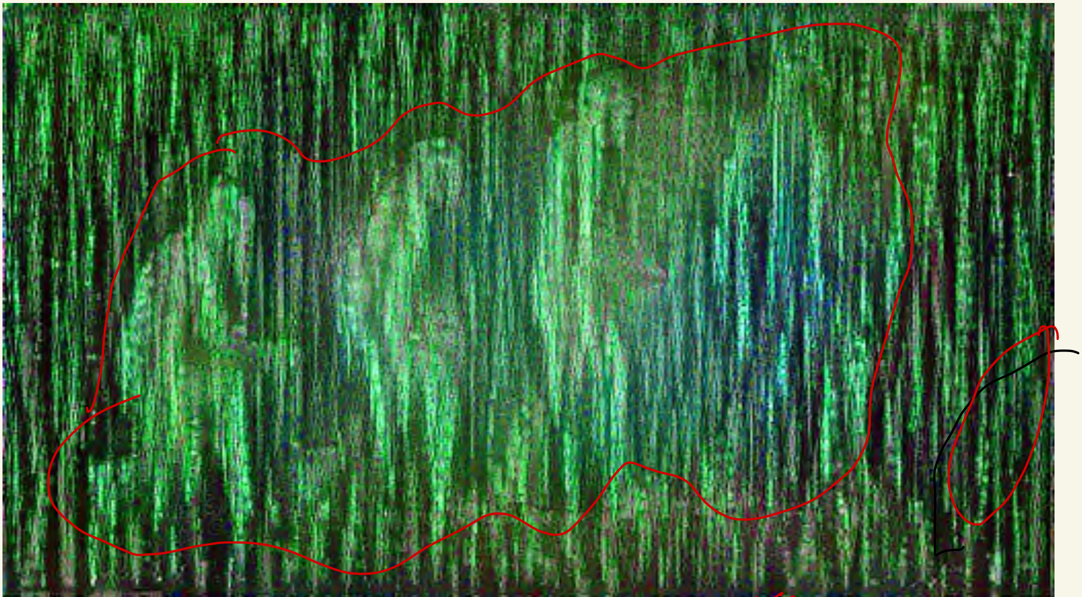
Jupyter → Saptarshi sir
→ Yudi sir
→ Jyoti
→ Satish (DB) ...
→



1) Use Case ...?

Sm
Ran
Scn

11) Why we need AI [Photoshop softw ----].

50-60. (5-8 prd
↳ 800 - 1200 × 8)

1000-100-0

1.) a) finding patter for Occurance Base Prob
b) Scene + AI camera [.....]

