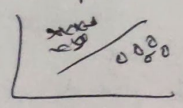
26 May Decision Tree

Story

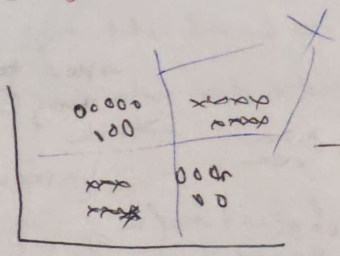
In Class
↳ D.T → Cl
→ Reg

→ Start with example



↳ by using Line

but if :-



→ why can't use here Logistic Regres

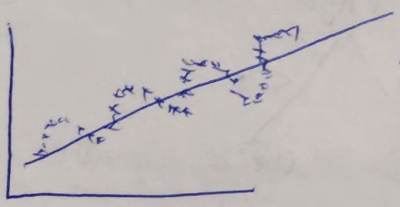
→ So we cannot using the line to Separate.

→ Problem with r.t classification

• why? Here data in non linear

↳ DT SVM

⇒ Problem for Regression



X no line use for this

Can use polynomial Regression

↳ No because it is not a ML Algo it is use for Preparse data OR DO feature engnr to fit in Linear Regression.

So use

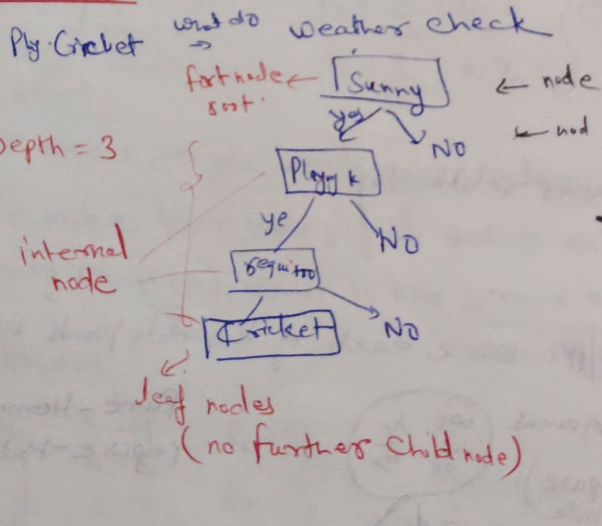
DT SVM

DT classifier — SVM classifiers

DT Regressor — SVM Regressor

• Decision Tree

Ex:- go out for Ply Cricket

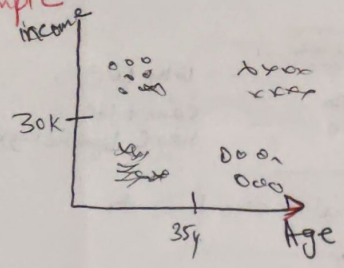


Can't say it is look like a tree

→ So we can say every node, some decision is being made

↳ So this is called as Decision tree

Example



→ How to solve this non linears using DT

→ think

cond ① income ≥ 30K & Age ≤ 30y
↳ group 1 (fraud)

② Income < 30K & Age ≤ 35 yrs
↳ group 2 (no fraud)

③ if income ≤ 30K & Age ≥ 35 yrs
↳ gr 3 → 0th fraudster

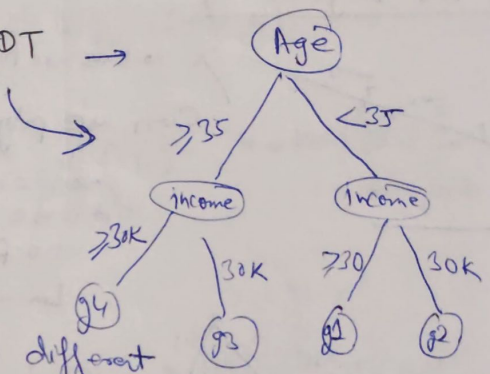
④ if income ≥ 30 & Age ≥ 35 y.
↳ gr 4 → fraudster

So we can write these Condition Linears/Logistic Regression

Any No

↳ then can we write this in DT →

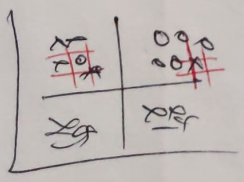
Ans ↳ yes.



So we can say

⇒ DT divide the data into different Partition or into Pockets (Nodes)

but sir we do by using 2 lines → so ok what if dp are mixed



what if dp like this then what you do

so → keep splitting/dividing until all the dp is of same class.

— Adv of DT

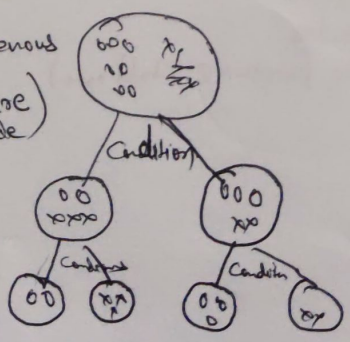
① Simple to Understand

② Capture the non-linear relationship

⇒ So what is intend of DT

• To create leaf node/pkt where each of the data points belong to one class.

Heterogenous data (impure node)



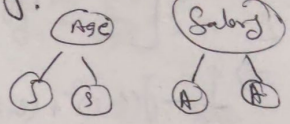
Pure - Homo (Same Class)
Impure - Hetro (mixture

Homogenous data (Pure Node)

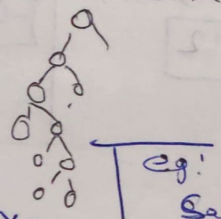
Now the point is → Let Say you split data based on some condition ⇒ but when you will know → this will splited or not splited.

Q1 When to Split? / How to Split → we know based on Condition then How you fit the conditions.
- On what factor condition should be use?

Q2 Which feature to be used for splitting?

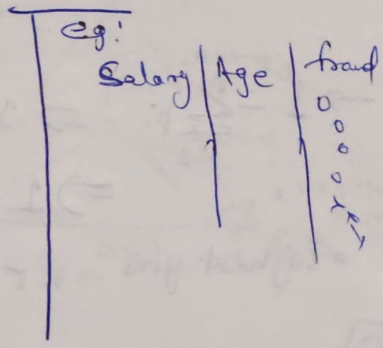


Q3 Till when to split? → who will be decide

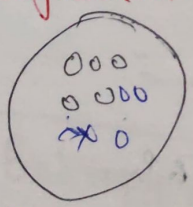


All
Ans. → Impurity measures

3 types
- Classification error
- Gini
- Entropy



In
1) Classification error



0 : 8
X : 2

$CE = 1 - \max p_i$ → propaplity of i class

$P_X = \frac{2}{10} = 0.2$

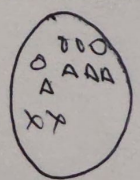
$P_O = \frac{8}{10} = 0.2$

$CE = 1 - \max(0.8, 0.2)$
$= 1 - 0.8$
$= 0.2$ Ans

What it means. → you want to pure node → yes

So → To make pure node, if I assign everything to majority class (O), what is the error rate?
↳ minority misclassified dps

→ For Multiclass



O - 4
X - 2
Δ - 4

$P_O = \frac{4}{10}$

$P_X = \frac{2}{10}$

$\Delta = \frac{4}{10}$

$1 - \max(0.4, 0.2, 0.4)$
$= 0.6$

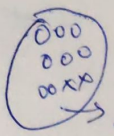
• [1,2,3] 1 to n
[0,1,2] 0 to n-1

② Gini

(i) $G.I = 1 - \sum_{i=0}^{n-1} (P_i)^2$

Meaning

i = from class 0 to $n-1$



$n-1 \to 2-1$

$1 - \sum_{i=0}^{1} (P_i)^2 = 1 - [(P_0)^2 + (P_x)^2]$

$P(0) = \frac{8}{10} \to 0.8$

$P(x) = \frac{2}{10} \to 0.2$

$\Rightarrow 1 - [(0.8)^2 + (0.2)^2]$

$\Rightarrow 1 - (0.64 + 0.04)$

$= 1 - 0.68 \Rightarrow \boxed{0.32}$

Same

(ii) $G.I = \sum_{i=0}^{n-1} P_i (1 - P_i)$

for Two class

$P_0(1-P_0) + P_x(1-P_x)$

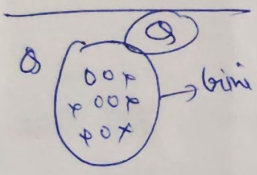
$P_0 + P_x = 1 \Rightarrow$ so $P_x \Rightarrow 1 - P_0$

$= 2P_0(1-P_0)$

$= 2 \times 0.8(1-0.8)$

$\Rightarrow 1.6(0.2)$

$= \boxed{0.32}$



$\Rightarrow$ Gini $\to 1 - \sum_{i=1}^{n} P_i \Rightarrow 1 - (P_0^2 + P_x^2) \Rightarrow 1 - ((0.5)^2 + 0.5^2)$

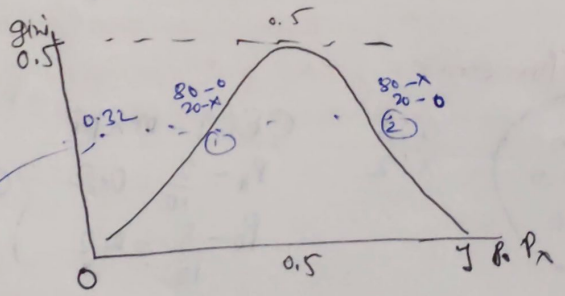
$\Rightarrow 1 - 0.5 = 0.5$

 $\Rightarrow 0.32$

• highest gini $\to 0.5$ ↗ when the same no of dps

which is more impure bcz the equal no of dp of all class

$\Rightarrow$ So



1 more observation

What is gini

0.32 0.32

$\Rightarrow$ So hence proof both are same

→ Adv
→ disAdv
→ use
→ Real world

def

③ Entropy:

$$E = -\sum_{i=1}^{n} p_i \log_2 p_i$$

$$= \text{(coin)} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}$$

$$= -2 \cdot \frac{1}{2}\log_2\frac{1}{2}$$

$$= \log_2\frac{1}{2}$$

$$= -\log_2 2^{-1} \Rightarrow -1 \times -1$$

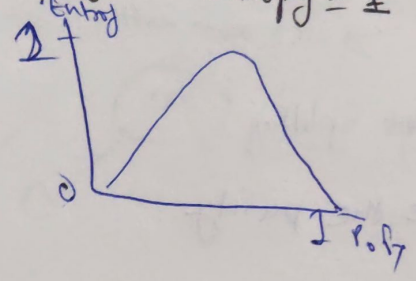
$$= 1 \text{ Ans}$$

Chemistry → Brownian motion

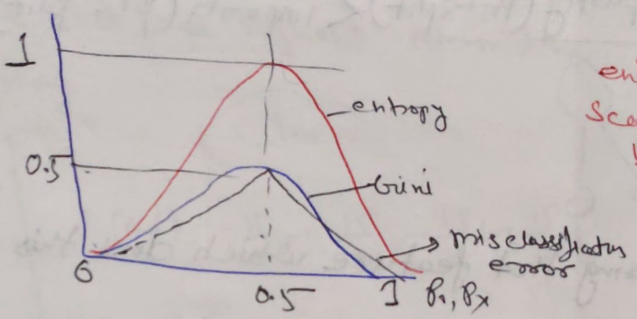
Phy → Stochastic Calculation

Entropy ↑ when Randomness ↑

So Highest entropy = 1



- Entropy - from 0 to 1
- Gini - 0 to 0.5
- misclassific error = 0 - 0.5



entropy scaled ↳ = Entropy / 2

⇒ For Multiclass classification

$$Gr.I = 1 - \sum_{i=1}^{n} p_i^2$$

$$= 1 - [p_1^2 + p_2^2 + p_3^2 \ldots p_n^2]$$

$$\text{Entropy} = -\sum_{i=1}^{N} p_i \log_2 p_i$$

$$- p_1 \log_2 p_1 - p_2 \log p_2 - p_3 \log p_3 \cdots p_n \log p_n$$

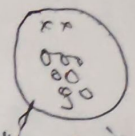
Adv
dis
usecase

(3) Which impurity measure to be used?

Gini
for small dataset → Entropy for large dataset – Gini
(computations)

(2) On what feature to split

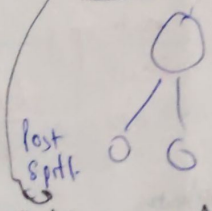
→ use Information gain



disorganisation / randomness

→ Aim to splitting
- We want pure nodes after splitting.
- Such a split that maximise the purity

- Impurity (Post split) < impurity (pre split)



Split using that feature which does this best

post Im < imp pre

Δimpurity ⇒ pre imp − post imp

if randomness decreases, purity increase, then we say Information gain.

Δ impurity

★ IG is a measure to define degree of disorganisation/ randomness in a system ⇒ IG → GI
Entropy
IG ⇒ Δ impurity

★ whichever feature will have maximum information, that feature will be used split

• Piston / ↳ semi conductor ↓ 1m - 150

Example

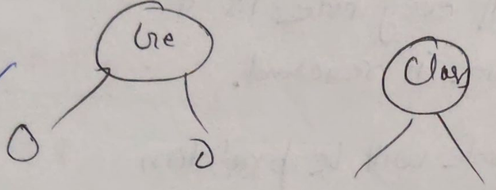
Gender	class	Cricket (Y/N)
M	10	yes
F	X	No
M	1X	ye
⋮		

Total Std = 30 → 50% ply cricket
Yes = 15
No = 15

gender → 10/2 → F → ply ckt (out of 10 → 2 ply)
↘ 20/13 → M (out of 20 30 play)

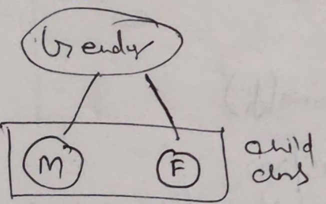
Class → 1X — 14/43% (out of 14 : 1X 43% ply)
↘ X — 16/57% (out of 16 cls X 57% ply crkt)

which cases IS L ←



Use → whichever feature has the highest information Gain, used for Splitting.

[In Gini, Entropy]



I.G = Δ impurity
En(Parent) − En(Child combined)

En Parent ⇒ $-\frac{15}{30}\log_2\frac{15}{30} - \frac{15}{30}\log_2\frac{15}{30}$
$= 1$

En(Child) Female $= -\frac{2}{10}\log_2\frac{2}{10} - \frac{8}{10}\log\frac{8}{10}$
$= 0.72$

En(Male) Chld $= -\frac{13}{20}\log_2\frac{13}{20} - \frac{7}{20}\log_2\frac{7}{20}$
$= 0.92$

No of Std is different in both Prob so use weighted

En(Child/Combined) = $\frac{\text{No of F}}{\text{Total Std}} \times E_{F\,Chld} + \frac{\text{No of M}}{\text{T. Std}} \times E_{male\ Child}$
⇓
Weighted Entropy
$\frac{10}{30}\times 0.72 + \frac{20}{30}\times 0.92 \Rightarrow 0.85$

I.G = $E_p - E_c \Rightarrow 1 - 0.85 = 0.15$

Class

$En(Parent) = -\frac{15}{30}\log_2\frac{15}{30} - \frac{15}{30}\log_2\frac{15}{30} = 1$

E(child)
$E(1X) = \frac{43}{100}\log_2\frac{43}{100} - \frac{57}{100}\log_2\frac{57}{100} = x_1$

$En(X) = -\frac{57}{100}\log_2\frac{57}{100} - \frac{43}{100}\log_2\frac{43}{100} \Rightarrow x_2$

$En(Combined) = \frac{14}{30}x_1 + \frac{16}{30}x_2$
$= 0.99$

I.G = $E_p - E_c$
$= 1 - 0.99$
$= 0.0.01$

Since IG G > IG
∴ we will split base on Gender

⇒ Algorithm of DT

Step1 → Recursive binary Spelitting/Partitoing the data into smaller subset.

Step2 → Select the feature to splint (I.G)

Step3 → Apply the Split

Step4 → Repeat the process for the subset of obtained

Step5 → Continue the process untill every node is a purenode/stopping criteria. is reached.

Step6 → Majority Value in the leaf node will be prediction

Eg: Age, Sala, Gender

→ feature is Categorical easy

→ What if feature is Continouse

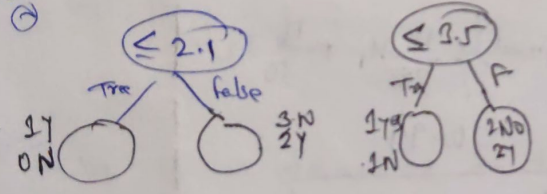
fi	output
2.1	Y
3.5	N
4.1	N
5.3	Y
6.8	Y

step. ① Sort the features(fi)

② treat each Value/row as a cutoff/threshold and try to built decision tree & whichever threshold gives you the best information gain that will be used for Splitting.

① Sort

②



★ DT is a Greedy Algorithm.

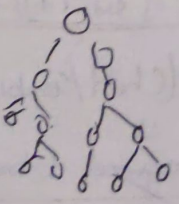
- It keep on Splitting untill every leaf node is pure node

⇓ memorize the data

⇓ Overfitting

reduce

Preprouning | Post pruning | Ensemble



Agenda
↳ Pickle
↳ Pre, Post Prun
↳ Implement
↳ DT Regg
↳ Practical
↳ SVM
↳ Naive Bayes
↳ Ensemble

1st June DT & SVM

Pickel → for use model

>>> import pickle (implements binary protocols for serializing & De-serializing a python object structure)

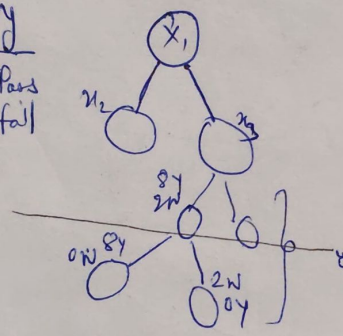
>>> picke.dump(model, open('model.pkl', 'wb'))
↳ Store it

>>> with open("model.pkl, 'rb') as file:
loaded_model = Pickle.load(file)
loaded_model.pridict( )

DT post purning & Pre purning.

DT is greedy Algorithms ⇒ So overfit the Model
↓
In Generalize Training A↑ Test↑
Training A↑ test↓

x_1	x_2	x_3	y
–	–	–	Pass / fail



↳ why you don't want to split from here
↳ because to Make Generalized Model.

Lasso/Ride
↳

⇒ Parameter to Create DT

1. Criteria — Gini / Entropy
2. Max_depth → Maximum depth of tree
   - The minimum no of samples needed to be considered as a leaf node default is 1
   - Visualisation
   - Hyperparameter tuning using CV
3. Splitter — Strategy (best, random) → best on information brain
   ↳ best on
4. min_samples_split → The min^m no of sample a node must Contain to consider split.
5.

Puring → Cutting

① Post pruning (Cutting of dt after it is created)
Cutting

↳ Construct the entire dt until leaf node
↳ Prune the dT
↳ works well with small dataset

★② Pre pruning
↳ limit the growth of DT
→ Hyperparameter tuning to selet best Paramter.

Practical

Decision Tree Regressor :

- Regression
  ↳ y → Continous.

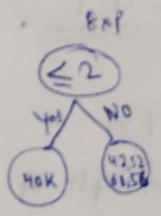
Classification
↳ y → Categorical

Calculate Variance ← Entropy
Variance reduction ← IG
impure

Exp	Career Exp	Salary
2.	Yes	40K
2.5	Yes	42K
3	No	52K
4	No	60K
4.5	Yes	58K

$\bar{y}_{Avg}$ 50K

use as each for split node



→ final aim
↳ To reduce the Variance

We want feature value give highest feature reduction

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

$$Var = \frac{1}{5}\sum_{i=1}^{n}\left((40-50)^2+(42-50)^2+(60-50)^2+(58-50)^2\right)$$

$$= \frac{1}{5}(100+64+4+100+36) \Rightarrow 60.8$$

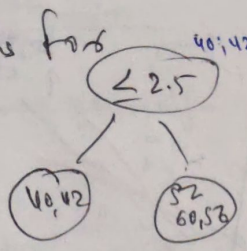
$$Var(\text{Left child}) = \frac{1}{1}(40-50)^2 \Rightarrow 100$$

$$Var(\text{Right child}) = \frac{1}{4}(8^2+2^2+10^2+8^2) \Rightarrow 51$$

$\Rightarrow 60 - \left(\frac{1}{5}\times 100 + \frac{4}{5}\times 51\right)$
$\Rightarrow 0.004$

Variance Reduction ⇒ $Var_{parent} - Var_{childcombined}$

then Again for



var.

