Machine Learning

6. Metrics

Yannick Le Cacheux

CentraleSupélec - Université Paris Saclay

September 2024

Outline

1 Metrics for classification

2 Metrics for regression

What accuracy is "good"?

- Is 90% accuracy good?

[1] Even though we have not covered state-of-the-art methods for image classification, *e.g.* convolutional neural networks or visual transformers, such methods exist and work really well for such tasks.

What accuracy is "good"?

- Is 90% accuracy good?

(a) A dog.

(b) A cat.

- It depends on the task
  - For instance: cat *vs* dog classification is fairly easy nowadays, so 90% accuracy is unimpressive[1]

[1] Even though we have not covered state-of-the-art methods for image classification, *e.g.* convolutional neural networks or visual transformers, such methods exist and work really well for such tasks.

What accuracy is "good"?

Figure: Chihuahuas and muffins

- The chihuahua *vs* muffin classification task is harder

What accuracy is "good"?

No general rule

The levels at which performance may be considered great, reasonable or disapointing depend on the task and the data.

- Binary classification for cat *vs* dog is easy
- Multi-class classification for "Siamese cat" *vs* "Himalayan cat" *vs* "Siberian cat" *vs* ... with dozens of possible cat species, including mixed-race samples, is harder
- Predicting whether Apple stock will be higher or lower a week from now is mind-boggingly hard, although it is again binary classification
- Predicting the outcome of an unbiased coin flip should be close to impossible[2]

[2] Or predicting the spin of an electron if you really want true randomness

Is accuracy always relevant?

Consider the following case:

- We have information about a patient
- We want to predict whether the patient will get a disease, *overfitticus neuralis*.
- 10% of people get the disease

- A model simply predicting "No" every time will have 90% accuracy
  - Is it a useful model?

Is accuracy always relevant?

- Even worse: assume the following "ground truth"
  - 70% of subjects have a 0% risk of getting the disease
  - 30% of subjects have a 33% risk of getting the disease

- The accuracy of the baseline "always no" model is 90%
- What is the highest possible accuracy?

Accuracy

Highest accuracy

The best achievable accuracy for the previous example is... 90%

- Why? Because even if we correctly identify the 30% of subjects with a risk of getting the disease, they are still more likely to *not* catch the disease.
- Yet, a model correctly identifying the subjects with a non zero risk of getting the disease is much more useful than a model predicting a 0% risk all the time

Moral of the story

Accuracy is not always a relevant metric to measure the performance of a classification model.

True and false positives / negatives

- Notations:
  - TP: True Positive (predicted 1 when correct answer is 1)
  - FP: False Positive (predicted 1 instead of 0)
  - TN: True Negative (predicted 0 when correct answer is 0)
  - FN: False Negative (predicted 0 instead of 1)

		Actual value	
		1	0
Pred.	1	TP	FP
	0	FN	TN

Table: Confusion Matrix

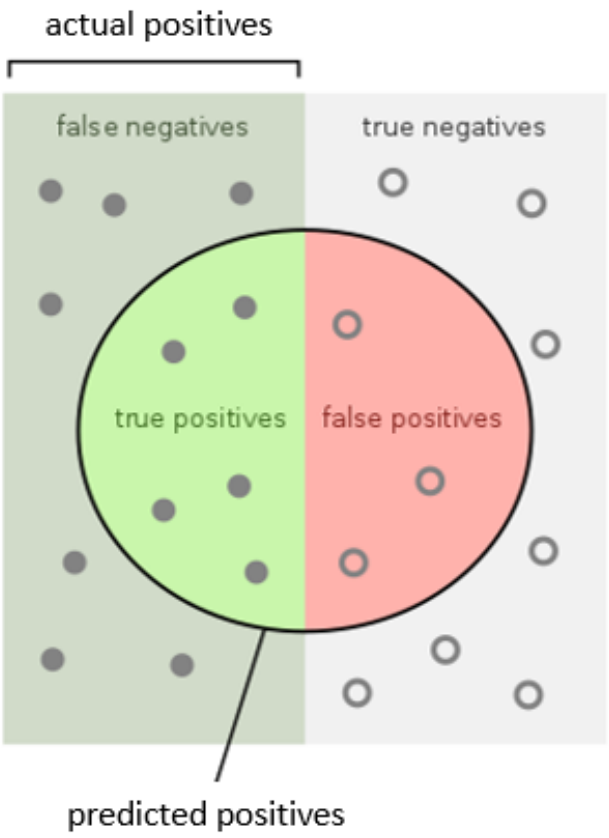


Figure: [3] Illustration of true positives, false positives etc

From Wikipedia

- *(Side question: what is the confusion matrix of the previous example?)*

Decision threshold

- We can change the decision threshold T we use to predict whether a patient will get the disease:
  - We predict *"yes"* (or 1) if the estimated probability $\hat{y} \in [0, 1]$ is $\geq T$, otherwise we predict *"no"* (or 0).