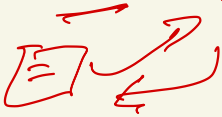
2.) Cong intellMerg

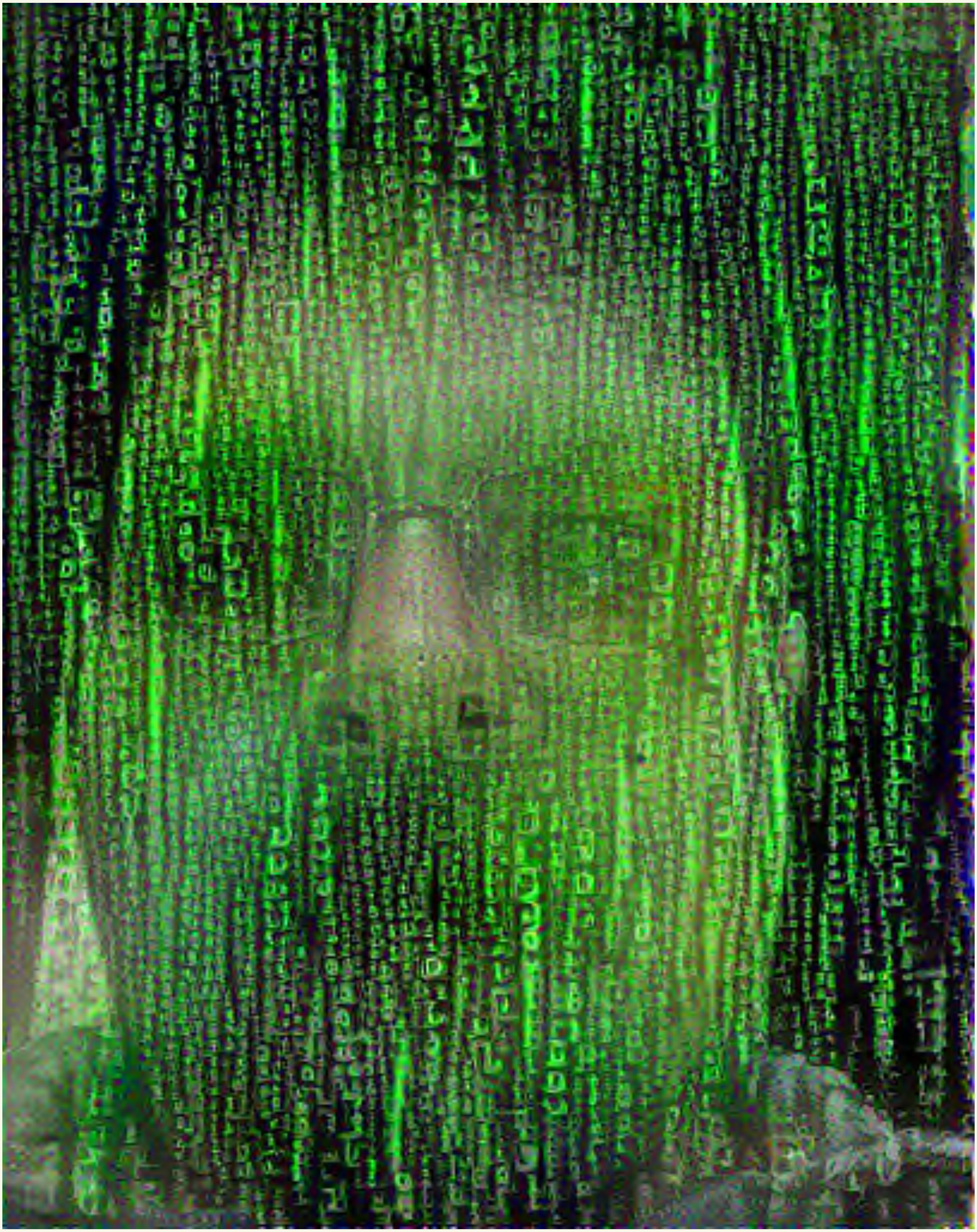
Prd Cus

IT 1 Sn

C++ Pr

Pr. a×b = C

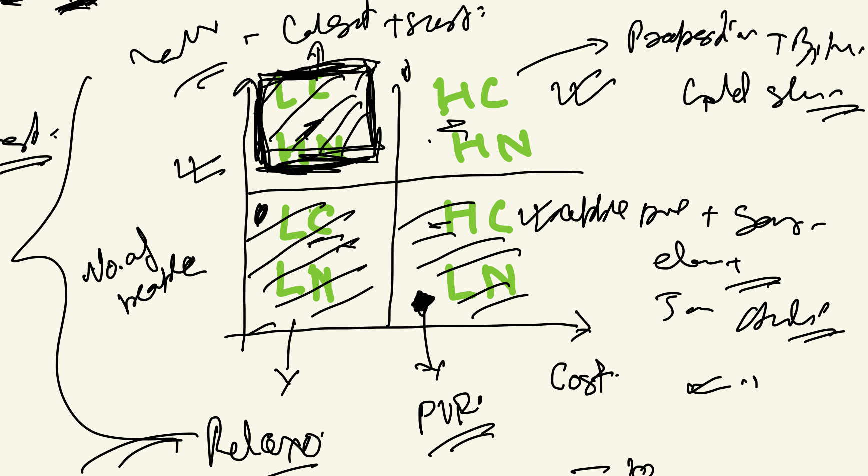




Any Algo:

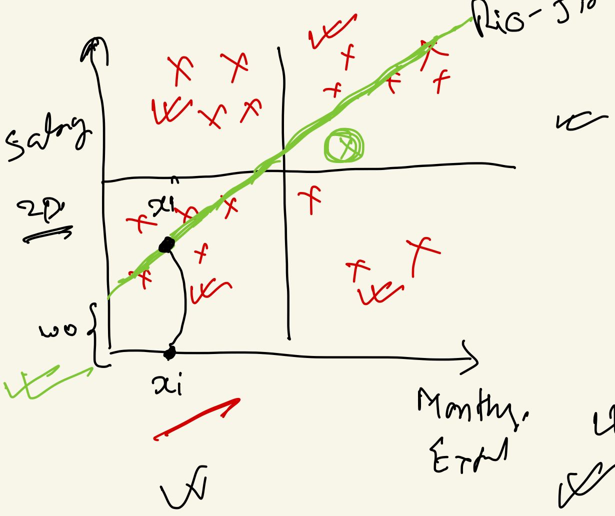
- Maths
  - Geo
  - Prob
  - Losses
- Code
  - Python / R / SAS - - - -

