Precision/Recall, F1 Score, Confusion Matrix

Topics

Classification metrics
Accuracy, and why it’s often not useful
Precision/Recall
F1 Score
Confusion Matrix

Classification

These metrics apply to classification not regression problems.

Classification: predict a class
Regression: Predict a number

Accuracy, and why it's not as useful

It's a fine measure if you have a balanced dataset. Or roughly the same number of every class in the dataset.

Accuracy fails if you have an unbalanced dataset, for instance

Cancer diagosis dataset

Or credit card fraud dataset

There are likely to be very few positive cancer diagnosis, or frauds in the above datasets.

For each of these datasets your model can always predict the majority class and score high accuracy. (The more unbalanced, the better the accuracy)

A better way- Precision/Recall/F1

Precision: Out of all found how accurate were predictions?

$$P = \frac{T_p}{T_p + F_p}$$

Recall: How accurate if you consider all?

$$R = \frac{T_p}{T_p + F_n}$$

F1 score: The harmonic mean of precision and recall

$$F1 = 2\frac{P \times R}{P + R}$$

Tp true positives
Fp false positives
Tn true negatives
Fn false negatives

Precision/Recall/F1

$$P = \frac{T_p}{T_p+F_p} \qquad R = \frac{T_p}{T_p+F_n} \qquad F1 = 2\frac{P \times R}{P+R}$$

Example: A database has 100 items, 60 are Positive,40 are Negative

Precision/Recall/F1

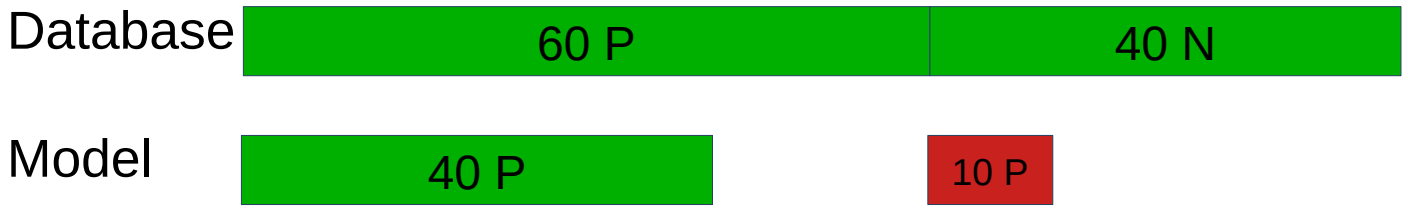
$$P = \frac{T_p}{T_p+F_p} \qquad R = \frac{T_p}{T_p+F_n} \qquad F1 = 2\frac{P \times R}{P+R}$$

Example: A database has 100 items, 60 are Positive,40 are Negative
Model predicts 50 positives, 40 are correct

Precision/Recall/F1

$$P = \frac{T_p}{T_p+F_p} \qquad R = \frac{T_p}{T_p+F_n} \qquad F1 = 2\frac{P \times R}{P+R}$$

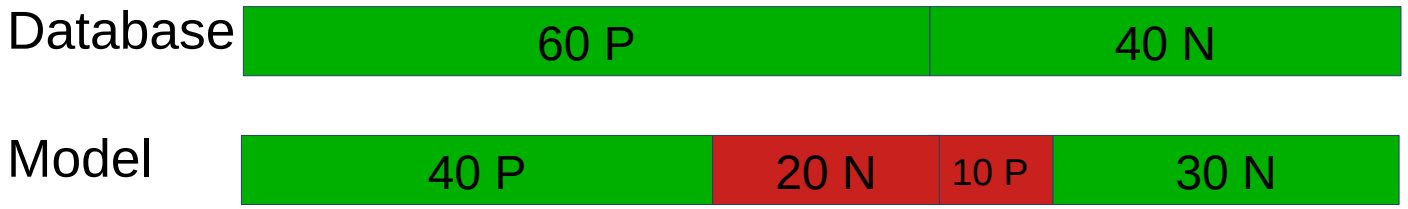
Example: A database has 100 items, 60 are Positive,40 are Negative
Model predicts 50 positives, 40 are correct



Precision/Recall/F1

$$P = \frac{T_p}{T_p+F_p} \qquad R = \frac{T_p}{T_p+F_n} \qquad F1 = 2\frac{P \times R}{P+R}$$

