



**Key Metrics:**

1. Accuracy.
2. Confusion Matrix: TPR, FPR, TNR, FNR
3. AUC-ROC
4. Precision, Recall, F1
5. Log-loss
6. Macro averaged & Micro averaged metrics
7. Cross-Entropy.
8. Jaccard Similarity.
9. R2
10. Root Mean Squared Error
11. Mean Absolute error.
12. Percentile Errors.
13. Logarithmic-metrics for regression
14. Explained Variance.

→ True Negative

→ False Positive //

$\rightarrow$  False negative

→ True positive

	P	N
P	True positive	False positive
N	False negative	True negative.









# Recall

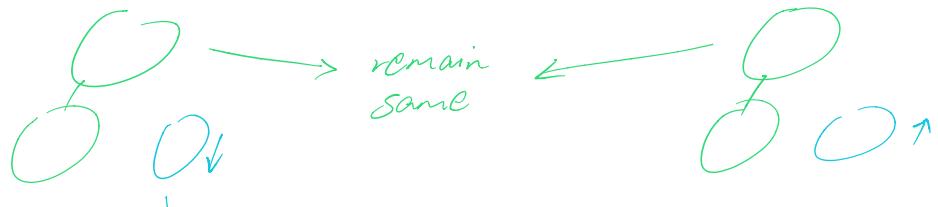
$$\frac{TP}{TP + FN}$$

Out of all positive points in offset, how much is did the model get right?

# precision

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

→ Out of all points that model predicted positive, how much are actually positive?



That means very few simple men are being predicted as pregnant.

&

Lots of pregnant women are being predicted as "NOT Pregnant"



$\alpha$  can be few.  
 $\beta$  can be high.  
And the result would still  
be same.

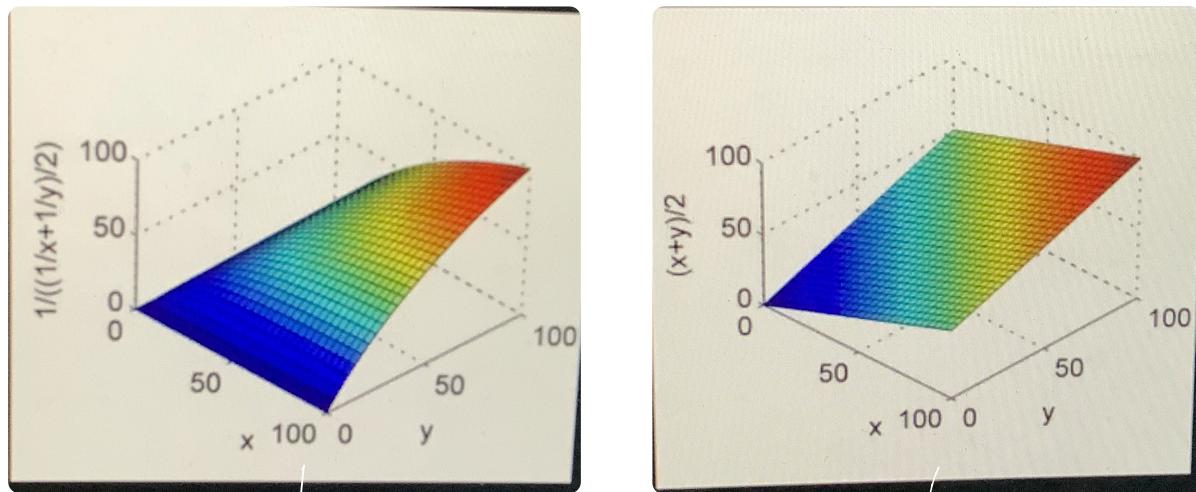
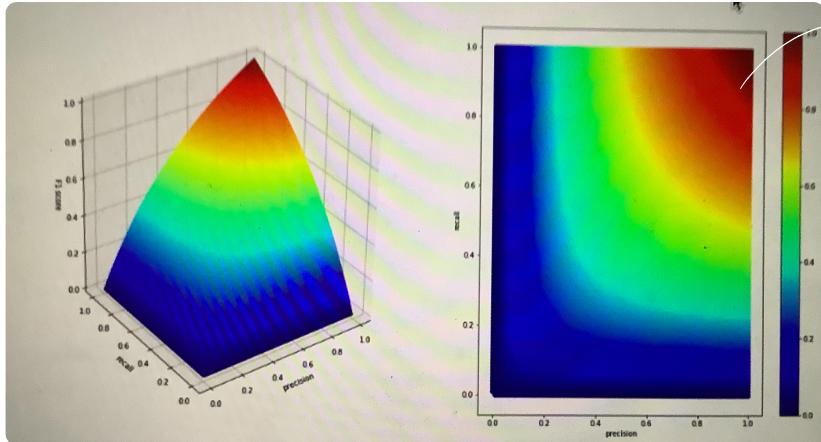
→ One input can overshadow the other  
one and cause the result to go up.

We want our result to be  $\min(A, B)$

but this is  
not differentiable.

So we use harmonic mean as our function

$\hookrightarrow$  F score



If we need, if the situation calls for it,  
we can change the metric's formula to whatever  
the fuck I want. Given that we can justify  
it choice.

↳

→ Custom metrics are used a lot in Kaggle.

→ Better stick to off. metrics because it becomes hard to explain to non-tech folks.

# Logloss:

$$\rightarrow -\frac{1}{N} \sum_{i=1}^N y_i \log p_i + (1-y_i) \log(1-p_i)$$

No fucks given about material as long as you can derive stuff.

Tools are not important, mindset is.

You know C, C++, Java, Python at a basic level? Fuck you.

We need someone with detailed knowledge of one language, that person can always pick up syntax of something else.

Any real company won't give a fuck about what language you use.

Where to avoid losses? Where is it a bad idea?

# Note → it can't be used by default if model doesn't give probability.

→ Logloss is a black box, no interpretability.

Hard to explain what the fuck is right or wrong with model.

(the worse case; some reference)

⇒ Logloss is an overkill for cases where we don't give a fuck about probability.



TLDR → Nothing works for all cases, it all depends.

for LIVE  
SESSION.

'0' '1' '✓' '1'  
"only  
concept"  
→ not "positive  
points concept"

[0.4  
0.6  
1.4  
1.6]

↓  
datapoints.

The AVC will be  
same for both.

Learn a concept  $\Rightarrow$  Make notes  $\Rightarrow$  Explain it  
 $\rightarrow$  to someone.

FÉYNMANN TECHNIQUE.