$Var_{Parent} = 60.8$

$Var_{left\ child} = \frac{1}{2}(100+64) \Rightarrow 82$

$Var_{right\ child} = \frac{1}{3}(4+36+100) \Rightarrow 46.66$

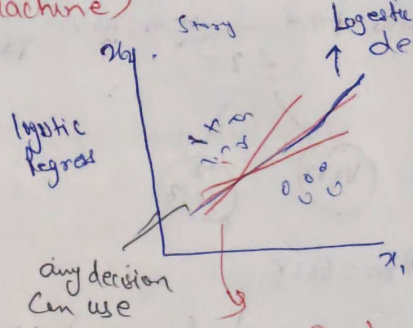
$Variance\ reduction \Rightarrow 60.8 - \left(\frac{2}{5} \times 82 + \frac{3}{5} \times 46.66\right)$

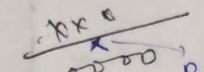
$\Rightarrow 0.004$

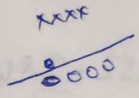
$\therefore$ Variance reduction for 2.5 is high so we this is we 1st for split

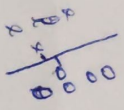
SVM (Support Vector Machine)

↳ SVC → classification
↳ SVR → Regressor



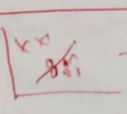
①  So I give u situation if it segregates both the Class yes
Any line Can be decision boundary
Prediction → Class 0 → Wrong Predn

②  → Wrong Prediction

③  Which scenario is best ①, ②, or ③ yes 3rd one so here come SVM.

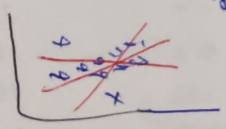
- Scaling not need in DT

and
- Logistic Regression Model doesn't care about margin/ Space across two class.
- If the data point changes slightly logreg will give error.

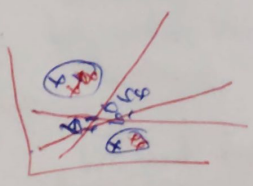
 → Prone to errors

So Sir Can we solve with DT → Ans Yes (And we are multiple option) OK
but SVM also solve this

- W.R.T Regression

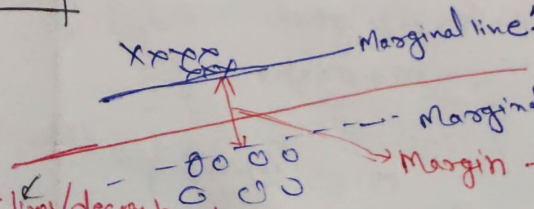


if not a out lier or if add for data -



↳ In SVM not impact by outlier but in Reg LR it reflect.

SVC



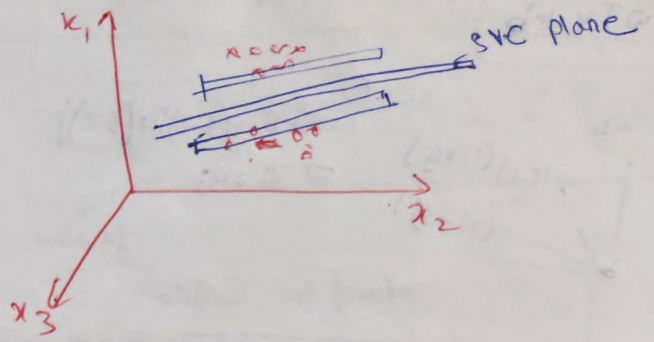
And what is the best line → is between the M line 1 & line 2 Point are Support Vectors

To make marginal line for both class you need atleast 2 Support Vectors

COMMON QUESTIONS

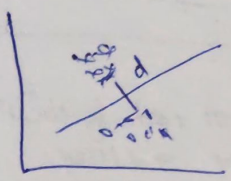
① Tell me about yourself?

→ for Multi class

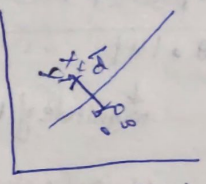


- There are no limit on SV.
  → atleast no of SV is 2
  → SVC is also known as Margin classifiers.
  → SV we for

Now Give you 2 scenerios



$d > d'$



- Choose model where marginal distance should maximum.

find max ↙

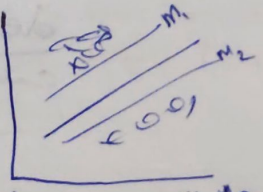
Step1 : find out all possible classifier

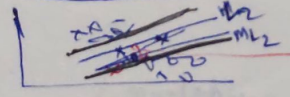
Step 2 : find out margin of each classifer

Step3 : Select the classifier which has maximum.

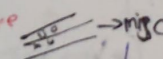
Maths behind
↳ SVM, C

Hard Margin : None of the dps are misclassified / overlapping

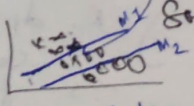
① 

② 

→ So Marginal distance will be small

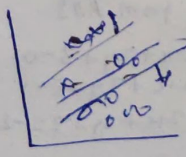


→ misclassified
so we here



Soft Margin

- you are ready to misclassify some of the dps in order to have good testing accuracy its a soft margin classifier.



C = 3 → control the mis class → decide by CV

↳ No of dps to sacrifice so that the best fit line is not overfitting.

→ eq of line $y = mx + c$, $\theta_0 + \theta_1 x$

$ax + by + c = 0$

$by = -ax - 0$

$y = \frac{-a}{b}x - \frac{c}{b}$

$y = Mx + C$

In 3D $y = \theta_0 + \theta_1 x$

$y = \theta_0 + \theta_1 x_1 + \theta_2 + \theta_2 \cdots \theta_n x_n$ → hyperplane

$y = b + w_1 x_1 + w_2 x_2 \cdots w_n x_n$ → w weights

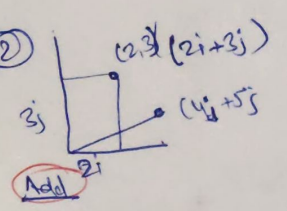
$y = b + [w_1 x_1 + \cdots w_n x_n]$ ↑ θ_0 & b bias

$w = \begin{bmatrix} w_1 \\ w_2 \\ w \end{bmatrix}$ $x = \begin{bmatrix} x_1 \\ x_2 \\ x \end{bmatrix}$ → multiply $w^T \cdot x \Rightarrow$

$[w_1 w_2 w_3 w_4] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = w_1x_1 + w_2x_2 + \cdots + w_4x_4$

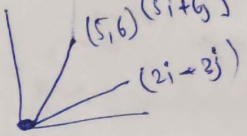
(1) $\Rightarrow \boxed{y = w^T x + b}$ ← use in SVM like this

(2)



(3) Vector Sub

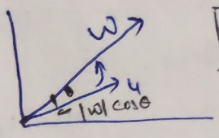
$\vec{A} - \vec{B}$



$\Rightarrow \vec{A} - \vec{B} = (5-2)i + (6-3)j$

$\Rightarrow 3i + 3j$

(4) Dot Product of two vector

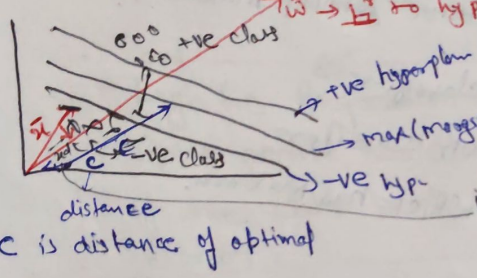


$w \cdot u$ → dot product means projection of u on w

$\rightarrow |w| \cos\theta \cdot |u|$

magnitude of w ↙ ↘ Magnitude of u

#



c is distance of optimal

$\vec{x}$ projection on w

if $x_d > c \rightarrow$ +ve class

$x_d < c \rightarrow$ -ve class

we know projection is dot product

$\vec{x} \cdot \vec{w} = c$

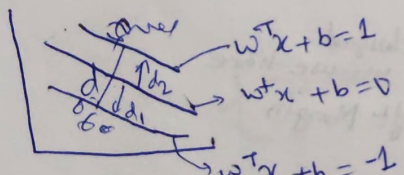
$\Rightarrow$

$w^T x + b = 0$

$$y = \begin{cases} +ve & \text{if } \vec{x} \cdot \vec{w} + b \geq 0 \\ -ve & \text{if } \vec{x} \cdot \vec{w} + b \leq 0 \end{cases}$$

desmos calculator

$\Rightarrow$



- Why both equal? ∵ d_1 & d_2 should be equidistance. So that's why we took equal distance from Marginal line.

So Maximise d Such that

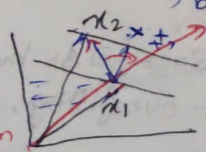
$y_i(\vec{w} \cdot \vec{x} + b) \geq 1$

- Why only 1 ? not (2, 3, 4, 1)
  - because the 1 is not change from RHS
    - $2x + 3y + 3 = 0$
    - $2x + 2y + 3 = 1$
    - $2x + 3y + 3 = -1$

If y_i $2x + 3y + 3 = -5$

$2x + 3y + 3 = -5$ } equal distance from 0

diff $\vec{x}_2 - \vec{x}_1$



So get shortest distance b/w the marginal plane, we need projection of $\vec{x}_2 - \vec{x}_1$ on unit vector 'w' to get d

$$\boxed{d = (\vec{x}_2 - \vec{x}_1) \cdot \frac{\vec{w}}{\|w\|}}$$

$$\Rightarrow \frac{x_2 \cdot \vec{w} - \vec{x}_1 \vec{w}}{\|w\|}$$

→ where x_1 & x_2 are support Vectors. they lie on marginal hyperplane

Constraint $y_i(\vec{w} \cdot \vec{x} + b) = 1$

for +ve class $y_i = 1 \rightarrow 1 \times (wx_1 + b) = 1 \Rightarrow wx_1 = 1 - b$

Putting ① & ②

$$\frac{(1-b)-(b-1)}{\|w\|} = \frac{2}{\|w\|} = d$$

for -ve class

$y_i = -1$

$-1(wx_2+b) = -1$

$wx_2 = -b-1$ — ②

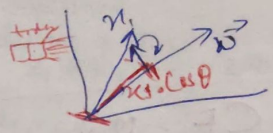
So Cost function

$$\underset{w,b}{\text{maximize}} \ \frac{2}{\|w\|} \quad \text{such that} \quad y_i(\vec{w}\cdot\vec{x}+b) \geq 1$$

magnitude of coefficient

Modified Cost function

$$\text{Minimise} \ \frac{\|w\|}{2} \ \text{by varying } w \text{ \& } b \quad y_i(wx_i+b) \geq 1$$



