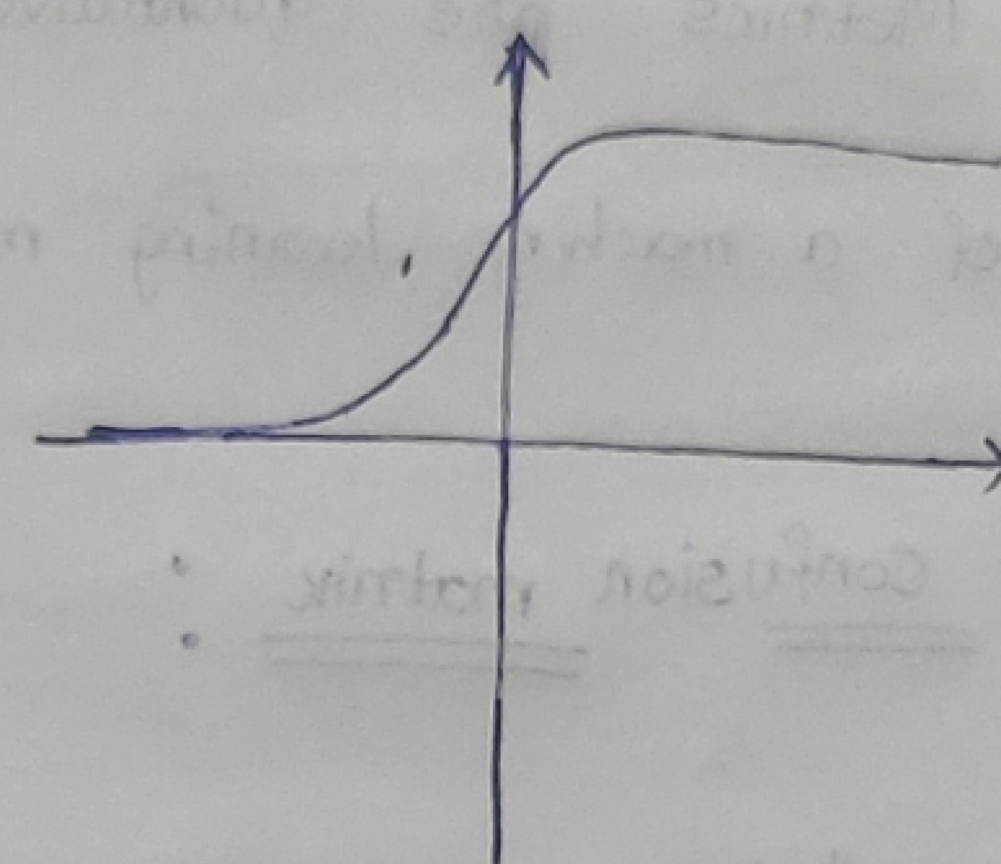
* Logistic regression :

When we want to Predict an outcome of true/false, 1/0 we use logistic regression cuz linear regression model give output as numbers for ex: 0·69, 0·82 etc which makes us fall in dilama but logistic regression gives the likelyhood of the outcome.

* Sigmoid function :

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



As the x value increaces sigmoid func reaches 1 but never becomes 1 vice versa when the x value ↓ it reaches 0 but never becomes "0".

$$z = b + w_1x_1 + w_2x_2 + \cdots + w_Nx_N$$

Logistic regression models are trained using the same Process as LR with 2 distinctions:

→ we use log loss instead of mse

→ Applying regularization is critical to Prevent overfitting

$$\text{Log Loss} = \sum_{(x,y)\in D} -y(\log y') - (1-y)\log(1-y')$$

* Classification :

we used sigmoid function to find out the Probability of the email being a spam but what if the case is to decide if the email is "spam" / "Not spam".

Classification is the task of Predicting which of a set of classes an example belong to.

* Metrics :

Metrics are quantative measures used to evaluate the Performance of a machine learning model.

* Thresholds & confusion matrix :

Let's assume that we want to integrate a ML model where it can Predict the email is a "spam" / "Not spam" as we are using Logistic regression we only get values / likelihood of being spam so if we want to seperate them we use "threshold Probability" called "Classification threshold". Examples with a Probability above threshold are then assigned to Positive class. (here, spam), examples with lower Probability are assigned to the -ve class

What happens if the Predicted score is equal to the classification threshold ?

It totally depends on the implementation chosen from the classification model.

the Keras library Predicts the -ve class if the thresholds are equal, but other tools / frome have different approach.

If we get a 0.95 and 0.51 as Predicted scores for 2 emails and if the threshold is set to 0.9 we only get 1 email as spam if we set threshold as 0.5 we get both emails are spam.

a 0.5 might seek like a intuitive threshold, it's not a good idea to classify the typo mistake as a spam it is worse than to send a spam mail to inbox compared to sending legitimate email to spam.

