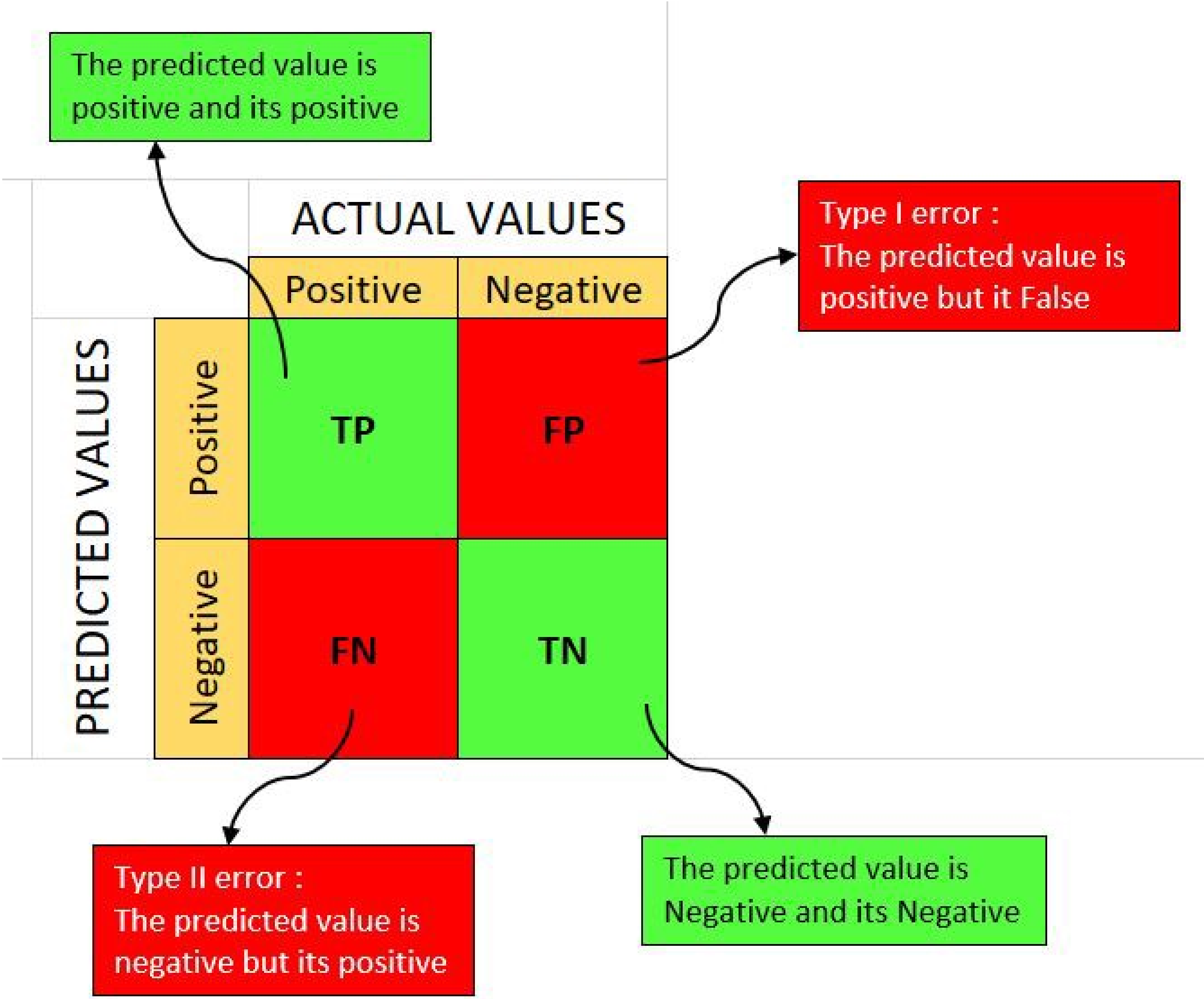
A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes.

The matrix compares the actual target values with those predicted by the machine learning model.



A good model is one which has high TP and TN rates, while low FP and FN rates.

If you have an imbalanced dataset to work with, it’s always better to use confusion matrix as your evaluation criteria for your machine learning model.

