Classification Metrics - 2

Which metrics are used in the medical domain?

They are used to measure how good a test is at correctly identifying the presence or absence of a disease.
In medical terms,

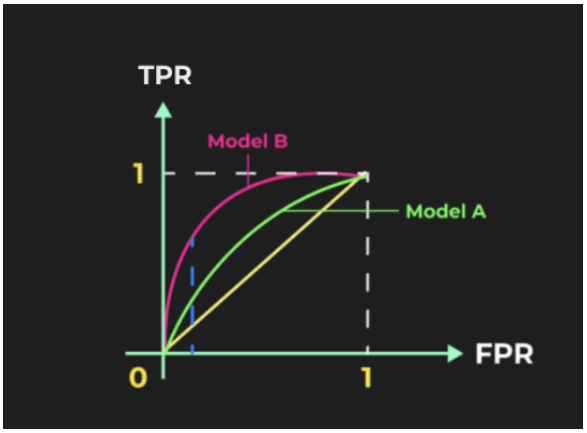
- **Sensitivity**: proportion of people with the disease who test positive for it
  - ⇒ The test is good to be used as a screening test
  - ⇒ There is a low chance of missing out on a person with a disease (low FN)
- **Specificity**: proportion of people without the disease who test negative for it
  - ⇒ It means that the test is good for confirmatory test
  - ⇒ There will be low FP

What is AU-ROC? Where is it used?

AU-ROC: Area Under Receiver Operating Characteristic Curve
- Used to find the best model by plotting TPR and FPR by sorting values of $\hat{y}_i$ and keeping them as the threshold for final prediction (y_{pred}).

How do we determine the better model using AU-ROC?

After plotting whichever curve has the most area covered tends to be the better model.



For ex:

- Here, AUC of model B > model A
- Hence, model B is better

Note:

- Unlike precision, recall, or F1-score, AU-ROC does not work well for highly imbalanced data.

What is the fundamental difference between AUC and the other metrics?

When we calculate Precision, Recall, or F1 score

- We calculate it for a certain threshold on $\hat{y}_i$
- This threshold is 0.5, by default

On the other hand, for AU-ROC

- we are calculating it using all possible thresholds

What will be the AUC of a random model?

The ROC curve will be diagonal. ⇒ Hence AUC will be 0.5

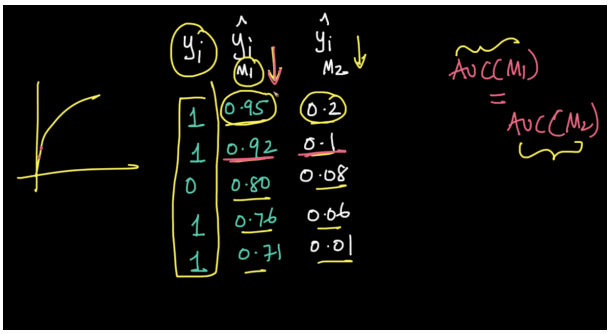
Does AUC depend on the actual values $\hat{y}_i$?

No. AU-ROC depends on how the ordering of the $\hat{y}_i$ is done and not on the actual values of $\hat{y}_i$.

For Example, say we have two models M1 and M2

- Actual y labels: [1, 1, 0, 1, 1]
- $\hat{y}_i$ for M1: $[0.95. 0.92,\ 0.80,\ 0.76,\ 0.71]$
- And $\hat{y}_i$ for M2: $[0.2,\ 0.1,\ 0.08,\ 0.06,\ 0.01]$

Since both have the same ordering of how $\hat{y}_i$ are arranged, hence AUC(M1) = AUC(M2).



AU-ROC is highly sensitive to imbalanced data, what metric can we use there?

We can use the Area under the Precision-Recall curve (AU-PRC).
This is a very good metric for imbalanced data.

How is PRC plotted?

- Precision on the y-axis
- Recall on the x-axis
- Similar to the ROC curve, we'll take each $\hat{y}_i$ as the threshold

  Then we take the area under the PRC curve to get AU-PRC.

How to use the Accuracy metric even when data is imbalanced?

Ans: G-Mean: When data is imbalanced, Geometric-Mean(G-Mean) measures model performance on both the majority and minority classes.

$$GMean = \sqrt{Specificity \times Sensitivity}$$

Which metric to use when? (Cheat Sheet)

- If we want probabilities of classes: **Log loss**
- If classes are balanced: **Accuracy**
- If classes are imbalanced:
  - If FP is more critical: **Precision.**
  - If FN is more critical: **Recall.**
  - **The F1 score** is a balance between precision and recall.
  - If our concern is both classes (TN and TP)**: AU-ROC**
- If severe imbalance**: PR AUC**

How are performance metrics different from loss functions?

