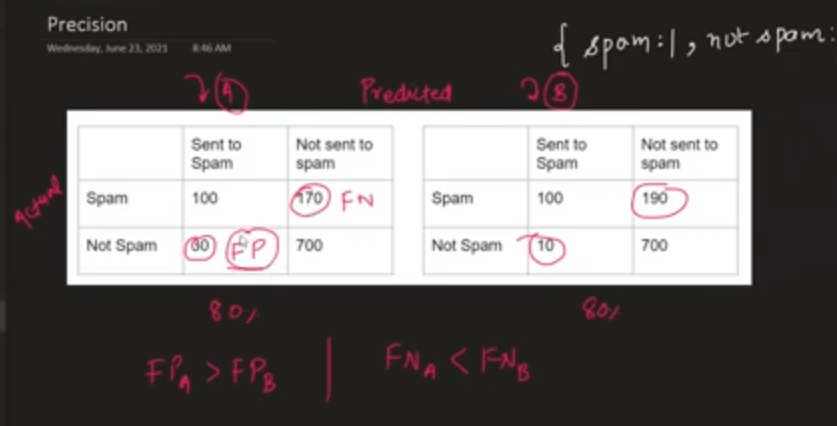
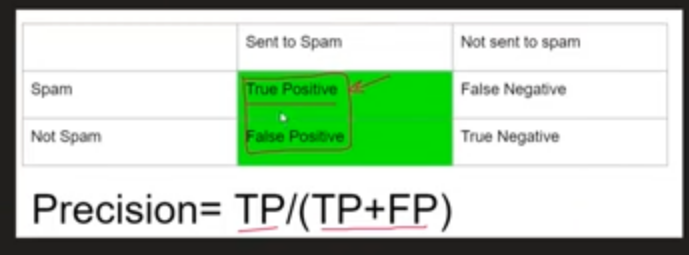
*Precision, Recall & F1 Score*

*Imbalanced Data ke time Accuracy use krna sahi nahi hai*

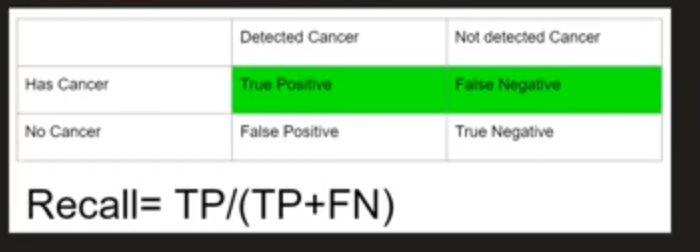
*Precision:*

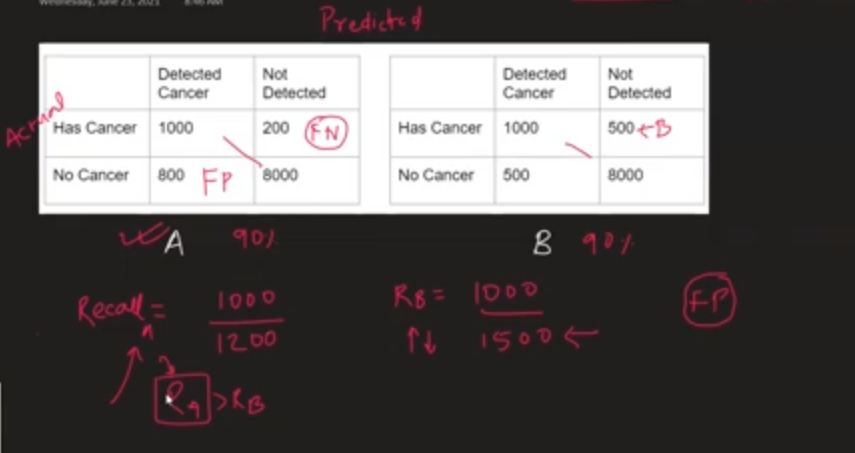
**

What is the Proportion of predicted Positive is truly Positive ? it is called Precision



Agar Type1 Galti zyaada High hai tou High Precision pakdenge humm





Agar Type2 Galti zyaada High hai tou High Recall pakdenge humm

The Problem:

Kabhi Kabhi kya hoga ki Hum Ese problem mai fass sakte jaha humme

Yeah smjhega hie nahi ki Type 1 Error zyaada Problamatic hai ki Type 2 Error Zyaada Problamatic hai.

