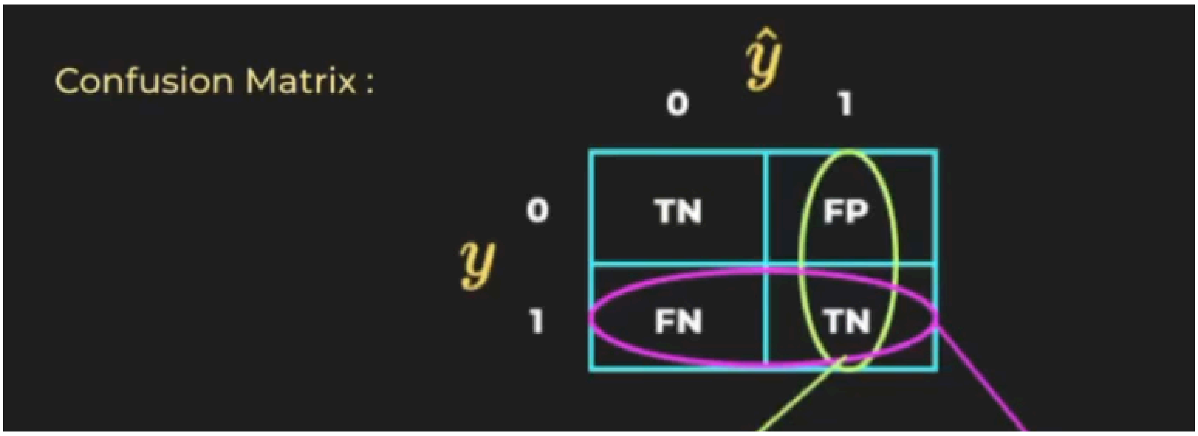
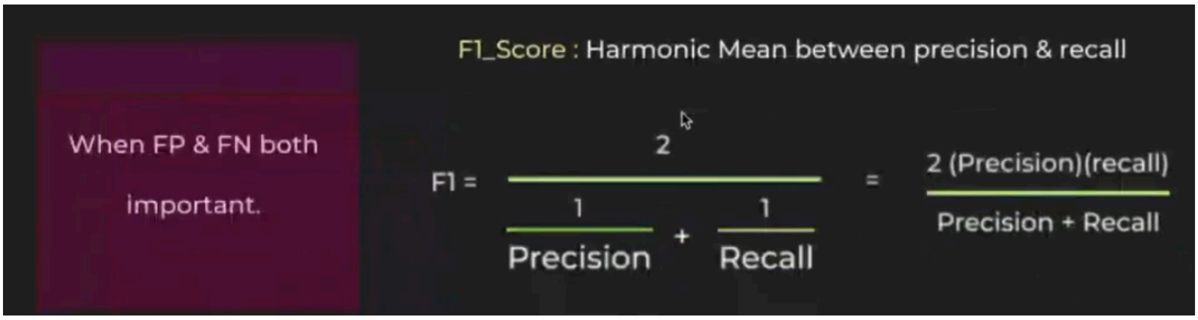
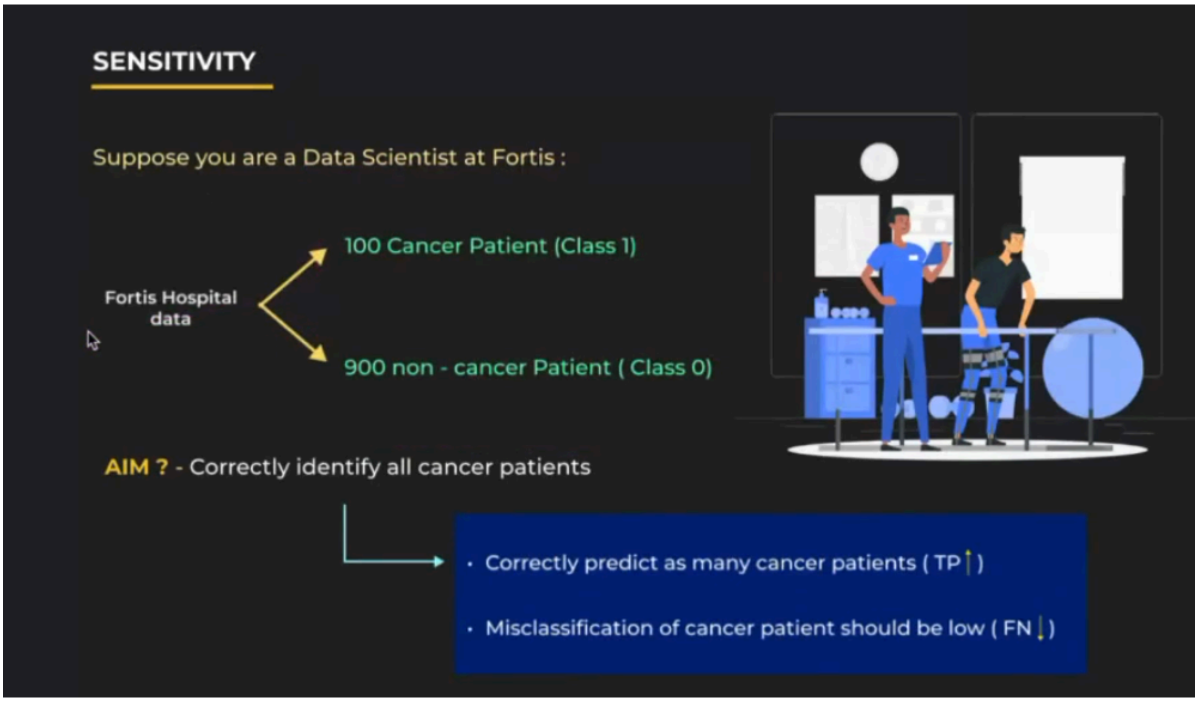
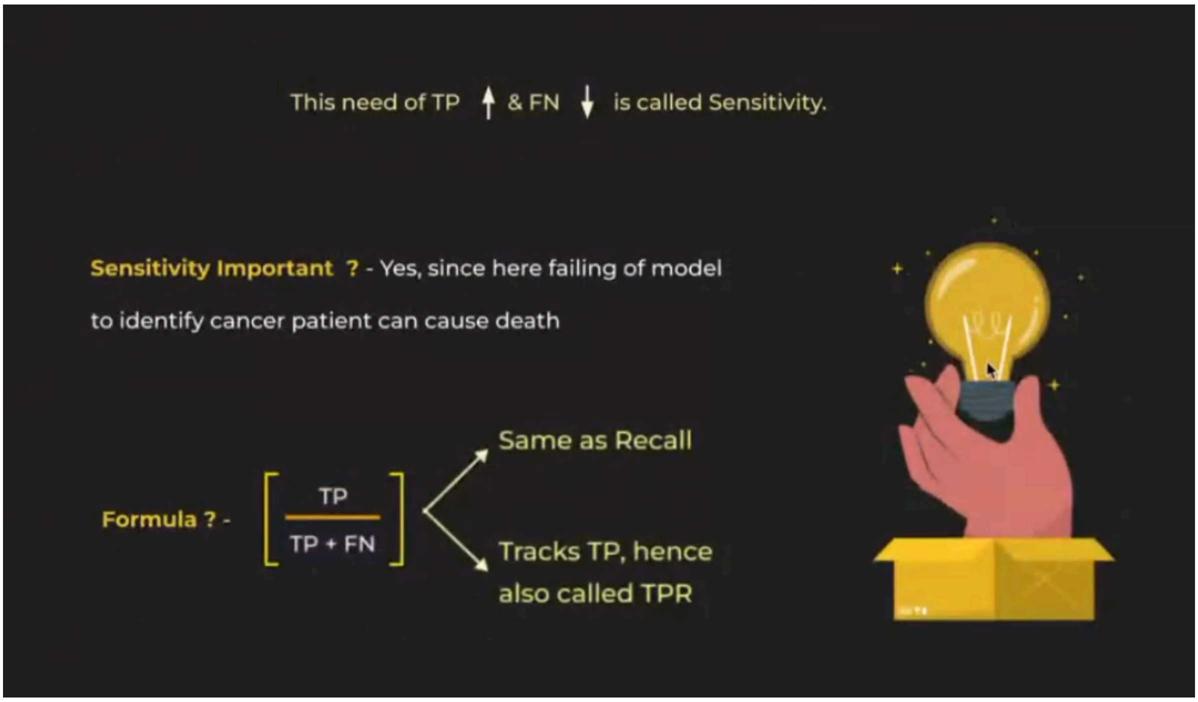
Recap :

