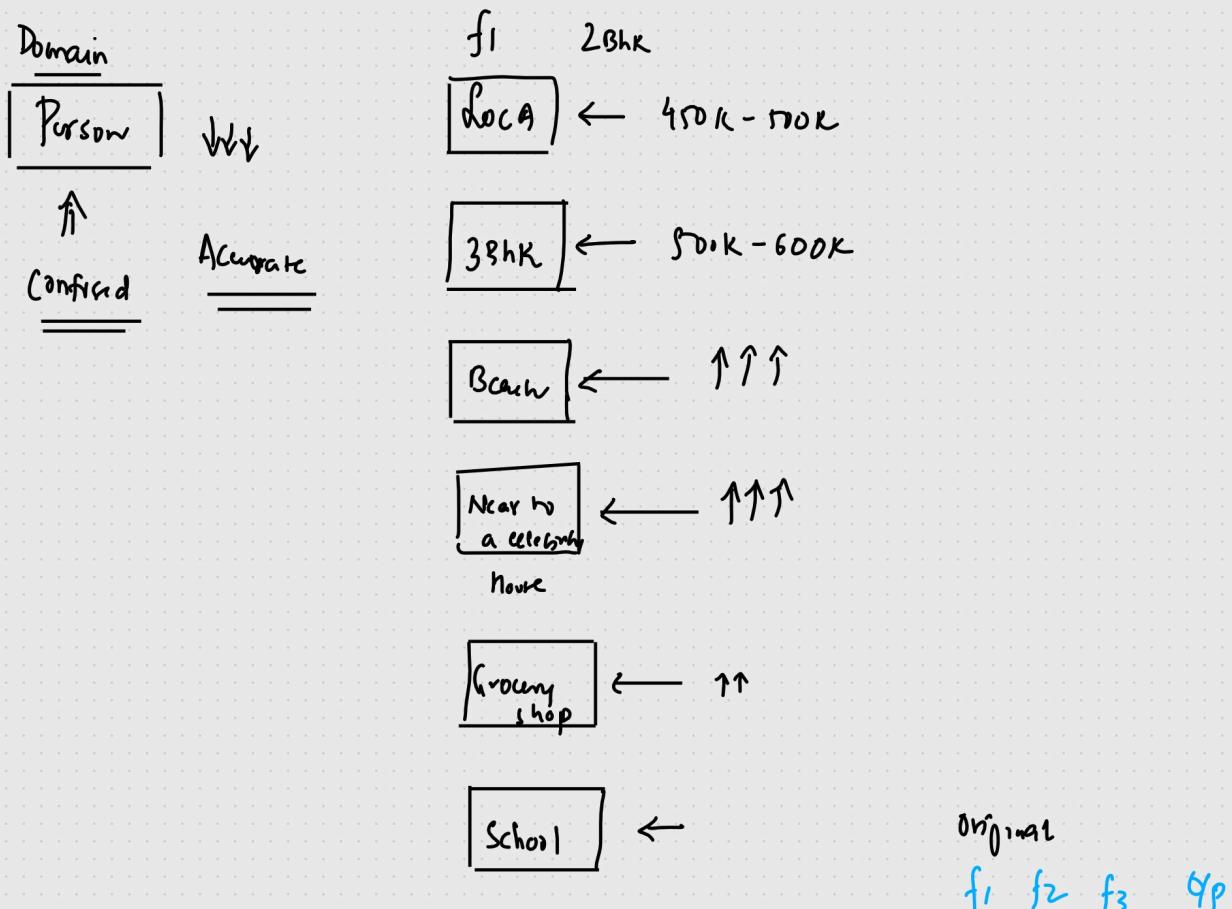
Unsupervised ML

- ① K Means Algorithm
 - ② Hierarchical Clustering
 - ③ DBScan Clustering
 - ④ Silhouette Scoring
- } Validate =

Principal Component Analysis (PCA) [Dimensionality Reduction]



② Model performance Degradation



Two different ways to remove curse of Dimensionality

① Feature Selection



② Dimensionality Reduction (PCA)



Sensor ↓ Feature Extraction

D₁ D₂ O/P



Imp features

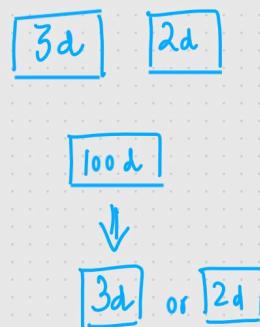
Y
Feature Extraction

Feature Selection Vs Feature Extraction

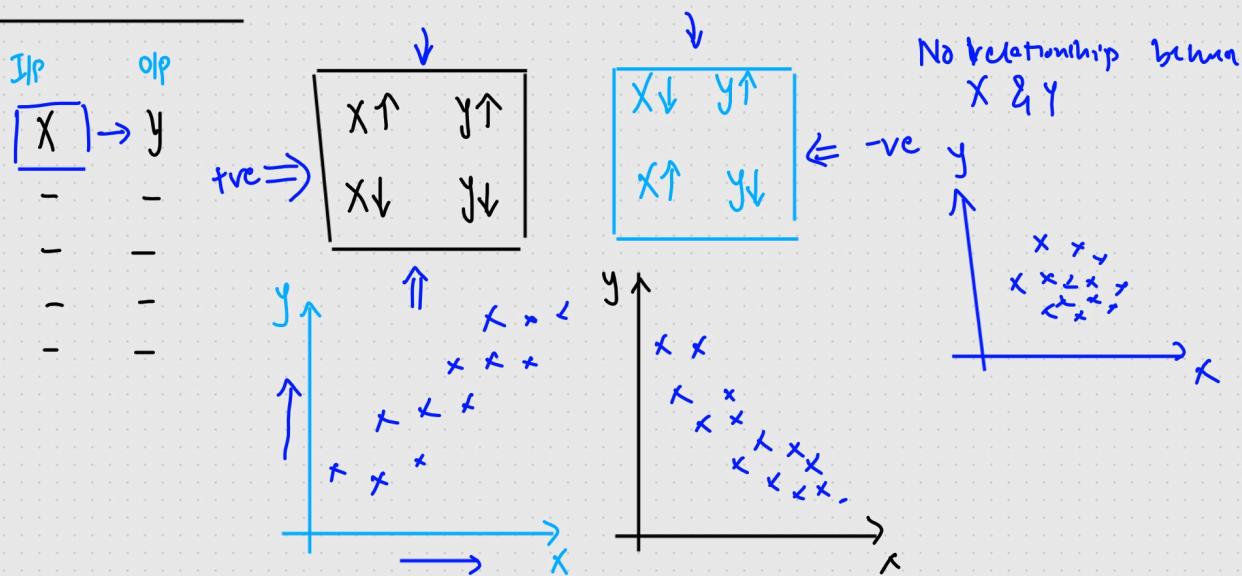
Dimensionality Reduction

① Why Dimensionality Reduction?

