Machine Learning
Module 4: Metrics

Marc-Olivier Boldi

Master in Management, Business Analytics, HEC UNIL

Spring 2025

Table of Contents

Table of Contents

Concept

A **metric** (score) is a numerical value measuring the performance of a model

To compare the different models, the metric *only* depends on the observed and predicted values or classes. There are metrics for regression and metrics for classification.
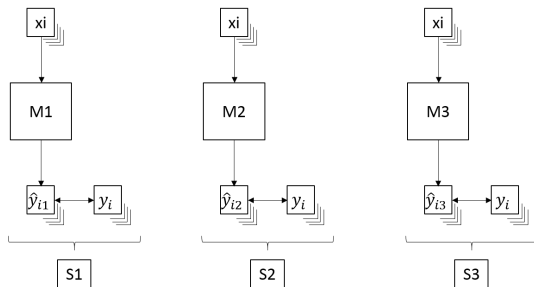
Table of Contents

Metrics for classifications

Metrics for classifications can be grouped as

- Metrics based on the predicted classes only (e.g., accuracy)
- Metrics based on the predicted probabilities (e.g., AUC)
- Metrics for binary classifications only (e.g., specificity, sensitivity, F_1)

Confusion matrix

The **confusion matrix** reports the frequencies of "observed" vs "predicted" classes. E.g., SVM with linear kernel and cost $C = 1$, on iris data:

```
           Reference
Prediction  setosa versicolor virginica
  setosa        50          0         0
  versicolor     0         48         2
  virginica      0          2        48
```

All the metrics based on predicted classes can be computed from the confusion matrix.

Accuracy

The **accuracy** is the proportion of correct predictions of the model

$$\mathcal{A} = \sum_i n_{ii} / \sum_{ij} n_{ij}$$

where n_{ij} is the number in the confusion matrix at line i and column j.

For the previous SVM model, the accuracy is

$$\mathcal{A} = \frac{50 + 48 + 48}{50 + 2 + 48 + 48 + 2} = 0.973$$

Cohen's kappa

Cohen's kappa is a measure which compares the observed accuracy to the accuracy that one would expect from a random model.

The **expected accuracy** from a random model is computed as the accuracy from a table containing the expected frequencies n_{ij} if the model was random[1].

This expected accuracy is also called **No Information Rate**.

---

[1]The calculations are the same as for the chi-squared test for independence.

Cohen's kappa

Suppose we have the following confusion matrix for a 3-class classification problem on $n = 182$ observations:

Observed confusion matrix (frequencies)							Observed confusion matrix (proportions)					
	Observations							Observations				
		A	B	C	Total				A	B	C	Total
Predictions	A	45	14	1	60		Predictions	A	24.7%	7.7%	0.5%	33.0%
	B	5	48	2	55			B	2.7%	26.4%	1.1%	30.2%
	C	10	2	55	67			C	5.5%	1.1%	30.2%	36.8%
	Total	60	64	58	182			Total	33.0%	35.2%	31.9%	100.0%

The **observed accuracy** is

$$A_o = \frac{45 + 48 + 55}{182} = 0.247 + 0.264 + 0.302 = 0.813.$$

Cohen's kappa

A random model would have proportions in each cell resulting from the independence between "Observation" and "Prediction". E.g.,

$$
\begin{aligned}
P(Obs = B \cap Pred = A) &= P(Obs = B)P(Pred = A) \\
&= 0.330 \times 0.352 \\
&= 0.116.
\end{aligned}
$$

Given that there are $n = 182$ observations, we would expect in this cell to contain a frequency of

$$
0.166 \times 182 = 21.1.
$$

Applying this to the whole table provides the table of expected frequencies under independence.

Cohen's kappa

Expected confusion matrix (proportions)						Expected confusion matrix (frequencies)					
	Observations						Observations				
		A	B	C	Total			A	B	C	Total
Predictions	A	10.9%	11.6%	10.5%	33.0%	Predictions	A	19.8	21.1	19.1	60.0
	B	10.0%	10.6%	9.6%	30.2%		B	18.1	19.3	17.5	55.0
	C	12.1%	12.9%	11.7%	36.8%		C	22.1	23.6	21.4	67.0
	Total	33.0%	35.2%	31.9%	100.0%		Total	60.0	64.0	58.0	182.0

The expected accuracy (under independence) is

$$A_e = \frac{19.8 + 19.3 + 21.4}{182} = 0.109 + 0.106 + 0.117 = 0.332$$

Therefore, a model without link between observations and predictions would reach an accuracy of 0.332.