Patient	$\hat{y}$	$T = 0.2$	$T = 0.4$	$T = 0.6$	$T = 0.8$
Alice	0.12	✗	✗	✗	✗
Bob	0.34	✓	✗	✗	✗
Carol	0.46	✓	✓	✗	✗
Dave	0.64	✓	✓	✓	✗
Eve	0.87	✓	✓	✓	✓

Table: Binary predictions ✓/✗ based on decision threshold T and estimated probability $\hat{y}$

- *(Side question: can this improve the accuracy of our previous example?)*

Receiver Operating Characteristic

Receiver Operating Characteristic (ROC) curve:

- We use the FP rate (FPR) as the x axis
- We use the TP rate (TPR) as the y axis
- We vary the threshold T between 0.0 and 1.0, and plot the corresponding FP and TP values

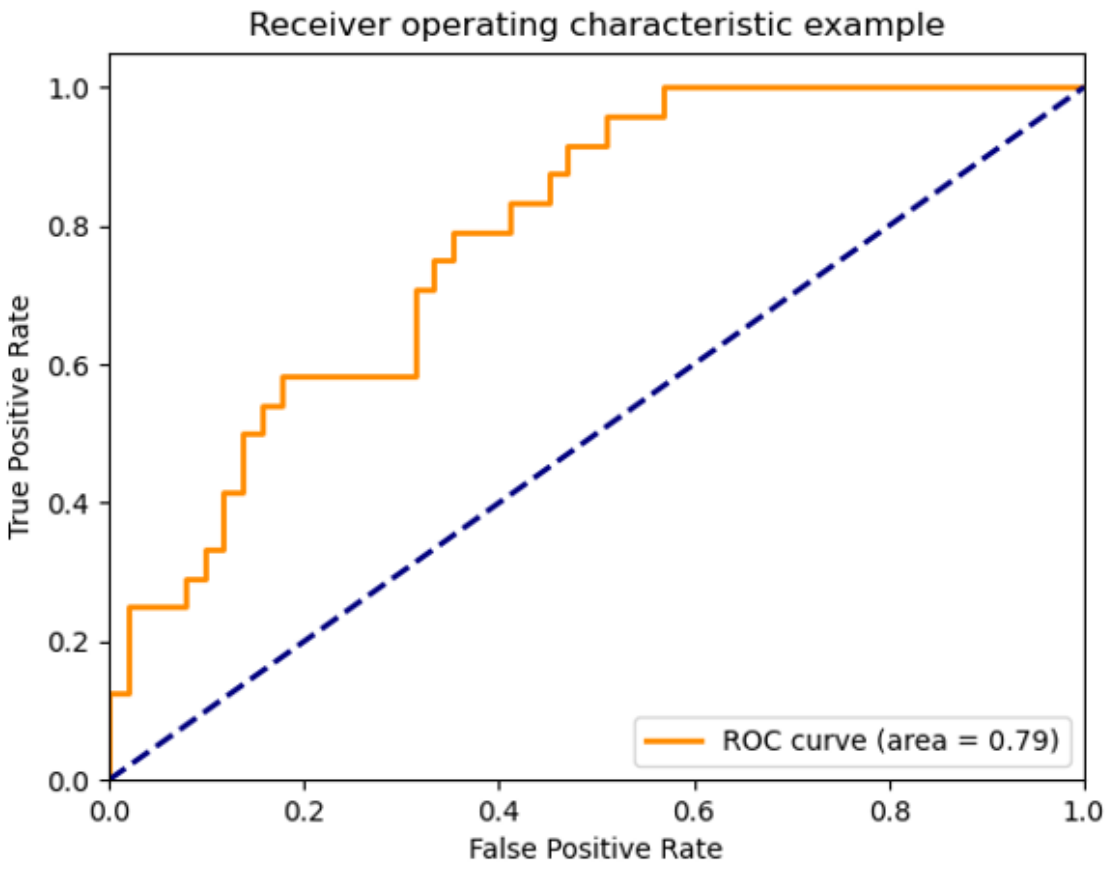


Figure: Example ROC curve, from scikit-learn documentation