- Ⓐ Prevent → Curse of Dimensionality
- Ⓑ Improve the performance of the model
- Ⓒ Visualize the data → understand the data



Feature Selection



$$\text{Cov}(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{N-1} = \text{tvc}$$

≈ 0

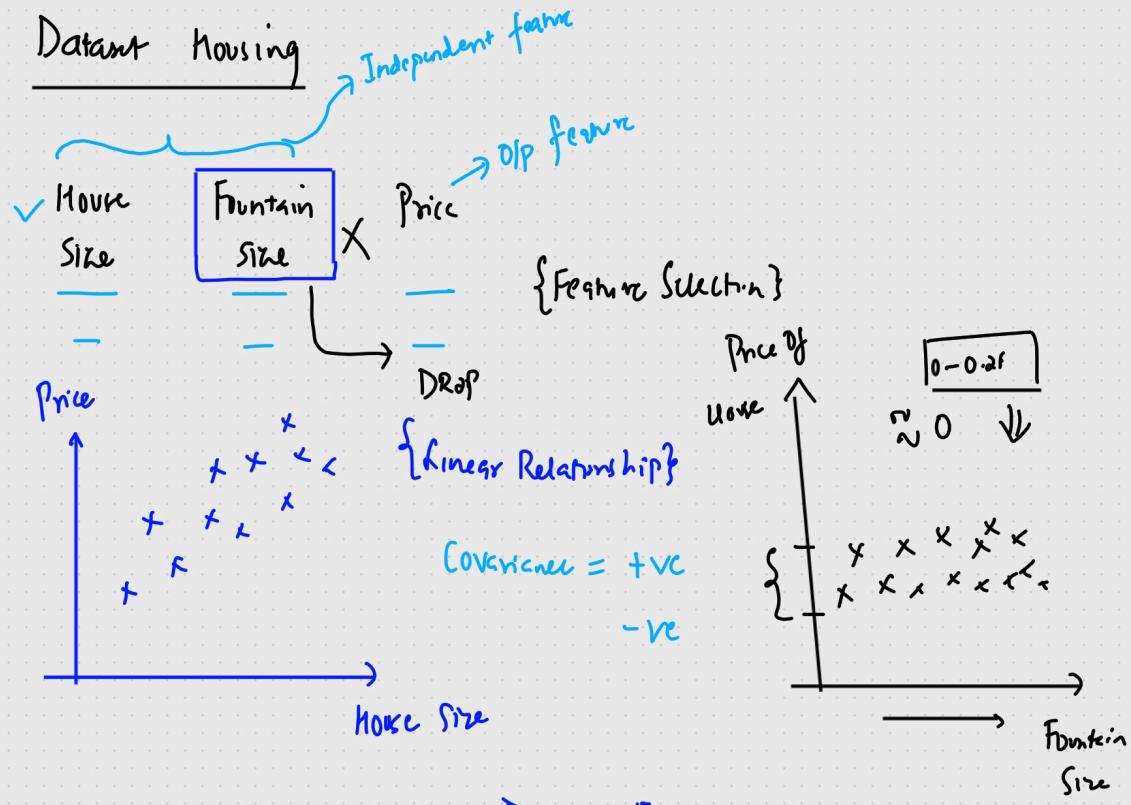
≈ 0 {No Relationship}

-ve correlated

$$\text{Pearson Correlation} = \frac{\text{Cov}(x, y)}{\sqrt{x} \cdot \sqrt{y}} = [-1 \text{ to } 1]$$

The more toward the
Value of +1 the

more +ve correlated X & Y is



Feature Extraction

{ Room Size | No. of Rooms }

Domain Expertise

+ve
-ve

2 feature → 1 feature

Dimensionality Reduction

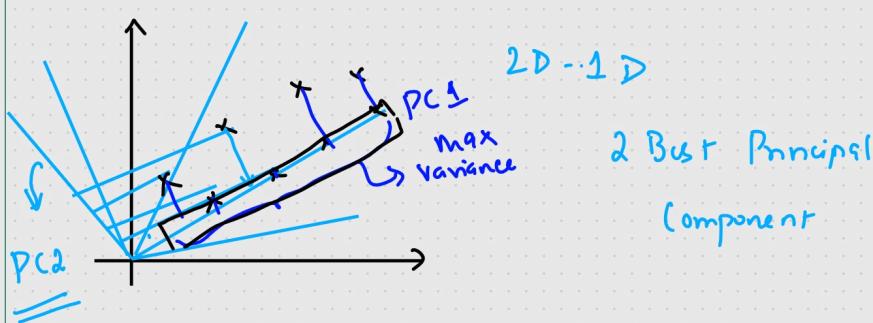
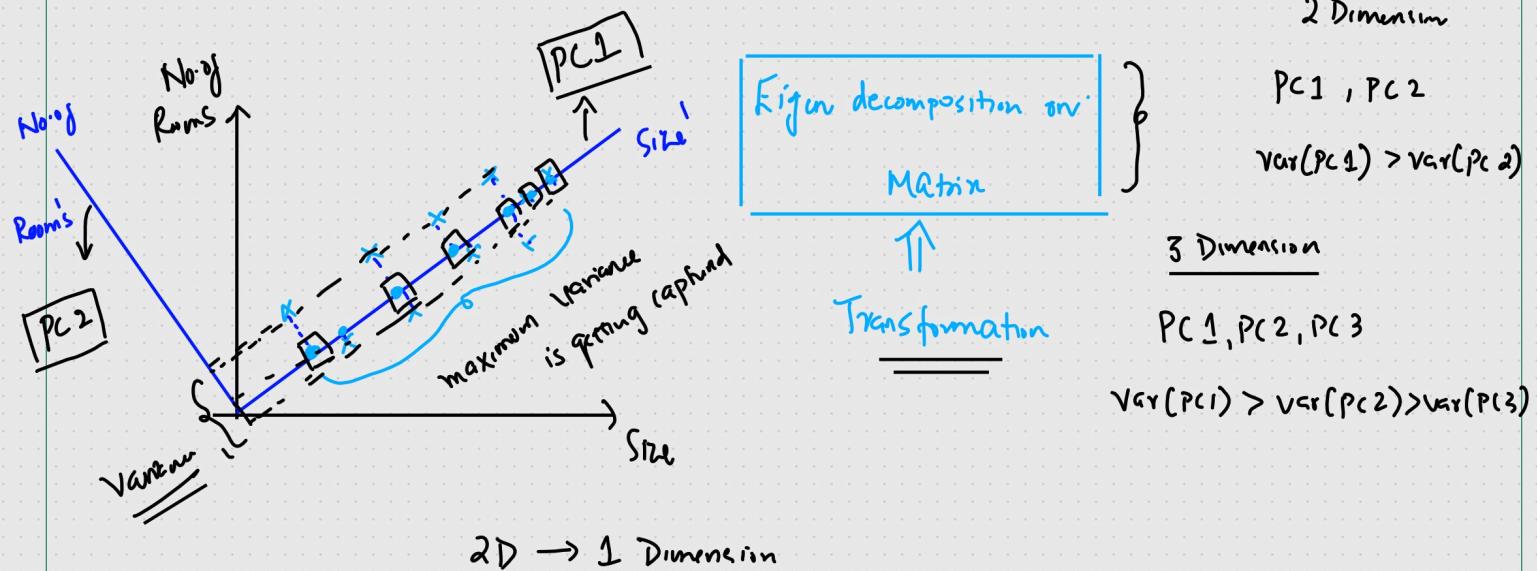
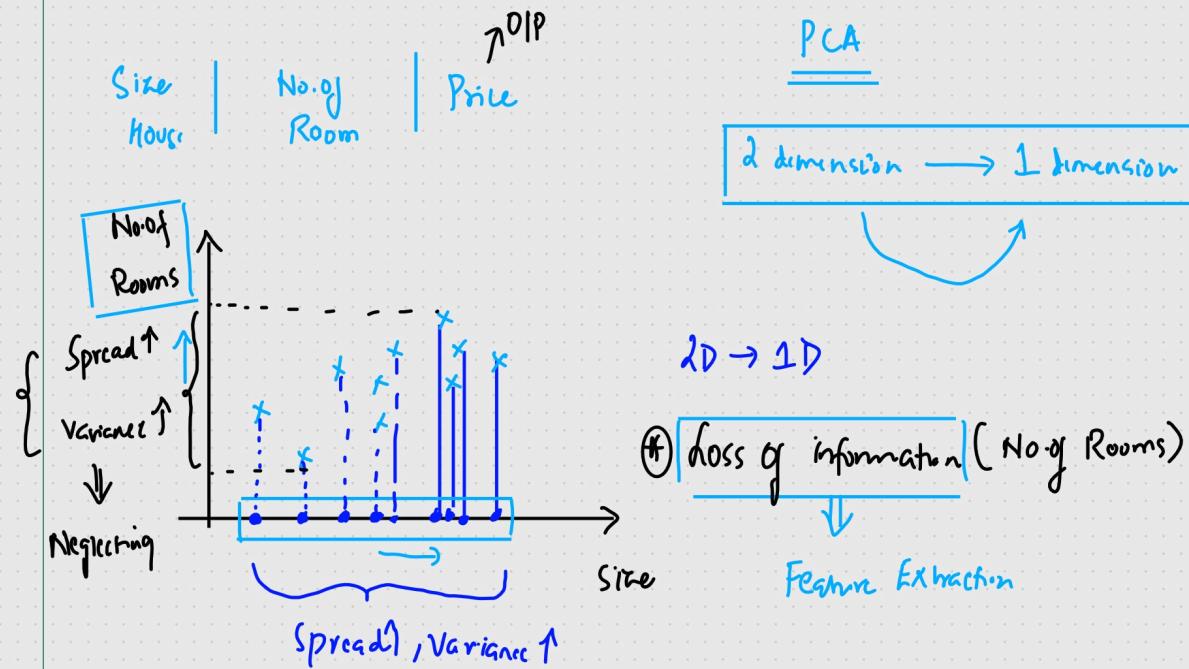
↓ ↓ Transformation To extract New feature