Domain expert:



→ 10

100



→ 100

$y = f(x)$

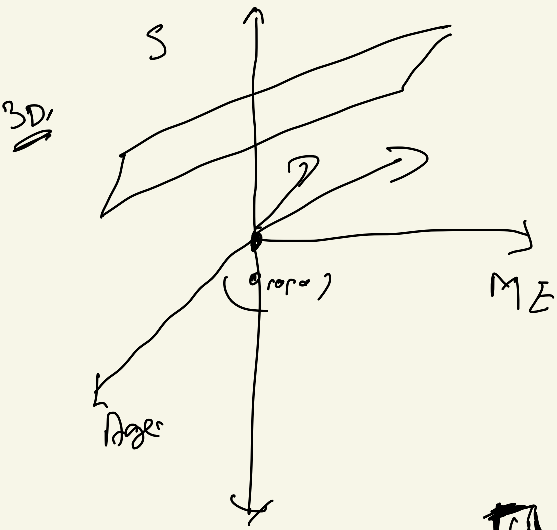
$y = f(100)$

$$\boxed{f(x) \Rightarrow w_0 + w_i x_i}$$

$\boxed{y} = mx + c$

$c = w_0$; $m = w_i$

$x = x_i$



$y = w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i3}$

$\boxed{y_i = w^T x_i + w_0}$

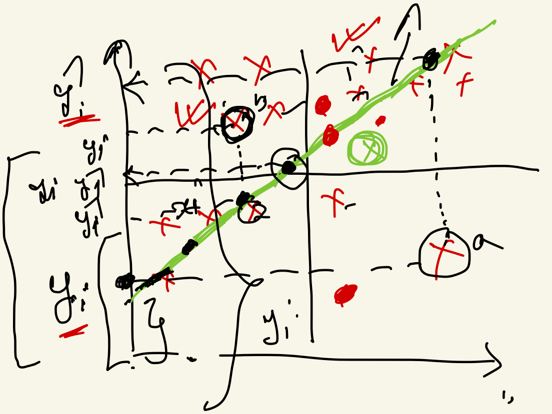
$y_i = w^T x_i$

$w_0 = 0$

$\boxed{y} = mx + c$

50 → P
500 → P ×10
5000 → P → ×1000

hyper



E = errorless

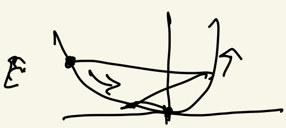
a → $y_i - \hat{y}_i$ $3-8 = -ve$

b → $y_i - \hat{y}_i$ $8-3 = +ve$

$(y_i - \hat{y}_i)^2$ Loss

aim $(y_i - \hat{y}_i)^2 = 0$

$y_i = \hat{y}_i \quad ^2$



[Gradient descent Algo]

→ features

Train 80% y

Test → predict $\hat{y}_i$

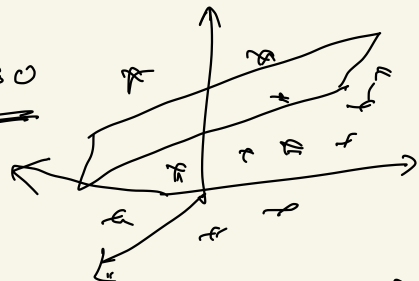
$\| y_{test} - y_{pred} \|$

200 × 6 → Dim

5 → a, b, c, d, e → independent

y → Target / dependent

80-50



$y = mx + c$ $[\hat{y}_i - y_{test}]^2$

↳ loss

100 → 80%

Train Test

70 - 30 %

60 - 40

$\boxed{80 - 20}$

$\hat{y}_i \leftarrow f(x_i)$

Lig leg: ⇒ Raw

⇓

Information

⇓ Predict

→ Numerical: median:

→ Categorical → onehot / get_dum

→ Scaling (N, C):

→ fit to model

→ Predict unseen data

→ Calcul the score:

→ onehot / get_dum

→ Flask (Predictions)

→ EDA (Plotting of data)

RAM

Model

ETL ⇒ DP ⇒ $\boxed{}$ → Test/param

DC

DS

hyper parameters

lr (n_F_ ...)