→ Cost function for Soft Margin will be

$$\underset{w,b}{\text{minimize}} \ \frac{\|w\|}{2} + C \sum_{1} \xi_{pi}$$

hing loss

misclassification error / margin error called

$\frac{\|w\|}{2}$: SVM error called

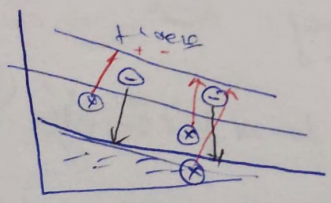
C : no of misclassified dp (Hyper parameter)

ξ_{pi} : zero for correctly classified dps; it is the distance of all misclassified dps due to new marginal plane

$C = 1$ in sklearn

$\xi \to s_{ki}$

ξ_{p-zero}



Such that

$$y_i(W^T x_i + b) \geq 1$$

+ve dp from there own plan & -ve dp distance from their own plan → So summation of all misclassified dp from their plan

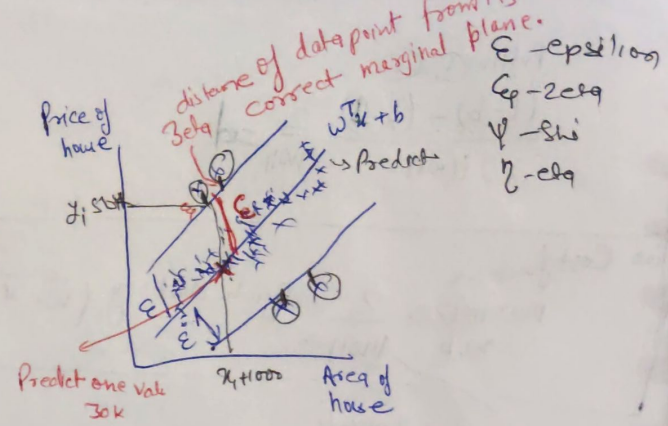
Support Vector Regressor

SVR

→ We want all dp's in surrounding of best fit line.

⇓

$w^T x + b + \epsilon$ to $w^T x + b - \epsilon$



for Regression

$$CF = \underset{w,b}{\text{Minimize}} \frac{\|w\|}{2} + C\sum_{i=1}^{n}(\zeta_i + \zeta)$$

Constraints

$$|y_i - y_{pred}| \leq \epsilon + \zeta_\varphi$$

$$|y_i - w^T x_i| \leq \epsilon + \zeta_\varphi$$

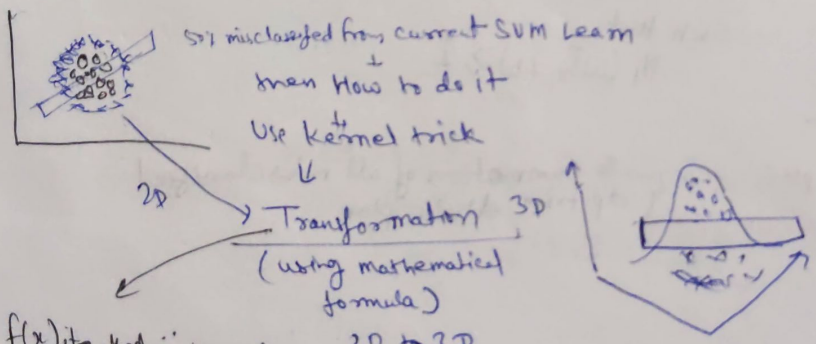
$y_{act} - y_{pred} = \text{Error} = \epsilon + \zeta_\varphi$ ← minimize this

→ It done in backed by gradient descent

- All the dp's in b/w of the marginal plan.

ζ_φ - is the distance of dp from its correct marginal plane.

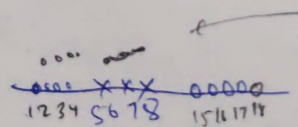
SVM → Kernel trick



f(x) it called ⇒ Kernel or Kernel transformation

eg

x	y
2	yes
3	no
4	yes
5	no



Idea → To change the data from 1D to 2D or increase the dimension

→ Kernel trick → Lazy → you are not sending the data in higher dimension you are using mathematical transformation to achieve it.

(called)

(1) Polynomial kernel:

$$f(x_1, x_2) = (x_1^T \cdot x_2 + c)^d$$

d is degree of polygon

c → constant

x_1, x_2 → features

$x_1^T \cdot x_2 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \cdot \begin{bmatrix} x_1 & x_2 \end{bmatrix}$

$\begin{bmatrix} x_1^2 & x_1 x_2 \\ x_2 x_1 & x_2^2 \end{bmatrix}$

Electric Vehicle Data Analysis

(1) Problem Statement
or orally

(2) radial basis feature (rbf)

- Create non-linear combination of features to bring feature higher dimension.

$$f(x_1, x_2) = e^{-\frac{||(x_1 - x_2)||^2}{2\sigma^2}}$$

$||x_1 - x_2||$ → Euclidean distance b/w x_1 & x_2

σ ⇒ Variance.

(3) Sigmoid kernal

$$f(x) = \frac{1}{1+e^{-x}}$$

* How to choose right kernel
↓
Hyperparameter tuning

Which on Choose

↳ (1) distinguishable → linear kernal

(2) use → Rbf kernal

(3) → use polynomial kernal

(4) Sigmoid → most for binary

PHD
* Bessel kernel → $f(x,y) = J_{(v+1)}\frac{(\sigma|x-y||)}{(x-y)^{-n(v+1)}}$
* ANOVA kernel → $f(x,y) = \sum_{n=1}^{n} \exp\left(-\sigma(x^k - y^n)^2\right)^q$

9 June Naive bayes & SVM

→ NB
→ Ensemble
↳ bagging / Boost
→ KNN
↳ PCA
↳ Unsupervise learning
↳ Anomaly detection

2 hr SVM implement

2:30

Naive Bayes (only for classification)

* Prob: how likely something is to happen $P = \frac{\text{No of favourable outcome}}{\text{Total no of outcome}}$

$P(H) = 1/2$

• Independent Events & dependent events.
(outcome of an event is dependent on another event.)

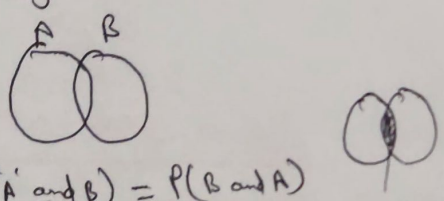
↓

outcome of one event is not effected by another Events

$$P(A/B) = P(A) * P(B/A)$$ → Conditional Probability

↓

* Bayes theorem



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

So $P(A/B) \cdot P(B) = P(B/A) \cdot P(A)$

Bayes theorem

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

$P(A/B)$ = Prob of event A given

$P(A)$ = Prob of event A, B has occurred

$P(B)$ = " " event B,

$P(B/A)$ = Prob of event B given A has occurred.

example

x_1	x_2	x_3	y

→ what is post A B → $P(y/x_1x_2x_3) = \frac{P(y) * P(x_1x_2x_3/y)}{P(x_1x_2x_3)}$

$$P(y/x_1x_2x_3) = \frac{P(y) * P(x_1/y) \cdot P(x_2/y) \cdot P(x_3/y)}{P(x_1) \cdot P(x_2) \cdot P(x_3)}$$

y has two possibility

$P = 0.7$ — Yes — $P(\text{yes}/x_1x_2x_3) = \frac{P(\text{yes}) * P(x_1/\text{yes}) * P(x_2/\text{yes}) * P(x_3/\text{yes})}{P(x_1) P(x_2) P(x_3)}$ — Constant

$P = 0.3$ — No — $P(\text{No}/x_1x_2x_3) = \frac{P(\text{No}) * P(x_1/\text{No}) * P(x_2/\text{No}) P(x_3/\text{No})}{P(x_1) P(x_2) P(x_3)}$ — constant

NB → P-0.7 (Yes)

* for multi class Classification

$$P(C_k|x) = \frac{P(x|C_k) \cdot P(C_k)}{P(x)}$$

$$P(C_1|x_1,x_2,x_3) = \frac{P(C_1) * P(x_1|C_1) * P(x_2|C_1) * P(x_3|C_1)}{P(x_1)\,P(x_2)\,P(x_3)}$$

$$P(C_2|x_1,x_2,x_3) = \frac{P(C_1) * P(x_1|C_2) * P(x_2|C_2) * P(x_3|C_2)}{P(x_1)\,P(x_2)\,P(x_3)}$$

$$P(C_3|x_1,x_2,x_3) = \frac{P(C_3) * P(x_1|C_3) * P(x_2|C_3) * P(x_3|C_3)}{P(x_1)\,P(x_2)\,P(x_3)}$$

Max Probability → Predicted class

↳ example

↳ Outlook data sets

3:10

Outlook	Yes	No	P(E/Y)	P(E/N)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0
Rain	3	2	3/9	2/5
	9	5		

for Temp

Temp	Yes	NO	P(E/y)	P(E/N)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cold	3	1	3/9	1/5
	9	5	1	

Play (Y/N):
Y - 9 P(Ys) = 9/15
N - 5 P(Ns) = 5/14

↳ $P(\text{yes}/\text{Sunny, Hot}) = P(Y) \cdot P(S|y) \cdot P(H|y)$

$= 9/14 \cdot 2/9 \cdot 2/9 \Rightarrow \frac{2}{63} \Rightarrow 0.031$

$P(N/S,H) = P(N) \cdot P(S|N) \cdot P(H|N)$

$= (5/14) \cdot 3/4 \times \frac{2}{5} = \frac{3}{35} = 0.085$

↳ No Normalized the data because both has no number of equal datapoint
So → Normalized the prob to get in percentage for P(y/Sunny, Hot)

$$P(y/\text{Sunny, Hot}) = \frac{0.031}{0.031 + 0.085} = 0.27 = 27\%.$$

$$P(N/\text{Sunny Hot}) = \frac{0.085}{0.031 + 0.085} = 0.73 \Rightarrow 73\%.$$

Prediction ⇒ The person will not go to play football

Why NB called → because here use Bayes theorem

↳ Why Naive → It assum that features are independent
→ No Multicollinearity

→ So NLP use NB → because of independents

Adv:
- based on probability
- fast Algorithm (only calculation)
- work well with less training data
- Performance is good

DisAdv
- Assumes features are independent
- It doesn't work for Regression problem.

⇒ Variant of NB

(1) Bernoulli NB

data f1 f2 f3 → Independent Variable follow Bernoulli, Barnoulli NB.