→ House Size | Price

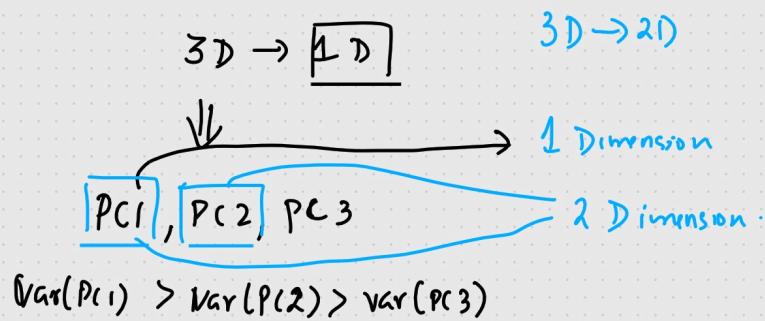
PCA Geometric Intuition

{ Dimensionality Reduction }

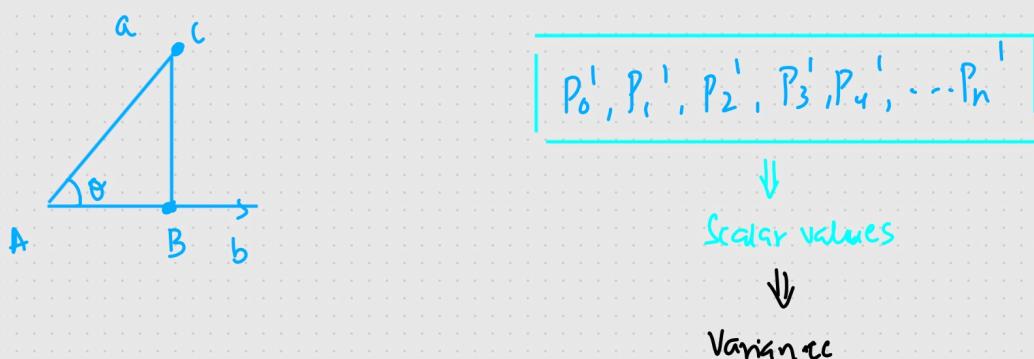
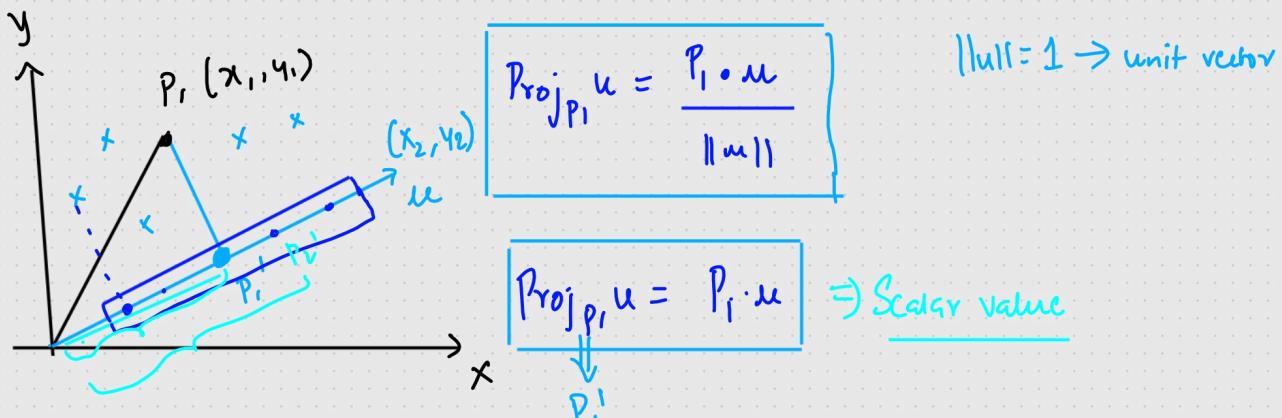
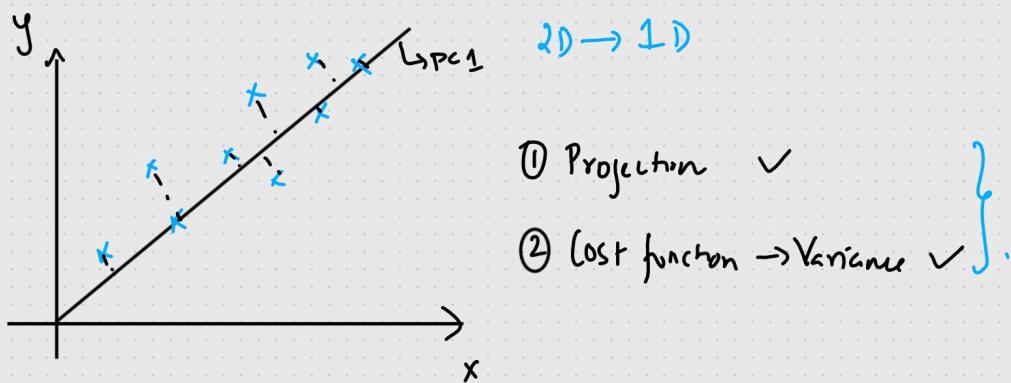
Moving Dataset



To get the best Principal Component which captures maximum variance



Maths Intuition behind PCA Algorithm



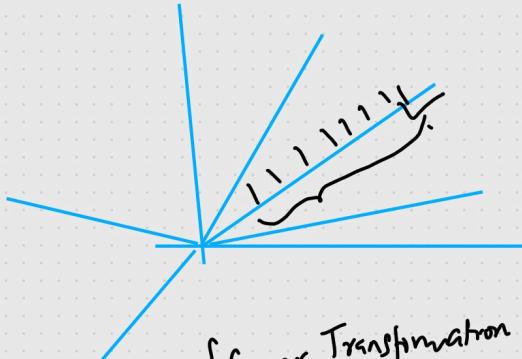
$$P_0^1, P_1^1, P_2^1, P_3^1, P_4^1, \dots, P_n^1$$

$$\downarrow \\ x_0^1, x_1^1, x_2^1, x_3^1, x_4^1, \dots, x_n^1$$

$$\text{Max Variance} = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n} \quad \left\{ \begin{array}{l} \text{Goal: Find the best} \\ \text{unit vector which} \\ \text{captures maximum variance} \end{array} \right\}$$

\downarrow
Cost function

Eigen vectors And Eigen values.