> Problem with accuracy - Does not work for imbalance data.










```
What to say when screening test identifies 92 Cancer patients out of 100?

a) test has high sensitivity
b) test has low sensitivity
c) test has no sensitivity
d) cannot be determined
```

Correct Answer a) test has high sensitivity

False Negative Rate (FNR) / Miss rate

What happens if model is insensitive ?

→ FN ↑ leading to **worsening of health condition of cancer patient**

This means FN should also be tracked

How ? - Measure change in FN which is called as FNR.

Formula ? - FNR = 1 - Sensitivity

$$= 1 - \frac{TP}{TP + FN}$$

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

↓

As patients are missed by the MODEL,

- FNR is called Miss Rate.

SPECIFICITY

Do TN & FP not play any role in medical firms ?

TN & FP plays important role

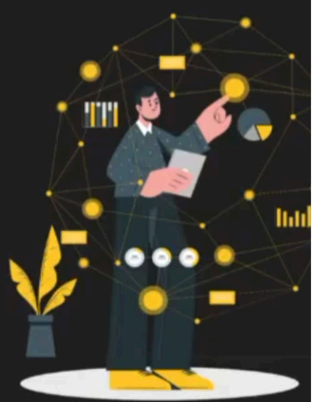
- → Model should correctly predict non - cancer patient **(TN ↑)**
- → Model should keep misclassification of non - cancer patient low (FP ↓)

Need of TN ↑ & FP ↓ is called **Specificity**

Analogy : Specificity is sensitivity for negative class (CLASS 0)

Specificity Important ? - High specificity avoids

1. Fruitless experience treatments
2. Reduces social stigma anxiety for a non - cancer patient



Formula ? - Specificity = $\frac{TN}{TN + FP}$ → Tracks TN , Hence called TNR

Imagine if the screening test identifies 600 Non-cancer patients as cancerous:

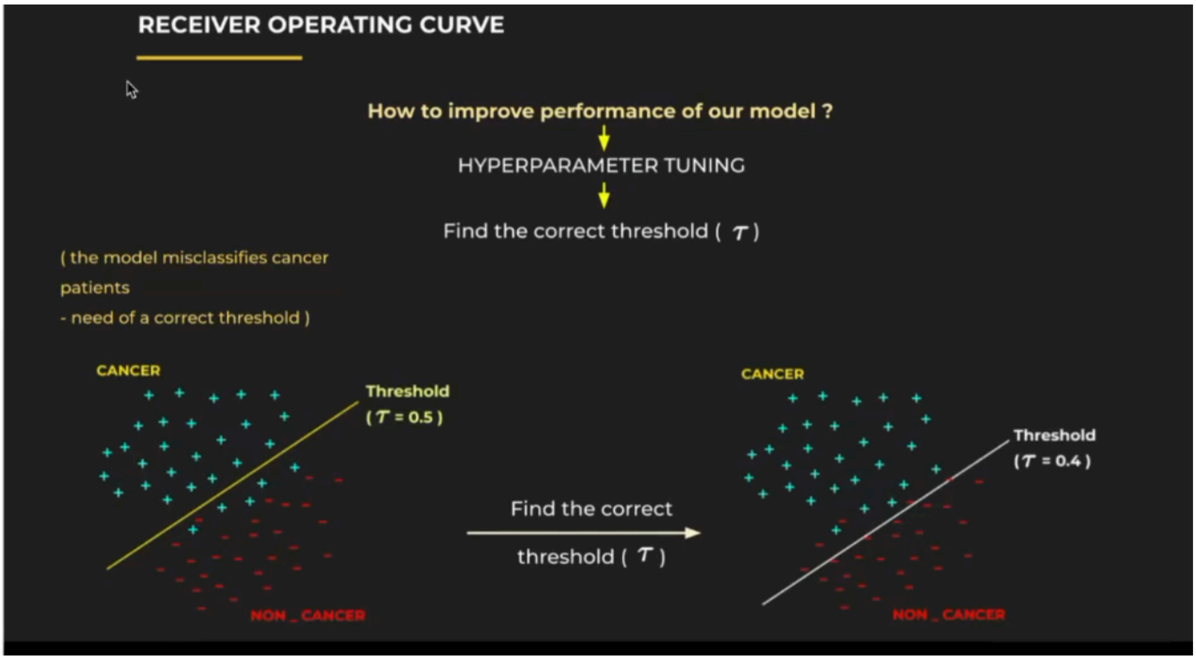
- This will cause fruitless treatments on these patients which are quite expensive.
- Also will create a social Stigma, anxiousness, Stresss to these patients
- Hence Specificity becomes crucial in order to avoid unnecessary expensive treatments, medications, social stigma and anxiety

Observation :

- Specificity is basically sensitivity defined for Negative class (class 0)
- Specificity tracks TN so TNR (True negative rate).
- FNR = 1 - TNR
- sensitivity tracks TP so TPR (True positive rate).
- FPR = 1 - TPR

ROC (Receiver Operating Characterstic Curve)

- If our model got 88% F1-score, what can be done to increase performance ?
  - Doing some hyperparameter tuning might help.
  - Adding regularization
- What if adding it does not do major improvement in model?
  - in our cancer patient example, we wanted to correctly classify cancer patients
  - and for this we use a threshold of 0.4 instead of 0.5



How to find the correct threshold τ ?

- Considering Validation data

Step : 1

n

X	Y	$P = P(Y^{(i)} = 1 \mid X^{(i)})$
$x^{(1)}$	$y^{(1)}$	$p^{(1)}$
$x^{(2)}$	$y^{(2)}$	$p^{(2)}$
$x^{(n)}$	$y^{(n)}$	$p^{(n)}$

Sort samples in descending order based on value $P^{(i)}$

Complexity : O(nlogn)

