The Journal of Open Source Software

SeqMetrics: a unified library for performance metrics calculation in Python

Fazila Rubab[1], **Sara Iftikhar**[1], and **Ather Abbas**[2]

1 researcher at Environmental AI Rsearch Group, Islamabad, Pakistan **2** postdoctoral fellow, KAUST

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Marcel Stimberg

Reviewers:

- @FATelarico
- @y1my1
- @SkafteNicki

Submitted: 16 February 2024
Published: unpublished

Summary

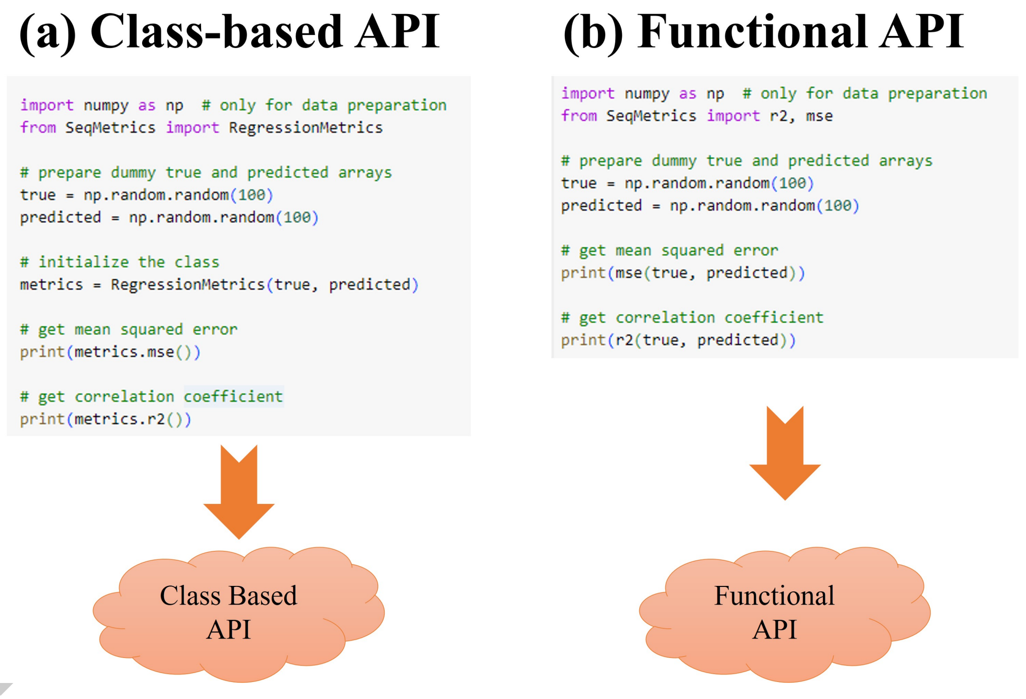
Current python infrastructure lacks a robust, unified and simplified library for error and performance metrics calculations. The SeqMetrics application responds to this critical need, providing a robust toolkit for evaluating regression and classification models with a comprehensive suite of performance metrics suitable for tabular and time series data. Designed for versatility, the library offers 112 regression and 22 classification metrics through both functional and class-based APIs. The design of the library ensures seamless integration into various coding environments. The web based GUI of SeqMetrics enhances user accessibility, allowing efficient input through data arrays or file imports. It serves to users of varied programming expertise, offering a user-friendly interface for rigorous model assessment. The library prioritizes computational efficiency and has minimal dependency with straightforward pip installation from PyPI. Rigorous testing of Seqmetrics ensures robustness, supported by extensive documentation for effective utilization by people from diverse backgrounds. Overall, SeqMetrics bridges the gap in Python's scientific analysis toolkit, contributing to data analysis.