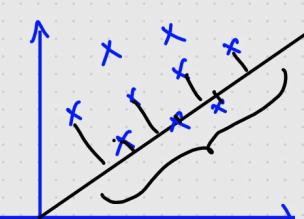


- ① Covariance Matrix between features
- ② Eigen vectors and Eigen values will be found out from this covariance matrix
- ③ Eigen vector \rightarrow Eigen value \rightarrow magnitude of the Eigen vector \rightarrow Capture the maximum variance

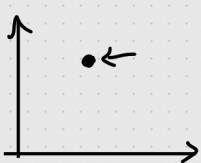
{Linear Transformation of matrix}

$$A v = \lambda v$$

Eigen vectors And Eigen values [Linear Transformation]



[Eigen decomposition of covariance Matrix]



Eigen vector & Eigen values

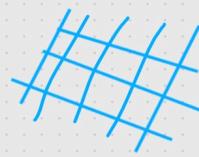
$$[] * [v] = \lambda * v$$

↓
Eigen
Value



$$A * v = \lambda * v$$

↑ v ↓



Eigen vector \rightarrow Maximum magnitude

Eigen vector \rightarrow Max Magnitude

Principal Component

Max Eigen vector

Max Var

Best Principal Component \rightarrow PC1

Steps to calculate Eigen value and vectors

① Covariance of features

$$\begin{pmatrix} X, Y \end{pmatrix} \quad Z$$

$$\downarrow x'$$

$$\text{Cov}(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{N-1}$$

$2 \times 2 \quad X \quad Y$

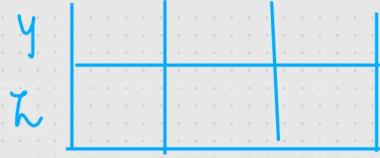
$$A = \begin{matrix} & X & Y \\ X & \text{Var}(X) & \text{Cov}(X, Y) \\ Y & \text{Cov}(Y, X) & \text{Var}(Y) \end{matrix}$$

$$\text{Cov}(X, X) = \text{Var}(X)$$

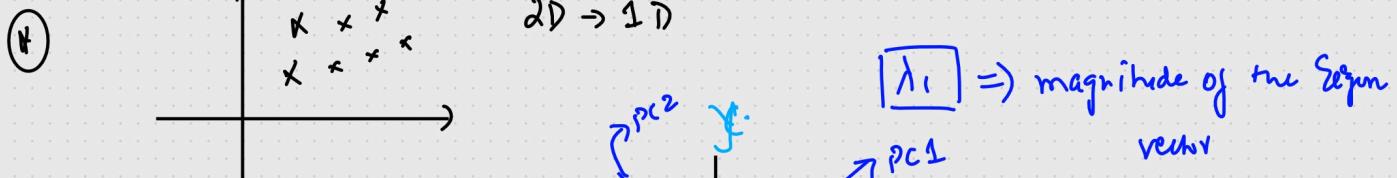
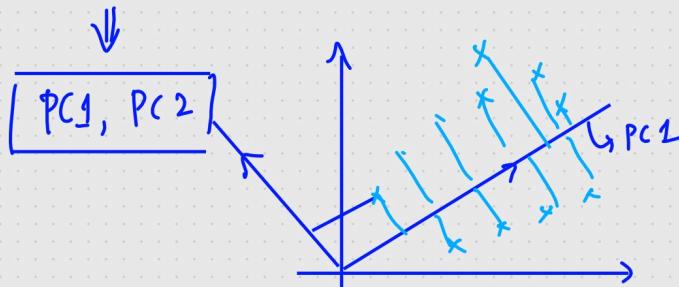
$$\text{Cov}(Y, Y) = \text{Var}(Y)$$

X Y Z

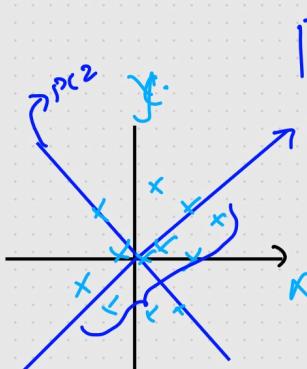
X	Y	Z

y
 π


 $f_1 \text{ & } f_2$
 $A \cdot v = \lambda \cdot v$
 $\lambda_1, \lambda_2, \lambda_3$
 λ_1, λ_2
 $\downarrow \quad \downarrow$
 $PC_1 \quad PC_2$
 $\lambda_1, \lambda_2 \rightarrow \text{Eigen values}$



① Standardize the data



② Covariance Matrix of X & Y

$$A = \begin{matrix} X & Y \\ X & \begin{array}{|c|c|} \hline \text{Var}(X) & \text{Cov}(X,Y) \\ \hline \text{Cov}(Y,X) & \text{Var}(Y) \\ \hline \end{array} \end{matrix} \quad 2 \times 2$$

③ Find out Eigen vectors And value

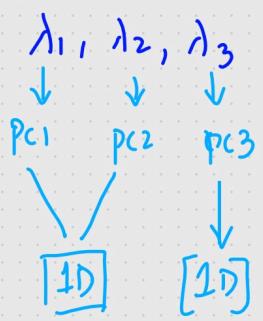
$$A v = \lambda v$$

$$\begin{bmatrix} \lambda_1, \lambda_2 \end{bmatrix} \Rightarrow \text{Eigen values.}$$

$$\downarrow \quad \downarrow$$

$$PC_1 \quad PC_2$$

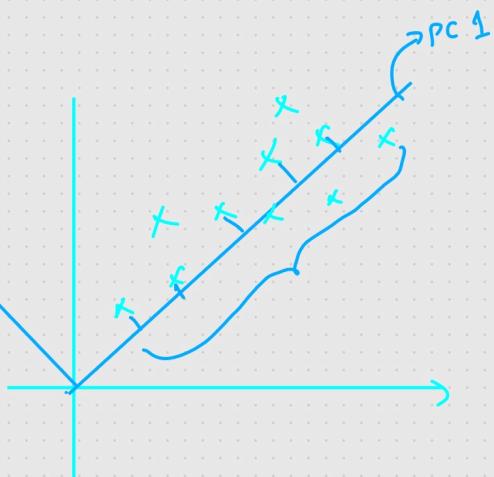
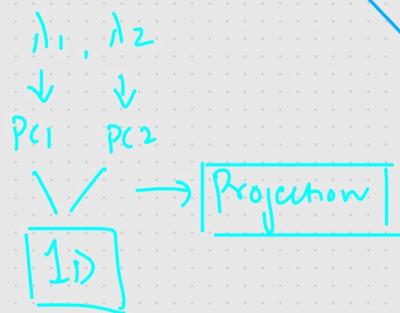
$3D \rightarrow 2D$