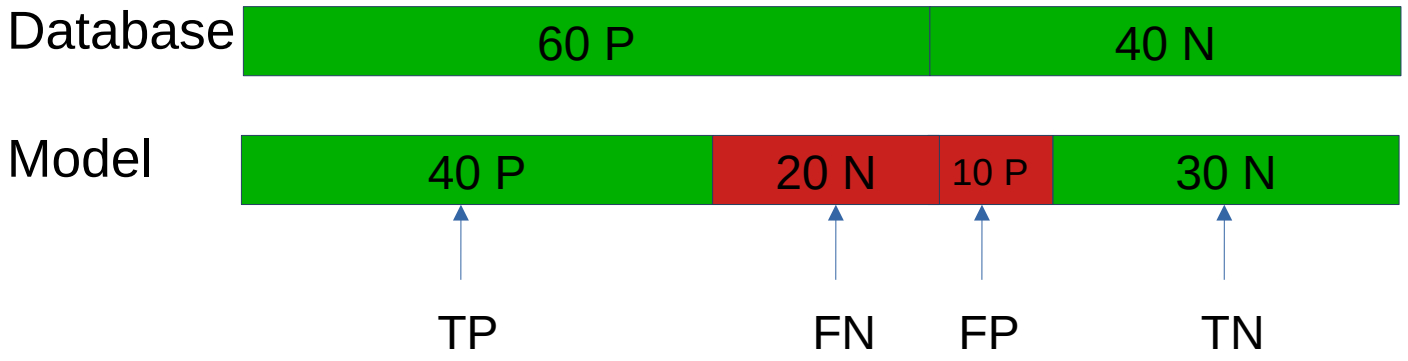
Example: A database has 100 items, 60 are Positive,40 are Negative
Model predicts 50 positives, 40 are correct



Precision/Recall/F1

$$P = \frac{T_p}{T_p+F_p} \qquad R = \frac{T_p}{T_p+F_n} \qquad F1 = 2\frac{P \times R}{P+R}$$

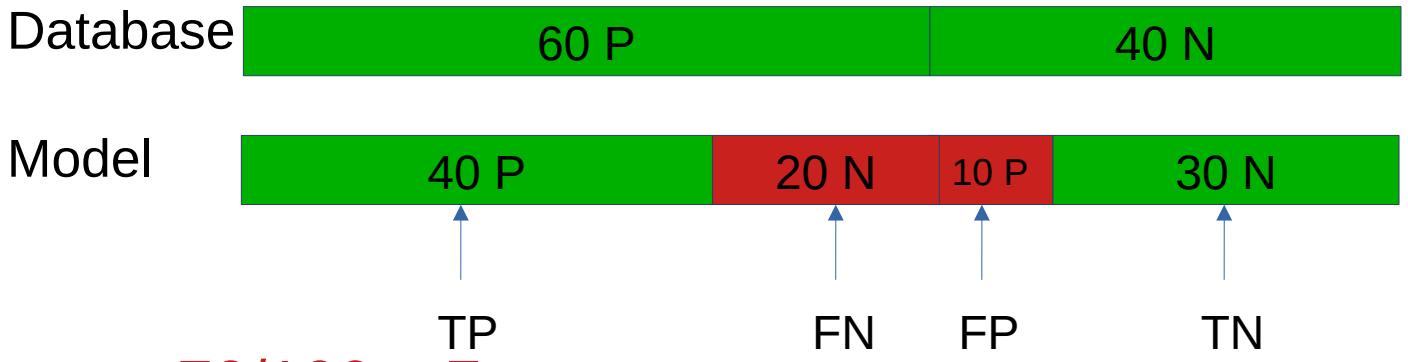
Example: A database has 100 items, 60 are Positive,40 are Negative
Model predicts 50 positives, 40 are correct



Precision/Recall/F1

$$P = \frac{T_p}{T_p+F_p} \qquad R = \frac{T_p}{T_p+F_n} \qquad F1 = 2\frac{P \times R}{P+R}$$

Example: A database has 100 items, 60 are Positive,40 are Negative
Model predicts 50 positives, 40 are correct



Accuracy= 70/100= .7
P=40/(40+10)=.8 (how accurate finding Positives?)
R= 40/(40+20) =.66 (how accurate overall?)
F1= 2*(.8*.66)/(.8+.66) = .723

Strive for high precision and recall

Precision/Recall/F1 for Multiple classes

When you have more than 2 classes?
Precision:
Just calculate TP and FP for each class and sum;

Precision/Recall/F1 for Multiple classes Example

A multiclass classification problem with 3 classes.
A,B,C
With a 1:1:100 class ratio (100 times as many C's as A's)

If a dataset has 10,000 C's it will have 100 A's and B's

Precision/Recall/F1 for Multiple classes Example

A multiclass classification problem with 3 classes.
A,B,C
With a 1:1:100 class ratio (100 times as many C's as A's)

If a dataset has 10,000 C's it will have 100 A's and B's

$$P = \frac{T_p}{T_p + F_p}$$

A model predicts 70 A's, 50 are correct and 20 wrong
Predicts 150 B's, 99 are correct and 51 wrong

Precision=(50 +99)/((50+99) + (20 + 51))
=.677

Confusion Matrix

Very simple display:
shows the number right and wrong for every class

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,res)
```

```
array([[42,  9],
       [ 1, 91]])
```

Summary

Accuracy is misleading, especially if your dataset is imbalanced
Precision/Recall
F1 Score
Confusion Matrix- just a visual way to check your models predictions against correct values for all classes