ROC curve example

Patient	$\hat{y}$	y	$T = 0.2$	$T = 0.4$	$T = 0.6$	$T = 0.8$
Alice	0.12	✗	✗	✗	✗	✗
Bob	0.34	✗	✓	✗	✗	✗
Carol	0.46	✓	✓	✓	✗	✗
Dave	0.64	✗	✓	✓	✓	✗
Eve	0.87	✓	✓	✓	✓	✓

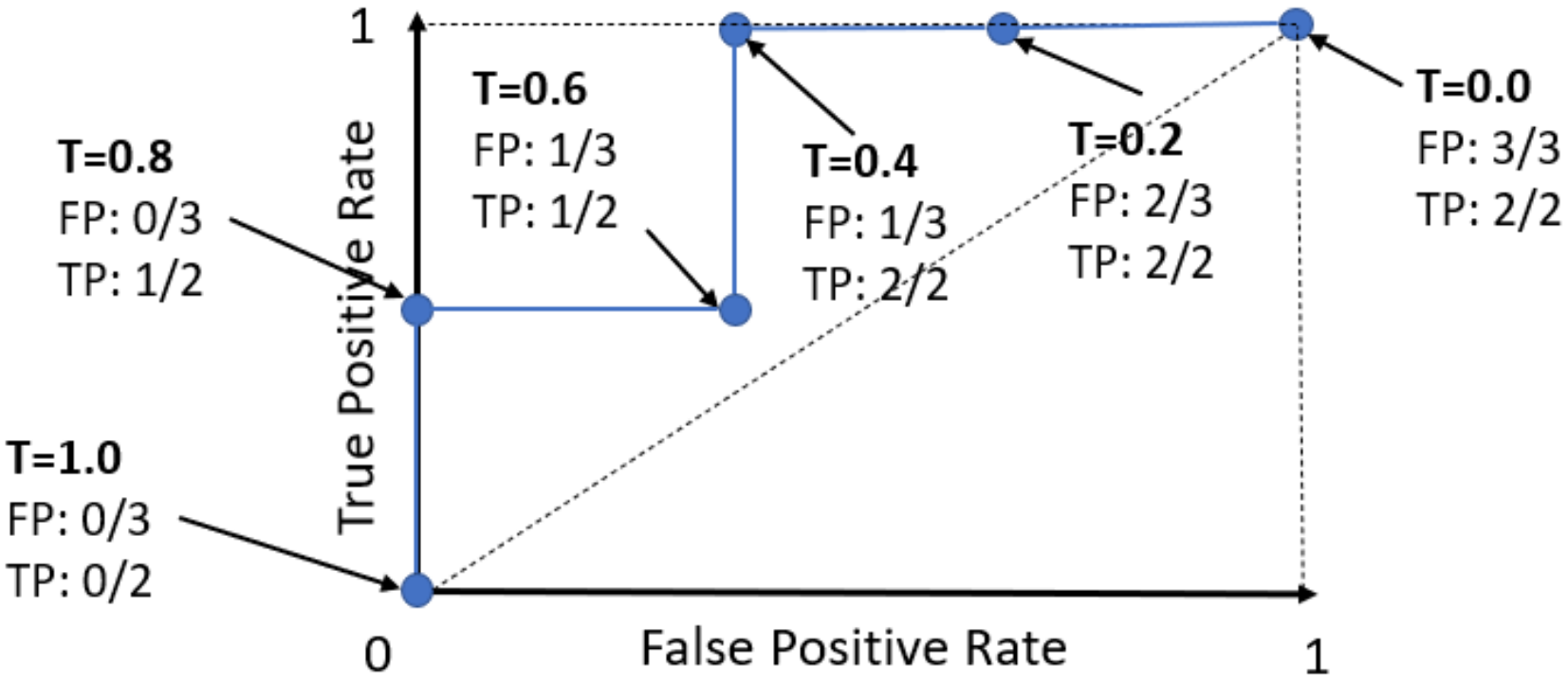


Figure: ROC curve of previous example

Area under a ROC curve

The *AUC* or Area Under the (ROC) Curve can be used as a metric

- AUC $= 0.5$ means the model makes random predictions
  - (TP rate grows as fast as FP rate)
- AUC $= 1.0$ means that there exists a threshold T such that all estimations $> T$ have class 1, and all $< T$ have class 0
  - It is then (theoretically) possible to have 100% accuracy
- AUC < 0.5 means?

