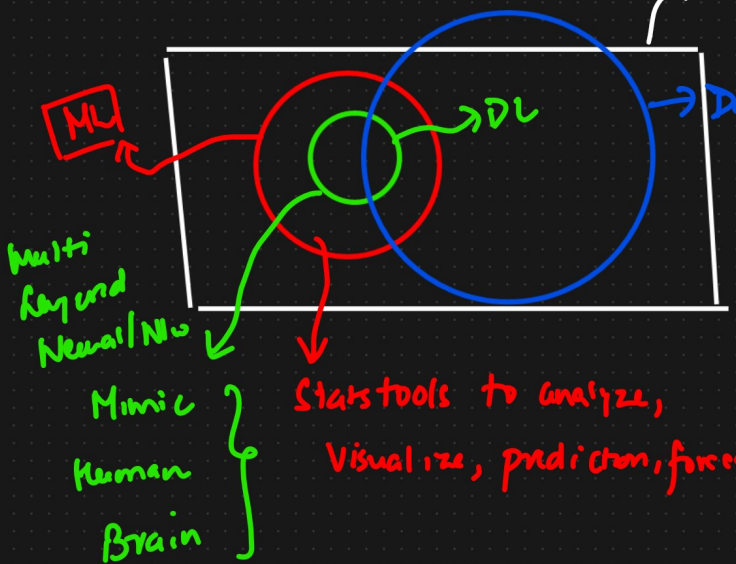
7 days Machine Learning Algorithms

Purpose : Clear the Interviews

Agenda

1. Introduction to ML (AI Vs ML Vs DL Vs DS)
2. Supervised ML and Unsupervised ML
3. Linear Regression (Maths & Geometric Intuition)
4. R^2 & Adjusted R^2
5. Ridge and Lasso Regression

① AI Vs ML Vs DL Vs DS



AI application

AI application is able to do it own task without any human intervention

Eg : Netflix → Action → Recommendation
↳ Comedy →

Amazon.in → Iphone → Headphones }

Self Driving Cars →

Machine & Deep learning

- Supervised ML
  - Regression
  - CLASSIFICATION
- Unsupervised ML → CLUSTERING
  - DIMENSIONALITY REDUCTION

Reinforcement

Supervised ML

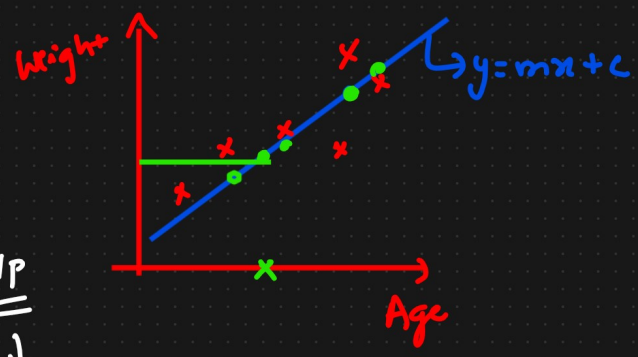
Age	Weight
24	62
25	63
21	72
27	62

O/P

Age → hypothesis → O/P → Weight

Independent features → Age

Dependent feature → Weight



① Regression Problem

Age	Weight
24	72
23	71
25	71.5
—	—

O/P → Continuous variable → Regression Problem

② CLASSIFICATION

Binary classification
Multi class classification

O/P

No. of hours	No. of play hours	No. of sleep	P/F
—	—	—	P
—	—	—	F
—	—	—	P
—	—	—	F

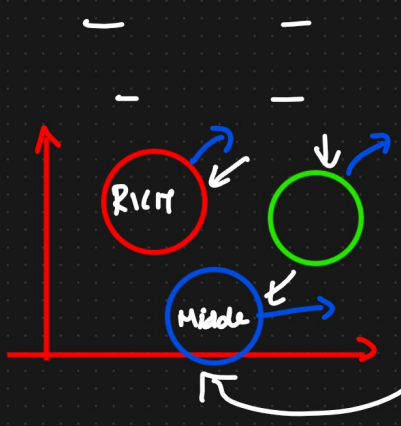
② Unsupervised ML

- Clustering
- Dimensionality Reduction

Salary	Age
—	—
—	—

{No Dependent variable}

Clustering → Customer Segmentation



Ad marketing

GROUPING

Customer Segmentation

Eg: Product 1
↳ Rich

Product 2
↳ Middle

② Dimensionality Reduction

1000 → lower Dimension
100

PCA, LDA

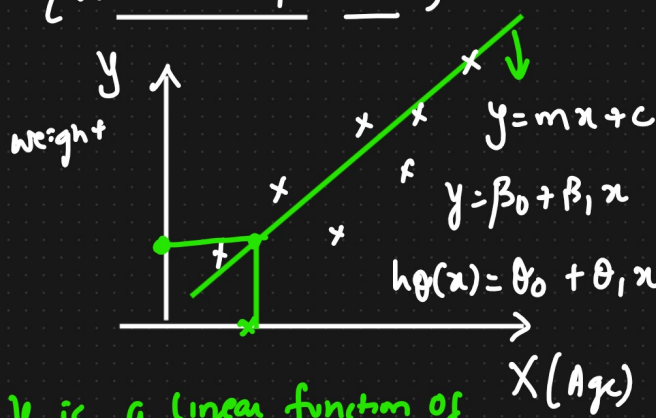
Supervised

① Linear Regression
② Ridge & Lasso
③ Logistic Reg
④ Decision Tree
⑤ AdaBoost
⑥ Random Forest
⑦ Gradient Boosting
⑧ Xgboost
⑨ Naive Bayes
⑩ SVM
⑪ KNN

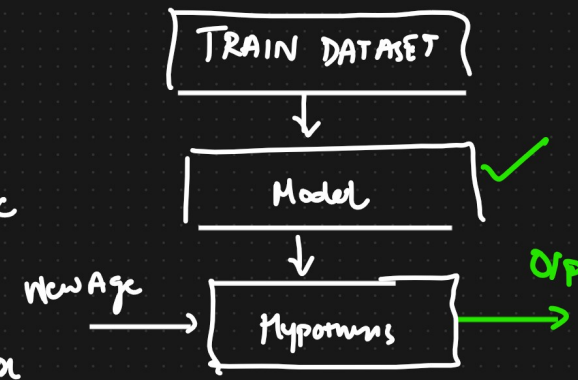
Unsupervised

① K means
② DBScan
③ Hierarchical
④ K Nearest Neighbor Class
⑤ PCA
⑥ LDA

① {Linear Regression}



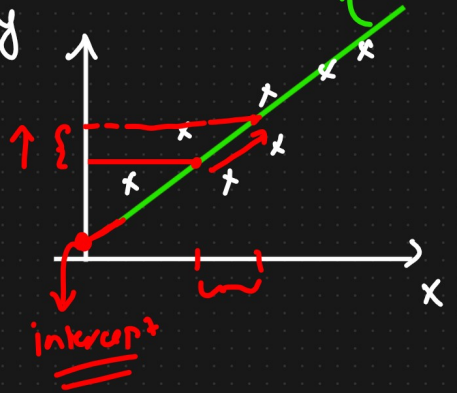
y is a linear function of x



Credits : Andrew NG

Equation of a Straight line

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



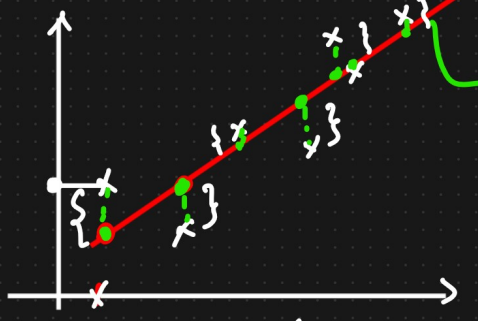
When $x = 0$

θ_0 = Intercept

θ_1 = Slope or Coefficient

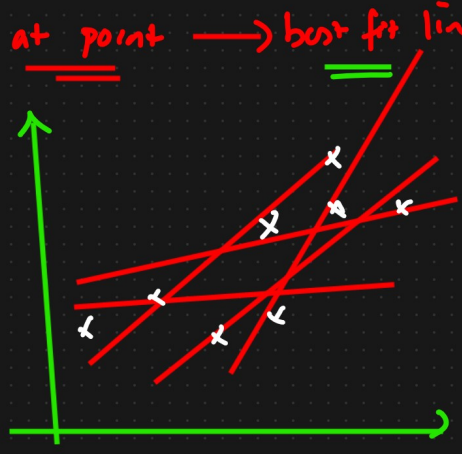
x_i = data points

Linear Regression



Minimal ↓↓↓↓

→ Best fit line



Hypothesis

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

{Derivation} → purpose

$$x^n = nx^{n-1}$$

$$\frac{\partial (x^2)}{\partial x} = \frac{2x}{x}$$

Cost function

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

→ Cost function

↳ Squared Error Function

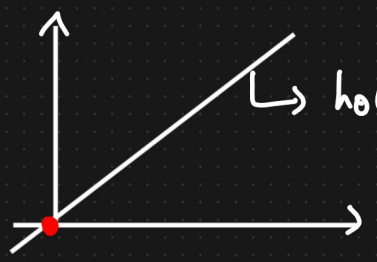
What we need to solve

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

$$\Downarrow$$

$$\underset{\theta_0,\theta_1}{\text{minimize}} \quad J(\theta_0,\theta_1)$$

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

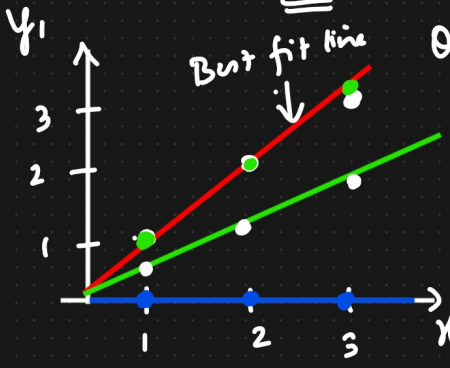


$h_\theta(x) = \theta_1 x$ ← slope

$\theta_1 = 1$ (1, 1)

$\theta_1 = 0.5$ (2, 2)

(3, 3)



$\theta_1 = 0$

$\theta_1 = 2$

$\theta_1 = 1$

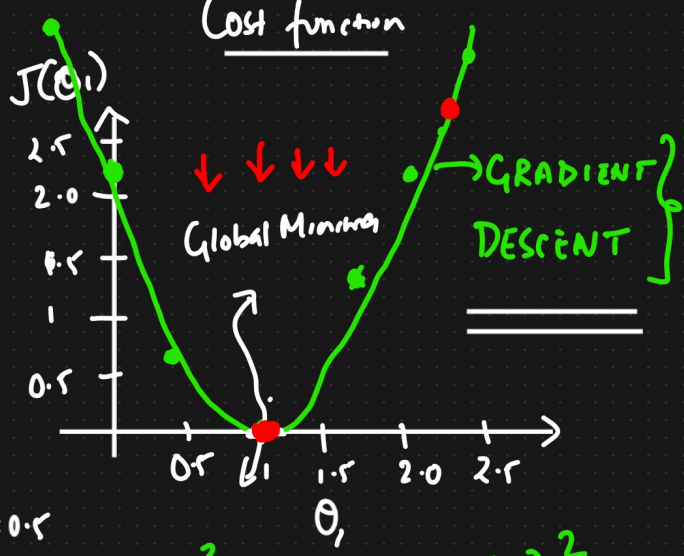
$J(\theta_1) = 0$

$$J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{3}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