M yo bn
F
M No bats

$\Rightarrow \begin{bmatrix} f_1 & f_2 & y \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ ⇒ it is called = Sparse Matrix (where max-M 0 and 1)

also it leads Overfitting
(because it not proper algos it only memorize thing)

(2) Multinomial NB

when inputs are → text data

eg: ↳ mail classification → spam/ham

eg: I am going to play football

you think play & football dependent Answer is No why because I also possibility to playing Played any possibility.
(but also include context)

eg:-

Email body	o/p output
you got a discount	spam
millions of Rs	spam
after leti	Ham

⇓

Charge (Numerical Value) Vectors ⇒ NLP technique
1. Tf-Idf
2. BOW
3. Word2Vec

- Multinomial and Bernoulli is use for textual data in NLP

(3) Gaussian NB

→ If features are continous & follow Normal Distribut

- Age/Sal/fn
- if some feature use Bernulli distribution or some Normal Distrn use that Variant which every feature is higher.

Lec-9 Random forest & Bagging 9 june

Which Model use
- Linear data → MLR, Logis
- Non linear → DT, SVM

In DT → overfitting
{ Pre pruning / Post → Ensembles

How it works in industry → All model use

- eg DS Jobs → M BA, M3 BA, M2 BA, M1 BA, Ajay DA

⇒ Ensembles and Its techniques

↳ Combine multiple Models

data
↳ model 1
↳ m2
↳ m2
↳ m3

→ Combine → Prediction (maybe Same Alg or diffrent)

for Prediction: more stable and accurate as compared to Individual model.

- One person might give you wrong advice
- multiple mentors → Chance of getting wrong Less/minimize

⇒ of Same Algo
1. DT_1 → max depth = 5
   DT_2 → " " = 10
   " " = 12

different algos Also
- logistic Reg
- SVC
- DTC(5)
- DT(5)
- NBC

- NOTA TTS
- Delitte
- ZS
upgrad

- So Ensemble → No necessary only one type of Algorithm

1. Parallel Technique (Bagging) → (Random forest)
2. Sequential tech (Boosting) → (Ada Boost, Gradient b, XGBoost)

→ DT_1 → DT_2 → D3 (wrong pred) → more weight on DT on wrong prediction data

(1) Parallel technique of Ensembles

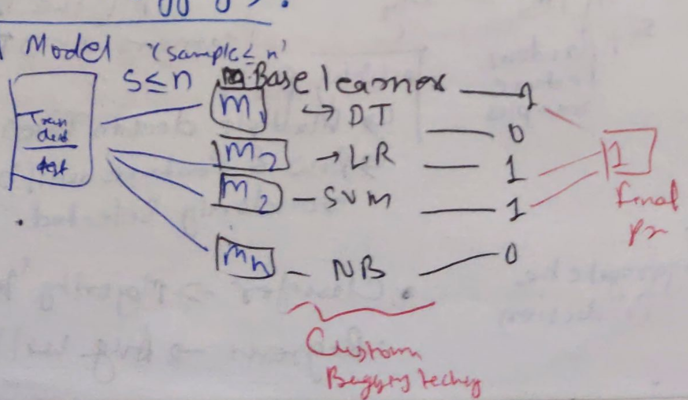
data → Log, tree, SVC, n

- Here all the models here are built parallely and independent of each others.

(2) S

* Bagging (Custom Bagging):

- Parallel Model (sample ≤ n)



Same family
DT, DT2, DC
Called Random forest

→ Custom Bagging class

use Pipeline & Column transformer

A Sequence of data transformation ⇒ Pipeline

↓↓

Set of all feature

↓↓

Colum Transform (group all the pipeline steps for each of the Column)

- Model training Pipline & Colum. ipynb
  - handing missing Value
  - data encoding
  - Feature scaling

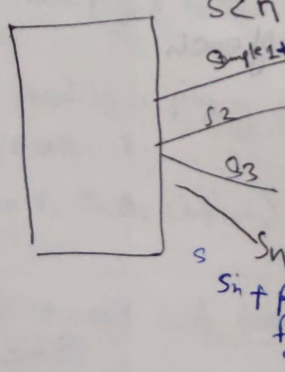
→ from sklearn.compose import ColumnTransformer

→ from sklearn.pipeline import Pipeline,

```
>>> num_pipeline = Pipeline(steps =[('imputation', SimpleImputer(strategy="median")),
                                    ('Scaling', StandardScaler())])

>>> Cat_pipeline = Pipeline(steps =[("imputation"), SimpleImputer(strategy
                                    ('encoding', OneHotEncoder())])  ="most_frequent")

=>> ColumnTransformer( [("num_pipeline", num_pipeline, num_cols),
Preprocessor           ("Cat_pipeline", Cat_pipeline, Cat_cols)])

X_Train = preprocessor.fit_tranform(X_train)
X_test = preprocessor.fit_tranform(y_train)
```

Random forest Classifier and Regressor

(Bagging technique)



data can be repeated across the sample.

Called → Bootstrap Sampling

→ IN RF
↳ All the base learners are DT.

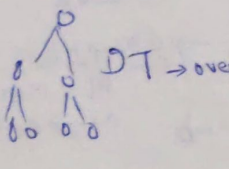
- Why RF?
  - ↳ Multiple decision Trees in parallel
  - ↳ Row & feature will be randomly Selected.

- Bagging: (
  - Bootstrap ↳ different Samples with replacement ↳ Bootstrap Sampling
  - Aggregating ↳ Aggregate the Prediction

- Classifier → Majority 'k' the prediction
- Regressor → Avg will be the Predi

→ High Variance
↳ use bagging
→ Reduce Bias
↳ use boosting

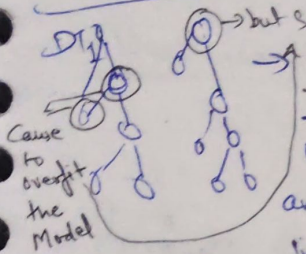
* Random forest reduces the Variance



DT → overfitting → high Variance

Random forest
↳ Random sampling of rows & features
→ data is splitted into small chunks (Randomly)

June 14 data to last

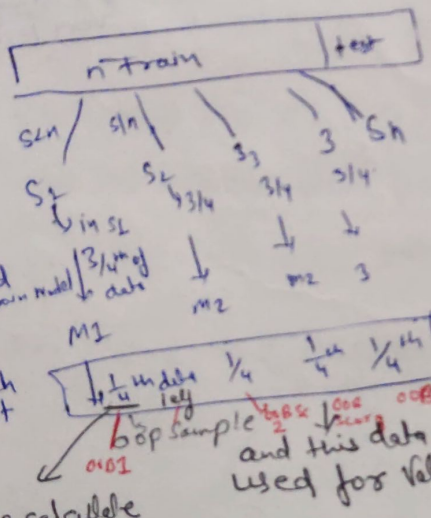


→ but same data not overfit here why
→ feature/Row which was causing an overfitted model in ONE DT might not be contributing to overfitting in other DTs.
↓↓
reduce the Variance

• Due to Random subset of rows and features each of the subset is different representation of itself.
↳ Each of random subset trains a diff. model.

⇒ OOB Score :

Out of Bag



So Also calculate Validation Accuracy Score → called → OOB Score

• In training itself OOB is low ~~if self OOB~~. Model is not performing well on train data.

14 June toolkit

→ Use Numpy in Real World
↳ go in depth
↳ where Use these thing

⇒ Reshape

• 3 diffrent way to create Array (
• if you have list of 2D [C

• Note book
↳ large ho jaye
So. Structure the Code

Escap + B
M → Markdown
Y ⇒ Code

escape
Shift+tab

⇒ for loop vs Vectorized operation, max operation?

```
>> list_2D = [[1,2,3],[4,5,6]]
>> arr = np.arr(list_2D)
>> type(list 2D)
>> type(arr) or arr.dtype ⇒ use for data type
```

• np.full() → np.full((n,m), val, dtype = "float")

np.ones((n,m)) * 5 → Concept of vectorization use
also use broadcasting
↳ use valid broadcasting

(Q2) Generate an array of 100 evenly spaced
np.linespace(1, 100, 100)

list(range(1,10))

(Q5) np.asarray & np.array, np.asanyarray
↳ यहां Copying से बचता है
→ Avoid copying array

→ ,!! Copy hoke Same के होते हैं than again wo array के लिए भी अलग space leta hai

when we use adv obj & also Maintain their functionality (Called Sub Classing) Properties
तब asanyarray Usefull होगा।

np.expand_dims() ↑ wes np.asanyarray)
↑ directly not use by you always very less.

np. id(arr1), id(arr2)
id(arr1), id(arr3)

(Q)
```
np.random.rand(3,3)
>> np.random.uniform(5, 20, (3,3))
                                  not include
>> np.random.randint(low, high, (5,6))
```

→ Compare all
arr[arr%2]
→ These type of thing is called filtering)

broadcasting → element wise → == operation
>> arr%2 == 0 ← boolean indexing
↓
Now then use filtering
arr[arr%2 == 0]

>> np.where
↙ not use index
जहाँ True और False
नहीं होता

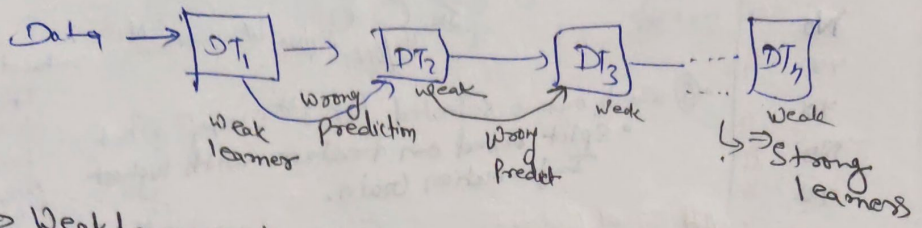
1D, 2D
[true]*3 + [false]*2

Q3 → 3D → max value find → np.random.randint (1, 10, (3,3,3)) 3D
argmax → index
max → number
Q3 → argmax() >> np.argmax(arr_3d, axis=2) Column
axis=1 → Row level
axis=0 → ?? find

→ using in classification Models
→ Max Prob

Ensembles
- Bagging → Base learners (DT)
- Boosting → Week learners (DT) → Add, gradient Boost

⇒ Boosting
1. ADA Boost
2. Gradient Boosting
3. XG Boosting



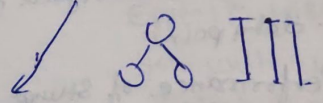
• what do you mean by base learn and week learners.

→ Weak learners :- Model is very simple and not properly from data.

• Principle of Boosting :- Building a first model on training data and build a second model to rectify the errors present in this first Model.