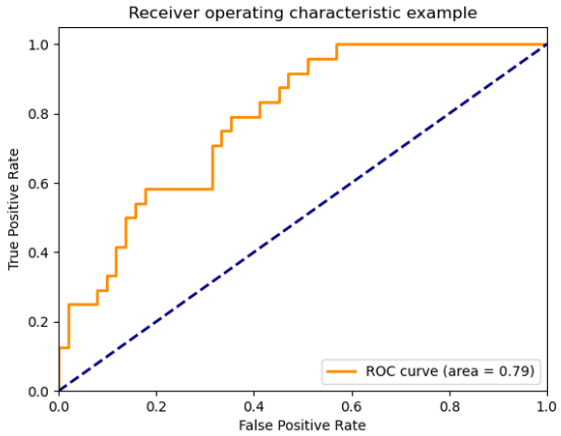


Figure: ROC curve with AUC 0.79. Dotted line corresponds to random predictions.

Area under a ROC curve

The *AUC* or Area Under the (ROC) Curve can be used as a metric

- AUC $= 0.5$ means the model makes random predictions
  - (TP rate grows as fast as FP rate)
- AUC $= 1.0$ means that there exists a threshold T such that all estimations $> T$ have class 1, and all $< T$ have class 0
  - It is then (theoretically) possible to have 100% accuracy
- AUC < 0.5 means that we probably messed up somewhere:
  - Predictions are worse than random chance
  - (but predicting the oppositive of what we do is actually useful!)

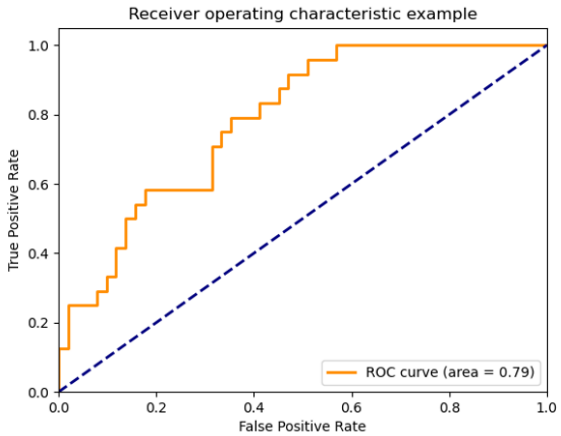


Figure: ROC curve with AUC 0.79. Dotted line corresponds to random predictions.

Back to previous example

- Assuming again the Ground Truth (GT) is: 30% of patients have a 33% risk of disease