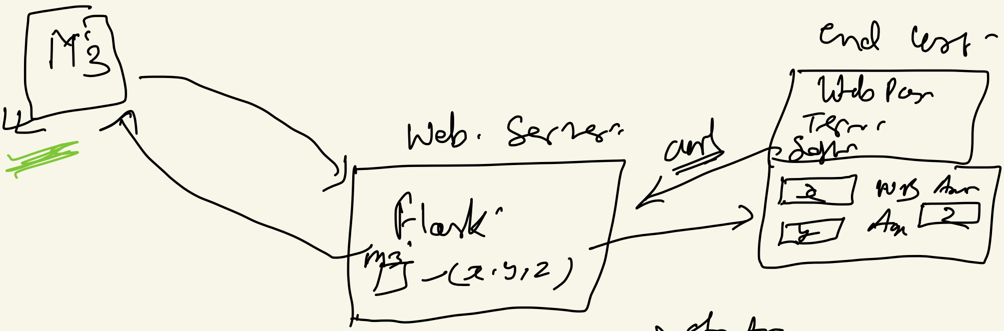
Persist [save it on hard].

import pickle.

x1	x2	x3	x4	x5	x6	x7	x8	y
-	-	-	-	-	-	-	-	-

$A_3 \Rightarrow M_3$

Ac	At
A1	M1
A2	M2
A3	M3
A4	M4
A5	M5



Flask wsgi → Platform

Django ✓ —— Predefined Structure

VM vs Docker

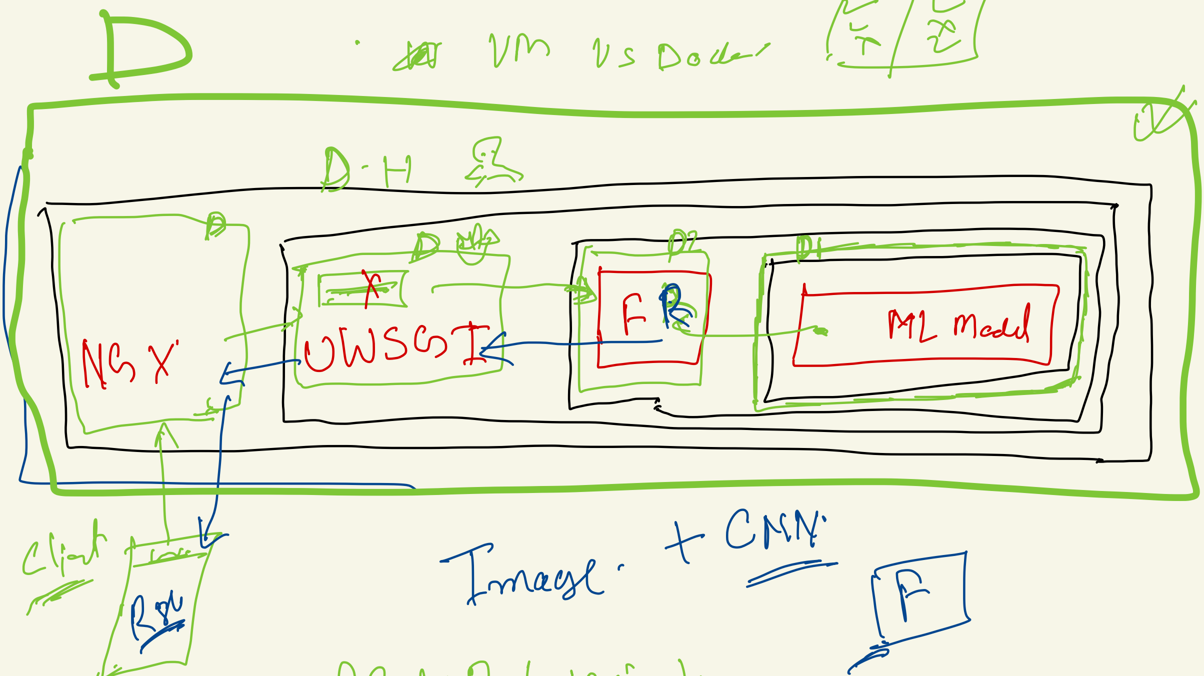


Image + CNN

F

OS + Py (-version).