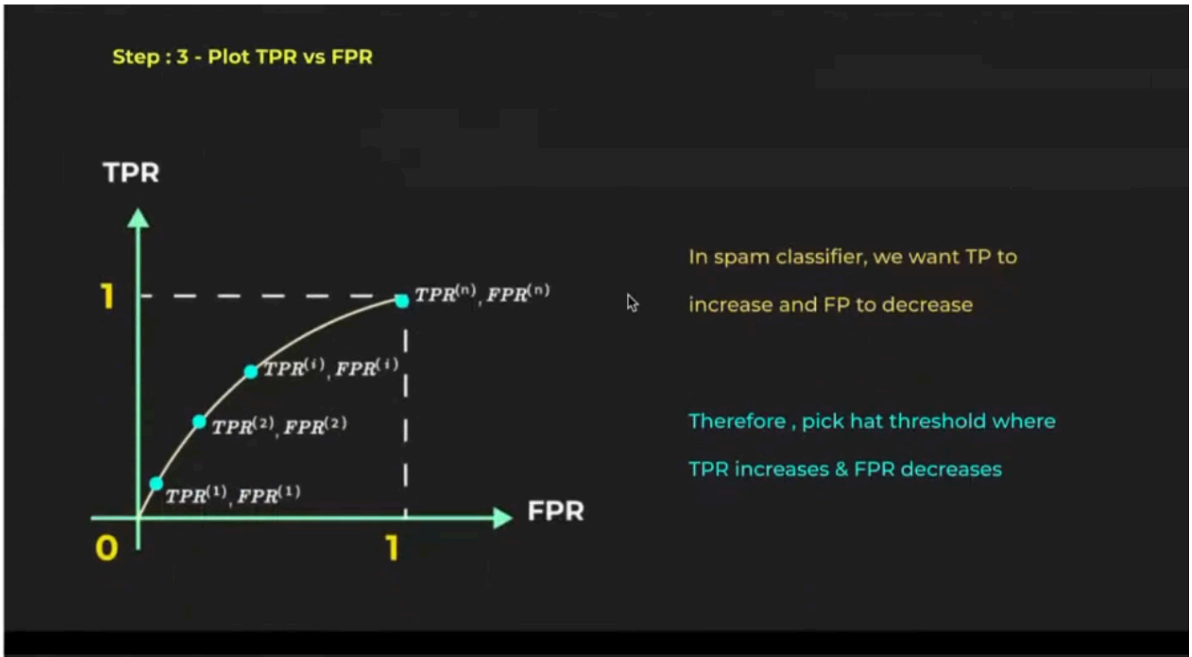
Step : 2 *Use every $p^{(i)}$ as threshold $(\tau^{(i)})$ for predicting $\hat{y}^{(i)}$*

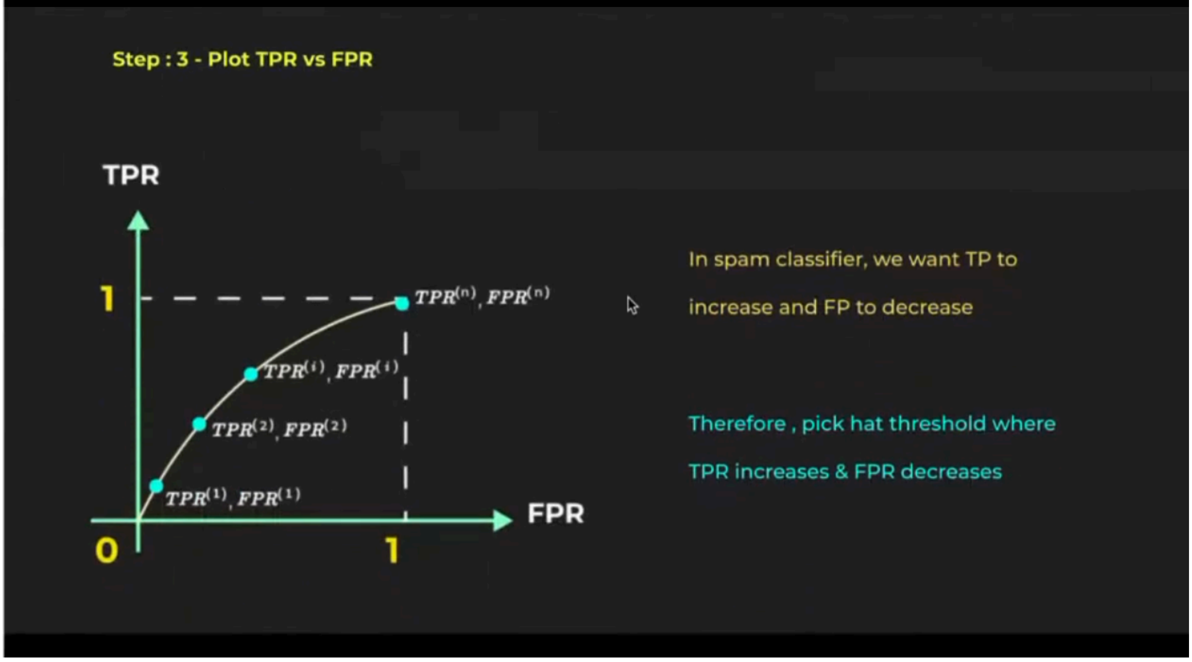
X	Y	P	$\hat{y}_\tau = p^{(1)}$	$\hat{y}_\tau = p^{(2)}$		$\hat{y}_\tau = p^{(n)}$
$x^{(1)}$	$y^{(1)}$	$p^{(1)}$	1	1		1
$x^{(2)}$	$y^{(2)}$	$p^{(2)}$	0	1	- - - - -	1
.	.	.	.	.		.
.	.	.	.	.		.
$x^{(n)}$	$y^{(n)}$	$p^{(n)}$	0	0		1

Based on $\hat{y}_\tau = p^{(i)}$, we find TPR & FPR

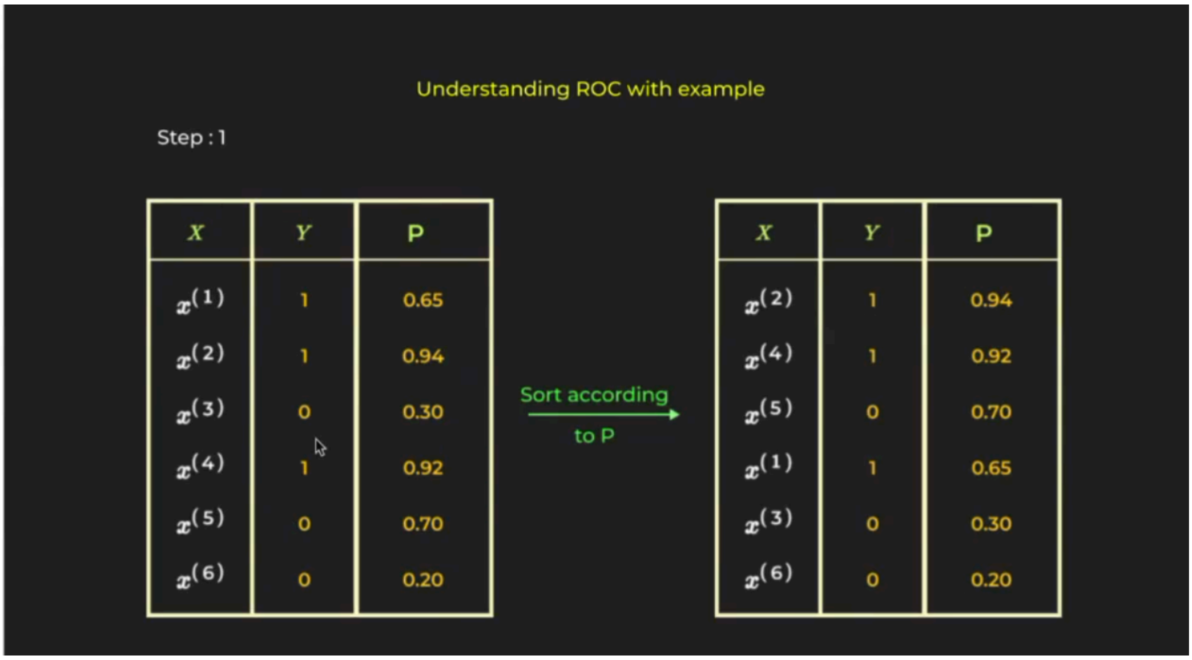
P	TPR	FPR
$p^{(1)}$	$TPR^{(1)}$	$FPR^{(1)}$
$p^{(2)}$	$TPR^{(2)}$	$FPR^{(2)}$
.	.	.
.	.	.
$p^{(n)}$	$TPR^{(n)}$	$FPR^{(n)}$

$$TPR = \frac{TP}{TP + FN} \quad , \quad FPR = \frac{FP}{FP + TN}$$





a



X	Y	P
$x^{(1)}$	1	0.65
$x^{(2)}$	1	0.94
$x^{(3)}$	0	0.30
$x^{(4)}$	1	0.92
$x^{(5)}$	0	0.70
$x^{(6)}$	0	0.20