Cohen's kappa

By definition, the Cohen's kappa is

$$\kappa = \frac{A_o - A_e}{1 - A_e}$$

- $\kappa = 0$ if $A_o = A_e$ when the accuracy of the model is the same as a random model (i.e., it is not a good model).
- $\kappa = 1$ if $A_o = 1$ when the accuracy of the model is 1 (i.e., it is a good model).
- Intermediate values are the excess of accuracy of the model over the random model, relative to the inaccuracy of the random model $(1 - A_e)$.

In the example,

$$\kappa = \frac{0.813 - 0.332}{1 - 0.332} = 0.72.$$

Metrics for binary classification

With a binary classification task $(0/1)$:

	Predict 1	Predict 0	Total
True 1	TP	FN	P
True 0	FP	TN	N
Total	PP	PN	M

where

- TP: # instances 1 that are predicted as 1: "true positives"
- FP: # instances 0 that are predicted as 1: "false positives"
- TN: # instances 0 that are predicted as 0: "true negatives"
- FN: # instances 1 that are predicted as 0: "false negatives"
- P: # instances 1: "positives".
- N: # instances 0: "negatives".
- PP: # instances predicted 1: "predicted positive".
- PN: # instances predicted 0: "predicted negative".
- The total number of instances is $M = TP + FP + TN + FN = P + N = PP + PN$.

Metrics for binary classification

A logistic regression was fitted to a data set[2]. The positive class is "Good". There are 700 "Good" and 300 "Bad" in the data set.

	Pred Good	Pred Bad	Total
Obs Good	686	14	700
Obs Bad	273	27	300
Total	959	41	1000

Then,

$$TP = 686, \ FP = 273, \ TN = 27, \ FN = 14$$

and

$$P = 700, \ N = 300, \ PP = 959, \ PN = 41.$$

---

[2]The well known German credit data

Sensitivity

The **sensitivity** is the proportion of correct predicted positive among the true positives:

$$\text{Sens} = \frac{TP}{TP + FN} = \frac{TP}{P}$$

It is the probability that the model recovers a positive instance:

$$Pr\left(\text{model predicts positive} \mid \text{positive}\right).$$

It is also referred to as **recall** or **true positive rate**. E.g., with the credit data

$$Sens = \frac{686}{686 + 14} = 0.98$$

The model is thus very good at recovering the positive instances.

Specificity

The **specificity** is the proportion of predicted negatives among the negatives:

$$\text{Spec} = \frac{TN}{TN + FP} = \frac{TN}{N}$$

Also called **True Negative Rate**, it is the probability that the model recovers a negative instance:

$$Pr\left(\text{model predicts negative} \mid \text{negative}\right).$$

E.g., with the credit data

$$TNR = \frac{27}{27 + 273} = 0.09$$

The model is poor at recovering the negative instances.

Positive Predictive Value

The **precision** also called **Positive Predictive Value**. It is the proportion of true positives among the predicted positive:

$$PPV = \frac{TP}{TP + FP} = \frac{TP}{PP}$$

The precision is the probability that the prediction is correct given that the model predicted positive:

$$Pr\left(\text{positive} \mid \text{model predicts positive}\right).$$

E.g.,

$$PPV = \frac{686}{273 + 686} = 0.715$$

Given that the model predicts a "positive", the chance it is indeed a positive is 0.715.

Negative Predictive Value

It is the equivalent of the precision but for the negative class.

$$NPV = \frac{TN}{FN + TN} = \frac{TN}{PN}$$

It is the probability that the prediction is correct given that the model predicted it as negative:

$$Pr\left(\text{negative} \mid \text{model predicts negative}\right).$$

E.g.,

$$NPV = \frac{27}{14 + 27} = 0.66$$

Given that the model predicts a "negative", the chance it is indeed a negative is 0.66.

Summary

Predicted

Truth	1	0	
1	TP	FN	Sensitivity, Recall
0	FP	TN	Specificity

Precision, PPV   NPV

Predicted

Truth	1	0	
1	686	14	Sensitivity=686/(686+14)=0.98
0	273	27	Specificity=27/(273+27)=0.093

PPV=686/(273+686)=0.715   NPV=27/(14+27)=0.66

Balanced accuracy

One issue with the accuracy is that it is a global indicator that strongly depends on the number of positive and negative cases in the data set.

For example, with 700 positive instances out of 1000, a model always predicting "Good" would have an accuracy of 0.70: a large accuracy for such a useless model.

This is because of unbalance in the outcomes in the data[3].

The **balanced accuracy** is the accuracy computed *as if* the data were balanced. It ends up being the average between sensitivity and specificity:

$$BA = \frac{Sens + Spec}{2}$$

E.g., $BA = (0.98 + 0.09)/2 = 0.535$, a low value: the model cannot recover negative cases.