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

$$J(\theta_1) = 0$$

Cost function



$\theta_1 = 0.5$

$$J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{3}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

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

$$= \frac{1}{2\times 3}\left[0.25 + 1 + 2.25\right] \approx 0.58$$

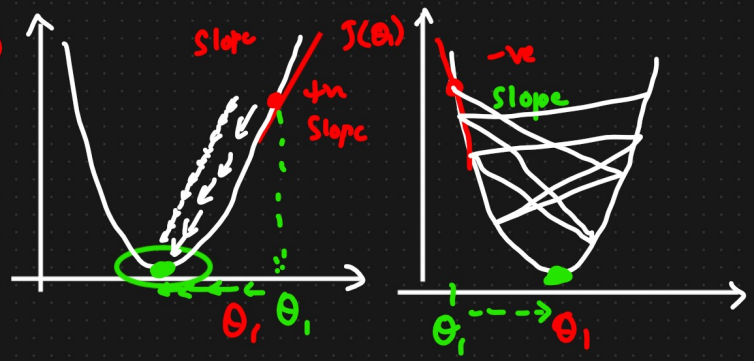
$$J(\theta_1) = \frac{1}{2m}\left[(0-1)^2 + (0-2)^2 + (0-3)^2\right]$$

$$= \frac{1}{6}\left[1+4+9\right]$$

$$\approx 2.3$$

$\alpha \Rightarrow$ Huge

$\alpha = 0.01$



Convergence Algorithm

Repeat until convergence

{

$$\theta_j := \theta_j - \alpha \underbrace{\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)}_{\text{derivative (slope)}}$$

}

α : Learning Rate

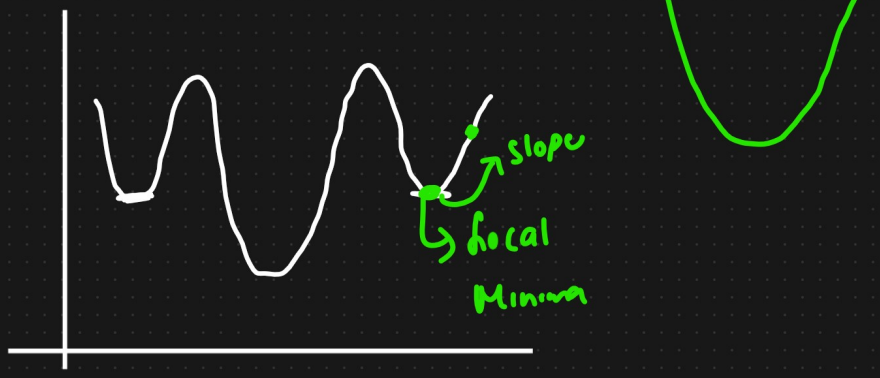
$\theta_1 := \theta_1 - \alpha(+ve)$

$\theta_1 := \theta_1 - \alpha(-ve)$

$\theta_1 := \theta_1 + \alpha(ve)$

$\theta_1 := \theta_1 - \alpha(0)$

$\theta_1 := \theta_1$



GRADIENT DESCENT Algorithm

Repeat until convergence

{

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

}

$j = 0$ and 1

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

$\frac{1}{2m}$ x^2 ✓ $\frac{2}{2m}$ x

Convergence Algorithm:

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

$$\begin{cases} j=0 \Rightarrow \dfrac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \dfrac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \\ j=1 \Rightarrow \dfrac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \dfrac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x^{(i)} \end{cases}$$

$x^2 \Downarrow 2x$

$\frac{\partial}{\partial \theta_1} (\theta_1 x) = x$

$\{\alpha = 0.001\}$ α = Learning Rate

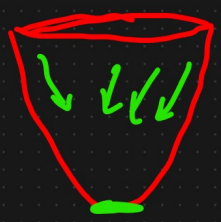
Repeat until convergence

{

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

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

}



Performance Metrics

R^2 and Adjusted R^2

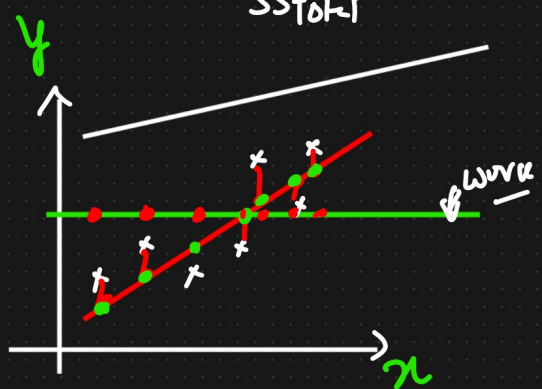
-ve ???

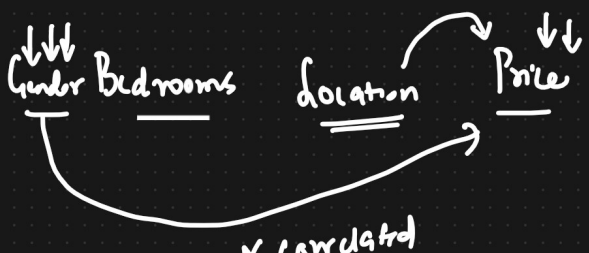
$h_\theta(x)$

$$R^2 = 1 - \frac{SS_{Res}}{SS_{Total}} = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} = 1 - \frac{\text{Low}}{\text{High}}$$

Small number = Big number = 90%

Higher





$R^2 = 85\%$

2 $R^2 = 90\%$

3 $R^2 = 91\%$ ✓ → Model

Adjusted R^2

p = features or predictors

$$R^2_{adjusted} = 1 - \frac{(1-R^2)(N-1)}{N-p-1}$$

(✓ ↑ Big ✓)

$p=2 > p=3$

N = No. of datapoint

p = No. of predictors

$p >>>$

$p=2$	$R^2 = 90\%$	$R^2_{adjusted} = 86\%$ ↓↓
$p=3$	$R^2 = 91\%$ ↑↑	$R^2_{adjusted} = 82\%$ ↓↓
	R^2 ↑↑↑	82%

86%

Day 2 – Live Machine Learning Algorithm

Agenda

1. Ridge and Lasso Regression
2. Assumption of Linear Regression
3. Logistic Regression
4. Confusion Matrix
5. Practicals Implementation

① Ridge And Lasso Regression

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

$\theta_0 : 0$

$J(\theta_0, \theta_1) = 0$ ↓↓↓

{Training data}

Overfitting

✓ (Low Bias)

Underfitting { High Bias, High Variance }

① Model Accuracy is bad with Training data

② Model Accuracy is also bad with Test data

Model performs well → Training data

Fails to perform well → Test Data ✓ (High Variance)

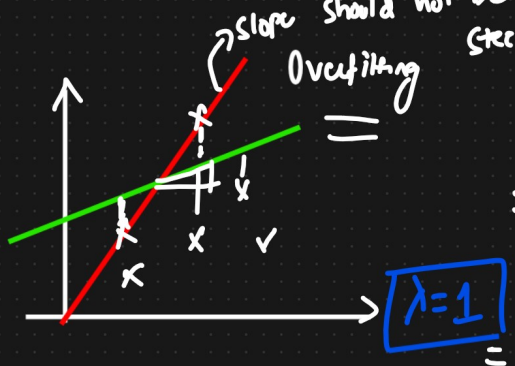
Model 1

Traing Acc = 90%

Test Acc = 80%

⇓

{ Overfitting
Low Bias, High Variance }



$\lambda = 1$

Model 2

Traing Acc = 92%

Test Acc = 91%

⇓

{ Generalized Model }

{ Low Bias
Low Variance }

$J(\theta_1) = 0$

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

$= (\hat{y}^i - y^{(i)})^2 + \lambda(\text{slope})^2$ ✓

$= 0 + 1(2)^2$

$= 4$ // ⇓⇓ bb

$= (\hat{y}^{(i)} - y^{(i)})^2 + \lambda(\text{slope})^2$

⇓

(Small value) $+ 1(1.5)^2$

$=$ (Small value) $+ 2.25$

≈ 3 ⇓⇓

Model 3

Traing Acc = 70%.

Test Acc = 65%

⇓

Underfitting

High Bias, High Variance

$h_\theta(x) = \hat{y}$ $\theta_1 = 2$ $\theta_0 = 0$

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

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

↳ slope

Ridge (L2 Regularization)

$\theta_1 \Rightarrow$

{ Prevent Overfitting }

situations { Hyperparameter }

R^2, adjusted R^2

$\lambda \rightarrow$ Hyperparameter ✓

Convergence

{ }

Lasso (L1 Regularization)

$= (\hat{y} - y)^2 + \lambda |\text{slope}|$ → feature selection

$|\theta_0 + \theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5 \cdots + \theta_n|$

$h_\theta(x) = \hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \cdots + \theta_n x_n$

① Preventing Overfitting } → { L1 Regulation }
② Feature Selection } → { L2 Regulerization }

{ λ → Cross validation }

⇓
Performance metrics
⇓

Ridge Regression (L2 Norm)

Cost function = $(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda (slope)^2$

④

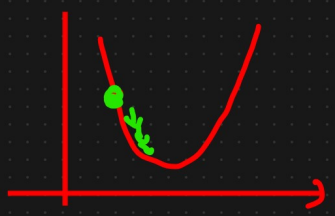
Purpose : Preventing Overfitting

$|\theta_0 + \theta_1 + \theta_2 + \theta_3 \dots + \theta_n|$

Lasso Regression (L1 Reg)

Cost function = $(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda |slope|$

Purpose : 1) Prevent Overfitting
2) Feature Selection



Assumption of Linear Regression

① Normal / Gaussian Distribution → Model will get trained well

✓ ② {Standardization {Scaling data} → Z-score $\mu = 0, \sigma = 1$}

③ Linearity

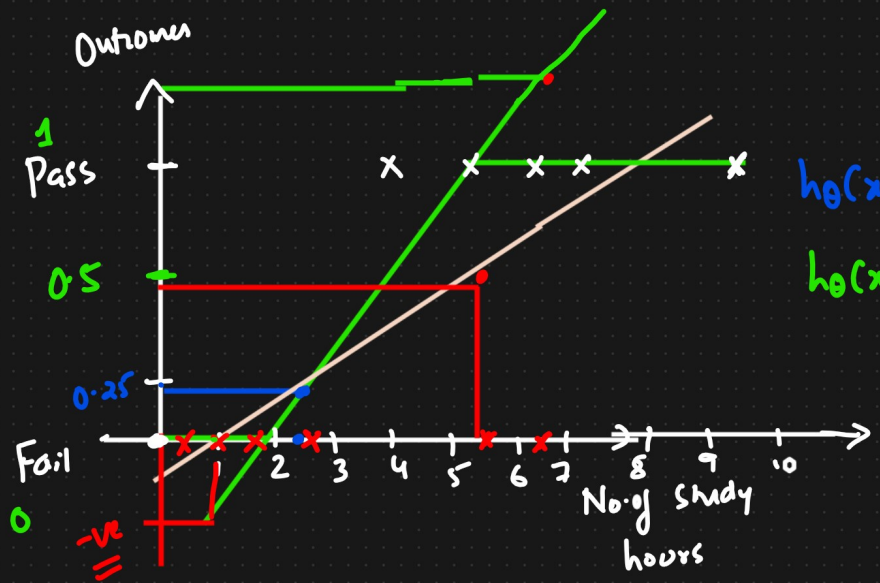
X_3 [X_1 X_2] [Y] (95%)

Variation Inflation factor?