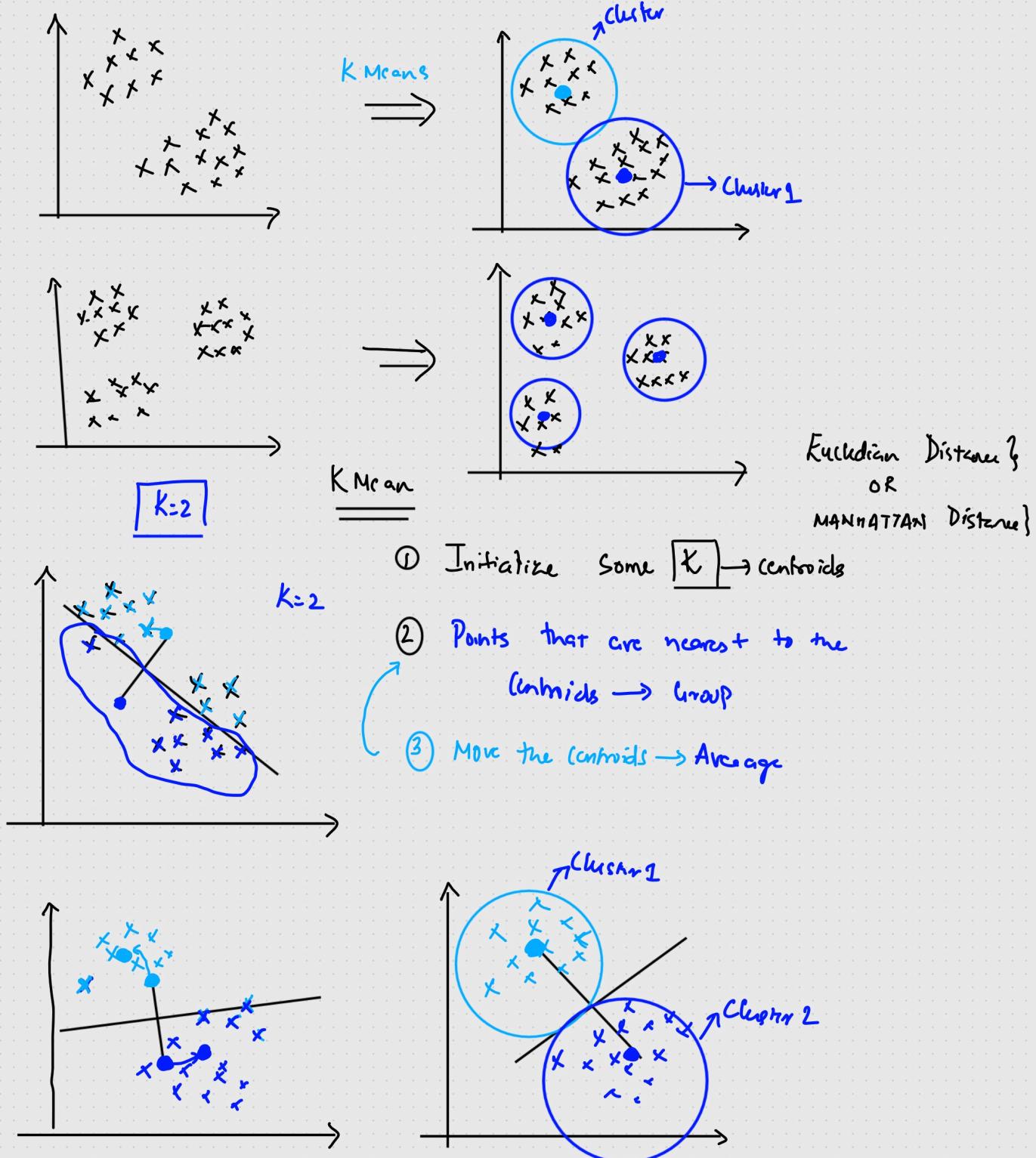
$3D \rightarrow 1D$



$2D \rightarrow 1D$



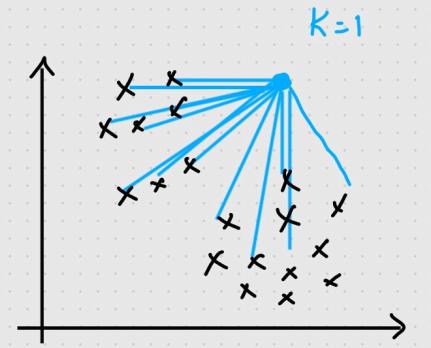
K Means Clustering Algorithm



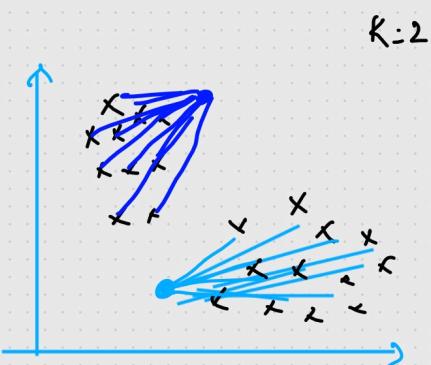
How do we select the K value?

WCSS = Within cluster sum of squares

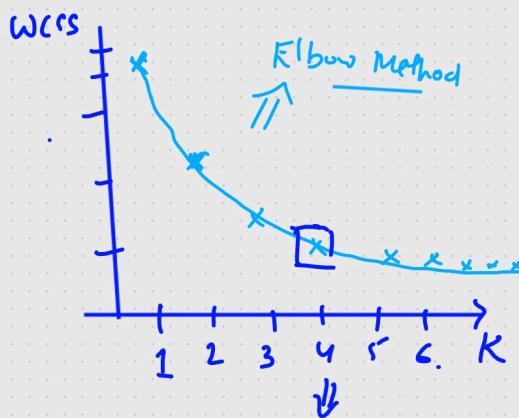
Initialize K=1 to 20



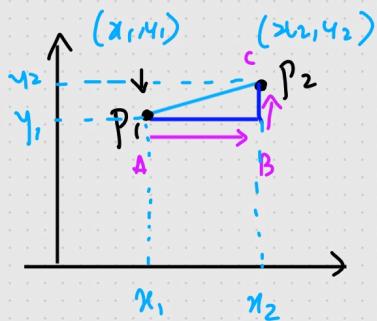
$$WCSS = \sum_{i=1}^n \left(\text{distance between points to nearest centroid} \right)^2$$



WCSS ↓



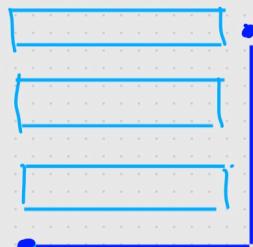
f) Euclidean Distance



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

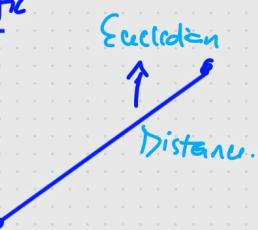
$$\text{Manhattan dist} = |x_2-x_1| + |y_2-y_1|$$