Statement of need

Performance metrics and errors measure the distance and similarity between two arrays (Botchkarev, 2018). Quantification and analysis of model performance through error and performance metrics is a crucial step in any scientific modeling exercise (Gleckler et al., 2008). However, Python lacks a unified library encompassing performance metrics across diverse fields for one-dimensional numerical data. Existing Python libraries for performance evaluation offer only a limited number of metrics. For instance, the metrics sub-module from Keras (Chollet & others, 2015) contains only 24 metrics, while scikit-learn's (Pedregosa et al., 2011) metrics module covers 45 metrics. Torchmetrics library, (Detlefsen et al., 2022) although contains 100+ metrics, however, it provides only 48 which are intended for 1-dimensional numerical data. Addressing this gap is imperative to provide researchers a unified platform for comprehensive model evaluation, streamlining their computational workflows and enhancing accuracy across various domains. The SeqMetrics application addresses the critical need for a robust and versatile toolkit for assessing and comparing regression and classification models performance across a spectrum of domains. With a comprehensive suite of performance metrics for sequential (tabular and time series) data, spanning traditional statistical measures to specialized atmospheric sciences metrics, the software serves as a valuable resource for researchers, analysts, and practitioners in fields such as hydrology, finance, and engineering. By providing a standardized platform for evaluating model performance through a diverse set of metrics, the application facilitates the rigorous validation and optimization of regression and classification models, contributing to informed decision-making processes and ensuring the reliability of predictive modeling in complex and dynamic systems.

API design

The SeqMetrics library offers a comprehensive suite of 112 regression metrics and 22 classification metrics to facilitate a detailed assessment of model performance. The regression metrics are for the data with continuous values such as R-squared, Mean Squared Error, and Root Mean Squared Error etc. On the other hand, the classification metrics are for categorical data which include accuracy, precision, recall, F1 score, and the area under the ROC curve among others. From programming perspective, the SeqMetrics library employs a modular architecture, offering a functional and class-based APIs for smooth integration across diverse coding environments. The class-based API, consists of 'RegressionMetrics' and 'ClassificationMetrics' classes (Figure 1a), which provide users with a structured approach for model evaluation. The user has to first initialize these classes by providing the data and then all the performance metrics are available from the respect instances of the classes. Conversely, the functional API, offers a more simplified approach to access these metrics without initializing the classes initially (Fig. 1b). All the functions which calculate performance metrics receive two arrays as obligatory input arguments and return a scaler value as output. With a unified API design and minimal dependency only on NumPy, the library prioritizes efficiency in computational tasks. It ensures straightforward installation via pip from the Python Package Index (PyPI), a widely adopted standard, which not only streamlines the process but also contributes to the overall efficiency of the application for the broader scientific community.



```python
import numpy as np  # only for data preparation
from SeqMetrics import RegressionMetrics

# prepare dummy true and predicted arrays
true = np.random.random(100)
predicted = np.random.random(100)

# initialize the class
metrics = RegressionMetrics(true, predicted)

# get mean squared error
print(metrics.mse())

# get correlation coefficient
print(metrics.r2())
```

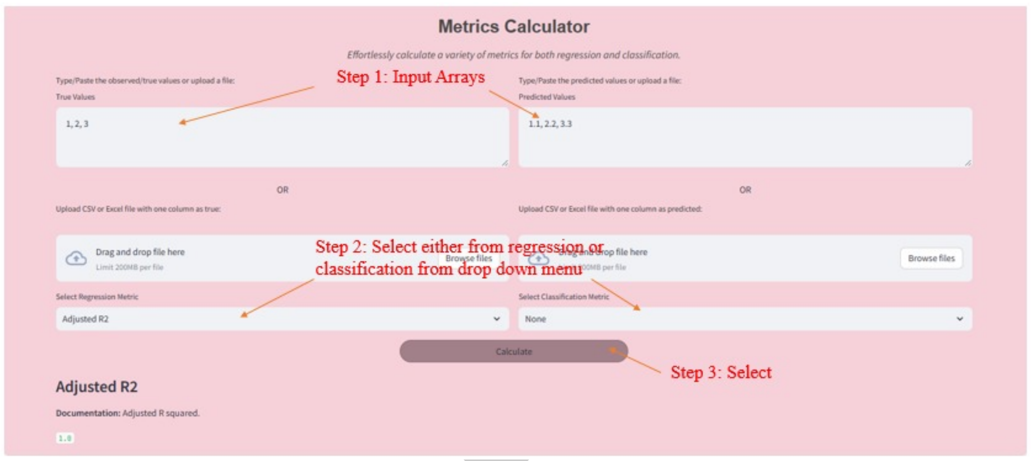
```python
import numpy as np  # only for data preparation
from SeqMetrics import r2, mse

# prepare dummy true and predicted arrays
true = np.random.random(100)
predicted = np.random.random(100)

# get mean squared error
print(mse(true, predicted))

# get correlation coefficient
print(r2(true, predicted))
```