④ Multi Collinearity

Logistic Regression (classification) → Binary classification

No. of study	No. of play	P/F
—	—	P
—	—	F



Linear Regression ?? ✓

$h_\theta(x) < 0.5 \Rightarrow 0 \rightarrow$ Fail

$h_\theta(x) \geq 0.5 \Rightarrow 1 \rightarrow$ PASS ✓

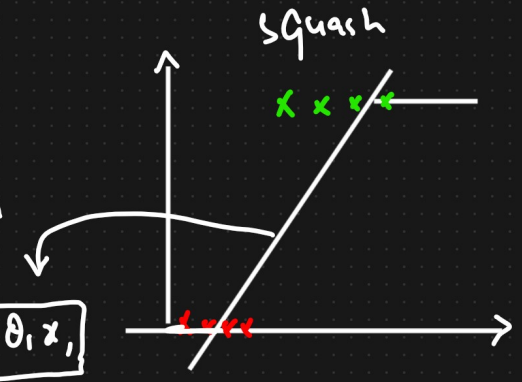
{Sigmoid function}

0 to 1 ✓

Decision Boundary Logistic Regression

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 - \cdots + \theta_n x_n$$

$$\boxed{h_\theta(x) = \theta^T x}$$



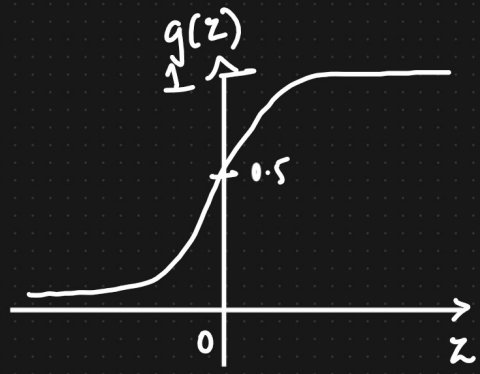
$$h_\theta(x) = \boxed{\theta_0 + \theta_1 x_1}$$

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

Let $z = \theta_0 + \theta_1 x$

$h_\theta(x) = g(z)$

$$h_\theta(x) = \frac{1}{1+e^{-z}}$$ → Sigmoid or Logistic function



$g(z) \geq 0.5$ when $z \geq 0$ ✓

$$\boxed{h_\theta(x) = \frac{1}{1+e^{-(\theta_0+\theta_1 x)}}}$$

Training Set

$\{(x^1, y^1), (x^2, y^2), (x^3, y^3) \dots (x^n, y^n)\}$

$y \in \{0, 1\} \rightarrow$ 2 o/p

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

$z = \theta_1 x$

Change parameter θ_1 ?

Cost function

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

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

Logistic Regression

$$h_\theta(x) = \frac{1}{1 + e^{-(\theta_1 x)}}$$

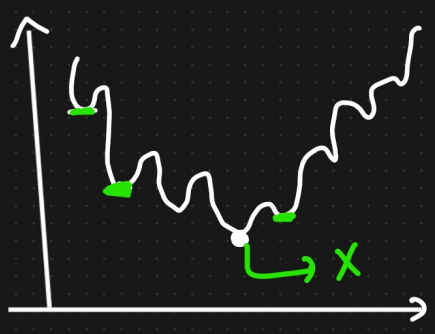
Logistic Regr Cost function $= \frac{1}{2} (h_\theta(x^{(1)}) - y^{(1)})^2$

We cannot use this cost function for Logistic

$$h_\theta(x) = \frac{1}{1 + e^{-(\theta_1 x)}}$$

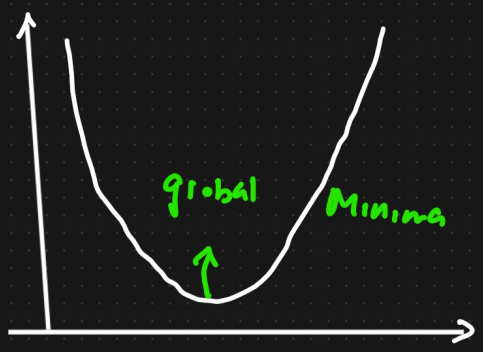
Gradient Descent

Non convex function



Gradient Descent

Convex function

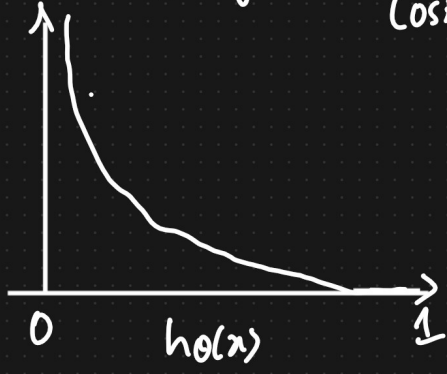


Logistic Regression Cost function

$$h_\theta(x) = \frac{1}{1+e^{-(\theta_1 x)}}$$

$$J(\theta_1) = \begin{cases} -\log(h_\theta(x^i)) & y=1 \\ -\log(1-h_\theta(x^i)) & y=0 \end{cases}$$

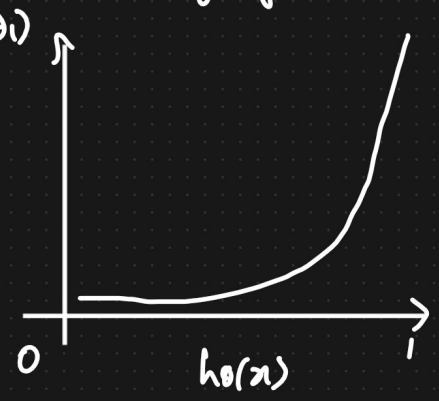
$J(\theta_1)$ if $y=1$



Cost = 0 if $y=1$, $h_\theta(x)=1$

Classification

$J(\theta_1)$ if $y=0$



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

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

$\Downarrow$ cost function.

if $y=1$

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

if $y=0$

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

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

↓ cost

$$h_\theta(x^i) = \frac{1}{1+e^{-\theta_1 x}}$$

Repeat until convergence

→ {

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

}

Performance Metrics {Classification problem}

x_1	x_2	Actual y	Pred $\hat{y}$ ← o/p
–	–	0	1
–	–	1	1
–	–	0	0
–	–	1	1
–	–	1	1
–	–	0	1
–	–	1	0

{Actual}

Pred	1	0
1	3	2
0	1	1

Predicted \ Actual	1	0
1	TP	FP↓
0	FN↓	TN

→ Confusion matrix

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

② 0 → 900, 1 → 100 } Imbalanced Dataset

$$= \frac{3+1}{3+2+1+1} = \frac{4}{7}$$

$$= 0.57 = 57\%$$

0 → 600, 1 → 400 } Balanced Data

0 : 900, 1 : 100 } Biased

{ Model → 0 = $\frac{900}{1000} = 90\%$ }

. TPR, Sensitivity

① Precision

$$\frac{TP}{TP + FP}$$

② Recall

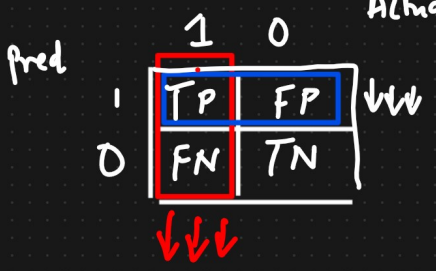
$$\frac{TP}{TP + FN}$$

③ F-Score.

{ Tom Stock market is going to crash } → Precision

{Spam classification} → Precision

{ HAS CANCER OR NOT } → Recall



$$F\text{-}\beta\text{eta} = (1+\beta^2)\frac{\text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}$$

$$\beta = 1 \quad = (1+1)\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1-Score

$$= \frac{2(\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

Harmonic mean

$$= \frac{2xy}{x+y}$$

$$\beta = 0.5 \quad (1+(0.5)^2)\frac{P \times R}{(0.25)P + R}$$

F0.5 Score

$$\beta = 2$$

FN >> FP.

F2 Score

3rd Day → Machine Learning Algorithms

Agenda

① Practicals → Simple Examples
② Naive Bayes Intuition
③ KNN algorithms

Previous Session

① Linear Regression
② Ridge & Lasso
③ Logistic Regression

⇒ Complex

① Naive Bayes Intuition {Classification}

⇓

{BAYES THEOREM}

Rolling a Dice

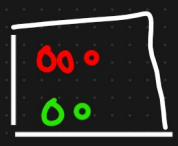
{1, 2, 3, 4, 5, 6}

$P(1) = \frac{1}{6}$ $P(3) = \frac{1}{6}$

$P(2) = \frac{1}{6}$

{Independent Events}

Dependent Event



First Event ✓

$P(R) = \frac{3}{5}$ → R ✓

Given Marble

$P(G) = \frac{2}{4} = \frac{1}{2}$ → G

Dependents

$P(G) = \frac{2}{5}$

$P(R) = \frac{3}{4}$

$P(R \text{ and } G) = P(R) * P(G/R)$ → Conditional probability

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

$\Rightarrow$ P(A and B) = P(B and A) {Yes}

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

$$P(B/A) = \frac{P(B) * P(A/B)}{P(A)}$$

Bayes Theorem

(RUX Naive Bayes

I/P

X_1 X_2 X_3 X_4 X_5 ... X_n → y o/p

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

Yes or No

$$= \frac{P(y) * P(x_1/y_1) * P(x_2/y_2) * P(x_3/y_3) \cdots P(x_n/y_n)}{P(x_1) * P(x_2) * P(x_3) * \cdots P(x_n)}$$

DATASET

x_1	x_2	x_3	x_4	y
				Yes ✓
				No ✓

$$P(y=Yes/x_i) = \frac{P(Yes) * P(x_1/Yes) * P(x_2/Yes) * P(x_3/Yes) * P(x_4/Yes)}{P(x_1) * P(x_2) * P(x_3) * P(x_4)}$$

Constant → Ignore # fixed

$$P(y=No/x_i) = \frac{P(No) * P(x_1/No) * P(x_2/No) * P(x_3/No) * P(x_4/No)}{P(x_1) * P(x_2) * P(x_3) * P(x_4)}$$

Constant → # fixed

x_i → Yes / No

$P(Yes|x_i) = 0.13$ $\quad P(No|x_i) = 0.05$

$\Downarrow$ $\quad \geq 0.5 \Rightarrow 1$

$< 0.5 \Rightarrow 0$

$$P(Yes|x_i) = \frac{0.13}{0.13+0.05} = 0.72 = 72\%$$

$P(No|x_i) = 1 - 0.72 = 0.28 = 28\%$

DATASET

Binary class

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

x_1 Outlook — $P(Sunny/Yes)$

	Yes	No	P(y)	P(No)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rain	3	2	3/9	2/5
Total	9	5		

Temperature

	Yes	No	P(y)	P(N)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cold	3	1	3/9	1/5
Total	9	5		

PLAY

		P(Yes)	P(N)
Yes	9	9/14	5/14
No	5		
Total	14		

→ Test (Sunny, Hot) → O/P

$$P(Yes|(Sunny, Hot)) = P(Yes) * P(Sunny/Yes) * P(Hot/Yes)$$

$P(\text{Sunny}) * P(\text{Hot})$

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

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

$$P(\text{No} \mid \text{Sunny, Hot}) = \frac{P(\text{No}) * P(\text{Sunny} \mid \text{No}) * P(\text{Hot} \mid \text{No})}{P(\text{Sunny}) * P(\text{Hot})} \rightarrow \text{Constant}$$

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

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

$P(\text{Yes} \mid \text{Sunny, Hot}) = 0.031 \quad = 1 - 0.73 = 0.27 = 27\%$

$P(\text{No} \mid \text{Sunny, Hot}) = 0.085 = \frac{0.085}{0.031 + 0.085} = 0.73 = 73\%$

$\rightarrow$ (Sunny, hot) $\longrightarrow$ Yes or No

Answer $\rightarrow$ No ✓

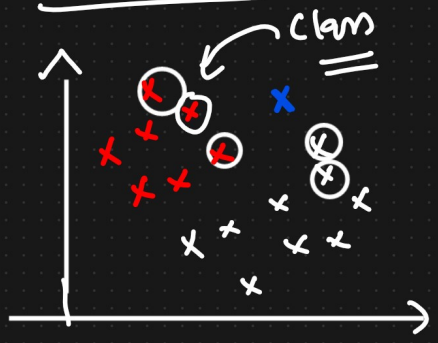
Assignment

('Overcast', 'Mild') $\longrightarrow$ Naive Bayes ?