Est. prob.	Disease
0.0	✗
0.0	✗
0.0	✗
0.0	✗
0.0	✗
0.0	✗
0.0	✗
0.3	✓
0.3	✗
0.3	✗

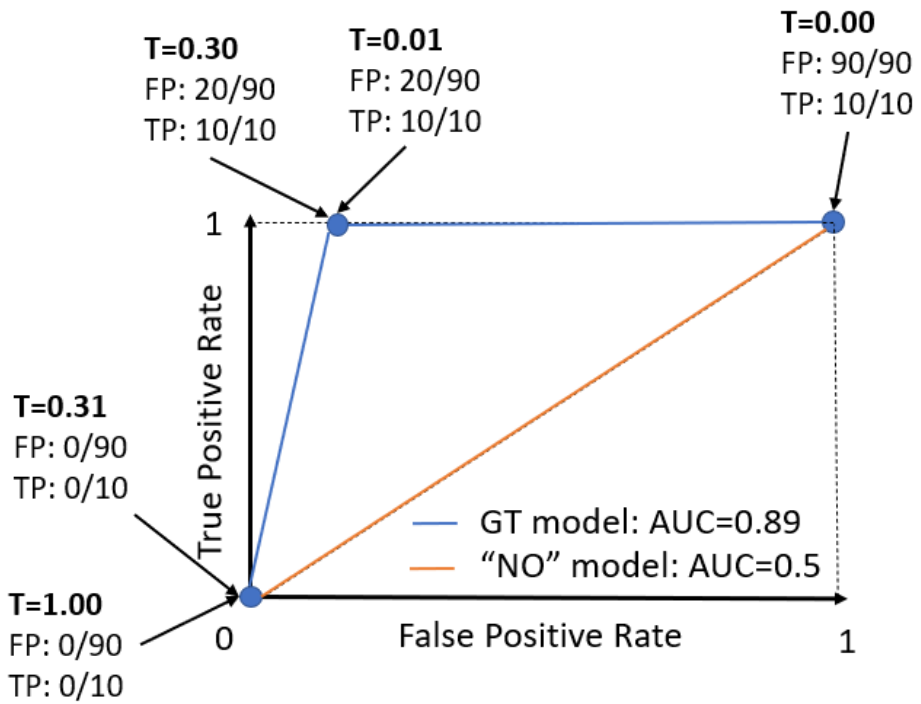
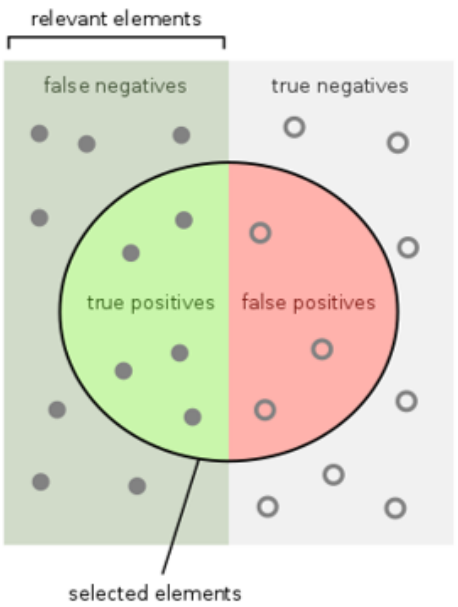


Figure: ROC curve of ground truth model *vs* baseline model

- The ROC curve tells us that GT model is better than the "NO" baseline!

Sensitivity / specificity

- ROC curves are sometimes called sensitivity / specificity curves



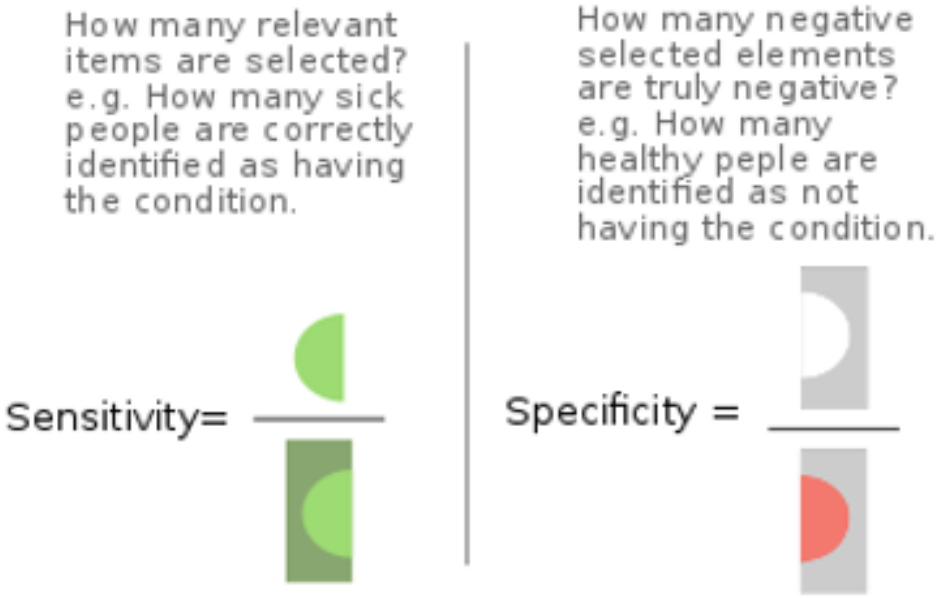


Figure: Illustration of sensitity and specificity[4]

$$\text{Sensitivity} = \frac{TP}{TP + FN} = TPR \quad \text{(True Positive Rate)} = \text{Recall}$$

$$\text{Specificity} = \frac{TN}{TN + FP} = TNR \quad \text{(True Negative Rate)} = 1 - FPR$$