- Loss functions are usually differentiable in the model's parameters.
- Performance metrics don't need to be differentiable.
- A metric that is differentiable can be used as a loss function also. Forex: MSE

Imbalance Data

Effect of Imbalance data on kNN

As the value of k increases, the data imbalance impacts the model predictions more and more

Effect of Imbalance Data on Logistic Regression

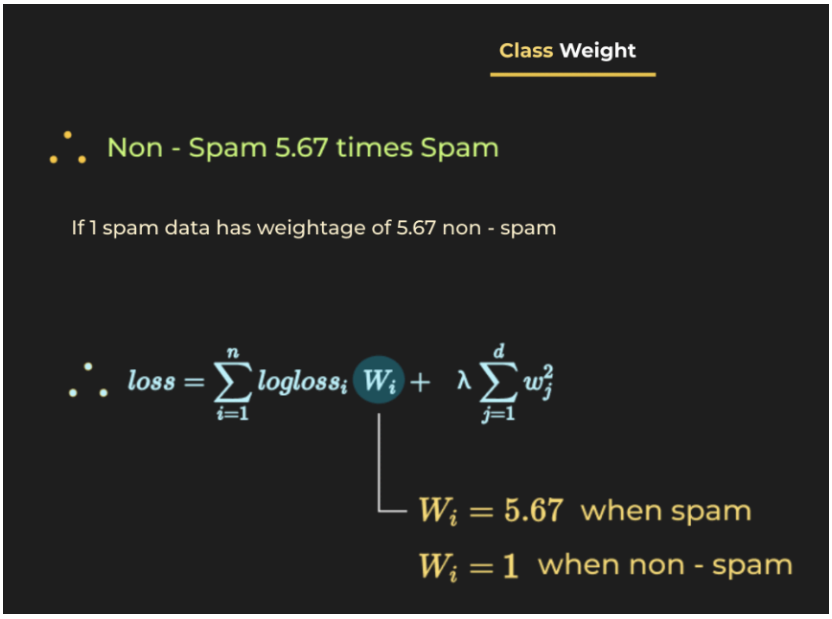
Suppose we have more -ve class samples than +ve class samples. Then the -ve class dominates the log loss function.

This makes the hyperplane π be pushed away from the -ve class sample such that it passes the +ve samples, thus making the model predict every point as -ve class.

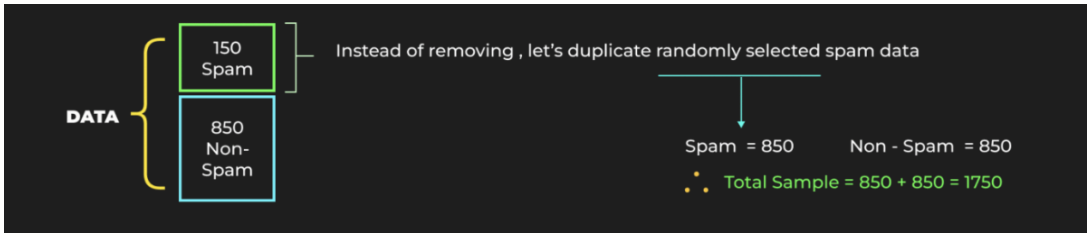
Handling Imbalance Data

A. **Weighted Loss:** Add a class weight to the loss function. Which increases the weightage loss value of the minority class

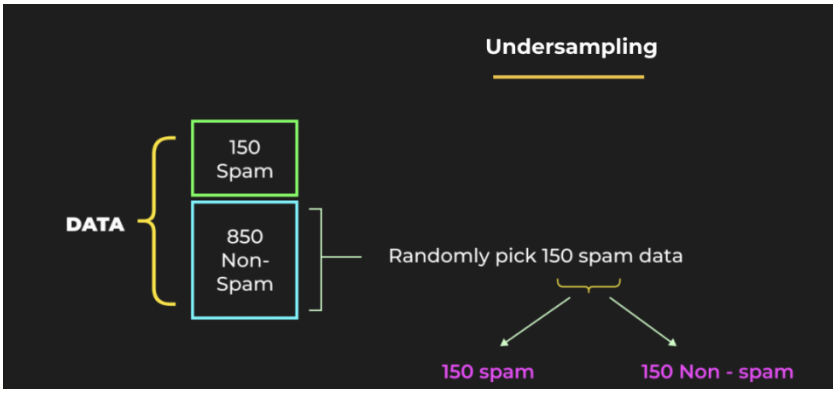
$$w_{minority} = \frac{Number\ of\ samples\ of\ Majority}{Number\ of\ samples\ of\ Minority}\ ;\ w_{majority} = 1$$



B. Oversampling: Replicating the minority class such that, the number of samples in the minority class is the same as the number of samples in the majority class.



C. Undersampling: Removing the number of samples of the majority class such that the number of samples in the minority class is the same as the number of samples in the majority class.



Note: Leads to Information loss that reduces model reliability. Hence only reliable if the number of samples is really large

D. SMOTE: Works similar to kNN. First, it selects a minority sample data point x_1. Then based on the value of k, find the distance between the k-nearest neighbor and the datapoint d.

Selects a random value between [0,1] for each k-neighbor, and multiplies with the distance vector which is added to the features of the datapoint x_1.

This creates a new sample data point x_{new}: $x_1 + [random\ value] \times d$

Thus creating synthetic minority-class data samples.