② KNN Algorithm {K Nearest Neighbour}

- Classification
- Regression

① Classification



K Nearest Neighbour

K = 5

Maximum No

Red = 3

White = 2

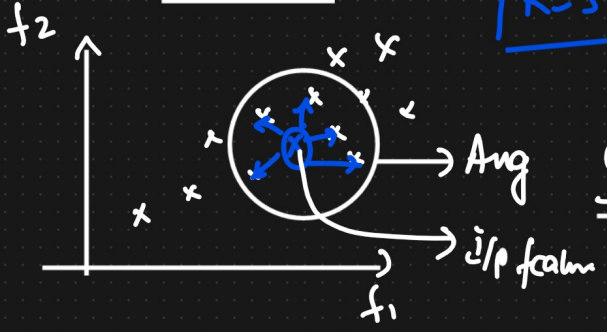
K = 1 to 20

↑

K = 5 → Hyperparameter

Error Rate ↑↑.

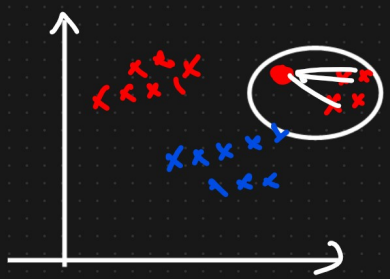
Regression



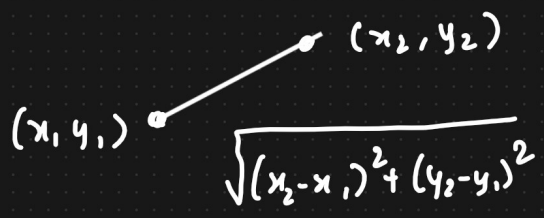
K Nearest

① Outliers

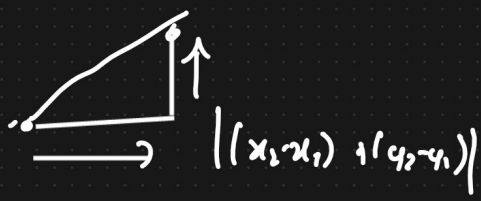
② Imbalanced



Euclidean Distance ✓



Manhattan Distance ✓



$|(x_2 - x_1)(y_2 - y_1)|$

Day 4 - Machine Learning Algorithms

Agenda

1. Decision Tree CLASSIFICATION
2. DECISION TREE REGRESSION
3. PRACTICAL IMPLEMENTATION
4. Ensemble Techniques

Agenda

{DAY 1, DAY 2, DAY 3}

⇓

Experience

Decision Tree {Solving many usecase}

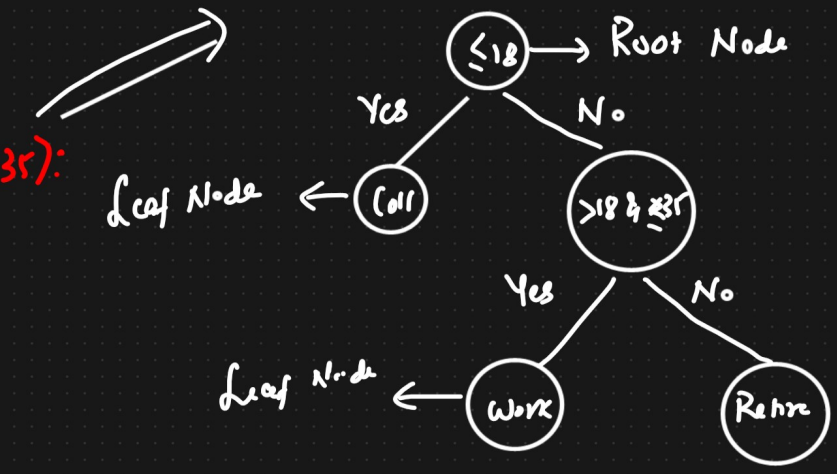
- Regression
- Classification

Regression & Classification

⇓

Decision Trees

```
if (age ≤ 18):
    Print ("College")
elif (age >18 and age ≤ 35):
    Print (Work)
else :
    Print ("Retire")
```



DECISION TREE

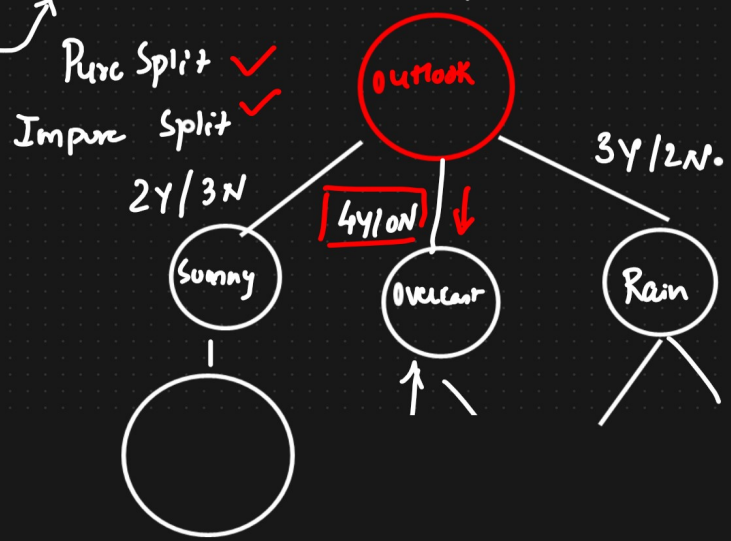
Nest if else ⇒ Decision Tree

CLASSIFICATION 9Y/5N

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Pure Split

Impure Split



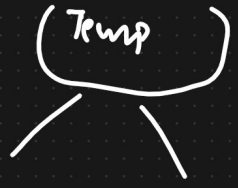
① Purity → Pure Split ??

↳ Entropy

↳ Gini Impurity

② How the features are selected

↳ Information Gain ?



Pure leaf Node

① Entropy

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

+ } Binary
− }

+ ⇒ Yes − ⇒ No

P_+ = Probability of Yes

→ Pure split

→ 0

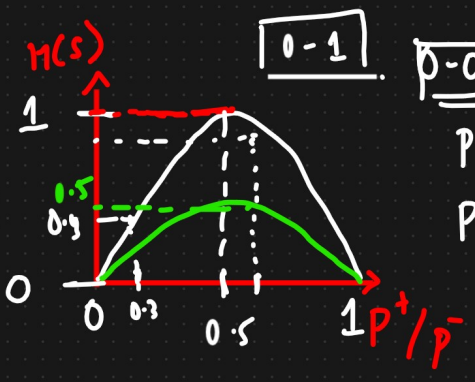
Entropy $H(S) = -\frac{3}{3}\log_2\frac{3}{3} - \frac{0}{3}\log_2\frac{0}{3}$

$= -1 \log_2 1$

$= 0$ → Pure split

① Gini Impurity

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



$H(S) = -\frac{3}{6}\log_2 3/6 - \frac{3}{6}\log_2 3/6$

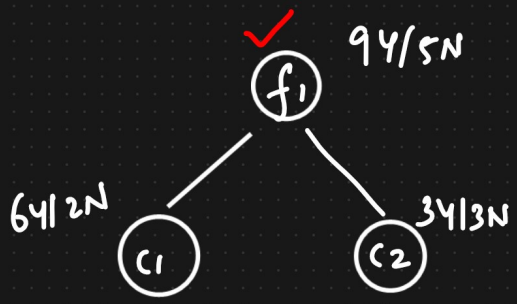
$= 1$ ✓

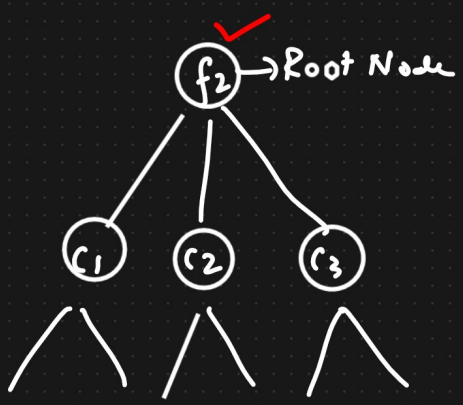
↳ Impure Split

Purity Test → Entropy

f1 f2 f3

② Which feature to take to split??