Dono Precision & Recall ke beechmai Tradeoff hota haii

If we can’t define ki type 1 danger hai ki type 2 then yeah solve krneko

There is a

3rd Matrix called F1 Score ( Jo ki Harmonic Mean hota hai )

[ Harmonic Mean ka property yeah hai ki vo hamesa Lower side rahega ]

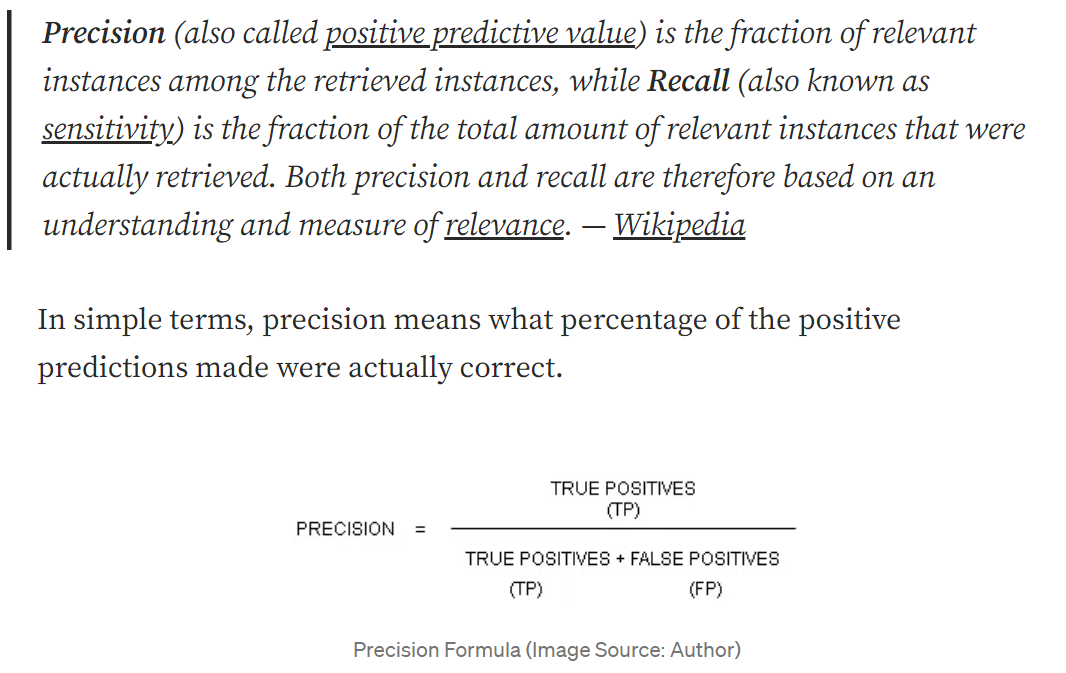
Tou hum F1 score panellize karnekeliye use karte haii

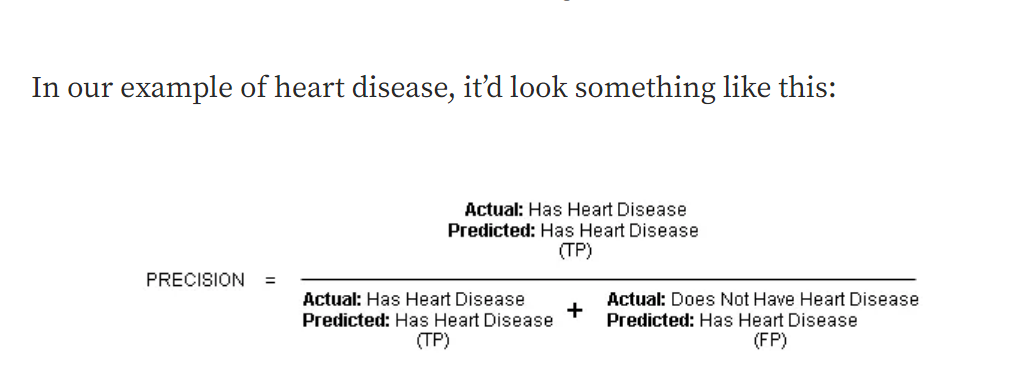


**Precision-Recall**

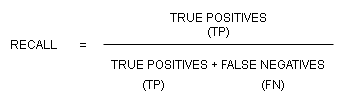
Now that you have understood what Confusion Matrix does, it’ll be easier to understand Precision-Recall.

We have already seen how accuracy can be misleading in some cases. Precision and Recall helps us further understand how strong the accuracy shown holds true for a particular problem.



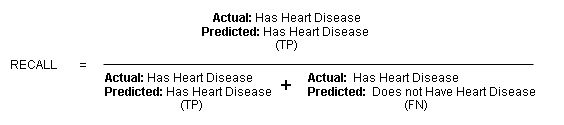


Recall in simple terms means, what percentage of actual positive predictions were correctly classified by the classifier.