X	Y	P
$x^{(2)}$	1	0.94
$x^{(4)}$	1	0.92
$x^{(5)}$	0	0.70
$x^{(1)}$	1	0.65
$x^{(3)}$	0	0.30
$x^{(6)}$	0	0.20

Step : 2 $Create\ \hat{y}_T = p^{(i)}$

X	Y	P	$\hat{y}_T = 0.94$	$\hat{y}_T = 0.92$	------	$\hat{y}_T = 0.20$
$x^{(2)}$	1	0.94	1	1		1
$x^{(4)}$	1	0.92	0	1		1
$x^{(5)}$	0	0.70	0	0		1
$x^{(1)}$	1	0.65	0	0		1
$x^{(3)}$	0	0.30	0	0		1
$x^{(6)}$	0	0.20	0	0		1

Find TPR , FPR - TP = 1 , FP = 1, FN = 2 , TN = 3

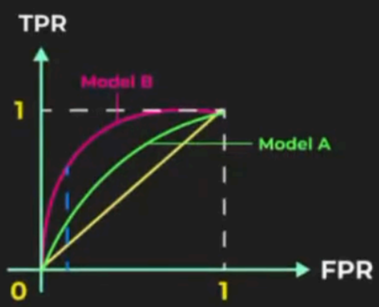
$$\therefore TPR = \frac{1}{1+2} = \frac{1}{3} = 0.33 \qquad FPR = \frac{0}{(0+3)} = 0$$

Repeating step - 2 for each P :

P	TPR	FPR
0.94	0.33	0
0.92	0.50	0
0.70	0.67	0.33
0.65	1	0.33
0.30	1	0.67
0.20	1	1

Pair of TPR & FPR for all 6 probabilities

Suppose we have 2 models → Model A, Model B } Which is better ?



Intuitively,

TPR(B) > TPR(A) for same FPR.

Hence , MODEL B is better

What will be the ROC curve for Random Model ?

As our example data is **BALANCED** $\rightarrow$ 3 +ve data, 3 -ve data

TP + FN = 3

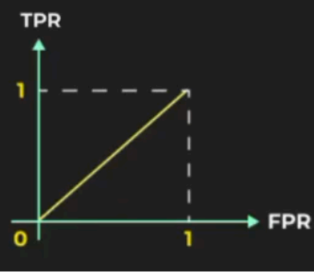
FP + TN = 3

Also as Random Model generates (1, 0) randomly

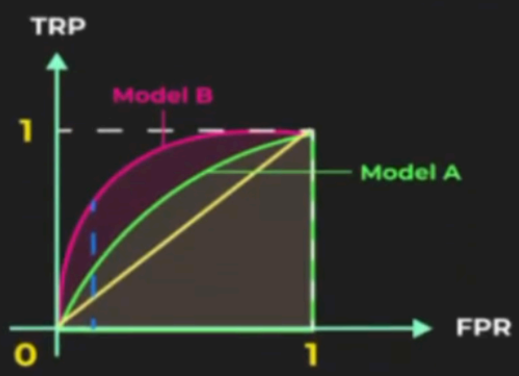
$\underbrace{\text{Random Model}}$

Produces K TP & K FP points $TPR = \frac{K}{3}$, $FPR = \frac{K}{3}$

Hence, TPR = FPR (y - axis) = (x - axis)



How to mathematically show Model B better ?