Confusion matrix :

	Actual Positive	Actual Negative
Predicted +ve	TP : A spam mail correctly classified as spam	FP : A not-spam classified as spam
Predicted -ve	FN : A spam mail mis-classified as not spam	TN : A not-spam email correctly classified as not-spam.

* Classification : Accuracy, recall, Precision and related metrics :

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

accuracy can serve as a corse-grained measure of model quality, for this reason, it is often the default evaluation metric.

However, when the dataset is imbalanced, or where one kind of mistake (FN/FP) is more costly than other, it's better to optimize one of the other metrics instead.

For heavily imbalanced datasheets, where one class appears very rarely say 1%, a model that predict 100% of the time would score 99% accuracy, despite being useless.

* Recall (or) True Positive Rate (TPR) :

The proportion of all actual positives that are classified correctly as positives, is also known as Recall.

$$\text{Recall (or TPR)} = \frac{\text{correctly classified actual positives}}{\text{all actual positives}} = \frac{TP}{TP+FN}$$

A hypothetical perfect model have zero false +ve's and $\therefore TPR = 1$

100% decision Rate.

In an imbalanced dataset where the no of actual +ve's is very low recall is more meaningful than accuracy because it measures the ability of the model to correctly identify all +ve instances.

* **False Positive Rate :**

$$FPR = \frac{\text{Incorrectly classified actual -ve's}}{\text{all actual negatives}} = \frac{FP}{FP + TN}$$

In an imbalanced dataset when no of actual -ve's is very, very low say 1-2 examples FPR is meaningless.

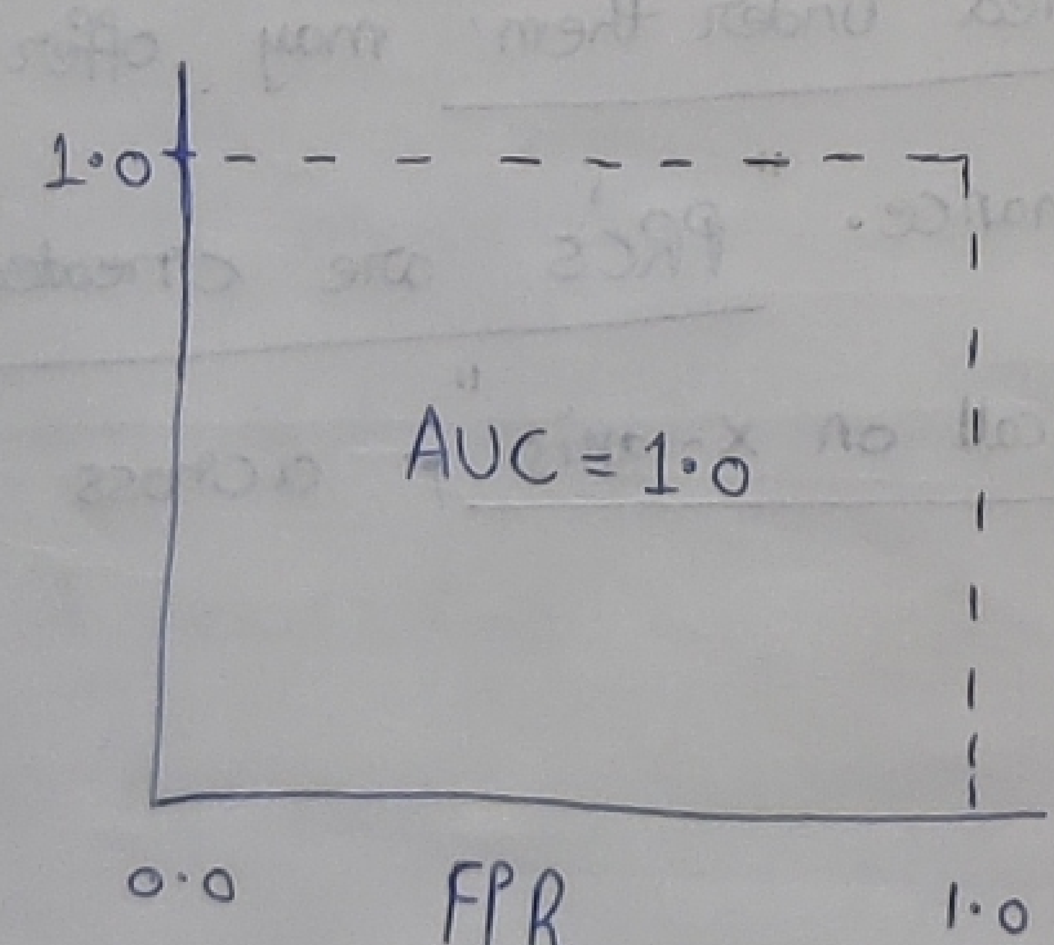
* **Precision :**

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

* **Classification : ROC & AUC :**

→ **Receiver - Operating characteristic curve (ROC) :**

* the ROC curve is visual representation of model performance across all thresholds.