---

[3]Note that the Cohen's kappa takes this into account.

The F_1 metric

This metrics is a (harmonic) mean of the precision and the sensitivity:

$$F_1 = 2 \times \frac{Prec \times Sens}{Prec + Sens}$$

E.g., for the credit data $Prec = 0.715$, $Sens = 0.98$, thus

$$F_1 = 2 \times \frac{0.715 \times 0.98}{0.715 + 0.98} = 0.83.$$

The limits are

- $F_1 = 0$ when either Prec or Sens is 0.
- $F_1 = 1$ when both Prec and Sens are 1.

The larger F_1, the better the model.

ROC curve and AUC

The **Receiver Operating Characteristic** (ROC) curve shows the sensitivities and the specificities of a model when *the threshold for the prediction varies*.

E.g., the sensitivity is 0.98, and the specificity is 0.09 with a threshold of $\lambda = 0.5$:

$$f(x_i; \theta) = \left\{ \begin{array}{ll} 1 & \text{if } p(x_i; \theta) \geq \lambda, \\ 0 & \text{if } p(x_i; \theta) < \lambda. \end{array} \right.$$

What happens when the threshold λ ranges from 0 to 1?

- If $\lambda = 0$, no cases are predicted as 0, thus *Sens* $= 1$ and *Spec* $= 0$.
- If $\lambda = 1$, no cases are predicted as 1, thus *Sens* $= 0$ and *Spec* $= 1$.

ROC curve and AUC

The ROC curve is obtained when λ varies between 0 and 1, $Spec = x$, and $y = Sens$. The specifity (x-axis) is usually in the reverse order.
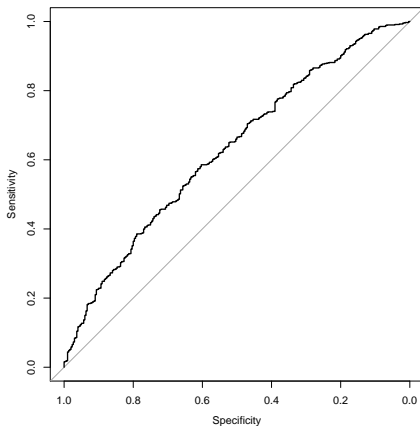
ROC curve and AUC

A perfect model has large sensitivity **and** specificity **for all** λ (except 0 and 1). This corresponds to a single point in the upper left corner of the ROC.

The diagonal in gray corresponds to the random model. The further the curve up that diagonal, the better the model.

The **Area Under the Curve** (AUC) is the area between the diagonal and the ROC curve. The larger it is, the better the model is.

E.g., $AUC = 0.628$.

The ROC curve provides a way to **tune** the threshold for the prediction under unbalanced class: select the threshold with maximum balanced accuracy (Youden Index point). See later.

Table of Contents

Root Mean Squared Error (RMSE)

The **RMSE** is the square root of the mean of the squares of the errors of prediction:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \{y_i - f(x_i; \theta)\}^2},$$

The lower the RMSE, the better the model.

MSE and MAE

Other metrics:

- The **Mean Squared Error** (MSE) is the RMSE without the square root

$$MSE = RMSE^2$$

  It is obviously equivalent to the RMSE.

- The **Mean Absolute Error** (MAE) is the average of the absolute errors:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - f(x_i; \theta)|$$

RMSE and MAE

Like the median to the mean, the MAE is more robust than the RMSE:

- One large error only will make the RMSE large.
- The MAE is less influenced by one error.

In practice, with:

- Model 1: mostly small errors except a few large ones,
- Model 2: only average errors, no large one.

Then, we will probably have

$$
\begin{aligned}
MAE(M1) &< MAE(M2) \\
RMSE(M2) &< RMSE(M1)
\end{aligned}
$$

The R^2

In the ML context, the R^2 (R **squared**) is the square of the correlation between the observations and the predictions.

$$R^2 = cor^2(y, f(x; \theta))$$

The R^2 is bounded between 0 and 1. The larger the R^2, the better the model.

The R^2 is an attempt to approximate / replace the usual R^2 of a linear regression. However, these two must not be confounded because they will often be different (though conveying the same idea).

Table of Contents

Final words

- There exist plethora of metrics with advantages and drawbacks.
- Some can be interpreted (accuracy, sensitivity, etc.), others cannot and are used only to compare models (F_1, *RMSE*, etc.).
- It is important to have a **baseline model** to compare the metric (e.g., random model for classification, average for regression).
- For regression, a **scatterplot of the observations** y_i **versus the predictions** $f(x_i; \theta)$ allows to analyze in more details the model quality. It is a must do!