IRON MAN → U.S

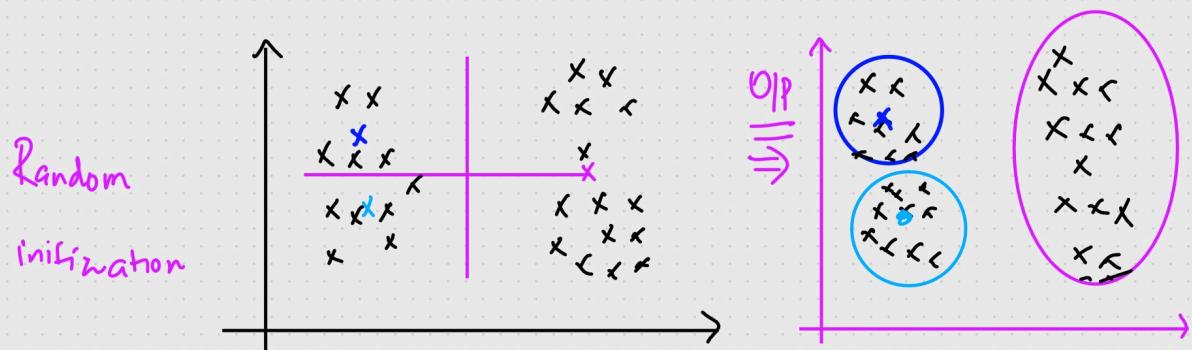
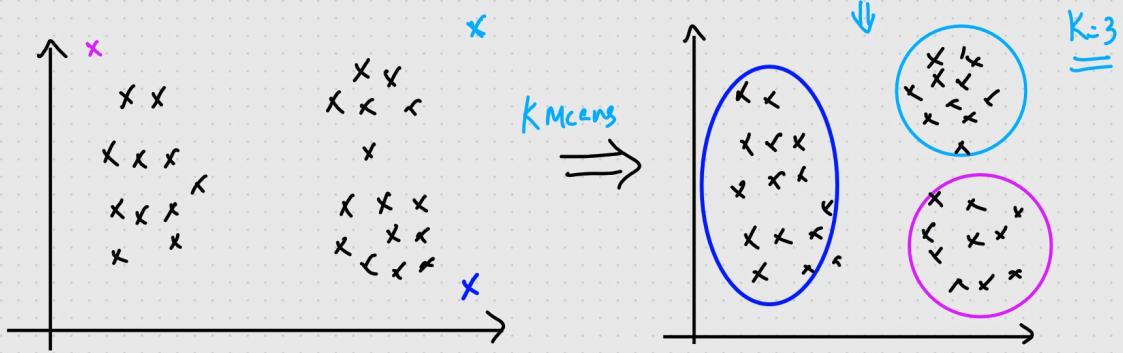


⇒ Manhattan distance

Air Traffic

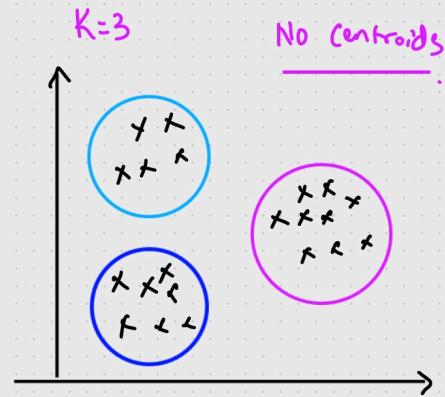
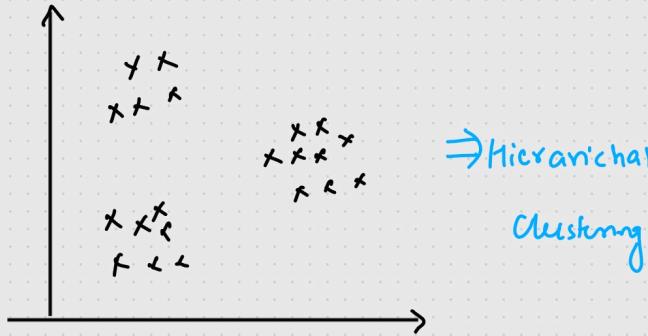


Random Initialization TRAP (Kmeans++)



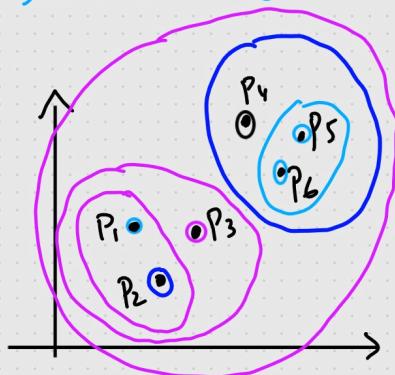
Kmeans++ Initialization Technique

Hierarchical Clustering



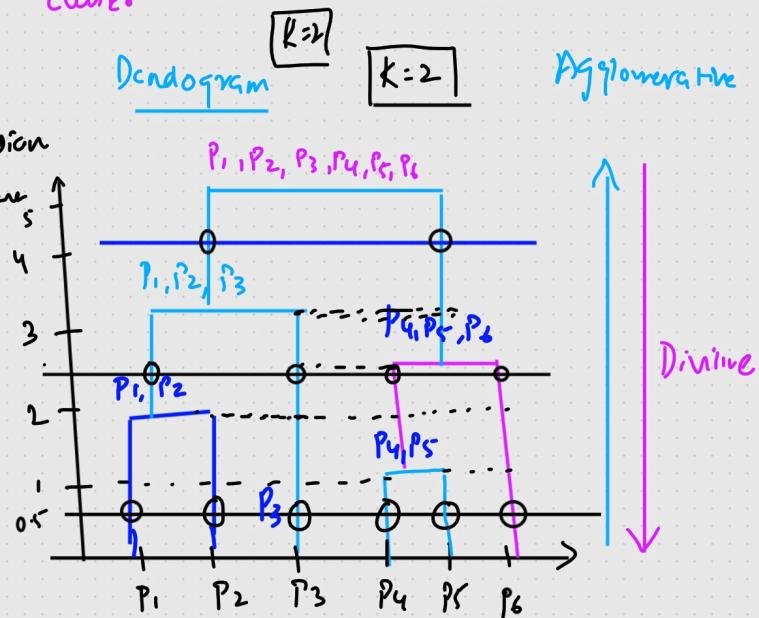
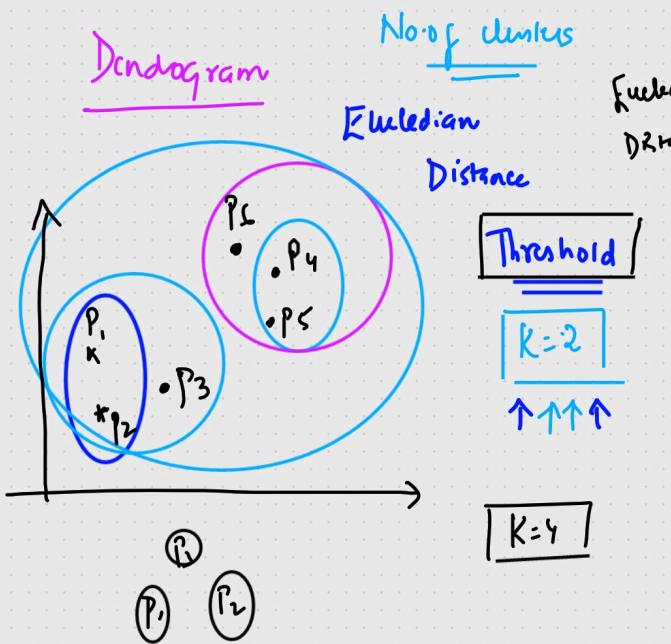
HC

- ① Agglomerative
 - ② Divisive
- } Geometric Intuition.