A confusion matrix is a tabular summary of the number of correct and incorrect predictions made by a classifier.

It is used to measure the performance of a classification model.

It can be used to evaluate the performance of a classification model through the calculation of performance metrics like accuracy, precision, recall, and F1-score.

We can measure model accuracy by two methods. Accuracy simply means the number of values correctly predicted.

1. Confusion Matrix

2. Classification Measure

Confusion Matrix

a. Understanding Confusion Matrix:

The following 4 are the basic terminology which will help us in determining the metrics we are looking for.

True Positives (TP): when the actual value is Positive and predicted is also Positive.

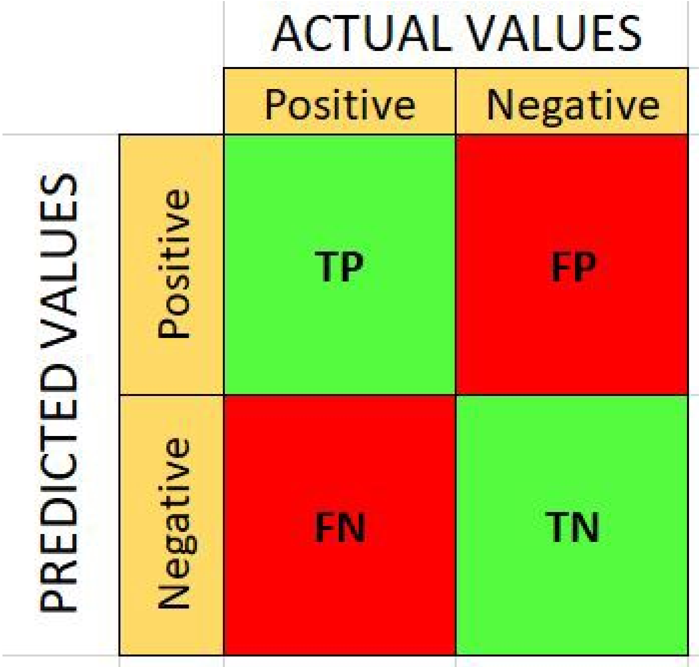
True negatives (TN): when the actual value is Negative and prediction is also Negative.

False positives (FP): When the actual is negative but prediction is Positive.

Also known as the Type 1 error

False negatives (FN): When the actual is Positive but the prediction is Negative.