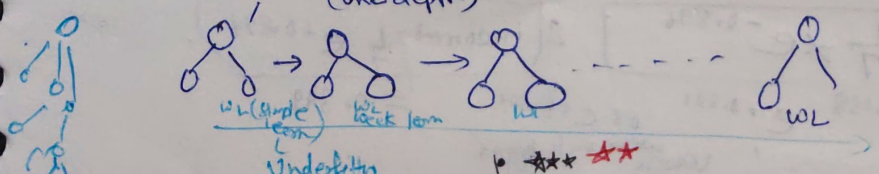
Ada Boost (Adaptive Boosting) (Yoav Freund & Robert Schapire in 1995)
→ Concept of Boosting before SVM we
→ work both for Classification & Regression
→ you know the Boosting work flow
→ Bagging → reduce variance but boosting reduces bias

• In Ada 1 concept → Decision Stumps :- 1 parent node & 2 child nodes that is.



A Decision tree with only one level of depth.

Here why we use binary not a fully grow tree (one depth)



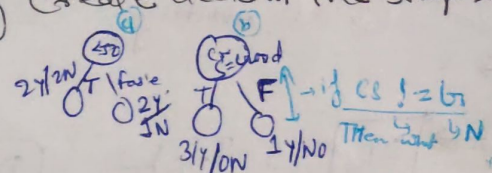
→ to Ada Boost solve the problem of high bias.

★★★ ★★
• each individual DT is a high bias model but error is transferred to next model leading to low bias at an overall level.

#Mathematical Explantion of AdaBoost

Salary	Credit Score	Approved	w_i
≤50K	B	No	1/7
≤50K	G	Yes	1/7
≤50K	G	Yes	1/7
>50K	B	No	1/7
>50K	G	Yes	1/7
>50K	N	Yes	1/7
≤50K	N	No	1/7
			Σ 1

Steps (1) Create decision tree Stump & predict



How many No & yes

So based on it This Predict Wrong because Actual is yes

- Which one is selected for 1st stump (a) or (b)
  - Split based on feature with highest Information Gain.

All equal Weights

Step2 → Assign equal weights to all the datapoints

Updated wt ex: 0.058, 0.058, 0.058, 0.058, 0.034, 0.058, 0.69

different in wt so we normalize the wt

Normalized wt = Update wt / total updated wt

0.058/0.69, 0.058/0.69, ..., 0.058/0.69 = 0.08

Step-3 → Total error & performance of stump.

Total Error = 1/7 (Sum of weights of misclassified dp's)

- So what is the performance of Model?

Performace of stump = $\frac{1}{2}\ln\left[\frac{1-TE}{TE}\right]$ → more TE, less

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

$\Rightarrow \approx 0.896$

- Boosting Prediction function

$f = \alpha_1 m_1 + \alpha_2 m_2 + \cdots \alpha_n m_n$

α = weight of Model which is performance of Model

* Performce of individual stump (Model) will be considered as weight for that specific model.

Steps: So Next step is to give more priority to wrog predition

Step-4 Give more weights to incorrect classified dp's & lesser weights to correct datapoint

* for correct classified: $wt_{new} = wt_{old} * e^{-\text{Performance of stump}}$

* for incorrect classified: $wt_{new} = wt_{old} * e^{+\text{Performance of Stumps}}$

So Correct → $\frac{1}{7} * e^{-0.896}$ & incorrect = $\frac{1}{7} e^{+0.896}$

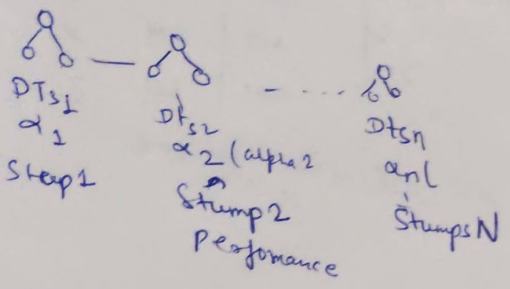
$= 0.058\ e^{-0.896}$ or $e^{+0.896}$ which one is bigger

$= 0.349$

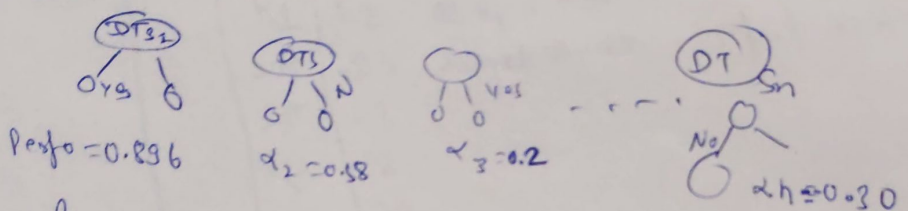
Error → transfer to next DTs to Reduce Bias

Row	Bin Assignment	Random No (0 to 0.98)
0.08	0 – 0.08	
0.08	0.08 – 016	
0.8	0.16 – 024	
0.08	0.24 – 0.32	
0.32	0.32 – 0.90	
	0.90 – 0.98	

Note
The row with higher bin range due to misclassified will be repeated
↓
it means more priority on misclassified.



final Prediction



$f = \alpha_1(m_1) + \alpha_2(m_2) + \alpha_3(m_3) + \ldots\ldots + \alpha_n(m_n)$

$\Rightarrow 0.896(\text{yes}) + 0.55(\text{No} + 0.8(\text{yes}) + \ldots 0.30(\text{No})$

$\Rightarrow 1.15(\text{yes}) + 0.25(\text{No})$

$\Rightarrow$ So final Prediction $\Rightarrow$ yes ✓

- when to stop trees → by define the no of tree)

Adaboost Regressor: All the steps will be same expect instead of Information gain, you will use variance reduction to select the decision stump.

Gradient Boosting

↳ Regression
↳ Classification

Exp	Degree	y_{act} Salary	$\hat{y}$ Predict
2	Btech	50K	75
3	M	70K	75
4	M	80K	75
5	Phd	100K	75

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

Step1 :- create a base model 75

Step2 : Compute the Residual/error from $\hat{y}$ actual.

Step3: Make a dt considering input as the indepent variable (Exp, degree) and target variable as R_1.

⇓

Using this decision tree make prediction for input variables.

$y_{act} - y_{predicted}$ → Residual Prediction

R_1	R_2
-25	-23
-5	-3
5	3
25	20

as R_1 is taken as target variable.

Why Gradient Called

⇒ Gradient Boosting Classifier

- ① → Base prediction eg: 0.5
- Instead of Variance Reduction using Information Gain in Decision tree.

XGBoost (Extreme Gradient Boosting)

Classifier / Regressor

Steps • Create a Base Model (0.5)

Step 2: Construct a decision tree

Step 3:- Calculate Similarity weight

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

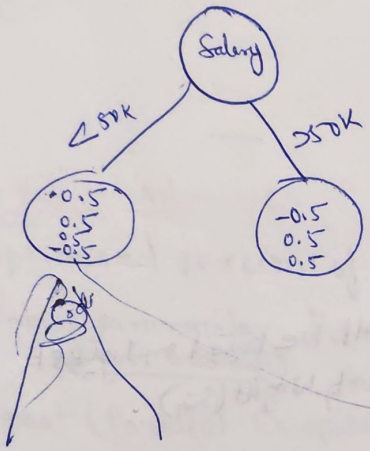
• XGBoost Classifier

Salary	Credit Score	Approved
<=50K	B	0
<=50K	G	1
<=50K	G	1
<=50K	B	0
>50K	G	1
>50K	N	1
>50K	N	0
<=50K		

Base model	Residual
0.5	0.5
0.5	0.5
0.5	0.5
0.5	-0.5
0.5	0.5
0.5	0.5
0.5	0.5
0.5	-0.5

Base Model → 0.5

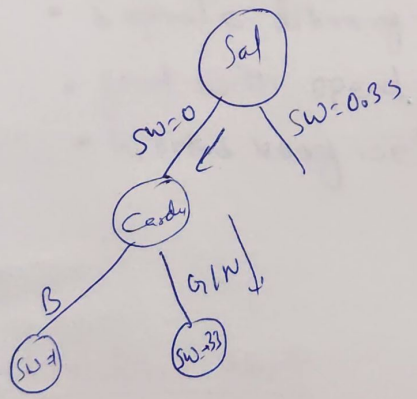
Residuals → (-0.5, -0.5, 0.5, 0.5, 0.5, 0.5)



$\lambda = 0$

$$\Rightarrow SW_{root} = \frac{[-0.5+0.5+0.5+(-0.5)+0.5+(-0.5)]^2}{0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)}$$

$$\Rightarrow \frac{(0.5)^2}{0.25 \times 7} = \frac{0.25}{1.75} = 0.14$$



$$SW_{Left\ child} = \frac{(-0.5+0.5+0.5-0.5)^2}{0.5 \times 3} = 0$$

$$SW_{Right\ child} = \frac{(-0.5+0.5+0.5)^2}{0.5 \times 3} = \frac{0.25}{0.75} = 0.33$$

$$SW_{gain} = (SW_{LC} + SW_{RC}) - Root$$
$$= (0+0.33) - 0.14$$
$$= 0.19$$

* Whichever feature has the highest Similarity Weight gain, the split will happen on that feature.