Steps

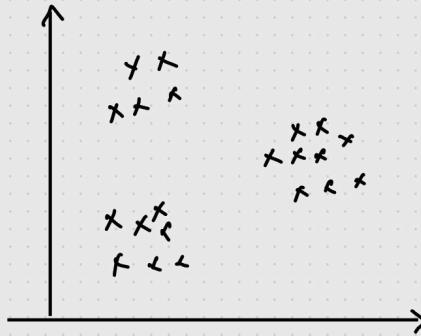
- ① For each point initially will consider it as a separate cluster
- ② Find the nearest point and create a new cluster
- ③ Keep on doing the same process until we get a single cluster



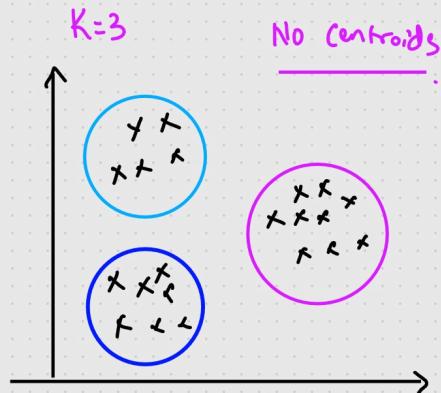
(k) Select the longest vertical line such that }
no horizontal line passes through it }

Threshold {Euclidean Distance} .

Hierarchical Clustering



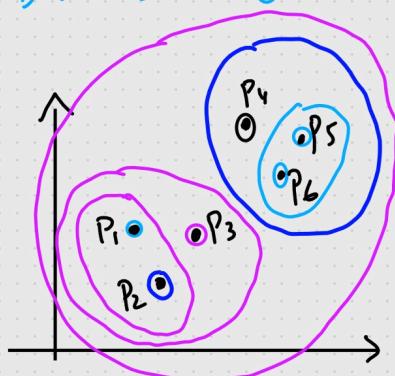
→ Hierarchical
Clustering



HC

- ① Agglomerative
- ② Divisive

⇒ Geometric Intuition.



Steps

- ① For each point initially will consider it as a separate cluster
- ② Find the nearest point and create a new cluster
- ③ Keep on doing the same process until we get a single cluster

Cosine Similarity

No. of clusters

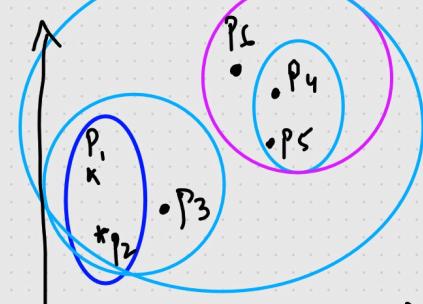
Dendrogram

Euclidean
Distance

Threshold

$K=2$
↑↑↑↑

$K=4$



Fusion

Distance

S

4

3

2

1

0.5

0

+

P₁ P₂ P₃ P₄ P₅ P₆

P₁, P₂, P₃

P₄, P₅, P₆

P₃

P₄

P₅

P₆

P₁

P₂

P₃

P₄

P₅

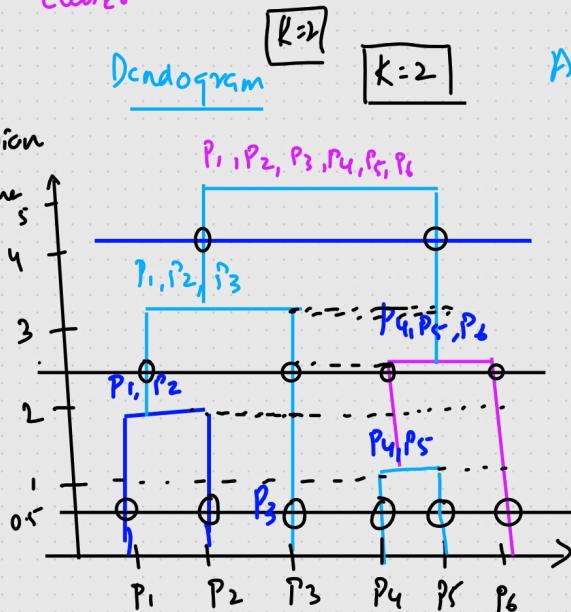
P₆

Dendrogram

$K=2$

$K=2$

Agglomerative



Divisive

(K) Select the longest vertical line such that }
no horizontal line passes through it }

Threshold {Euclidean Distance} .

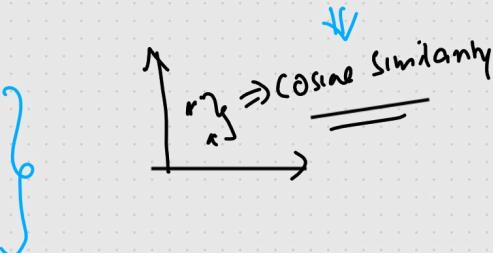
K Means Vs Hierarchical Clustering

Scalability And Flexibility

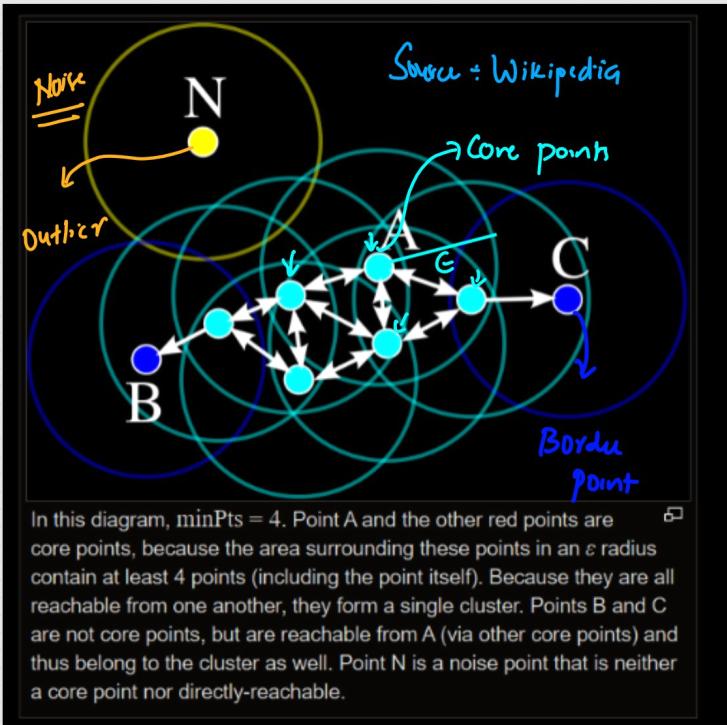
- ① Dataset size → Huge → K Means
Small → Hierarchical Clustering



- ② Kmean → Numerical data
Hierarchical clustering → Variety of data.
- ③ Centroids → Elbow method → No. of centroids
→ No. of clusters



DBSCAN CLUSTERING.