Also known as the Type 2 error



The target variable has two values: Positive or Negative

The columns represent the actual values of the target variable

The rows represent the predicted values of the target variable

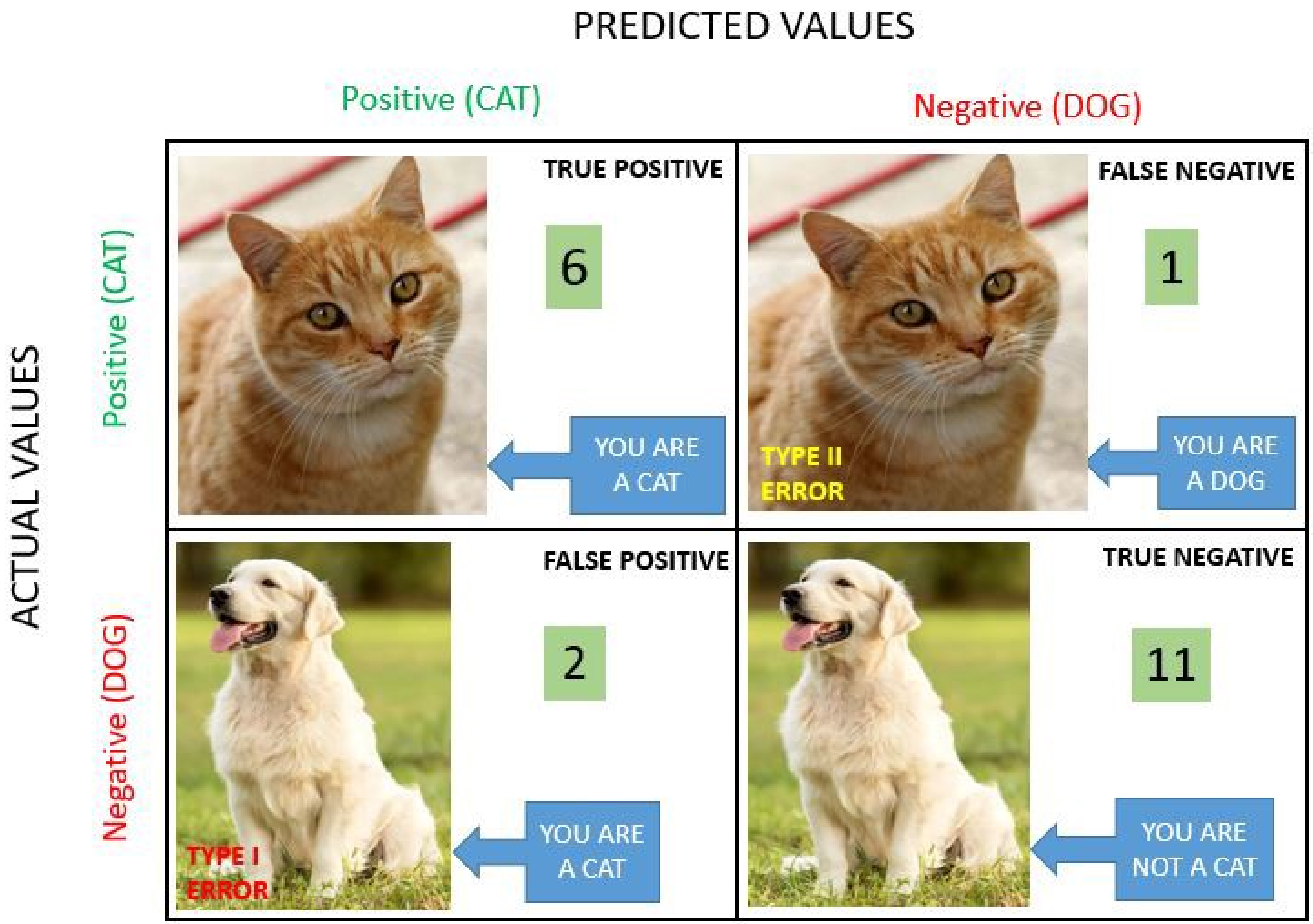
b. Understanding Confusion Matrix in an easier way:

Let’s take an example:

We have a total of 20 cats and dogs and our model predicts whether it is a cat or not.

```
Actual_values = ['dog', 'cat', 'dog', 'cat', 'dog', 'dog', 'cat', 'dog', 'cat', 'dog', 'dog', 'dog', 'dog', 'cat', 'dog', 'dog', 'cat

Predicted_values = ['dog', 'dog', 'dog', 'cat', 'dog', 'dog', 'cat', 'cat', 'cat', 'cat', 'dog', 'dog', 'dog', 'cat', 'dog', 'dog', '
```



True Positive (TP) = 6

You predicted positive and it’s true. You predicted that an animal is a cat and it actually is.

True Negative (TN) = 11

You predicted negative and it’s true. You predicted that animal is not a cat and it actually is not (it’s a dog).

False Positive (Type 1 Error) (FP) = 2

You predicted positive and it’s false. You predicted that animal is a cat but it actually is not (it’s a dog).

False Negative (Type 2 Error) (FN) = 1

You predicted negative and it’s false. You predicted that animal is not a cat but it actually is.

2. Classification Measure

Basically, it is an extended version of the confusion matrix. There are measures other than the confusion matrix which can help achieve better understanding and analysis of our model and its performance.

a. Accuracy

b. Precision

c. Recall (TPR, Sensitivity)