Precision / recall

- Another commonly used (pair of) metric is Precision and Recall

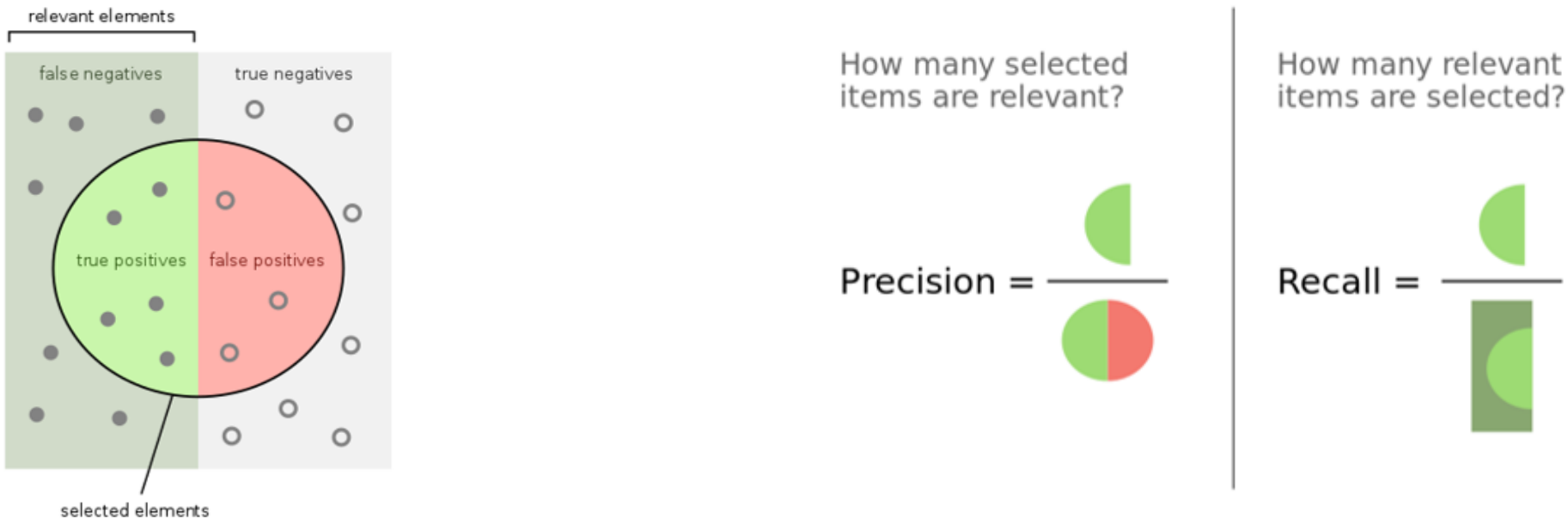


Figure: Illustration of precision and recall[6]

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{Recall} = \frac{TP}{TP + FN}$$

[6] From Wikipedia

Precision / recall

- Can we use only recall (or precision) as a metric?

[7] From the scikit-learn documentation

Precision / recall

- Can we use only recall (or precision) as a metric?
  - No: decreasing the classification "threshold" will usually increase recall (as more samples are "selected") but decrease precision
  - There is a trade-off between the two
- We can also plot precision-recall curves:

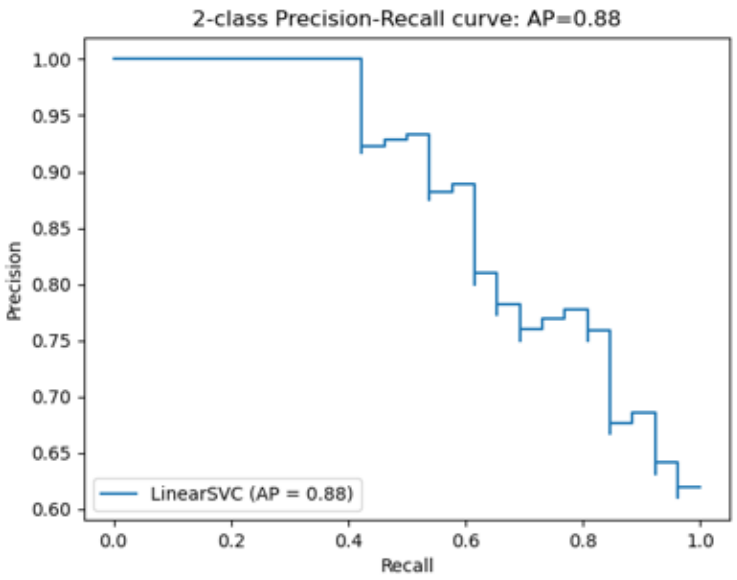


Figure: Precision recall curve[7]