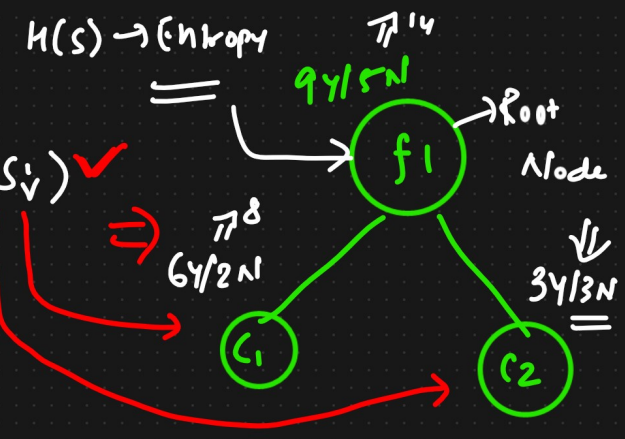




Information Gain

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

(Root Node)



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

$$= -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right)$$

$$\approx = \boxed{0.94}$$

$$H(C_1) = \frac{-6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8}$$

$$\boxed{H(C_1) = 0.81} \qquad \boxed{H(C_2) = 1}$$

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

$$Gain(S, f_1) = 0.049$$

$$Gain(S, f_2) = 0.051$$

Using which feature should I start splitting first

$$Gain(S, f_2) >> Gain(S, f_1)$$

② GINI Impurity ✓

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

$$= 1 - \left[(p_+)^2 + (p_-)^2\right]$$

{Entropy}

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

$$= 1 - \left[\frac{1}{2}\right] = \underline{0.5}$$

n = 2 output { Yes, No

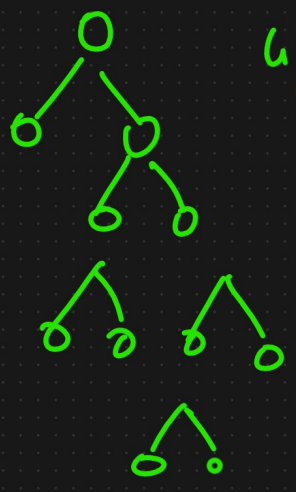
2Y/2N ⇒ Impure Split

Entropy = 1

Gini Impurity = 0.5

Entropy → fog

100



Gini Impurity → Simple Maths

Fast

Gini >> Entropy

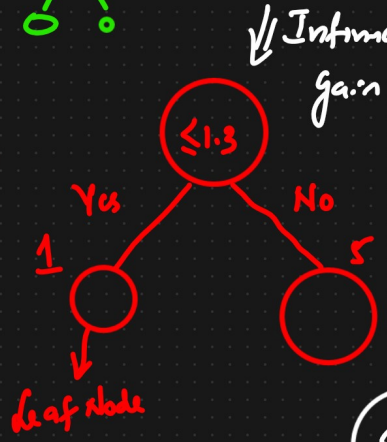
→ Continuous

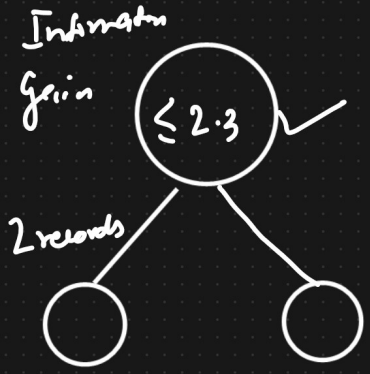
f1	O/P
2.3	
1.3	
4	
5	
7	
3	

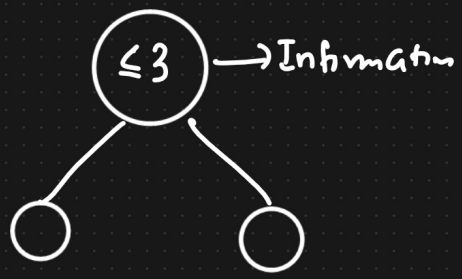
⇒

f1
1.3
2.3
3
4
5
7

O/P →



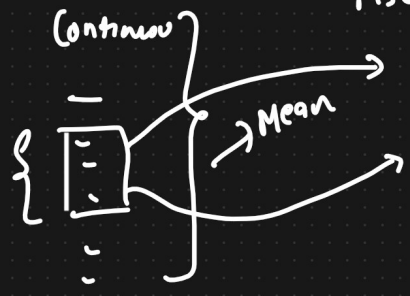


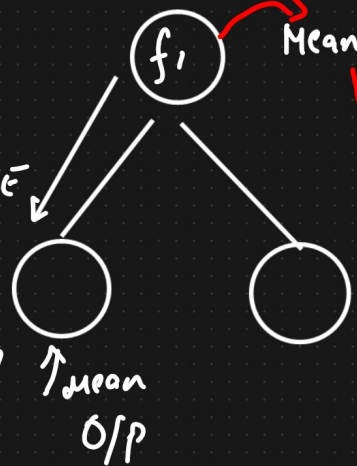


Best Info gain

Decision Tree Regressor

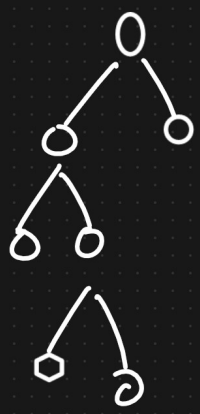
f1	f2	O/P





MSE or MAE

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



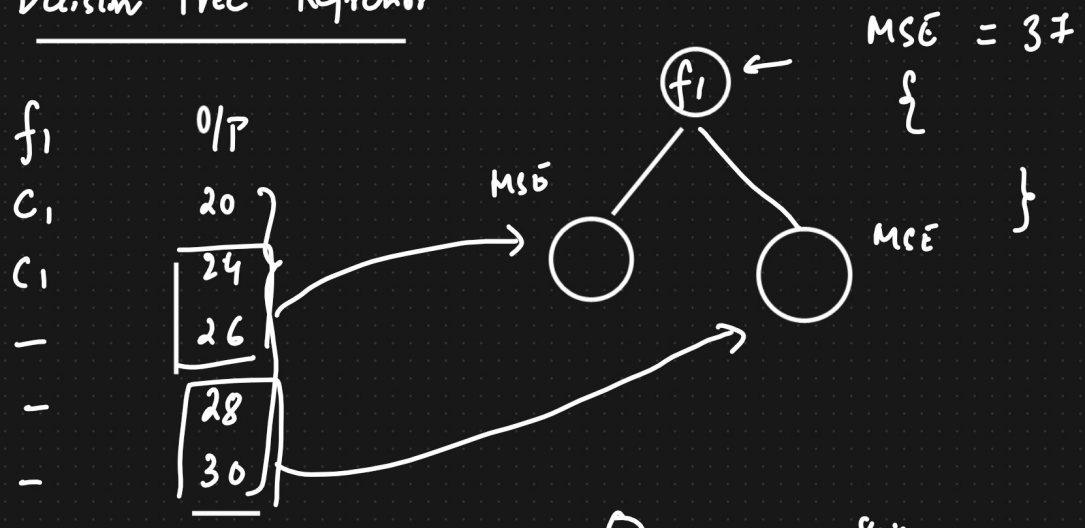
Overfitting

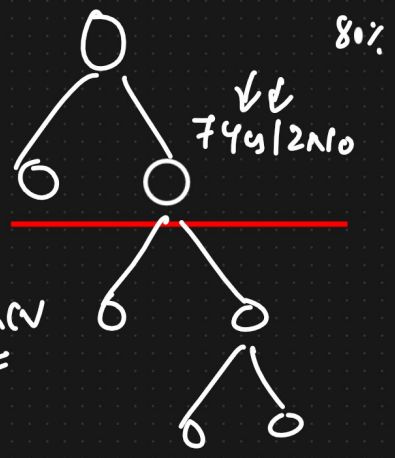
Hyperparameter

Decision → Overfitting

{ ① Post Pruning
② Pre Pruning }

Decision Tree Regressor





post pruning

pre pruning

Hyperparameter

GridSearchCV

max_depth, max_leaf

Day 5 — Machine Learning Algorithms

Agenda

1. Ensemble Techniques → Bagging, Boosting
2. Random Forest
3. AdaBoost
4. Xgboost → Youtube channel

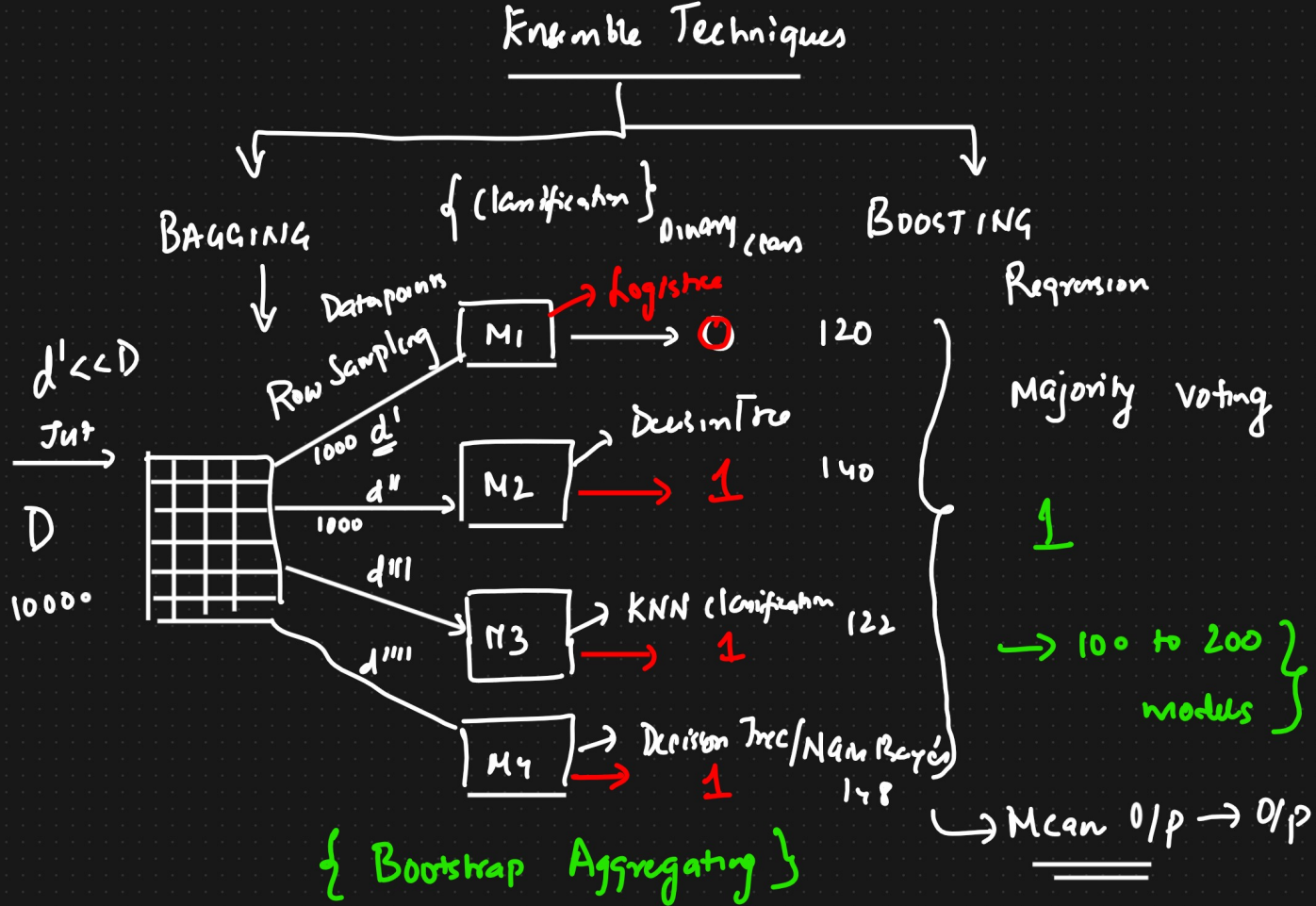
{ DJANGO, FLASK, EDA, Deep Learning, Stats, ML, NLP }

Ensemble Techniques

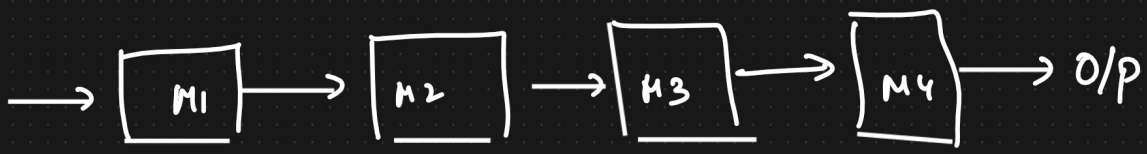
① Classification & Regression

↳ 1 Algorithm → Cl, Reg

Multiple Algorithms to Solve a problem?