d. F1-Score

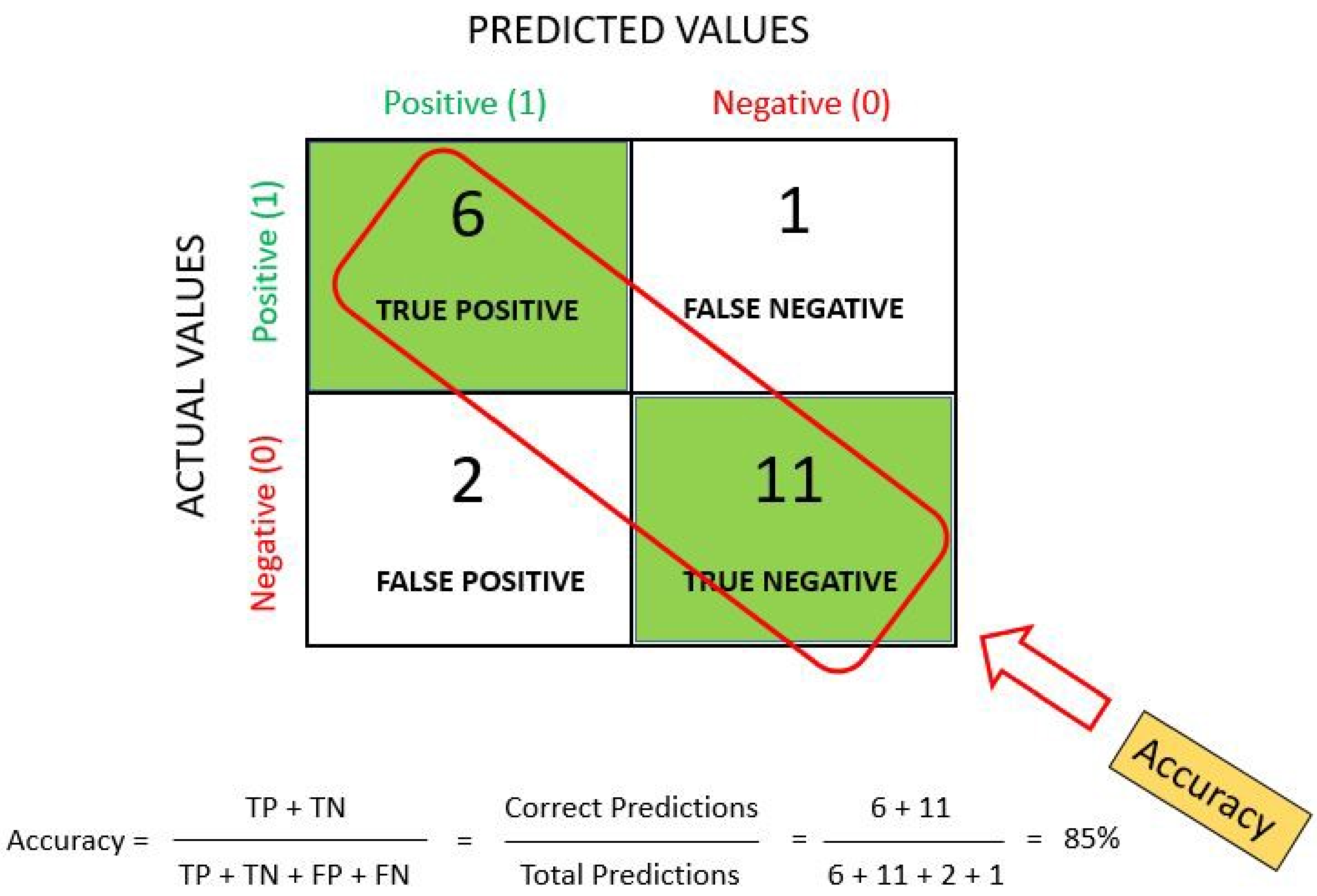
e. FPR (Type I Error)

f. FNR (Type II Error)

a. Accuracy:

Accuracy simply measures how often the classifier makes the correct prediction. It's the ratio between the number of correct predictions and the total number of predictions.

