

Machine Learning

Module 4: Metrics

Marc-Olivier Boldi

Master in Management, Business Analytics, HEC UNIL

Spring 2024

Table of Contents

- 1 Concept
- 2 Metrics for classification
- 3 Metrics for regression
- 4 Final words

Table of Contents

- 1 Concept
- 2 Metrics for classification
- 3 Metrics for regression
- 4 Final words

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 two types of metrics: for regression and for classification.

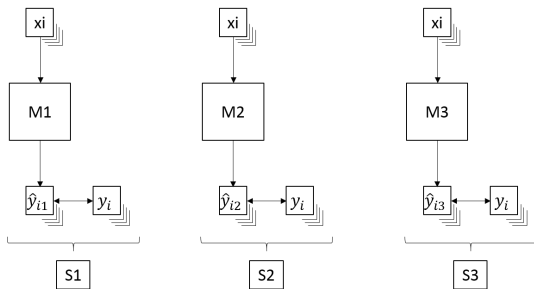


Table of Contents

- 1 Concept
- 2 Metrics for classification
- 3 Metrics for regression
- 4 Final words

Metrics for classifications

Metrics for classifications can be grouped as

- Metrics based on the predicted classes only
- Metrics based on the predicted probabilities
- Metrics for binary classifications only
- General metrics for multi-classes

Confusion matrix

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

Prediction	Reference		
	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 kappa¹ 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 random². The expected accuracy is also called **No Information Rate**.

¹Often used in psychology

²The calculations are the same as for the chi-square test for independence. ▶

Cohen's kappa

With the iris data, we have the following observations from the model:

Observed confusion matrix (frequencies)						Observed confusion matrix (proportions)					
		Observations						Observations			
		Setosa	Versicolor	Virginica	Total			Setosa	Versicolor	Virginica	Total
Predictions	Setosa	50	0	0	50	Predictions	Setosa	0.333	0.000	0.000	0.333
	Versicolor	0	48	2	50		Versicolor	0.000	0.320	0.013	0.333
	Virginica	0	2	48	50		Virginica	0.000	0.013	0.320	0.333
	Total	50	50	50	150		Total	0.333	0.333	0.333	1.000

The **observed accuracy** is $A_o = 0.973$.

Cohen's kappa

A random model would have balanced proportions in each cell:

Expected confusion matrix (proportions)						Expected confusion matrix (frequencies)					
		Observations						Observations			
		Setosa	Versicolor	Virginica	Total			Setosa	Versicolor	Virginica	Total
Predictions	Setosa	0.111	0.111	0.111	0.333	Predictions	Setosa	16.667	16.667	16.667	50.000
	Versicolor	0.111	0.111	0.111	0.333		Versicolor	16.667	16.667	16.667	50.000
	Virginica	0.111	0.111	0.111	0.333		Virginica	16.667	16.667	16.667	50.000
	Total	0.333	0.333	0.333	1.000		Total	50.000	50.000	50.000	150.000

where

$$0.111 = 0.333 \times 0.333.$$

The expected accuracy is $A_e = 0.111 + 0.111 + 0.111 = 0.333$.

The expected frequencies are not of direct use. These are the expected proportions multiplied by the total amount of instances:

$$16.67 = 0.111 \times 150.$$

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., not a good model).
- $\kappa = 1$ if $A_o = 1$ when the accuracy of the model is 1 (i.e., 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,

$$\kappa = \frac{0.973 - 0.333}{1 - 0.333} = 0.96.$$

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 set³. The positive class is "Good". There are 700 "Good" and 300 "Bad" in the data set.

Obs	Pred	
	Good	Bad
Good	686	14
Bad	273	27

Then,

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

and

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

³The 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:

$$Pr(\text{model predicts positive} | \text{positive}).$$

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

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

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:

$$Pr(\text{model predicts negative} | \text{negative}).$$

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:

$$Pr(\text{positive} | \text{model predicts positive}).$$

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:

$$Pr(\text{negative} | \text{model predicts negative}).$$

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

Precision, PPV (under TP) NPV (under TN)

		Predicted		
		1	0	
Truth	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 data⁴.

The **balance 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.

⁴Note 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** 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 $\lambda = 0.5$:

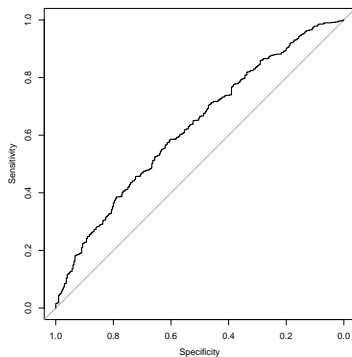
$$f(x_i; \theta) = \begin{cases} 1 & \text{if } p(x_i; \theta) \geq 0.5, \\ 0 & \text{if } p(x_i; \theta) < 0.5. \end{cases}$$

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 specificities are 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

- 1 Concept
- 2 Metrics for classification
- 3 Metrics for regression**
- 4 Final words

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.

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

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

- 1 Concept
- 2 Metrics for classification
- 3 Metrics for regression
- 4 Final words**

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!