→ O/P = σ(base + α(SW))

⇒ σ(0 + 0.2(0.33))

⇒ $\frac{1}{1+e^{-0.33}}$

⇒ 0.508

• XGBoost Classifier → O/P ⇒ σ(Base Model + α(SW))

output

$\frac{1}{1+e^{-x}}$ ↓ $\log\left(\frac{p}{1-p}\right)$ ↓

XGBoost Regressor:

Exp	Loan	Salary	Base Prediction	Residual
2	yes	40	51	-11
2.5	yes	42	51	-9
3	No	52	51	1
4	No	60	51	9
4.5	Yes	62	51	11

Base Prediction $= \frac{(40+42+52+60+62)}{5}$

$\Rightarrow 51.2 \Rightarrow 51$

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

SW_{LC} (Left child) $= \frac{(-11)^2}{1+1} = \frac{121}{2} = 60.5$

SW_{RC} (Right child) $= \frac{(-9+1+9+11)^2}{4+1} = \frac{144}{5} = 28.9$

$SW_{gain} = (SW_{child}) - SW_{root}$

$= (60.5 + 28.8) - 0.166$

$= 89.133$

~~Same for the Experience feature Calculate.~~

Now calculate with Exp. 2.5

-11,9,1,9,11 → Exp

≤2.5 → -11,-9

>2.5 → 1,9,11

$\rightarrow SW_{root} = 0.166$

$\frac{(-11-9)^2}{2+1} = \frac{400}{3} = 133.3$

$\rightarrow \frac{(1+9+11)^2}{3+1} = \frac{21^2}{4} = 110.25$

$SW_{gain} \Rightarrow 133.3 + 110.25 - 0.166$

≈ 244

*Split will be based on highest Similarity Weight (SW).

here calculate for 2

R, [-11,-9,1,9,11]

Exp

≤2 → -11

>2 → -9,1,9,11

(Similarity weight)

$*SW_{gain} = \frac{(\Sigma R)^2}{\Sigma Pr(1-pr) + \lambda}$

Pr = Probability

$*SW_{reg} = \frac{(\Sigma R)^2}{\text{No of Residuals} + \lambda}$

$\lambda = 0$

$$Sw = \frac{\Sigma (\text{residual})^2}{\Sigma Pr(1-Pr) + \lambda}$$

↑ hyperparameter

↑↑ Sw ↓↓

↓↓

Chances are there Swgain can become negative

↓↓

No Swgain ⇒ Stop Splitting

★ $\Sigma Pr(1-Pr)$ ⇒ Cover Value

(any Sw less $Pr(1-Pr)$, we stop splitting)

⇒ XGBoost Advantage

(Extreme)

- optimised version of Gradient Boost
- Inbuit parameter λ (regularization) (L2 in Gradient)
- fast (Parallel Computation)
- Seprate library itself
- good with speed & accuracy
- Works very well with large dataset.

Gradient Boost

- Computationally expensive
- prone to overfitting (better than Ada Boost)
-

★ Ada Boost

- for small dataset
- Sensitive to noise & outlier (Bcz 1 depth use so)

★★ What is significance of lambda in XGB, B

↓↓

what is the means

- regularization parameter.

Remember

- LR/MLR → Ridge, lasso
- Log → L_1, L_2
- SVM → C
- SVR → C
- DTC/DTR → ccp α
- Boosting → α ↑

- Boosted Random forest

- Black box & White box Model.

↳ Model which has explainability power

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$

↳ White Box Model

→ Optimized time Complexity

N → No of training sample

M → No of features

T → is no of tree

d → max depth

- XGB → $O(T \cdot NM \log N)$
- Gradient Boosting → $O(T \cdot NM \log N)$

in practice this is very very fast

↓↓

- Seprate library
- parallel computing

- Ada Boost ⇒ N, M, T

$O(T \cdot NM)$

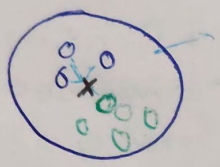
22 June KNN

K-Nearest Neighbour

① KNN Classifier
② KNN Regressor

KNN Classifier

Start with example:

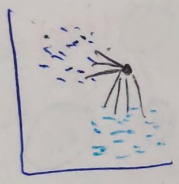


4 nearest point
blue 3
green 1
majority ⇒ blue
so Blue Class ✓

this is test data
No assign the X to class based on distance of 3 dps

Scenario 2
instead of 4 nearest dp take 7 nearest dp.
4 → then Ans → green
K = nearest data point

- So As K changes the class of new dp might also change
⇓
K is a hyperparameter



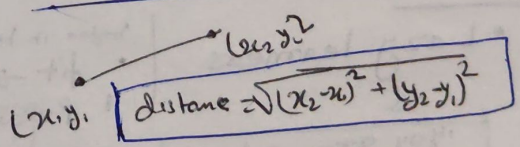
⇒ Sort the distance
0 - 1 } 4 near dps
K=4 0 - 2 } →
0 - 3 }
0 - 3
0 - 7
0 - 9
0 - 9

binary class → K as odd value
3, 5, 7, ...

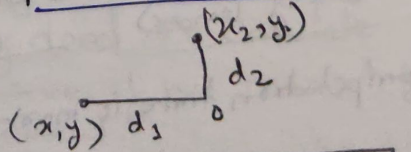
- Algorithms
① Plot the dp in n-d space
② Initialize the K-Value (No of neighbours you want to consider)
K ∈ 1 to ∞ (Generally K>3)
- Calculate the distance of new dps wrt to all dp's
- sort the distance
- based on K find class of that K nearest dp's
③ Find the mode of the class
④ Assign the class.

- Distance

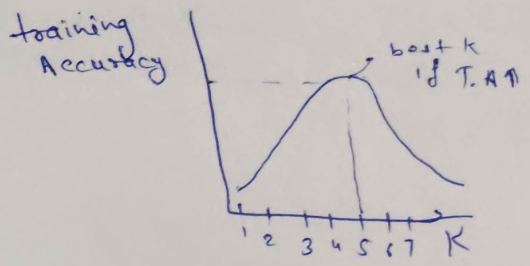
① Euclidean distance



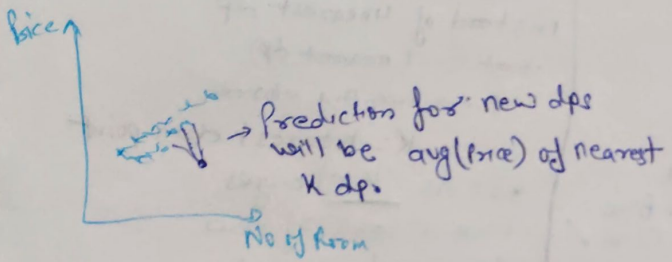
② Manhattan distance



- How to decide K



KNN Regressor



Advantage

- Easy to understand/very intuitive.
- Performance of model in terms evaluative metric is good

Dis Adv

- Lazy learners

"you are calculating parameters of model in real time - knn (Lazy learners)

before in best fit line we
- fit → Coeff's
- Predict.

$B_0 + B_1 X_1 + B_2 X_2 \ldots$

- All the model parameters are calculated while training & used for Prediction. (eager learners)

- Computation time is more.

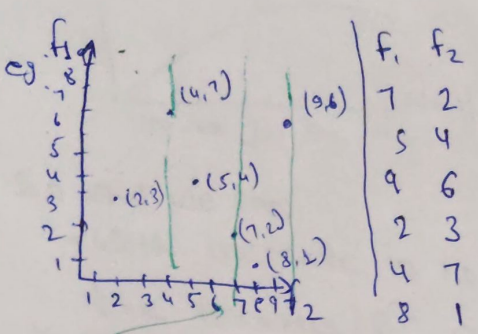
ML
- eager Learns
- Lazy Learns
  - ① Locally Weighted Learning
  - ② Case-Based Reasoning
  - ③ Lazy Bayesian rules

if small dataset → use KNN (No Industry use → because lazy learner

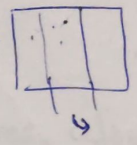
Why KNN lazy
- In real time the distance of test data from each of dp is calculated ⇒ Brute force

Variants of KNN (to optimize the distance)
- K-D Tree
- Ball tree

So as a work in Corporate you optimize the calculation How think.



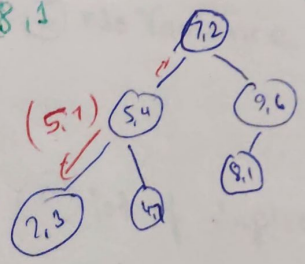
f_1	f_2
7	2
5	4
9	6
2	3
4	7
8	1



f1 - 2, 4, 5, 7, 8, 9
→ median = 7
So partition

(9, 6)
(8, 1)

So 1st partition



1. KD-tree
   - partition the data in binary tree

Step1 :- Sort the feature f_1 & f_2.

Step2 :- Calculate the median of f_1 and f_2.

Step3 : Partition the data based on medium

Step4 : Partion all the data recursively in each of subpartition.
(Partition until in one partition only an dp is there.

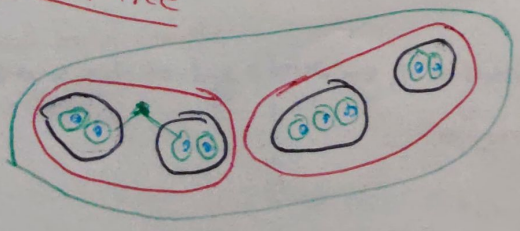
↳ What is the Advantage there

if new dp come (5, 7) So directly come when you know the data
↳ easly track the data points neighbours

- So instead of Calculating Using Searching

1:49

2. Ball tree



Advantage:
- only need to Calculate distance of the nearest group element. (this also use hierarchy)

Cosine Similarity
range (0 to 1)

1 - similar
0 - ...

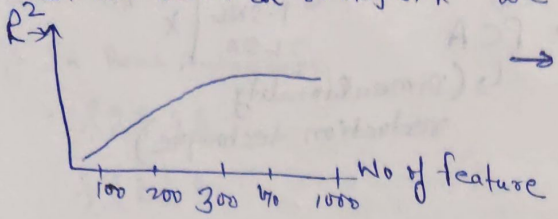
KNN as missing Value Imputer

```
KNNImputer(n_neighbors=2)
from sklearn.impute import KNNImputer
```

23rd June : PCA (Principle Component Analysis)

Example

When we learn R^2 or Adjust R^2 we see



→ 20 feature | 50 f | 100 f | 250 f | 1000 f
Model 1 | M2 | M3 | M4 | M5

$Acc/R^2 < R^2 < R^2_{Acc} < R^2/Acc = Acc/R^2$

So what we see

- With increase in no of features, after one point of the Accuracy / R^2 Stop increasing.

→ So

(Q) Question Why it happen so ?

↳ Maybe

(1) → Multi Collinearity ($f_1 f_2 f_3 \approx f_4$)

(2) → All the entries of features are exacty same

(3) → ~~two~~ feature are Same

f_1	f_2
1	1
1	1
:	:
1	1

(4) No Variance f_1

f_1
1.01
1.02
1.03
1.04
:

f_i { 1 1 : 1 } no variance in data

(and we know ML all about learn Pattern)

(5) lot of duplicate entries

f_1
1
1
:
2
2
2

- So what point I made that the feature we using might be not that much useful.

We know [with increase in no of feature performance of model degrades

↓

(Q) When increase the no of feature why Performance decrease ?

Performance → Execution time, Evaluation

Real world example.

you want to buy a home → go to brocker → buy 3BHK → 60L

↳ then you say → near Airport → 70L

again add Beach → 80L, Mall → 90L, Mkt, Hospital

→ if you add More Brocker confuse what will be price

like that model also confuse when no of feature.

It is curse of dimensionality ~~with~~ increase in no of feature the Performance of Model ~~de~~

⇒ To Remove Curse of Dimensionality:

(1) Feature Selection
- Correlation, Dif, P. value, Select K feature, Information Gain, Covariance, RfE.