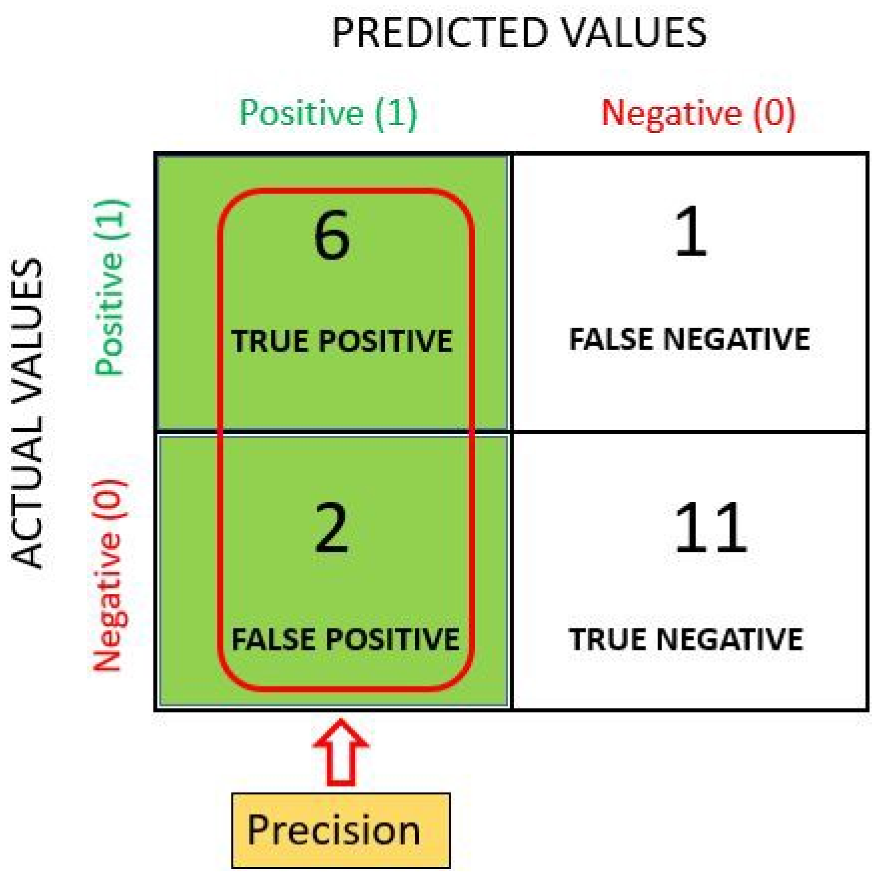
The accuracy metric is not suited for imbalanced classes. Accuracy has its own disadvantages, for imbalanced data, when the model predicts



b. Precision:

It is a measure of correctness that is achieved in true prediction.

In simple words, it tells us how many predictions are actually positive out of all the total positive predicted.



$$\text{Precision} = \frac{TP}{TP + FP} = \frac{\text{Predictions Actually Positive}}{\text{Total Predicted positive}} = \frac{6}{6+2} = 0.75$$

Ex 1:- In Spam Detection : Need to focus on precision

Suppose mail is not a spam but model is predicted as spam : FP (False Positive). We always try to reduce FP.

Ex 2:- Precision is important in music or video recommendation systems, e-commerce websites, etc.

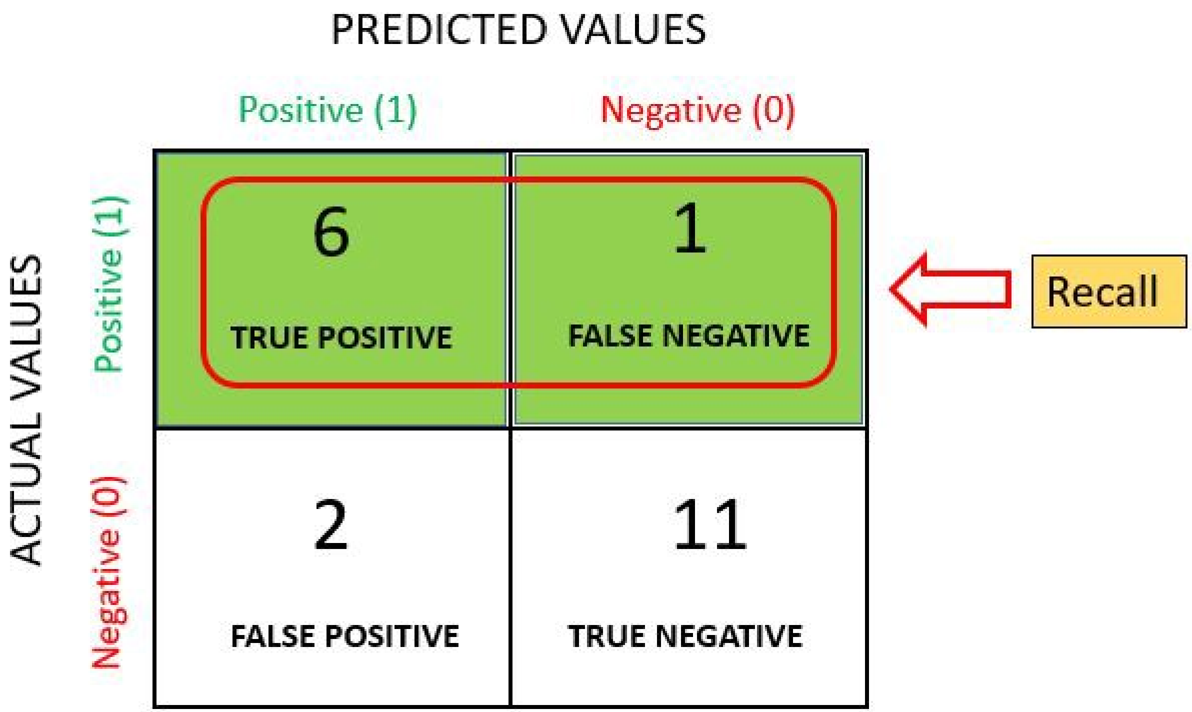
Wrong results could lead to customer churn and be harmful to the business