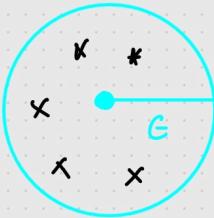
● → Core point
● → border point
● → Outlier

● → Non linear
Clustering

$$\text{minpts} = 4 \quad \epsilon = \text{radius}$$

Core point

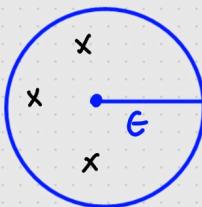
- ① No. of points within the ϵ should be greater ≥ 4



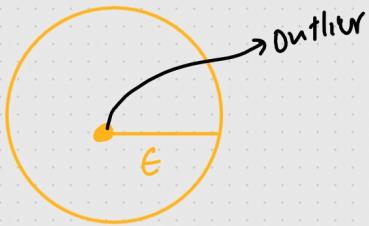
$$\text{minpts} = 4$$

Border point

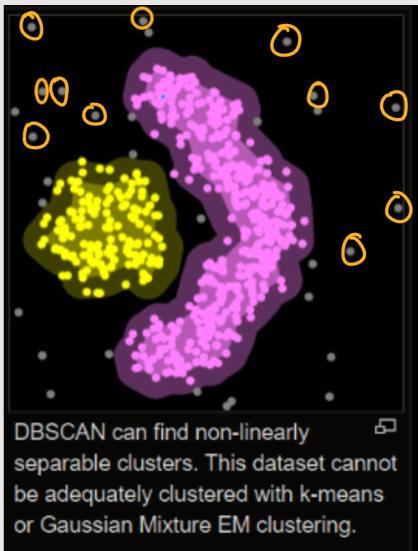
no. of data points within this radius will be less than minpts



Outlier (Noise)



Some Examples after we apply DBScan Clustering



The left image depicts a more traditional clustering method that does not account for multi-dimensionality. Whereas the right image shows how DBSCAN can contort the data into different shapes and dimensions in order to find similar clusters.

Silhouette (clustering)

①

For data point $i \in C_I$ (data point i in the cluster C_I), let

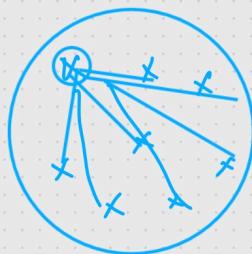
$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i, j)$$

$a(i)$



be the mean distance between i and all other data points in the same cluster, where $|C_I|$ is the number of points belonging to cluster i , and $d(i, j)$ is the distance between data points i and j in the cluster C_I (we divide by $|C_I| - 1$ because we do not include the distance $d(i, i)$ in the sum). We can interpret $a(i)$ as a measure of how well i is assigned to its cluster (the smaller the value, the better the assignment).

$a(i)$



②

We then define the mean dissimilarity of point i to some cluster C_J as the mean of the distance from i to all points in C_J (where $C_J \neq C_I$).

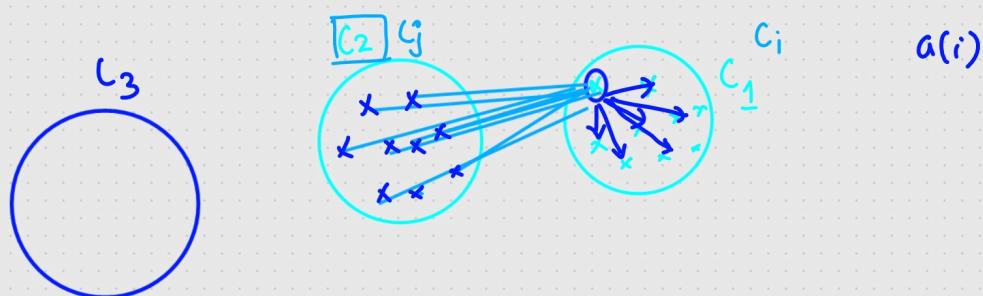
For each data point $i \in C_I$, we now define

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i, j)$$

$a(i) < b(i)$

to be the *smallest* (hence the `min` operator in the formula) mean distance of i to all points in any other cluster, of which i is not a member. The cluster with this smallest mean dissimilarity is said to be the "neighboring cluster" of i because it is the next best fit cluster for point i .

$a(i)$



② Silhouette Score

We now define a *silhouette* (value) of one data point i

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_I| > 1 \quad [-1 \text{ to } 1]$$

and

$$s(i) = 0, \text{ if } |C_I| = 1$$

Which can be also written as:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

From the above definition it is clear that

$$\{-1 \leq s(i) \leq 1\}$$

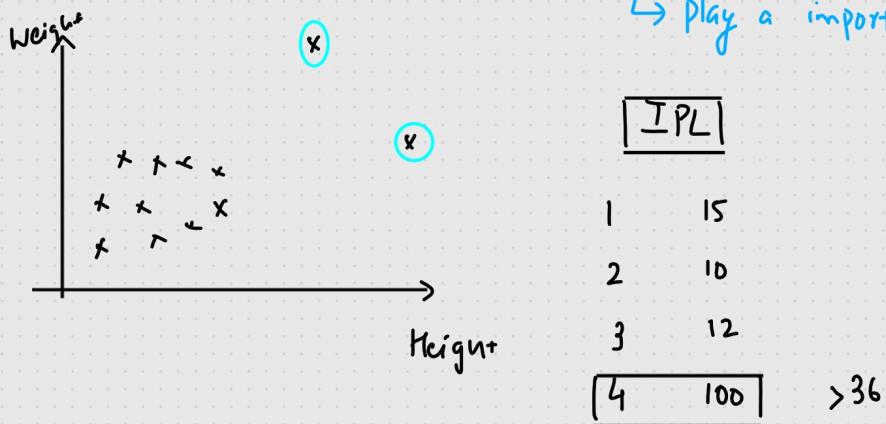
More near to 1 better

clustering model we

have created

Anomaly Detection [To detect Outliers]

↳ play a important role

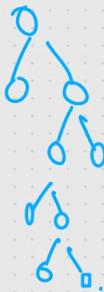
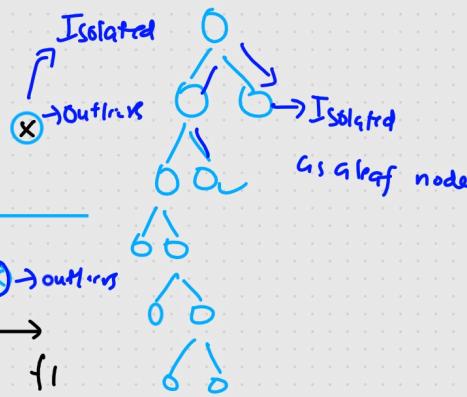
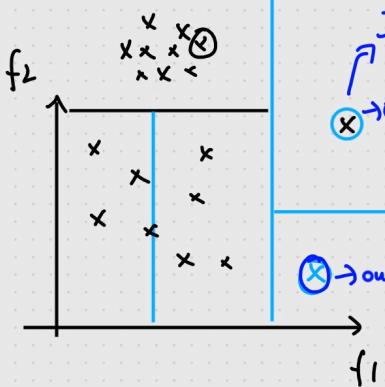


① Isolation Forest [Decision Trees].

Many Trees

Isolated Trees

$$\begin{matrix} f_1 & f_2 & f_3 & f_4 \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{matrix}$$



Anomaly Score

Mathematical Formula : Compute anomaly score. for a new point

$$S(x, m) = 2^{-\frac{E(h(x))}{c(m)}}$$

m = no. of data points
 x = data point.

$E(h(x))$ = Average search depth for x from the isolate tree.

$c(m)$ = Average depth of $h(x)$

[Threshold > 0.5]

$E(h(x)) \ll c(m) \Rightarrow S(x, m) \approx 1 \Rightarrow$ Anomaly score \Rightarrow Outliers

$E(h(x)) \gg c(m) \Rightarrow S(x, m) \approx 0.5 \Rightarrow$ Normal data point.

Local Outlier factor Anomaly Detection.