Boosting



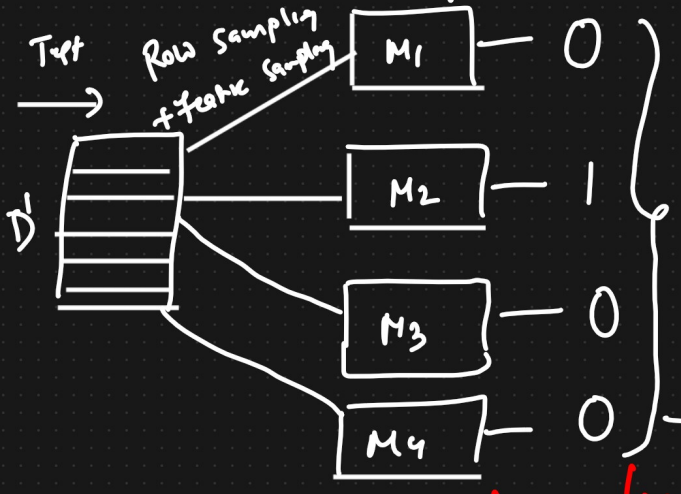
Weak learner = Weak learner = weak learner = Weak learner → {STRONG LEARNER}

BAGGING → ① RANDOM FOREST CLASSIFIER ② Random Forest Regression

BOOSTING → ① Adaboost ② Gradient ③ Xgboost

① Random Forest classifier and Regressor

Random Forest → Decision Trees

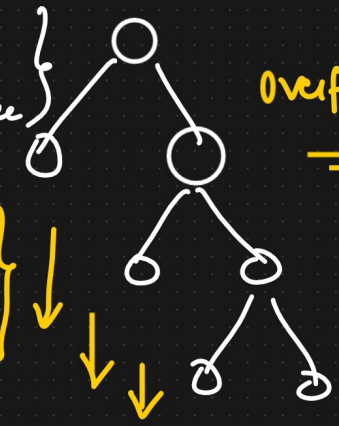


f_1 f_2 f_3 f_4

What is the main problem of DT?

Low Bias, High Variance → Overfitting??

Low Bias ✓ Low Variance ✓



Majority voting → O/p = 0

① Normalization ?!

⇓ Or Decision Tree

No.

② KNN { Standardization } ??

↓ ⇓

Yes Impacted by Outlier??

{ Eucledian , Manhattan }

③ Random Forest → Outliers ⇒ No → { check it google }

Bagging = Random Forest

Custom Bagging

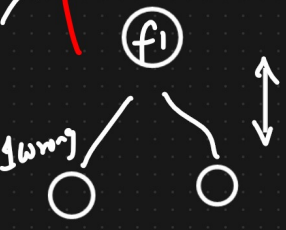
② Boosting

i) Adaboost → Decision Tree

Overall = 1

f1	f2	f3	fx	O/p	Weight	
–	–	–	–	Yes ✓	1/7	0.05
–	–	–	–	No ✓	1/7	0.05
–	–	–	–	– ✓	1/7	0.05
X –	–	–	–	–	1/7	↑0.349
–	–	–	–	– ✓	1/7	0.05
–	–	–	–	– ✓	1/7	0.05
–	–	–	–	– ✓	1/7	0.05

STUMPS



information gain & entropy

f2 f3

① { Weak Learner }

Total Error (TE) = 1/7

② Performance of Stump

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

$= \frac{1}{2}\log_e\left(\frac{1-1/7}{1/7}\right)$

$= 0.895$

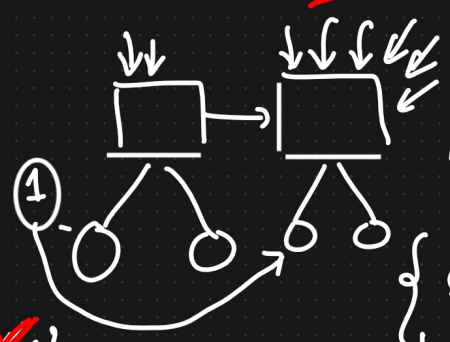
③ New Sample Weight = Weight * e^{-Ps} (Correct Records)

$= \frac{1}{7} \times e^{-0.895} = 0.05$

Incorrect record = Weight * e^{Ps}

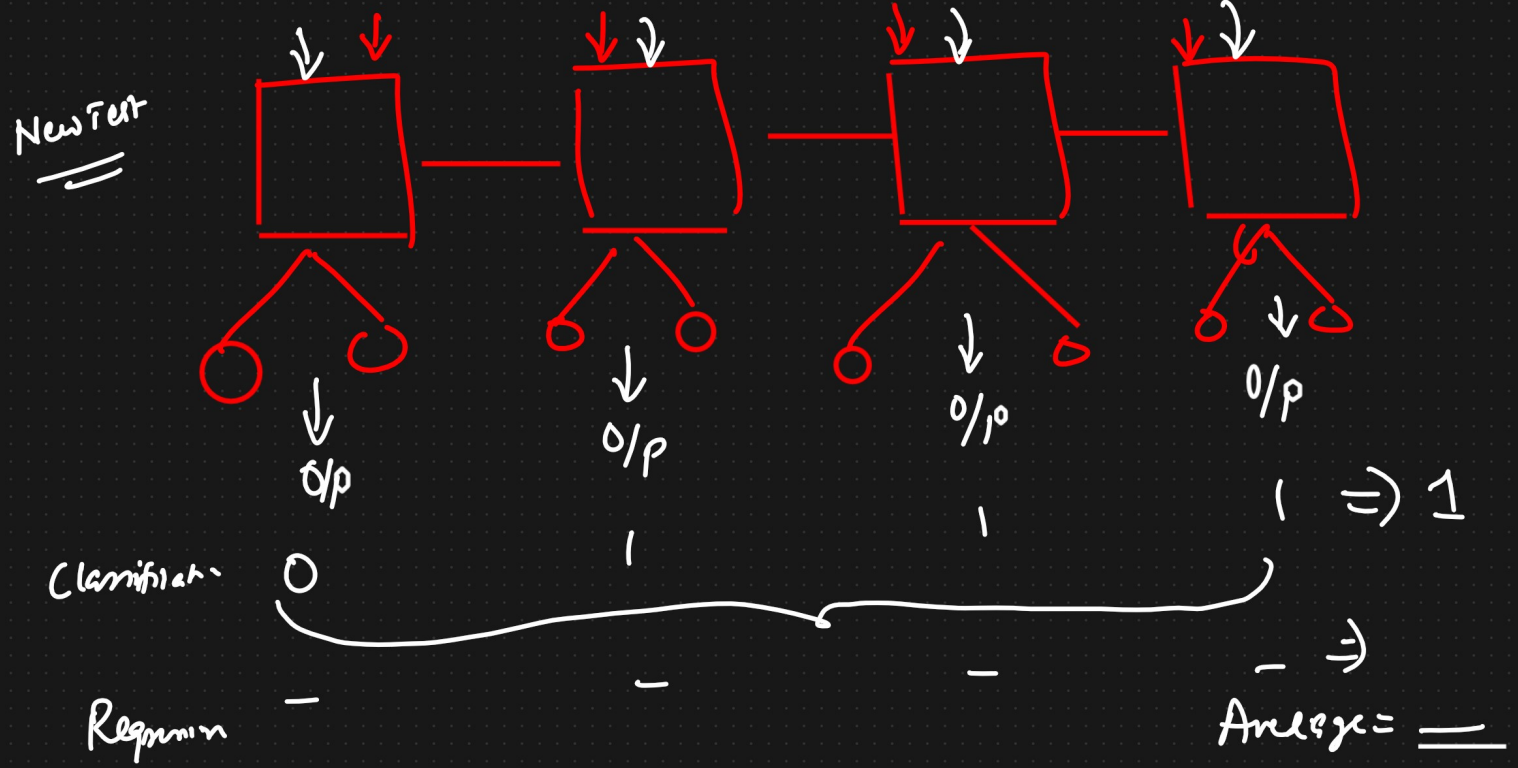
$= 1/7 \times e^{0.895} = 0.349$

New weight	Is it 1 ? Normalized weight	Buckets
0.05 : 0.649	0.07	[0 - 0.07]
0.05 : 0.649	0.07	[0.07 - 0.14]
0.05 : 0.649	0.07	[0.14 - 0.21]
→ 0.349	0.537	{ [0.21 - 0.747] ✓ }
0.05	0.07	[0.747 - 0.751]
0.05	0.07	[—]
0.05	0.07	[—]
0.649	≈ 1	



Randomly . Create some number between [0-1]

0.537
0.21
0.747



Black modes VS White box Modes

ANN → Black Box

→ Linear Regression → White box

Random Forest → Black box

Decision Tree → White box

Day 6 – Machine Learning Algorithms

Unsupervised ML

1. K Means Clustering
2. Hicrarichal clustering
3. Silhoutte Score
4. DBScan Clustering

Agenda

1. Svm & svR
2. Xgboost
3. PCA

Unsupervised ML

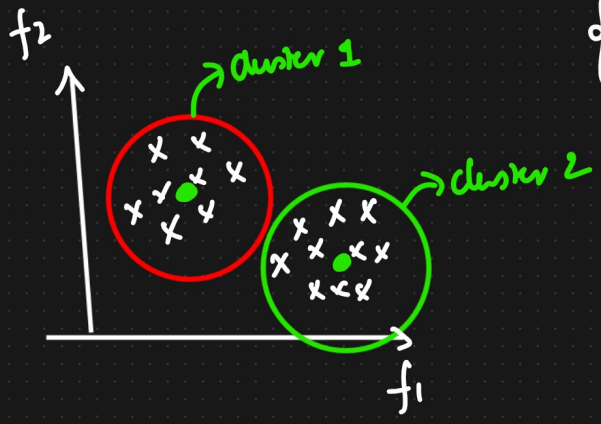
O/p

Clusters

{ Similar kind of data }

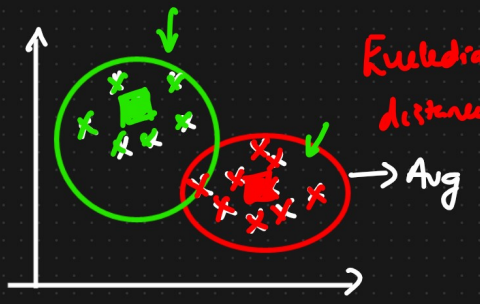
f_1	f_2
–	–
–	–
–	–
–	–

K Means Clustering



Custom Ensemble Technique

K Means → K = centroids



K = 2

High dimension

K = 2

centroids

① We try K values ⇒ suitable K=2

② Initialize K number of centroids ✓

③ Compute the avg to update centroids ✓

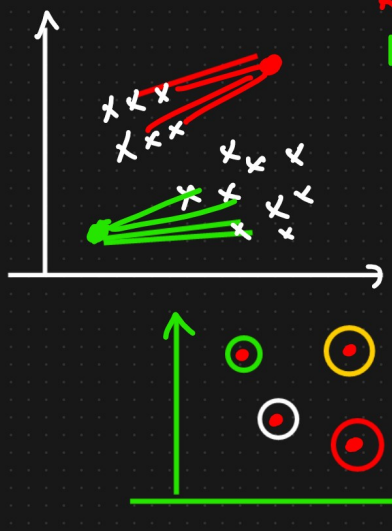
Validating

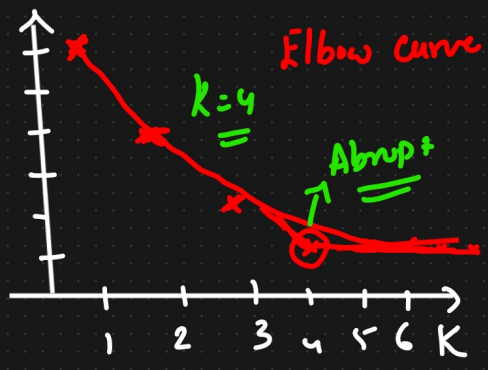
Within cluster sum of square

Elbow method (K value)