(2) Feature Extraction
- PCA (Dimensionality reduction technique)
  - ↳ T-SNE
  - ↳ LDA
  - also put not see X

Disadvantage
- dropping data
- Time Consuming

Q. Why we should remove Curse Dimensionality?
1. To improve performance of Model
2. Visualize the data → for insight (for human eyes $f_1, f_2, f_3 \dots f_{10}$ → 10D → 2D, 3D, 2D as Human)
3. Prevents from overfitting
4. Better Interpretation
5. To Remove Curse of Dimer

31:00

* Feature Selection *

→ Corr, Cov

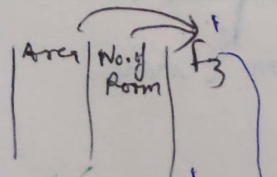
	Area	No. of Room	Price of house
eg			

if we choose or see corr to y & is $f_1, f_2 \dots f_4 \dots$ then what we do with rest of the feature maybe we drop it.

So what we do if not drop

Do

* Feature Extraction : Do data transformation to extract new feature which represents both the feature.



So How we can do this

How you do because you have domain knowledge (common sense)

House Area	Price

lets	# No of Bathroom	distance Airport	Distance from University

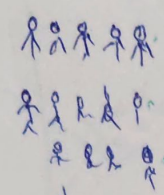
→ So when more feature domain expert also confuse

(Why We PCA Understand then Core PCA

Why PCA come

⇒ PCA

eg:- In Batch Photograph



Total

So how many person caputer in Camera only 4

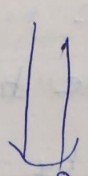
So what I convey is

↳ Variance/Spread is more compare to this side you understand why

Camera man this side

So How many person will Capture → All person

How in Real world works

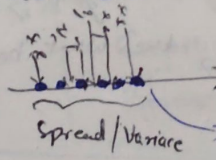


So Same Analogy to Ours features.

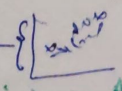
- Now 1 more point you calculate corr & decide to drop Area of house

then what is one thing left. → No of Rooms Only

So only X-axis



Now are data point in X-axis and lost this Variance

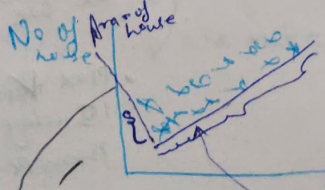


And we know ML all about Pattern So Avoid losing this of information

What is Conclsion

Remove feature loose the data

↳ So Now I do

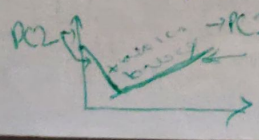


So what we do to capture or variance without loosing data

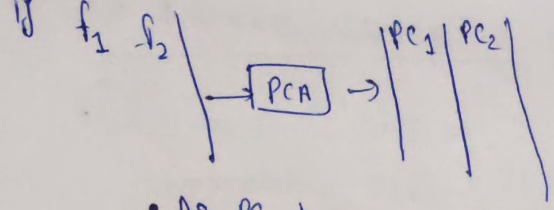
So can we change the axis like this

So know this where we loose more data

So know if use No of Rooms we cover 80% of Varian of Area of house → 2D to 1D → (No of rooms only) without lossing much of Variance/data))



→ only change the axis Coordinate Rotations



- As PC_1 has maximum variance you will select only PC_1 for model training.

(Q) So why not we PC_2 also too → if use both then why we Reduce.

(Q) Sir if we have 1000 feature what we do

$f_1\ f_2\ f_3 \cdots f_{1000}$ ⇝ $PC_1 \mid PC_2 \mid PC_3 \cdots PC_{1000}$

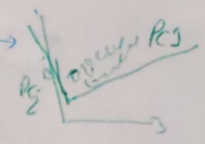
but always remember these 1000 PC has 2 Characteristics

(1) All of these PC's will be ⊥ to each others.

y why → PC2 axis are always ⊥ to each other's.

(2) The Variance Capture by

$PC1 > PC_2 > PC_3 > PC_4 \cdots > P_n$

→ 

(3) No of PCs = No of features

(Q) (if both equal why we use

↳ $f_1\ f_2 \cdots f_{1000} \mid y \mid \rightarrow PC_1 \mid PC_2 \mid \cdots PC_{1000} \mid$

$r \times \frac{1000}{d}$ $r \times 1000$ PC_1 ⇒ then what benefit

use 2nd point ↗ $PC1 > PC_2 > PC_3 > \cdots$

$PC_1\ PC_2\ PC_3 \mid P_c$

70% 20% 9%

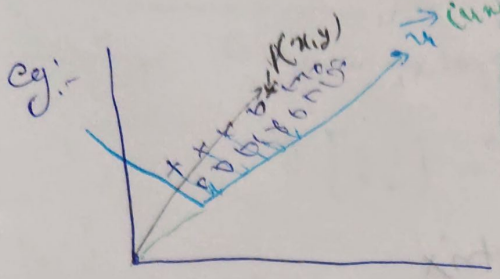
4D → 3d use PC_1, PC_2,

So How many PC we select

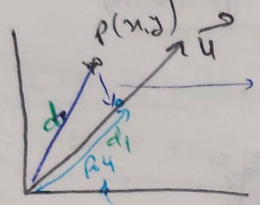
- Max Variance is along PC_1 > then PC_2 >

PCA Maths

* Axis transformation → use → Eigen decomposition of Covariance matrix



- So how we change this axis
  → so we need projection
  → I want a unit vector where I project every data point.
  ↓ So project every data point



you want to projection in $\vec{u}$ using dot product.

$$\text{Proj of P on } u = \frac{p \cdot u}{\|u\|}$$ → P also see in SVM

$\|u\| \Rightarrow 1$

$$\text{Proj of P on } u = p \cdot u$$

d, d_1, d_2 ... are distance from origin in unit vector

$d_1, d_2, d_3, \ldots$ are distance

So what is projection here

↓ you want that unit vector where variance / spread is max.

↓ So you know the variance formula, a ⇒ $\text{Var} = \sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n}$

↓

Aim: to find that unit vector which capture maximum variance after projection.

These are understanding

So know what is Co-variance $\sum_{i=1}^{n} \frac{(x_i - \bar{x})(y_i - \bar{y})}{n}$

what we do if $x_1 | x_2$ → we make → co-variance matrix

	x_1	x_2
x_1	$\text{Cov}(x_1, x_1)$	$\text{Cov}(x_1, x_2)$
x_2	$\text{Cov}(x_2, x_1)$	$\text{Cov}(x_2, x_2)$

or

$\text{Var } x_1$	$\text{Cov}(x_1, x_2)$
$\text{Cov}(x_1, x_2)$	$\text{Var } x_2$

↓ Covariance matrix or Variance / Covariant matrix

if 3 feature the 3 by 3 matrix so on.

So why we calculate this → to capture the maxim Variance.

↓ So whole idea is to capture Spread.

Theorem if you decompose a covariance matrix of feature, you will get eigen value & Eigen vector and eigen vector

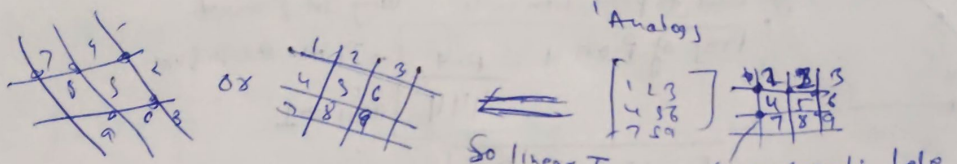
with highest magnitud of eigen Value Capture the maximum variance/spread.

↳ 2nd highest magnitud → 2nd highest Var
→ 3rd highest mo ,, → 3rd ,, ,,
→ 4th ,, ,, → 4th ,, ,,

So thats why $PC1 > PC2 > PC_3 \dots PC$.

(3) then what is the decomposition of matrix

↳ So Go head with concept → that is Linear transformation of matrix



Analogy

So linear Trans Say then same Vectors/matrix look like this (Changing the orientation)

So as coordinate

* Linear transformation

↳ A matrix transformation that brings changes in the co-ordinate system.

$A\vec{V} = \lambda \cdot \vec{V}$

A → Matrix, $\vec{V}$ → eigen Vector, λ → eigen Value, $\vec{V}$ → eigen Vector

(when we use co-varia matrix as A, Eigen decomposition of covariance matrix)

eigen Value, eigen Vector

How it Create.

GeoGebra.org

↳ Linear algebra Visualization

↳ show magnitude

1:47

$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}$

*

⇒ Linear trasformation pointless

- Many vectors were changing both magnitude & direction.

→ few vectors changed only magnitude & not the direction

↳ these vectors are called as eigen vector

eg $\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}$ → eigen (1,0) → (3,0) (same direction, magnitude change)

(-1,-1) → (-4,-4) (same direction, magnitude change)

eigen vector =

Eigen Value are change in magnitude for eigen Vector.

$\vec{AV} = \lambda \vec{V}$

Co-variance — Scalar

##

- The vector which only changes magnitude and not direction will be equals to dimensions of matrix / no. of features.

No of feature = No of PC

DIWF

$AV = \lambda V$ → vector → PC_1, PC_2, PC_3

λ ↓ Stretch / squish

Step to Calculate Eigen Value / Eigenvector (PC's) np.linalg.svd.

Step-1 :- Standardize the data (make the data mean centered)

Step-2 :- Covariance matrix (df.cov())

Step-3 :- Eigen decomposition (np.linalg.svd)

Step-4 : once you decompose $AV = \lambda V$

λ → eigen Value

V → eigen Vector (direction not changed)

1000 feature → 1000 PCs

$PC_1 > PC_2 > PC_3 \dots PC_n$

* Max var will be captured by first few PCs

if more than 90 variance good to go

Implementation: iris data (load_iris)

pca.fit_transform(x_train)

pca.components_ → covariance

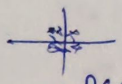
→ pca.explained_variance_ratio_

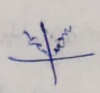
to choose how much PCA.

sum(--) or np.cumsum(explained_variance_ratio_)

⇒ Disadvantage of PCA

- Variance spread across all axis is same → PCA will fail.



- PCA will loose the Pattern  → 

MNIST → handwritten image (0-9)

PCA MNIST dataset

kaggle. ↳ 5K x 785 cols

↳ 50 →

Plt.imshow

How → before / After

↳ PCA do ↳ perform ML then

Mnist dataset with PCA + Randomforest

1 2 3 4 ...