Recall:

It is a measure of actual observations which are predicted correctly, i.e. how many observations of positive class are actually predicted as positive. It is also known as Sensitivity. Recall is a valid choice of evaluation metric when we want to capture as many positives as possible.

Recall is defined as the ratio of the total number of correctly classified positive classes divide by the total number of positive classes.

Or, out of all the positive classes, how much we have predicted correctly. Recall should be high(ideally 1).



$$\text{Recall} = \frac{TP}{TP + FN} = \frac{\text{Predictions Actually Positive}}{\text{Total Actual positive}} = \frac{6}{6+1} = 0.85$$

Ex 1:- suppose person having cancer (or) not? He is suffering from cancer but model predicted as not suffering from cancer

Ex 2:- Recall is important in medical cases where it doesn't matter whether we raise a false alarm but the actual positive cases should not go undetected!

F-measure / F1-Score

The F1 score is a number between 0 and 1 and is the harmonic mean of precision and recall. We use harmonic mean because it is not sensitive to extremely large values, unlike simple averages.

F1 score sort of maintains a balance between the precision and recall for your classifier. If your precision is low, the F1 is low and if the recall is low again your F1 score is low.

There will be cases where there is no clear distinction between whether Precision is more important or Recall. We combine them!

In practice, when we try to increase the precision of our model, the recall goes down and vice-versa. The F1-score captures both the trends in a single value.

$$\text{F1-Score} = 2* \frac{(\text{Recall}*\text{Precision})}{(\text{Recall} + \text{Precision})} = 2* \frac{(0.85*0.75)}{(0.85+0.75)} = 0.79$$

F-score should be high(ideally 1).

```
Actual_values = ['dog', 'cat', 'dog', 'cat', 'dog', 'dog', 'cat', 'dog', 'cat', 'dog', 'dog', 'dog', 'dog', 'cat', 'dog', 'dog', 'cat
```

```
Predicted_values = ['dog', 'dog', 'dog', 'cat', 'dog', 'dog', 'cat', 'cat', 'cat', 'cat', 'dog', 'dog', 'dog', 'cat', 'dog', 'dog', '
```

```
from sklearn.metrics import confusion_matrix
```

```
final_results = confusion_matrix(Actual_values,Predicted_values)
```

```
from sklearn.metrics import accuracy_score
accuracy=accuracy_score(Actual_values,Predicted_values)
```

```
accuracy
```

```
0.85
```

```
from sklearn.metrics import classification_report
report=classification_report(Actual_values,Predicted_values)
```

```
print(final_results)
```

```
[[ 6  1]
 [ 2 11]]
```

```
report
```

```
'              precision    recall  f1-score   support\n\n         cat       0.75      0.86      0.80         7\n         dog
0.92      0.85      0.88        13\n\n    accuracy                           0.85        20\n   macro avg       0.83      0.85
0.84        20\nweighted avg       0.86      0.85      0.85        20\n'
```

✓ 0s completed at 8:50 PM