K=

for i=1,10

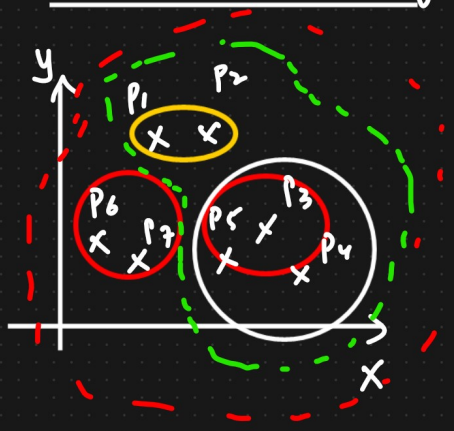


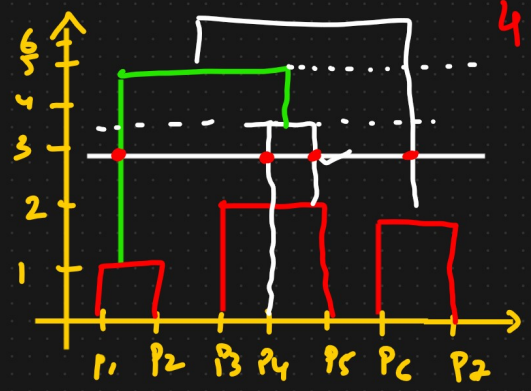


② Hierarchical Clustering

You need find the longest vertical line that has no horizontal line passed through it.

Dendrogram





Dataset is small

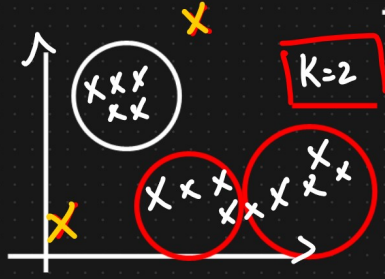
Dataset is large

Kmeans

Max Time is taken by KMeans or

Hierarchal clustering ??

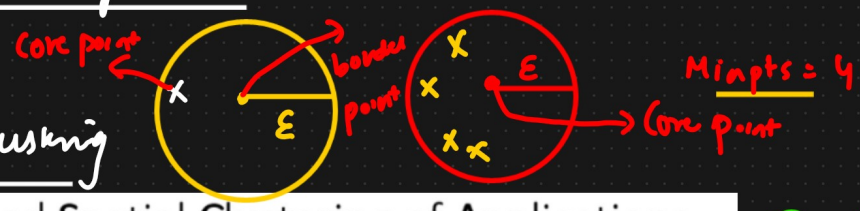
↳ Max Time



KMeans ++

Validate Clustering Models

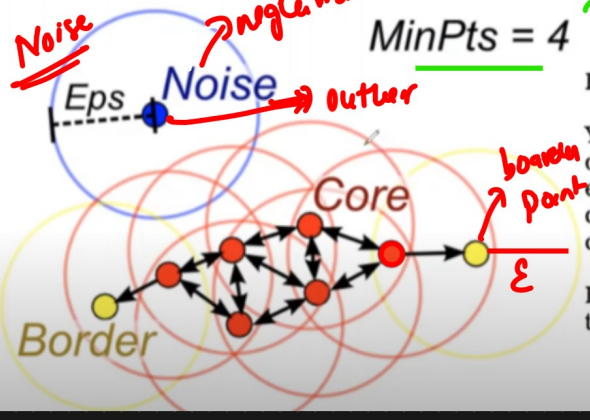
DBScan Clustering



Density-Based Spatial Clustering of Applications with Noise(DBSCAN)

Epsilon → Hyperparameter

MinPts = 4



Red: Core Points

Yellow: Border points. Still part of the cluster because it's within epsilon of a core point, but not does not meet the min_points criteria

Blue: Noise point. Not assigned to a cluster

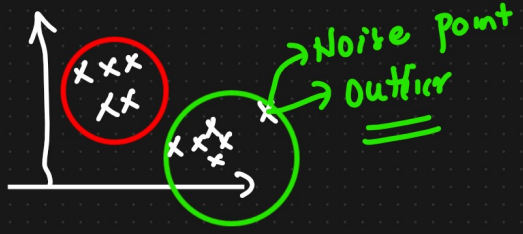
① Epsilon

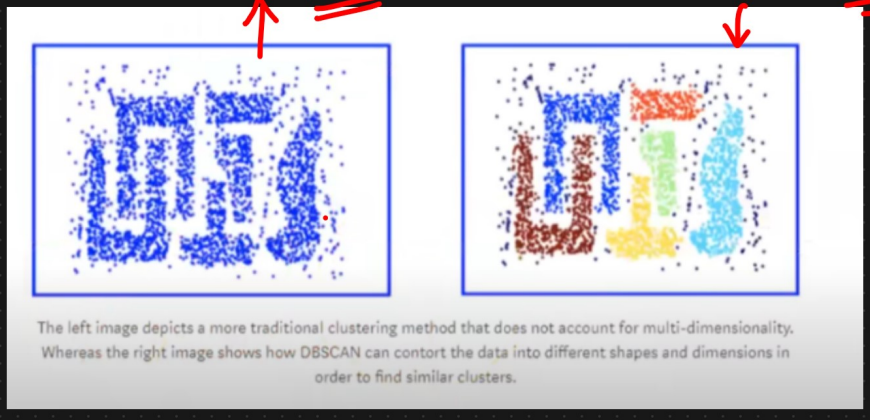
② Min pts

③ Core points

④ Border points

⑤ Noise point



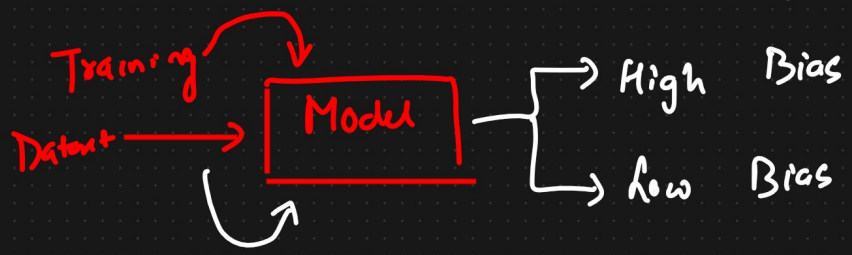


The left image depicts a more traditional clustering method that does not account for multi-dimensionality. Whereas the right image shows how DBSCAN can contort the data into different shapes and dimensions in order to find similar clusters.

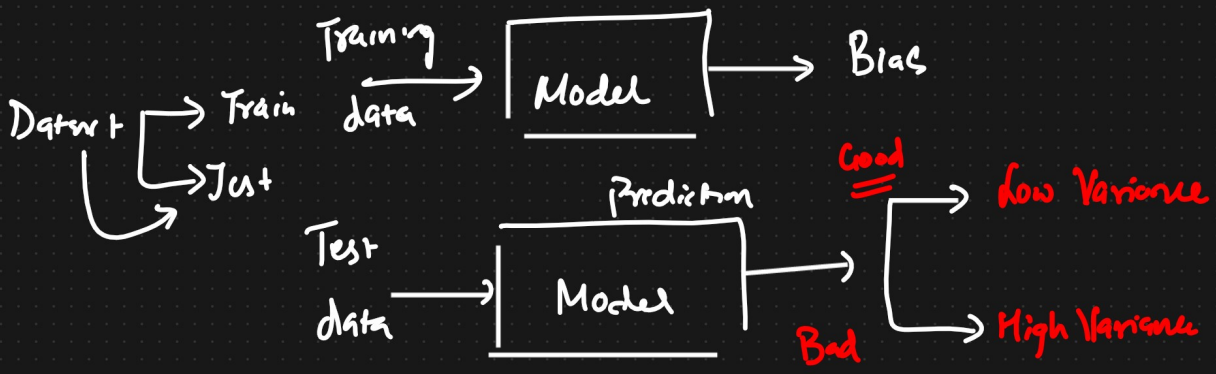
Defn of Bias And Variance

Training Dataset = 90%
Test Dataset = 70%
} ⇒ Overfitting
⇓
{ low Bias ✓ }
{ High Variance ✓ }

Bias : It is a phenomenon that skews the result of an algorithm in favor or against an idea.
↳ Training dataset



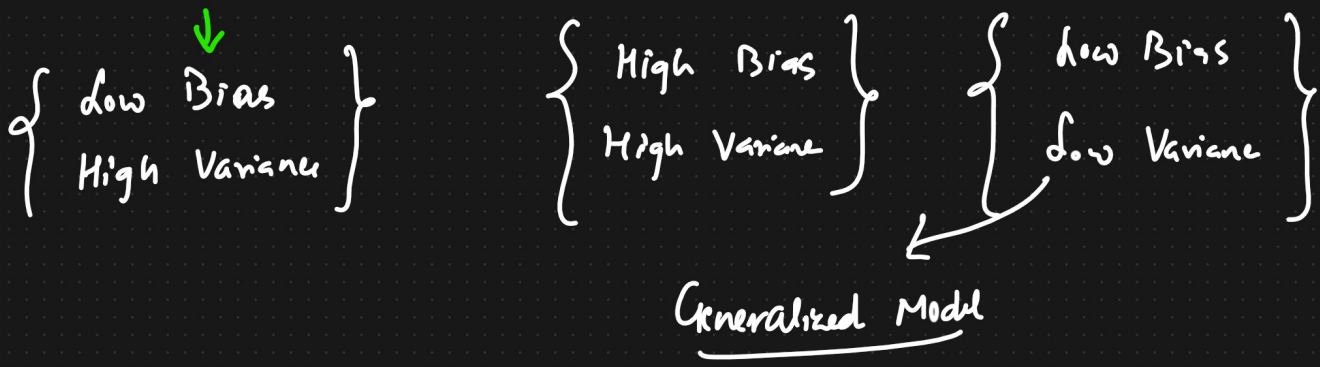
Variance : Variance refers to the changes in the model when using different portions of the training or test data



Model 1
Training Acc = 90%
Test Acc = 75%

Model 2
Train Acc = 60%
Test Acc = 55%

Model 3
Train Acc = 90%
Test = 92%


Low Bias
High Variance
High Bias
High Variance
Low Bias
Low Variance
Generalized Model

Day 7 : Xgboost CLASSIFIER AND REGRESSOR

Agenda

1. Xgboost classifier
2. Xgboost Regressor
3. SVM
4. SVR

$$\log\left(\frac{p}{1-p}\right) = \log\left(\frac{0.5}{0.5}\right) = 0$$

1. Xgboost Classifier

Extreme Gradient Boosting

{ Dataset }

Salary	Credit	Approval	Residual
<=50	B	0	-0.5
<=50	G	1	0.5
<=50	G	1	0.5
>50	B	0	-0.5
>50	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5
		$\lambda=0$	λ ?!