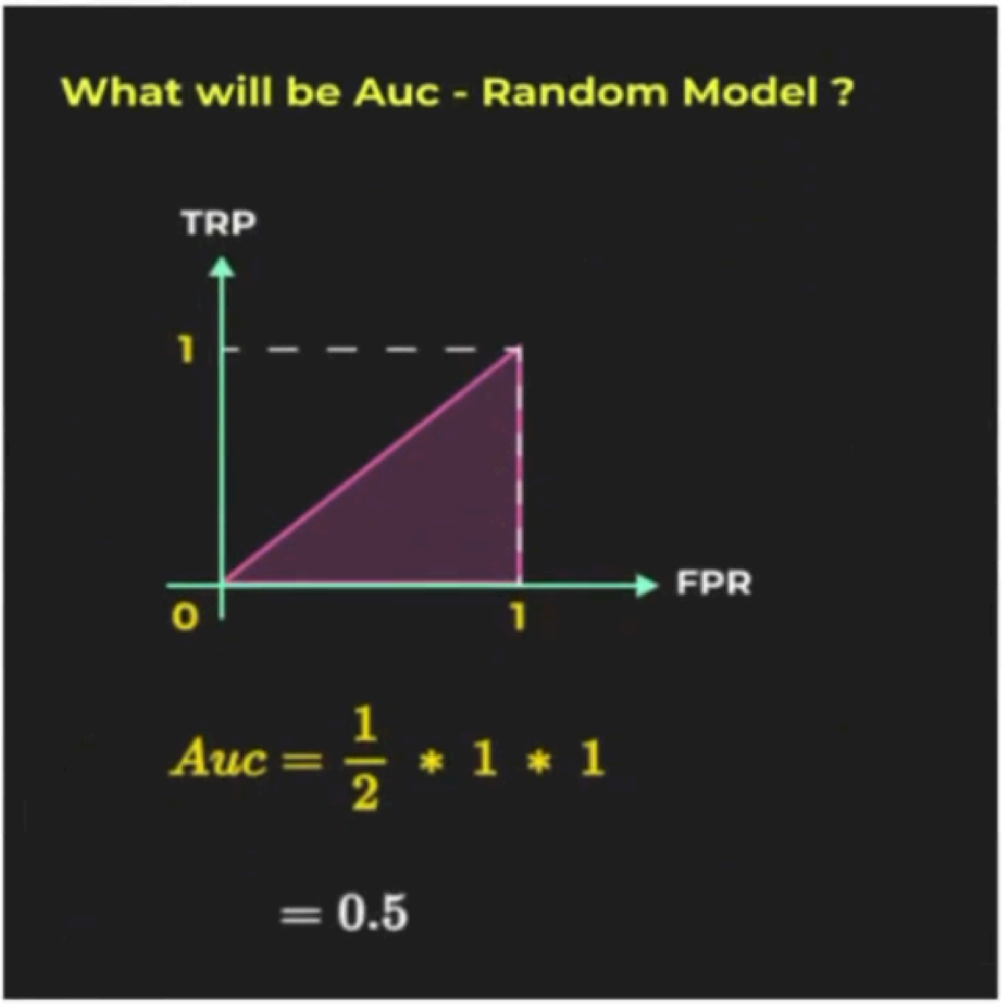


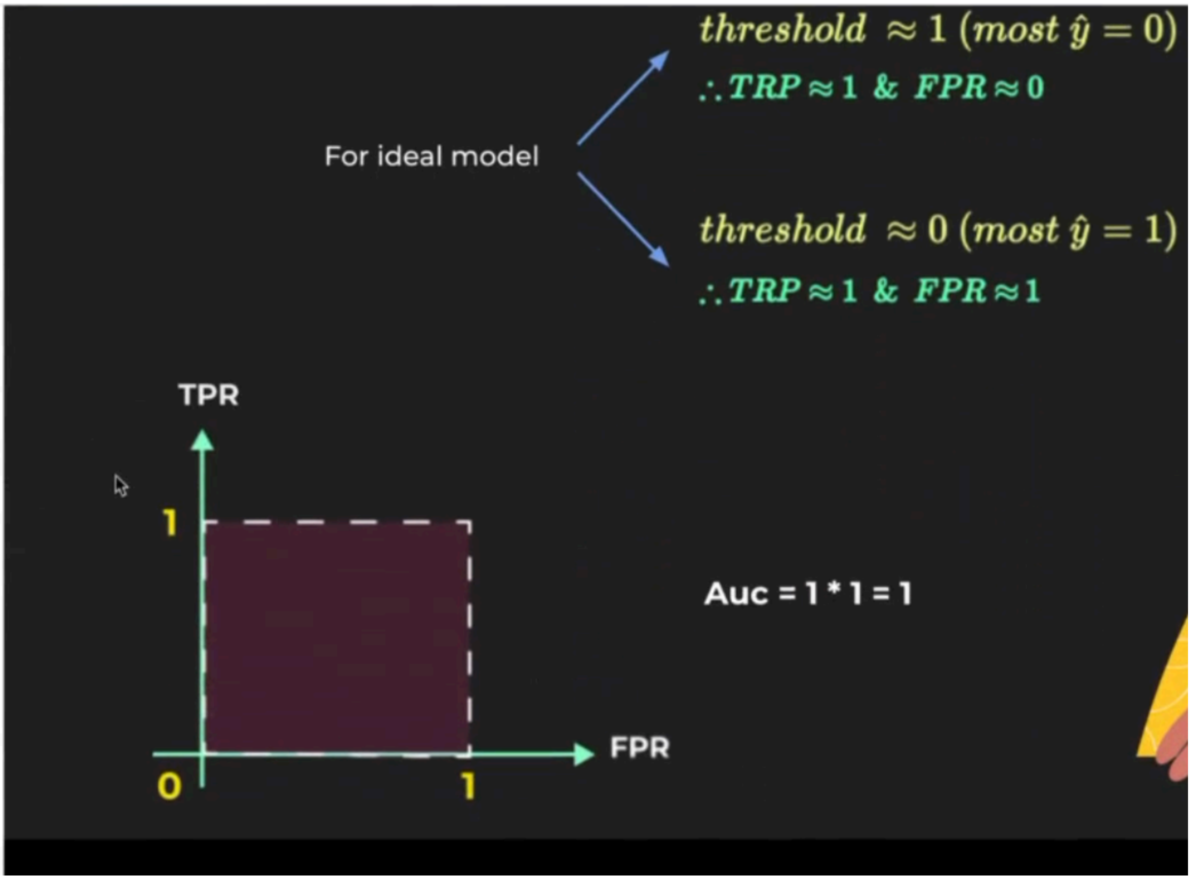
Let's take area under the ROC curve (All - ROC / Auc)

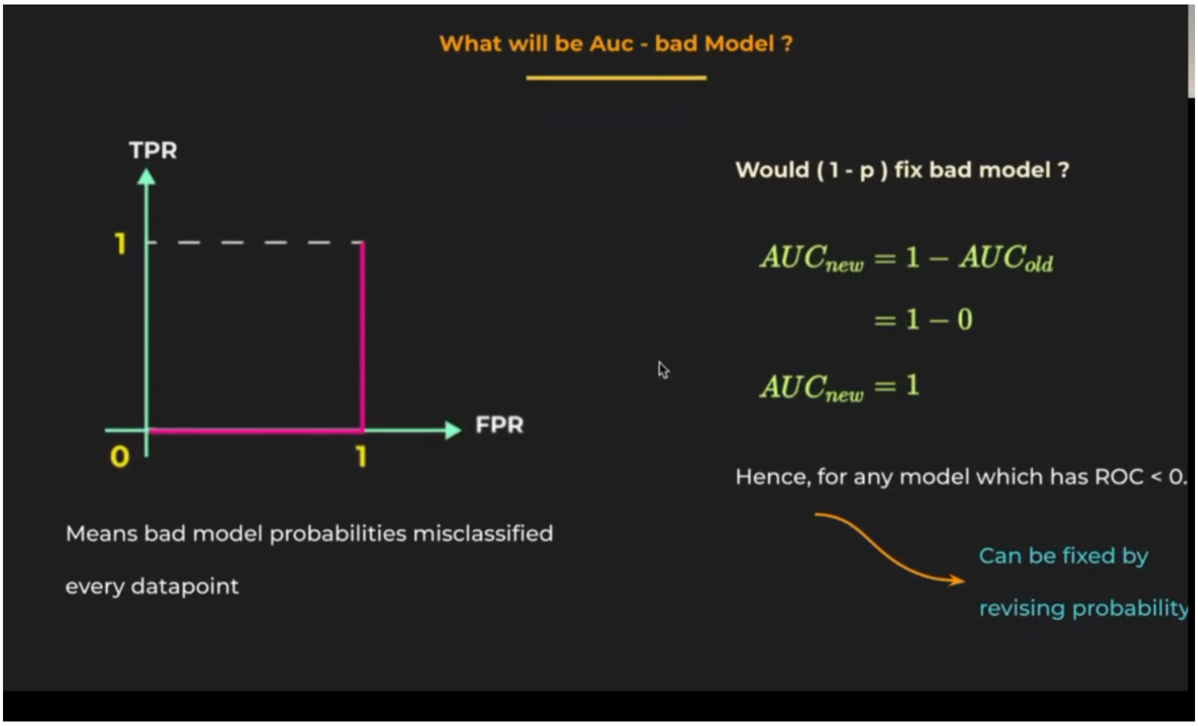
Clearly ,

Auc Model B > Auc Model A

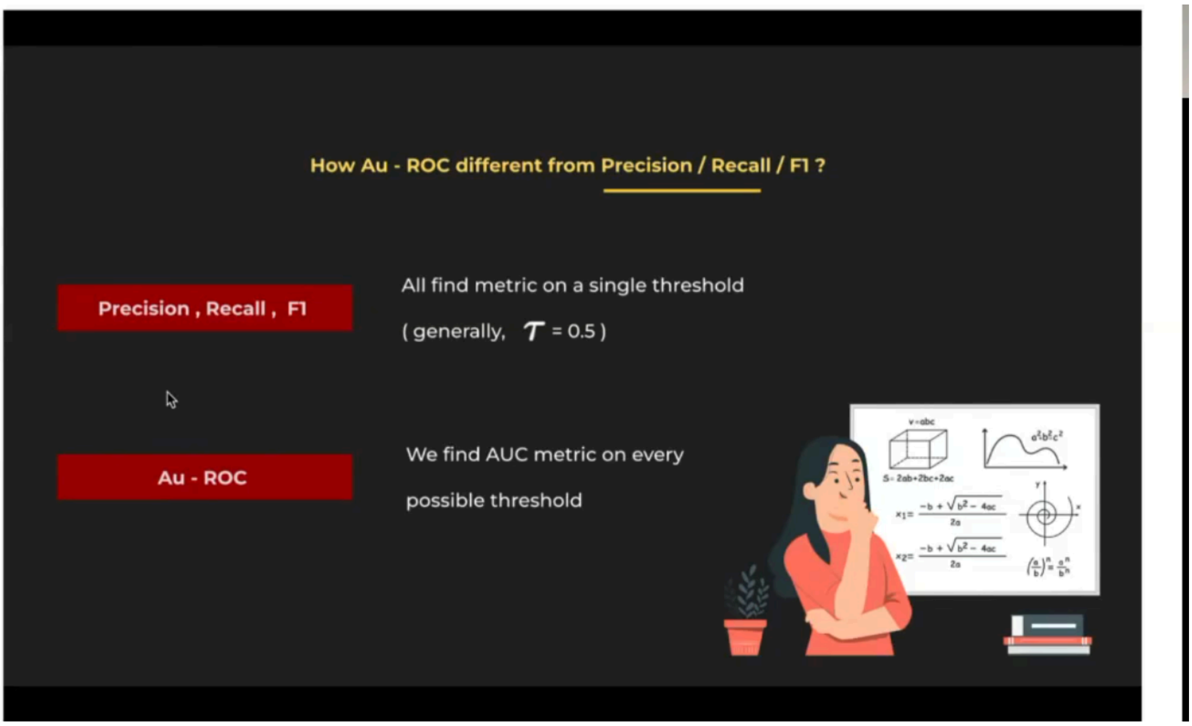
Hence, Model B better


What will be Auc - Random Model ?
TRP
1
0
1
FPR
$Auc = \frac{1}{2} * 1 * 1$
$= 0.5$