[7] From the scikit-learn documentation

F1 score

- Another commonly used metric is the $F1$-score, which is the *harmonic mean* of precision (P) and recall (R):

$$F1(P, R) = 2 \cdot \frac{P \cdot R}{P + R}$$

- It incentivizes a good balance between P and R

$H(70, 70) = 70$ $\quad$ $H(80, 60) = 69$ $\quad$ $H(50, 90) = 64$ $\quad$ $H(40, 100) = 57$

Other metrics for classification

- There are many other metrics for different situations, for example ranking metrics such as mean average precision, recall@K...
- There are also similar metrics for multi-class classification, for multi-label classification...

Many possibilities

As usual, there are too many possibilities for us to cover everything during class

Outline

1 Metrics for classification

2 Metrics for regression

Metrics for regression

- We have already seen Mean Square Error (MSE):

$$\text{MSE} = \frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2$$

- The Root Mean Square Error (RMSE) "rescales" the MSE to be on the same scale as our errors:

$$\text{MSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}$$

- It is sometimes more intuitive to use the Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{N}\sum_{n=1}^{N}|y_n - \hat{y}_n|$$

Explained variance

- All previous metrics are sensitive to the scale of the targets
- If we want a metric which does not depend on the scale of the data, we can use Explained Variance[8]:

$$\text{EV} = 1 - \frac{\text{Var}[y - \hat{y}]}{\text{Var}[y]} = 1 - \frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{\sum_{n=1}^{N} y_n^2}$$

- Explained variance:
  - is 1.0 if the predictions are perfect: $\hat{y}_n = y_n \forall n$
  - is 0.0 if we always predict the mean of the targets: $\hat{y}_n = \frac{1}{N}\sum_n y_n \forall n$
  - has a worst possible value of?

[8] We have already met explained variance in lab 3 on PCA

Explained variance

- All previous metrics are sensitive to the scale of the targets
- If we want a metric which does not depend on the scale of the data, we can use Explained Variance[8]:

$$\text{EV} = 1 - \frac{\text{Var}[y - \hat{y}]}{\text{Var}[y]} = 1 - \frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{\sum_{n=1}^{N} y_n^2}$$

- Explained variance:
  - is 1.0 if the predictions are perfect: $\hat{y}_n = y_n \forall n$
  - is 0.0 if we always predict the mean of the targets: $\hat{y}_n = \frac{1}{N}\sum_n y_n \forall n$
  - can tend to $-\infty$ if we make *really* bad predictions

[8] We have already met explained variance in lab 3 on PCA

R^2 score

- The R^2 score is close to the explained variance, and can be interpreted roughly similarly:

$$R^2 = 1 - \frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{\sum_{n=1}^{N}(y_n - \frac{1}{N}\sum_n y_n)^2}$$

R^2:

- is 1.0 for perfect predictions
- is 0.0 if we always predict the mean of the target
- if the mean error is 0, R^2 is the same as explained variance

Relative errors

- Sometimes, we are interested in the relative scale of the error with respect to individual targets. MAE does not reflect that:

$$\text{MAE} = \frac{1}{N}\sum_{n=1}^{N}|y_n - \hat{y}_n|$$

- We can use Mean Average Percentage Error (MAPE):

$$\text{MAPE} = \frac{1}{N}\sum_{n=1}^{N}|\frac{y_n - \hat{y}_n}{y_n}|$$

Relative errors

- Sometimes, we are interested in the relative scale of the error with respect to individual targets. MAE does not reflect that:

$$\text{MAE} = \frac{1}{N}\sum_{n=1}^{N}|y_n - \hat{y}_n|$$

- We can use Mean Average Percentage Error (MAPE):

$$\text{MAPE} = \frac{1}{N}\sum_{n=1}^{N}|\frac{y_n - \hat{y}_n}{y_n}|$$

- If we want errors on larger y_n to have more impact, we can use Weighted Average Percentage Error (WAPE) instead:

$$\text{WAPE} = \frac{1}{N}\frac{\sum_{n=1}^{N}|y_n - \hat{y}_n|}{\sum_{n=1}^{N}|y_n|}$$