metrica: an R package to evaluate prediction performance of regression and classification point-forecast models

Adrian A. Correndo¹¶, Luiz H. Moro Rosso², Carlos H. Hernandez¹, Leonardo M. Bastos³, Luciana Nieto¹, Dean Holzworth⁴, and Ignacio A. Ciampitti¹

1 Department of Agronomy, Kansas State University, Manhattan, KS, USA. 2 Private Consultant, Brasil. 3 Department of Crop and Soil Sciences, University of Georgia, Athens, GA, USA. 4 CSIRO Agriculture and Food, Australia. ¶ Corresponding author

DOI: 10.21105/joss.04655

Summary

The metrica R package (Correndo et al., 2022) is an open-source software designed to facilitate the quantitative and visual assessment of prediction performance of point-forecast simulation models for continuous (regression) and categorical variables (classification). The package ensembles a series of 80+ functions that account for multiple aspects of the agreement between predicted and observed values. Without the need of advanced skills on programming, metrica enables users to automate the estimation of multiple prediction performance metrics including goodness of fit, error metrics, error decomposition, model efficiency, indices of agreement, and to produce stylish data visualization outputs. This article introduces metrica, an R package developed with the main objective of contributing to transparent and reproducible evaluation of point-forecast models performance.

Statement of need

Evaluating the prediction quality is a crucial step for any simulation model, for which a myriad of metrics and visualization techniques have been developed (Tedeschi, 2006; Wallach et al., 2019; Yang et al., 2014). Nonetheless, to conduct a comprehensive assessment of the predicted-observed agreement in R (R Core Team, 2021), users normally have to rely on multiple packages, and even on self-defined functions, which increases the risk of involuntary mistakes due to the need of fluctuating syntax and data wrangling.

As the reproducibility of data analysis continues to be a challenge for science (Seibold, 2022), developing open source software like metrica offers a step toward a transparent and reproducible process to assist researchers in evaluating models performance. We decided to create metrica in R (R Core Team, 2021) due to its substantial role in data science (Thieme,
Under its open-source philosophy, R empowers the democratization of statistical computing (Hackenberger, 2020) by hosting and globally distributing cutting-edge algorithms through the Comprehensive R Archive Network (CRAN).

Finally, it is noteworthy that in the area of agricultural sciences, although point-forecast simulation models such as the Agricultural Production Systems simulator (APSIM) (Holzworth et al., 2014, 2018) count with tools to facilitate the integration into R through packages such as apsimx (Miguez, 2022), the assessment of its prediction quality is not yet integrated for R users. Therefore, we aim for metrica to offer users of simulation models for agriculture, plant, and soil sciences a toolbox for assessing the performance of regression and classification point-forecast models.

Package features

For regression models, metrica includes four plotting functions (scatter, tiles, density, & Bland-Altman plots) using ggplot2 (Wickham, 2016), and 48 prediction performance metrics. For classification models (two-class or multi-class), it includes one function to visualize a confusion matrix, and 26 functions of prediction scores. The full list of metrics with description, formula, and literature sources is presented in the package documentation at:

- Regression metrics vignette.
- Classification metrics vignette.

To the best of our knowledge, metrica covers several functions not supported, or partially supported by similar R packages (or components) designed for model evaluation such as yardstick (Kuhn & Vaughan, 2022) from tidymodels (Kuhn & Wickham, 2020), the measuring performance components from caret (Kuhn, 2022) or mlr3 (Lang et al., 2019), Metrics (Hamner & Frasco, 2018), hydroGOF (Zambrano-Bigiarini, 2020), cvms (Olsen & Zachariae, 2021), scoringutils (Bosse et al., 2020), or performance (Lüdecke et al., 2021). Unique features include:

- one of the most extensive collections of prediction performance metrics for regression and classification models in R.
- working under both vectorized (calling variables with $) or tabulated forms (Wickham et al., 2019).
- controlling the output format as a list (tidy = FALSE) or as a table (tidy = TRUE).
- for classification, functions automatically recognizing two-class or multi-class data; and specifically for multi-class cases, several metrics can be estimated for each class (atom = TRUE)(Ferri et al., 2009), (Ben-David, 2007), including balanced and imbalanced scenarios (Kubat et al., 1997).
- for regression, implementing a symmetric linear regression (standardized major axis-SMA-, (Warton et al., 2006)) to describe: i) pattern of the bivariate relationship with linear parameters (B0_sma, B1_sma), and ii) degree of predicted-observed agreement by using SMA-line to decompose the mean-squared-error (MSE) into lack of accuracy (MLA, PLA, RMLA) and lack of precision (MLP, PLP, RMLP) components (Correndo et al., 2021).
- offering MSE decomposition approaches described by (Kobayashi & Salam, 2000) (SB, SDSL, LCS), and (Smith & Rose, 1995) (Ub, Uc, Ue).
- including multiple indices of agreement and model efficiency such as: i) index of agreement d (Willmott, 1981), and its modified d1 (Willmott et al., 1985) and refined d1r (Willmott et al., 2012) variants, ii) Nash–Sutcliffe model efficiency (NSE) (Nash & Sutcliffe, 1970) and its improved variants E1 (Legates & McCabe Jr., 1999), Erel (Krause et al., 2005), and Kling-Gupta model efficiency (KGE) (Kling et al., 2012), iii) Robinson’s index of agreement (d).
agreement (RAC) (Robinson, 1957, 1959), iv) Ji & Gallo agreement coefficient (AC) (Ji & Gallo, 2006), v) Duveiller’s lambda (Duveiller et al., 2016), vi) distance correlation (dcorr) (Székely et al., 2007), or vii) maximal information coefficient (MIC) (Reshef et al., 2011), among others.

- importing files from APSIM Classic with `import_apsim_out()`), and from APSIM Next Generation with the `import_apsim_db()` function.

Using the functions

There are two core arguments to all `metrica` functions: (i) obs (Oi; observed, a.k.a. actual, measured, truth, target, label), and (ii) pred (Pi; predicted, a.k.a. simulated, fitted, modeled, estimate) values.

For regression, specific functions such as `scatter_plot()` require defining the axis orientation (e.g. predicted vs. observed -PO- or observed vs. predicted -OP-). For two-class models, the pos_level argument serves to indicate the alphanumeric order of the “positive level”. For multi-class classification, some functions present the atom argument (TRUE / FALSE), which controls the output to be an overall average estimate across all classes (default), or class-wise.

Example 1: Regression (continuous variables)

Figure 1 is an output example of regression performance analysis using the `scatter_plot()` function for the native dataset called `wheat`.

![Scatter Plot](image)

Example 2: Classification (categorical variables)

Figure 2 is an output example of classification performance analysis using the confusion_matrix() function for the native dataset called maize_phenology.

![Confusion Matrix](image)

**Multiclass Confusion Matrix**

<table>
<thead>
<tr>
<th>Predicted</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>V10</th>
<th>V12</th>
<th>V13</th>
<th>V14</th>
<th>V15</th>
<th>V16</th>
<th>V18</th>
<th>V19</th>
<th>V9</th>
<th>VT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observed</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Performance metrics:
- accuracy = 0.88
- precision = 0.83
- recall = 0.84
- specificity = 0.99
- f1score = 0.84
- gmean = 0.91
- khat = 0.85

Figure 2: Confusion matrix plot using metrica::confusion_matrix().

Documentation & License

The complete documentation and vignettes of metrica are available online at https://adriancorrendo.github.io/metrica/. The package is under the MIT License (https://opensource.org/licenses/MIT). Source code is available at GitHub (https://github.com/adriancorrendo/metrica) along with its corresponding section to report issues and suggestions (https://github.com/adriancorrendo/metrica/issues).

Acknowledgements

Authors gratefully acknowledge the financial support from the Feed the Future Innovation Lab for Collaborative Research on Sustainable Intensification (SIIL) at Kansas State University through funding United States Agency for International Development (USAID) under the Cooperative Agreement (Grant number AID-OAA-L-14-00006).

References