Recall Formula (Image Source: Author)

In our example, it’d look something like this:



Recall Example (Image Source: Author)

It basically asks, of all the patients that have heart disease, how many were classified as having heart disease?

The formula may seem kinda identical at first, but once you get the gist, it’ll be harder to get confused between the two.

Precision-Recall is also available in the scikit learn package. You can look up the official documentation [here](https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html).

***Precision-Recall Trade-Off***

Suppose we train a Logistic Regression Classifier to identify whether the patient has heart disease or not. It will predict that the patient has heart disease if the probability (threshold) is greater than or equal to 0.5 and the patient doesn’t have heart disease if the probability is less than 0.5.

Now, if we want to build a model in such a way that it predicts if the patient has heart disease only if it is very confident in the hypothesis, we might have to increase the threshold to 0.7 or 0.8.

In this scenario, we end up with a classifier having **high precision** and**low recall**. Higher precision because now the classifier is more confident that the patient has heart disease. Lower recall because now that the classifier’s threshold is set so high, there will be fewer patients classified as having heart disease.

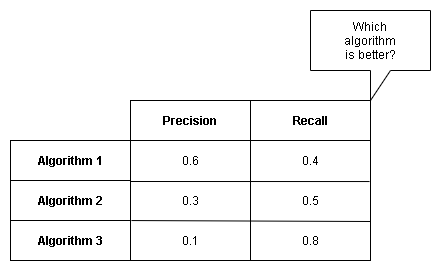
The alternative to this is we build a model in such a way that it won’t miss any possible cases of a patient having heart disease (to avoid false negatives). If a patient having heart disease goes unnoticed by the model, it could prove fatal. In this case, we decrease the threshold to 0.2 or 0.3 so that even if there is a slight chance that the patient may have heart disease, it raises an alarm and further diagnosis can be done to prove the hypothesis.

What we have here is an example of **high recall** and **low precision**. Higher recall because we will be classifying a larger number of patients having heart disease. Lower precision because out of a large number of patients predicted having heart disease some of them won’t actually have heart disease upon further diagnosis.

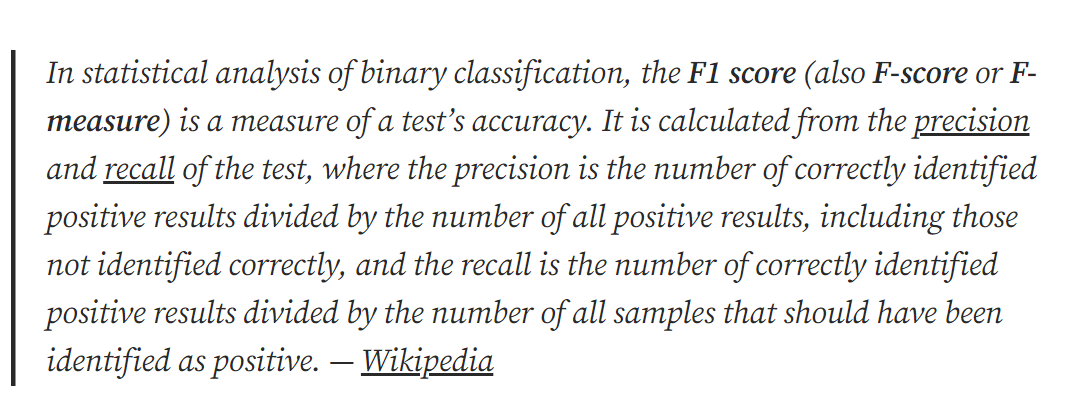
Generally speaking, the precision-recall values keep changing as you increase or decrease the threshold. Building a model with higher precision or recall depends on the problem statement you’re dealing with and its requirements.

# F1-Score

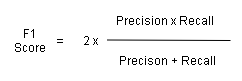
Precision-Recall values can be very useful to understand the performance of a specific algorithm and also helps in producing results based on the requirements. But when it comes to comparing several algorithms trained on the same data, it becomes difficult to understand which algorithm suits the data better solely based on the Precision-Recall values.



Hence there is a need for a metric that takes the precision-recall values and provides a standardized representation of those values.



F1 score can also be described as the harmonic mean or weighted average of precision and recall.



F1 Score Formula (Image Source: Author)

Having a precision or recall value as 0 is not desirable and hence it will give us the F1 score of 0 (lowest). On the other hand, if both the precision and recall value is 1, it’ll give us the F1 score of 1 indicating perfect precision-recall values. All the other intermediate values of the F1 score ranges between 0 and 1.

F1 Score is also available in the scikit learn package. You can look up the official documentation [here](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html).

*Multi-class Precision and Recall*