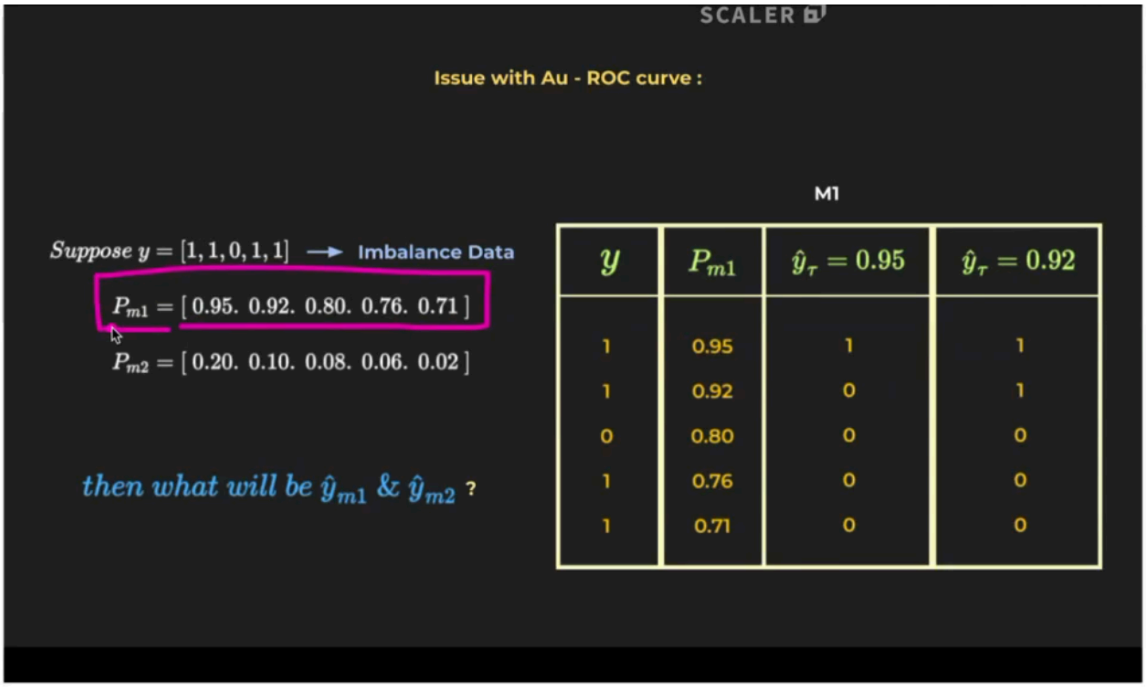

For ideal model
$threshold \approx 1$ ($most\ \hat{y} = 0$)
$\therefore TRP \approx 1\ \&\ FPR \approx 0$
$threshold \approx 0$ ($most\ \hat{y} = 1$)
$\therefore TRP \approx 1\ \&\ FPR \approx 1$
TPR
1
0
1
FPR
Auc = 1 * 1 = 1

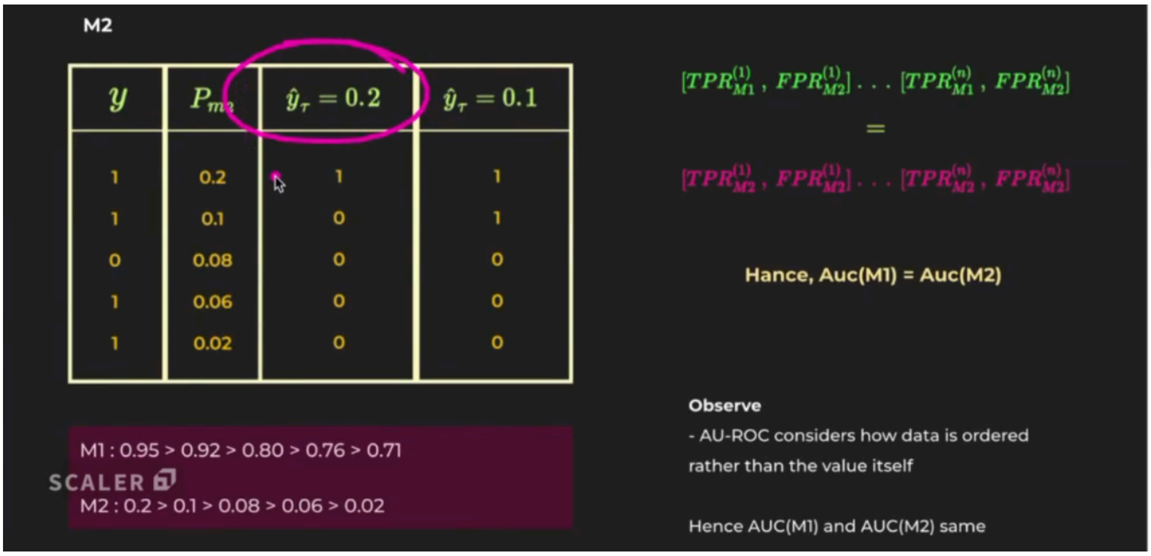


> Important :



> Issue with Au - ROC curve :





> in imbalance dataset auc ROC sometimes not work well.

```
What does it mean when two ROC curves overlap?
a) The two models have same thresholds
b) The two models have similar performance.
c) The two models have opposite performance.
d) The two models have different performance metrics.
```

> Ans: option(b)

Code

```
[ ] from sklearn.metrics import roc_curve, roc_auc_score
```

```
[ ] probability = model.predict_proba(X_test)
```

```
[ ] probability
```

```
array([[0.62081718, 0.37918282],
       [0.99302717, 0.00697283],
       [0.29952017, 0.70047983],
       ...,
       [0.93473179, 0.06526821],
       [0.93713658, 0.06286342],
       [0.74608365, 0.25391635]])
```

Observe

`Probability` variable contains 2 probability $P(Y = 1|X)$ and $P(Y = 0|X)$

But for thresholding we need only one probability, what can be done ?