* Area under the curve (AUC) :

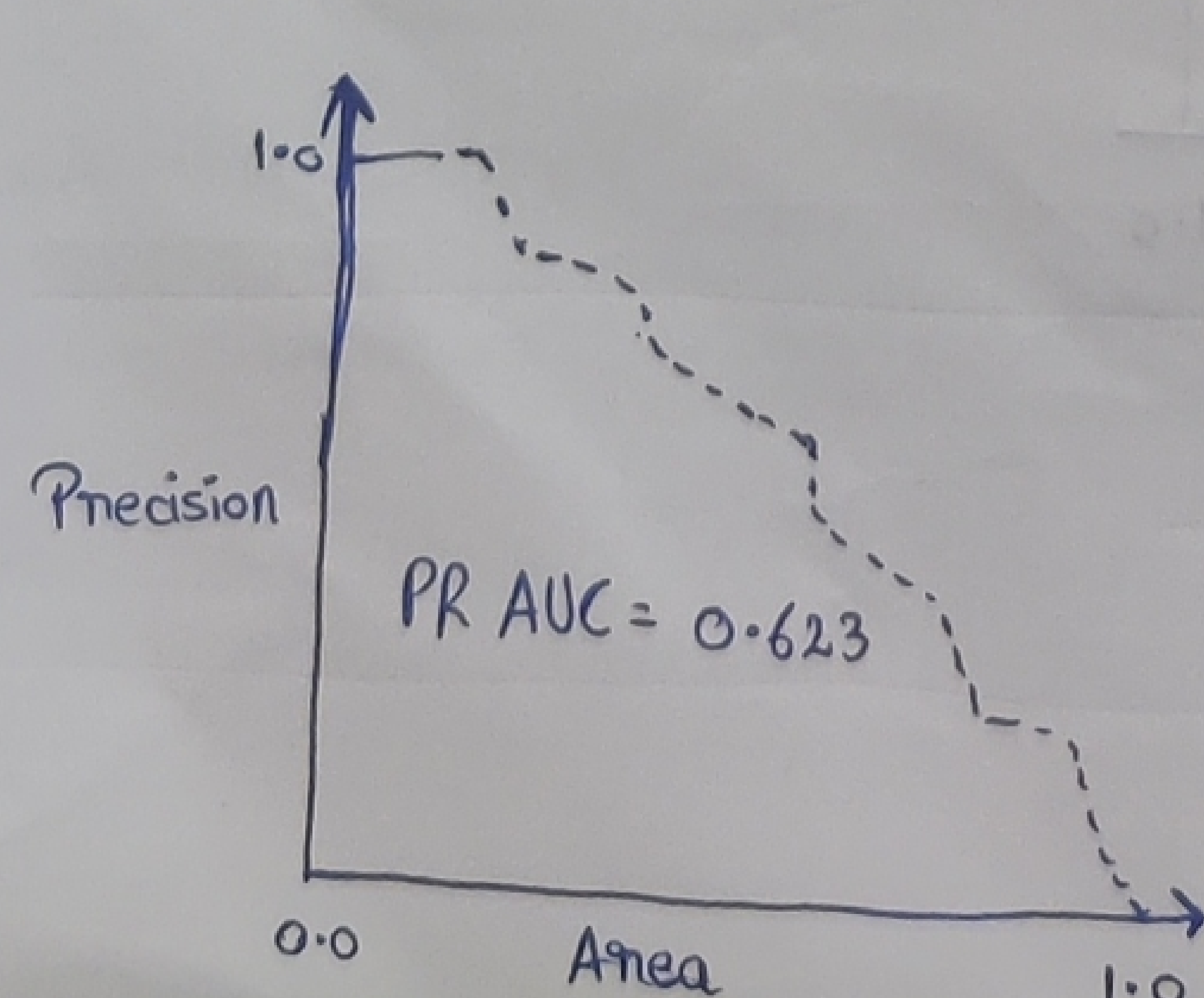
The AUC represents that the Probability that the model, if given a randomly chosen +ve and -ve examples, will rank the +ve higher than the negative.

In more concern terms, a spam classifier with AUC of 1.0 always assigns a random email a higher Probability of being spam than a random legitimate email. The actual classification of email depends on the threshold we choose.