Win X

Linux ✓

Kali X

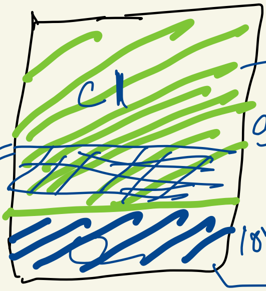
Ubuntu ✓

Centos X

$$Acc = \frac{\text{No. of correctly classified}}{\text{Total no.}} = \frac{8}{10} = 80\%$$

$$\frac{7}{10} = 70\%$$

2-class

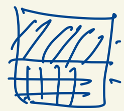


→ T/F ; 0/1

Y/N

x_q → A
↳ B

down sample



$$\boxed{acc = 90\% \text{ or } 0.9}$$

C1+C2 C1. $a_1 = \frac{90}{100} = 90\%$

C2 $a_2 = \frac{10}{100} = 10\%$

up sampling

down sample

10% → 42, 44, 43

30% → a ↓

$\frac{40}{50}$

90
10

20
10

60 - A
40 - B.

Integration author a1 → 0 ← 1

FB → BL

~~SN~~ | ~~EXP~~ | Location | N.Y.E | Grade | Y.J. | P.A | TA | — | ... | ~~Bas~~

20 – 24%

60 – 70%

Regression → Now on

Classification

| 0-5 | 5-10 | 0-2 | 2-5 | 5+ |

75% – 80%

Categorical

AF

O

OT

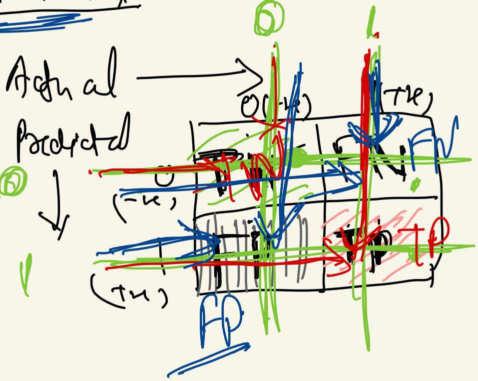
001

101

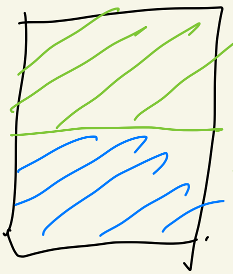
110

10000

Confusion Matrix



TN	A - 0	MP - 0
TP	A - 1	MP - 1
FP	A - 0	MP - 1
FN	A - 1	MP - 0



0 — -ve

1 — +ve

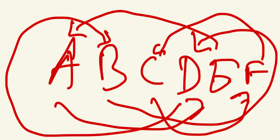
FN → $x_q \rightarrow \boxed{-ve}$ — Negative data incorrectly.

FP → $x_q \rightarrow \boxed{+ve}$ — Posive data incorrect.

$x_q \rightarrow$ -ve → -ve → TN
↳ +ve → FP

$x_q \rightarrow \boxed{+ve}$ → -ve → FN
↳ +ve → TP

		Predicted				
		A	B	C	D	E
Actual	A	TP_A				TN_A
	B		TP_B			
	C			TP_C		
	D				TP_D	
	E	TN_E				TP_E



TPR → TP/P

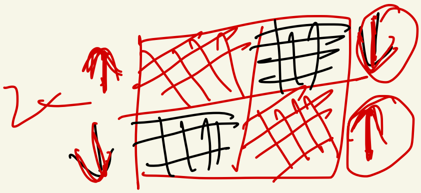
TNR → TN/N

FPR → FP/N

FNR → FN/P

A \ P	0	1
0	TN	FN
1	FP	TP

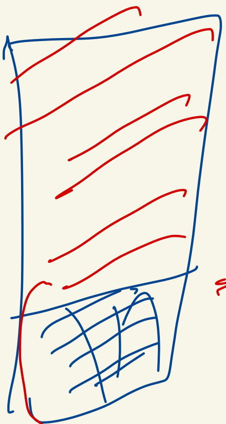
It has falsely predicted the positive values



P

A

	0	1
0	TN	FN
1	FP	TN



UB

ACC

ACC

Good

good

1000

900

100

UB

	0	1
0	900	100
1	0	0

$TPR = \frac{0}{100} = 0\%$

$FPR = \frac{0}{900} = 0\%$

$TNR = \frac{900}{900} \to 100\%$

$FNR = \frac{100}{100}$ 100%.

TN	FN
FP	TP

900	0
0	100

Best of the Best model.