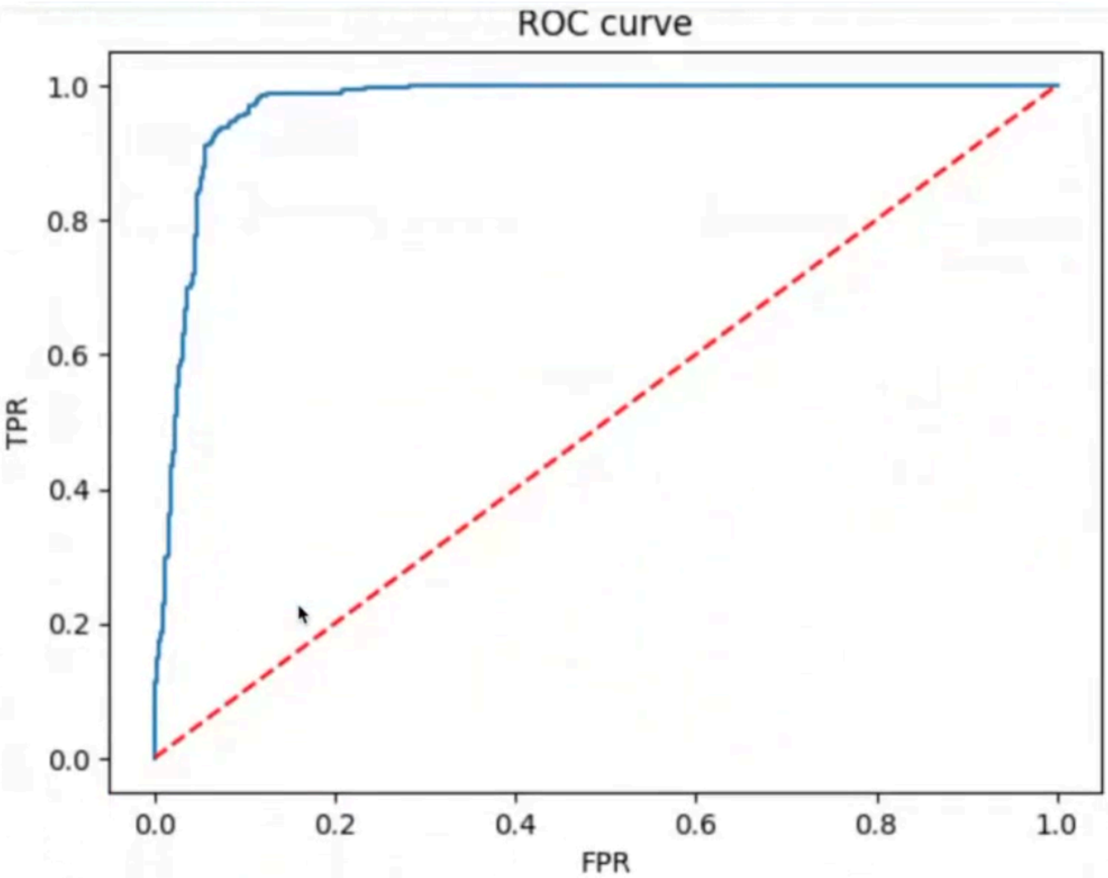
Ans: lets consider only $p = P(Y = 1|X)$

```
[ ] probabilites = probability[:,1]
```

```
[ ] fpr, tpr, thr = roc_curve(y_test,probabilites)
```

```
[ ] plt.plot(fpr,tpr)

    #random model
    plt.plot(fpr,fpr,'--',color='red' )
    plt.title('ROC curve')
    plt.xlabel('FPR')
    plt.ylabel('TPR')
    plt.show()
```




```
[ ] # AUC
    roc_auc_score(y_test,probabilites)
```

```
0.9689813348349934
```

observe

We get an AU-ROC value of 97.8%

- showing that the model is performing very good

But recall our F1 Score was just 88%

- which made us believe our model was a decent one

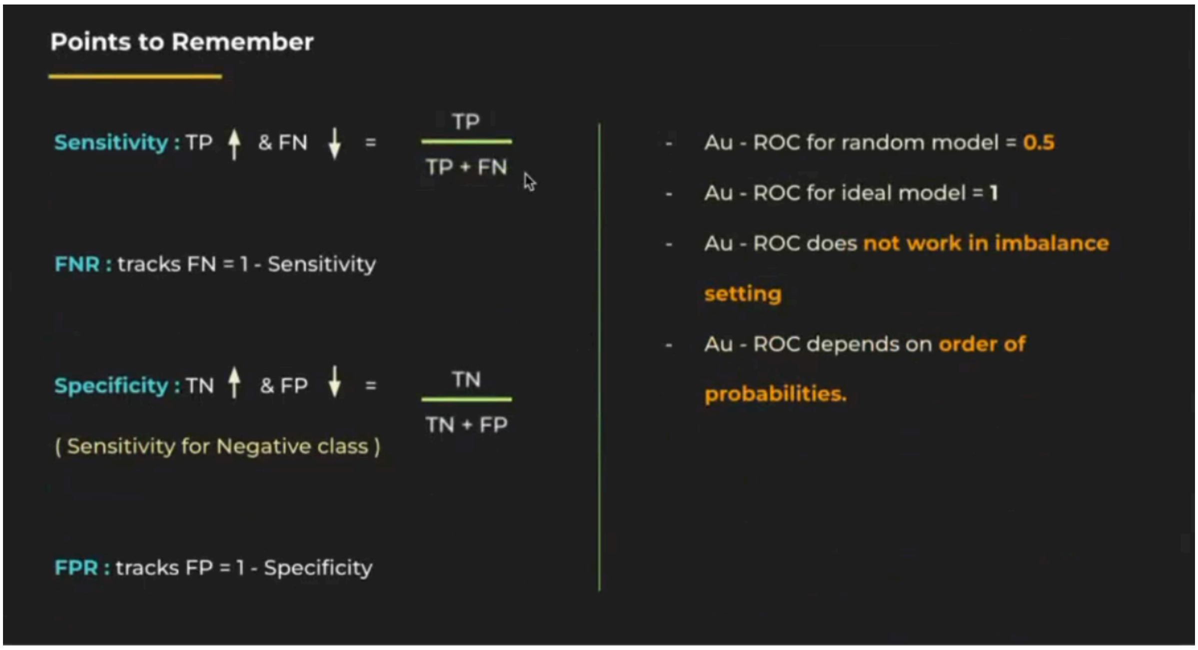
Why a huge difference in model performance when using AU-ROC and F1 score ?

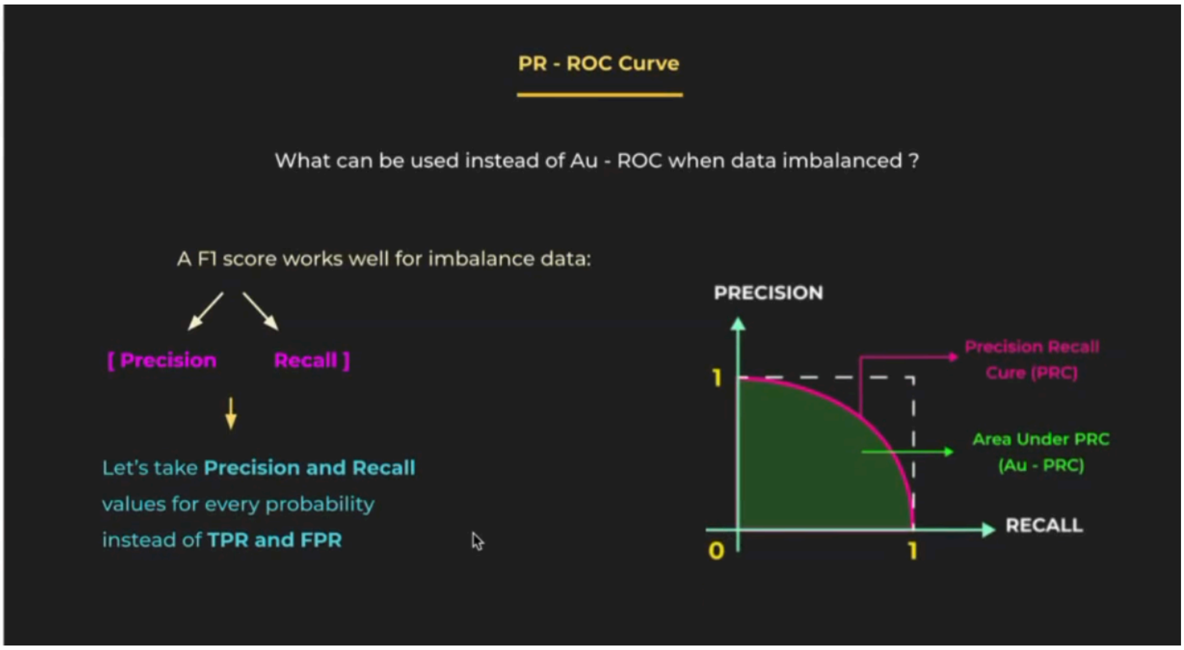
ans: Recall our data is imbalance:(70% ⟶ class 0 and 30% ⟶ class1):

- ROC curves provide an aggregate measure of model performance across all possible classificati
- Hence it can make a **poor model on the minority class appear better**
  - by focusing more on the performance on the majority class.