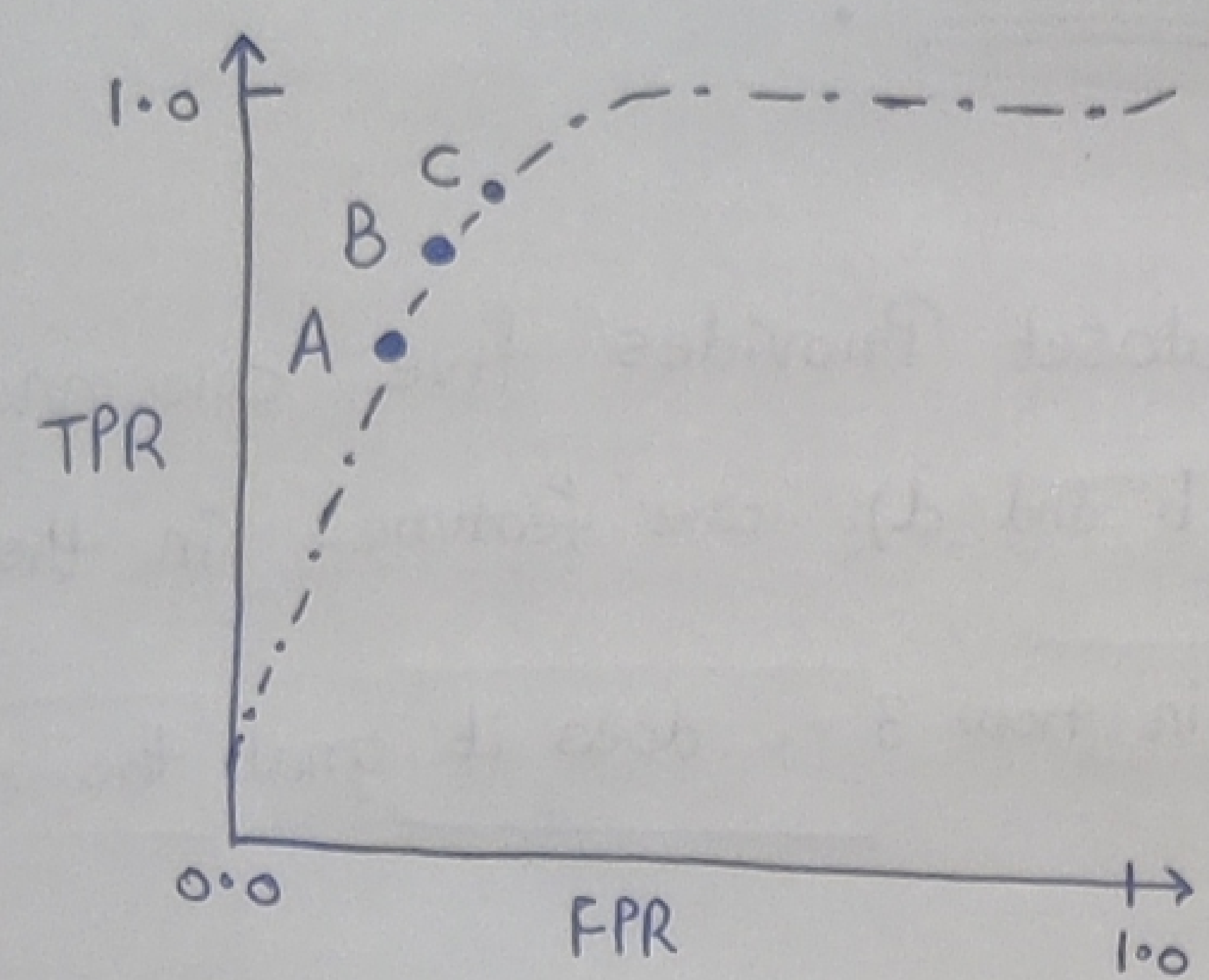
A spam classifier with AUC of 0.5 assigns a random spam email a higher Proba of being spam than a random legitimate email only half the time.

* Precision Recall curve :

AUC and ROC work well for comparing models when the dataset is roughly balanced between classes. "when the data set is imbalanced, Precision recall curve (PRC's) and area under them" may offer better comparitive visualization of model Performance. "PRC's are created by Plotting Precision on Y-axis and Recall on X-axis", across all thresholds.



consider the Points A, B and C in the following diagram each representing a threshold :



If FPR are highly costly, we choose a value that gives a lower FPR, like Point A even if TPR is reduced. conversely, If FP are cheap and FN are high costly we choose threshold with higher TPR value like Point C. If we want to choose a balance b/w them we choose Point B

* Prediction Bias : Prediction bias is the diff b/w the mean of a model's Predictions and the mean of ground-truth labels in the data.

Ex : If a model is trained on dataset where 5% of the emails are spam should Predict an average, that 5% of the emails it classifies as spam. If the model Predicts 50% of the time the email as spam then there is something wrong with the training dataset / model itself.