Base model → pr = 0.5 } Sim: 0.94

[-0.5, 0.5, 0.5, -0.5, 0.5, 0.5, -0.5]

Binary DT

Salary

<=50 : Sim = 0 → [-0.5, 0.5, 0.5, -0.5]

>50 : 0.33 → [-0.5, 0.5, 0.5]

Credit

B : Sim = 1 → [-0.5]

G, N : Sim = 0.33 → [0.5, 0.5, -0.5]

B, N

$$Sim = \frac{0.25}{0.5(1-0.5)} = \frac{0.25}{0.25} = 1$$

$$\frac{0.25}{0.75}$$

1. Create a Binary Decision Tree using the feature
2. Calculate Similarity weight

$$= \frac{\Sigma (Residual)^2}{\Sigma [Pr(1-Pr) + \lambda}$$

$$1 + 0.33 - 0 = 1.33$$

$$\sigma\left[0 + \alpha(1)\right]$$ (learning)

$\alpha = 0.01$

Sigmoid

0 to 1 ✓

③ Information gain

New Reward → O/p

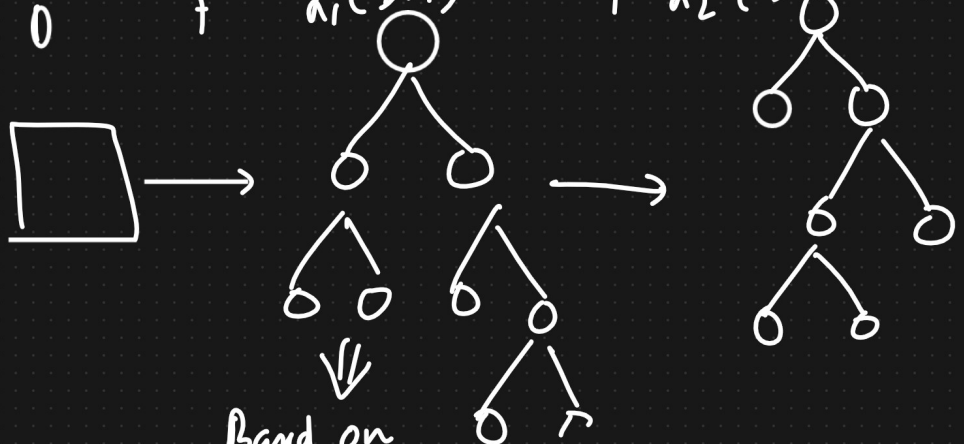
$$\left[\sigma\left[0+\alpha_1(DT_1)+\alpha_2(DT_2)+\alpha_3(DT_3)+\alpha_4(DT_4)+\cdots\alpha_n(DT_n)\right]\right]$$

XgBoost → BLACK BOX Model

Pre pruning

$\lambda \rightarrow$ (cross validation)

$$0 + \alpha_1(DT_1) + \alpha_2(DT_2) + \alpha_3(DT_3) + \cdots + \alpha_n(DT_n)$$



Based on independent features

$$\frac{(-11-9+1+9+11)^2}{5+1} \cdot \frac{1}{6} =$$

Base Model → 51

Sim = Y6

[-11K, -9, 1, 9, 11]

EXP

≤ 2 : [-11] → Sim = 65.5

$\frac{121}{1+1} = 60.5$

> 2 : [-9, 1, 9, 11] → 28.8

$\frac{(-9+1+9+11)^2}{4+1} = \frac{144}{5} = 28.8$

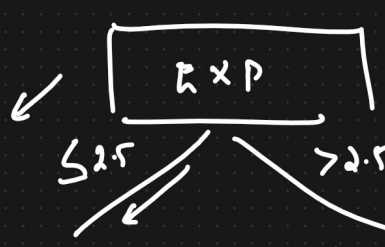
② Xgboost Regressor

Exp	Gap	Salary (→ O/P)	Res
2	Yes	40K	-11K
2.5	Yes	42K	-9K
3	No	52K	1K
4	No	60K	9K
4.5	Yes	62K	11K

$\lambda = 1$

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

Information = $60.5 + 28.8 - \frac{1}{6} = 89.13$



Y/c ℓDT1

133.33

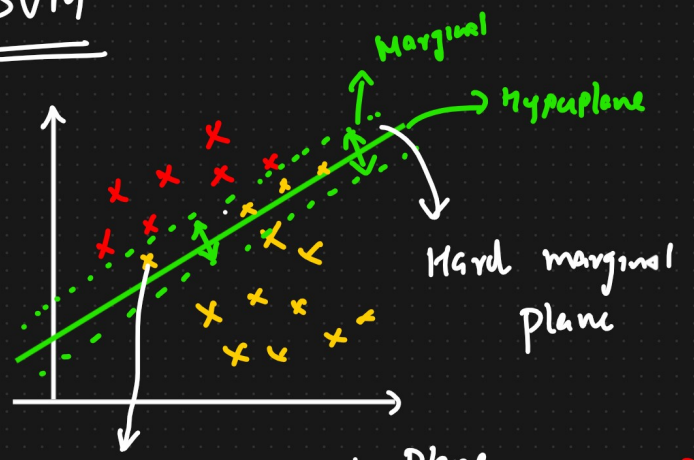
110.25

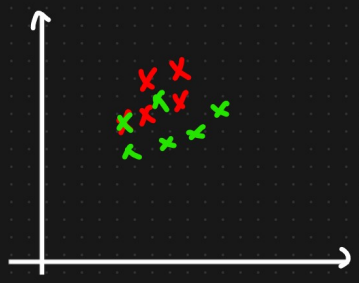
$\frac{-11-9}{2} = -10$ [-11, -9] [1, 9, 11] $\Rightarrow \frac{1+9+11}{3} = \frac{21}{3} = 7$

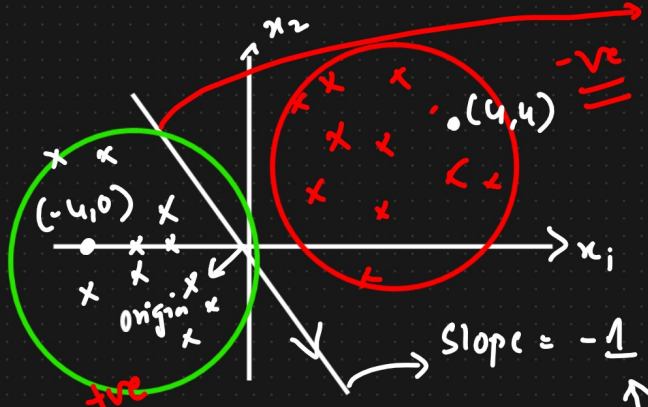
$$O/p = 51 + \alpha_1(-10) + \alpha_2(DT_2) + \alpha_3(DT_3) - \cdots + \alpha_n(DT_n)$$

① {EDA & Feature Engineering} ✓

③ SVM







$y = mx + c$ → Equation of a straight line (slope, intercept)

$ax + by + c = 0$

$y = \frac{-c}{b} - \frac{ax}{b}$

$m = -a/b \quad c = \frac{-c}{b}$

$y = mx + c$

$y = w_1 x + w_2 x_2 + \cdots + b$

$\omega \to -1$

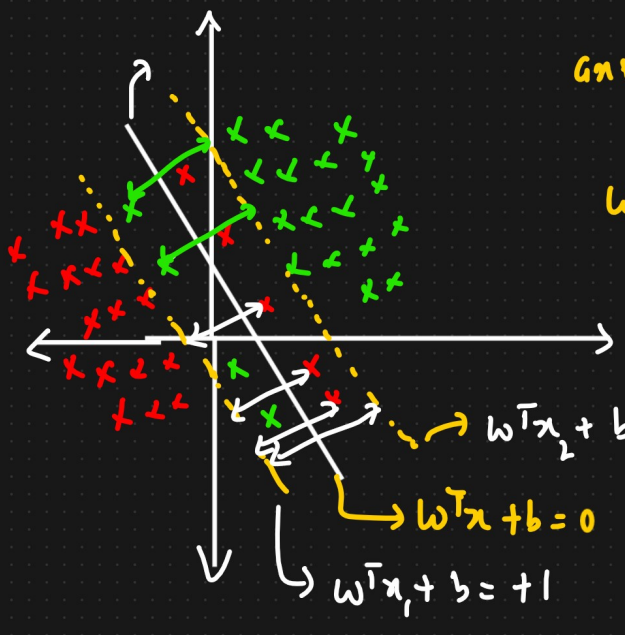
$$\boxed{y = \omega^T x + b}$$

$$y = \begin{bmatrix} -1 \\ 0 \end{bmatrix} [-4 \;\; 0]$$

$= 4 \Rightarrow$ +ve value

$$y = \begin{bmatrix} -1 \\ 0 \end{bmatrix} [4, 4]$$

$= -4 + 0 = -4$



$ax + by + c = 0$

$\Downarrow$

$w^T x + b = 0$

$\vec{w}$ Vectors + magnitude

$w^T x_1 + b = 1$

$w^T x_2 + b = -1$

$\underline{\quad\quad\quad\quad\quad\quad\quad\quad}$

$w^T (x_1 - x_2) = 2$

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

$\Downarrow$

$\frac{2}{||w||}$ $\Rightarrow$ Marginal

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

Such that y_i $\Downarrow$

$+1$ $\quad w^T x + b \geq 1$

-1 $\quad w^T x + b \leq -1$

$\Downarrow$

Major Aim $\quad y_i * (w^T x_i + b) \geq 1$

$\Downarrow$

for correct point

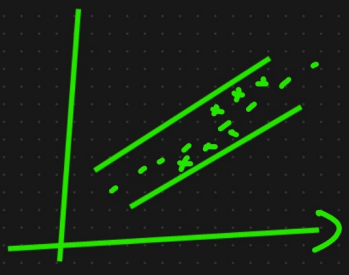
Maximize (w, b) $\frac{2}{||w||}$	$\Leftrightarrow$	Min (w, b) $\frac{||w||}{2}$

Min (w, b) $\frac{||w||}{2} + C_i \sum_{i=1}^{n} \xi_i$

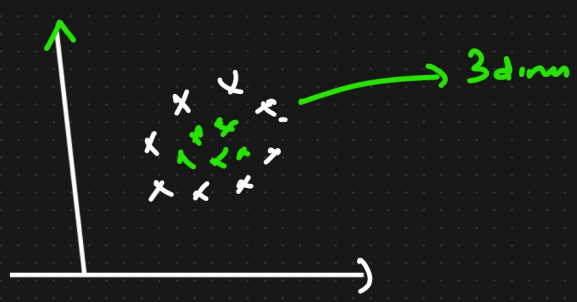
ξ_i → Summation of the distance of the wrong datapoints

C_i → {6} → How many Errors we can have

SVR = Explore



SVM Kernel



Practically =