Note: When data is highly imbalanced,

- AU-ROC is not prefered





Quiz

```
Which evaluation metric can be directly derived from the AUC-PRC?
a) Accuracy
b) F1 score
c) Precision
d) Recall
```

Correct Answer b) F1 score

Explanation As we have precision and recall for each threshold:

- We can find F1-Score for the same

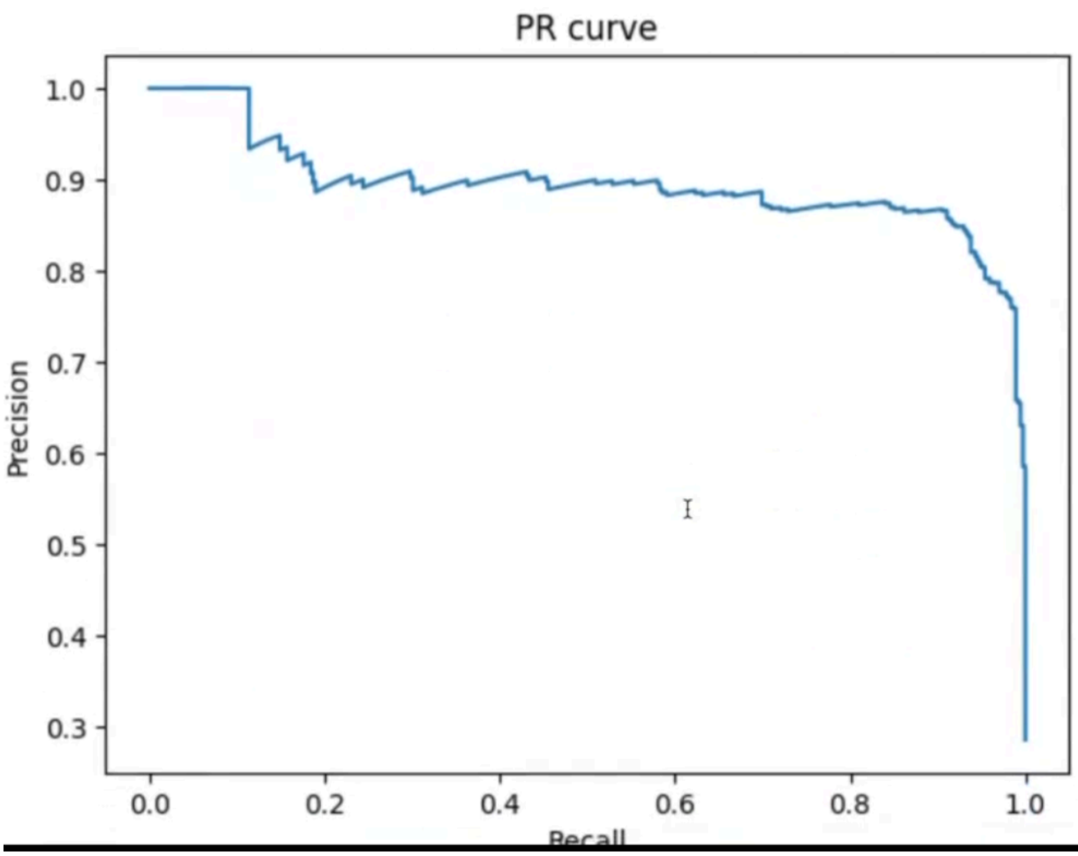
> Precision Recall Curve code

```
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import auc
```

```
precision, recall, thr = precision_recall_curve(y_test, probabilites)
```

```
plt.plot(recall, precision)

plt.xlabel('Recall')
plt.ylabel('Precision')
plt.title('PR curve')
plt.show()
```



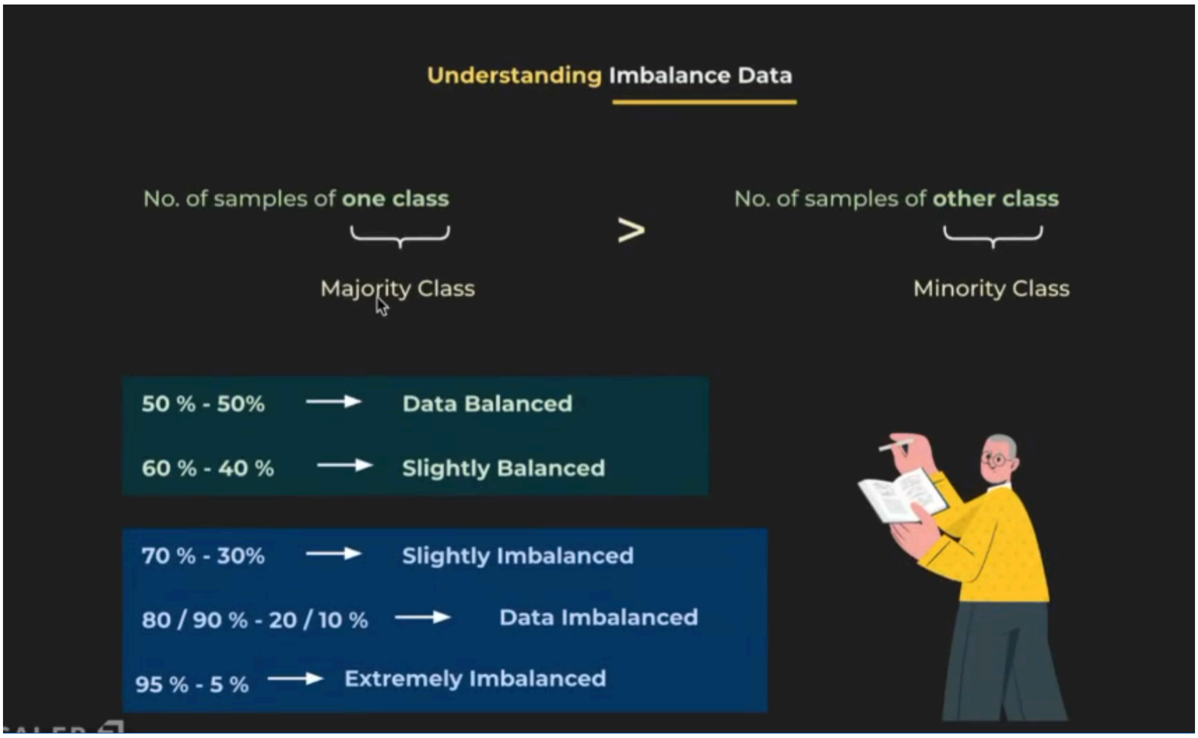

```
auc(recall, precision)
```

```
0.894367935686 1058
```

observe

Now the **AU-PRC** comes close to F1 score

- Showing that **PRC** worked just fine in imbalanced data


Understanding Imbalance Data
No. of samples of one class
Majority Class
>
No. of samples of other class
Minority Class
50 % - 50% → Data Balanced
60 % - 40 % → Slightly Balanced
70 % - 30% → Slightly Imbalanced
80 / 90 % - 20 / 10 % → Data Imbalanced
95 % - 5 % → Extremely Imbalanced