Figure 1: Comparison of class-based and functional API

Graphical User Interface

The SeqMetrics graphical user interface (GUI) offers a user-friendly and intuitive platform for seamless error calculation. This interface is built and deployed using streamlit at https://seqmetrics.streamlit.app . There are 2 ways of providing input in this web-based GUI. The first method includes providing it the input data arrays by copying and pasting the true and predicted arrays (Fig. 2). Another way is by importing CSV or Excel /files into the interface (Fig 3). This streamlines the process of entering true and predicted values for evaluation. The

GUI provides a clean and organized layout, guiding users through the evaluation workflow with clear instructions. With its simplicity and ease of use, the GUI empowers users to perform regression and classification model assessments effortlessly. Whether you're a seasoned data scientist or a beginner, the SeqMetrics GUI enhances accessibility and efficiency in evaluating model performance without compromising on robustness and precision. Therefore, the design of the SeqMetrics is equally beneficial for advanced programmers as well as for those with limited programming knowledge.

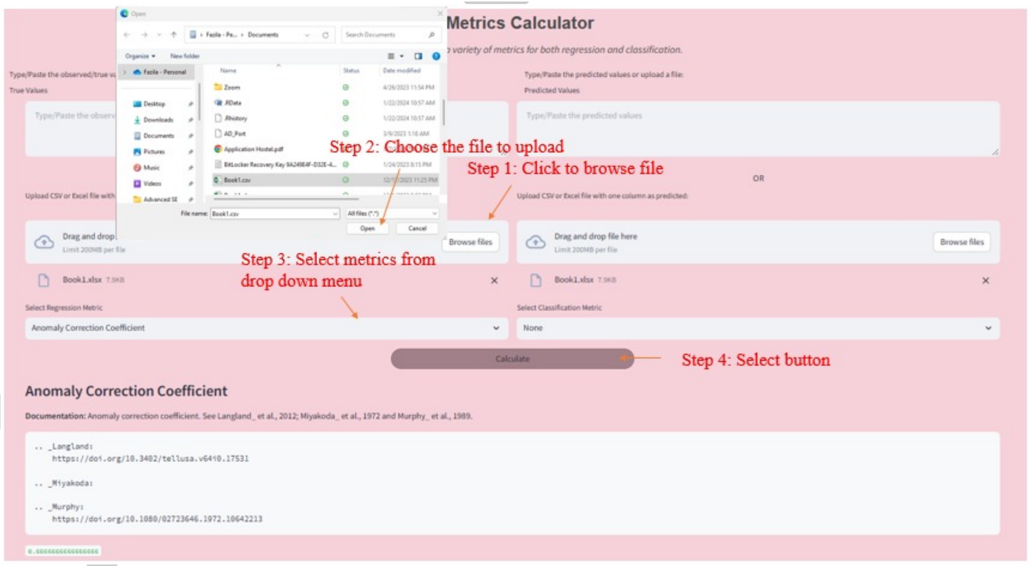


Metrics Calculator
Effortlessly calculate a variety of metrics for both regression and classification.
Step 1: Input Arrays
Type/Paste the observed/true values or upload a file:
True Values
1,2,3
Type/Paste the predicted values or upload a file:
Predicted Values
1.1, 2.2, 3.3
OR
Upload CSV or Excel file with one column as true:
Upload CSV or Excel file with one column as predicted:
Drag and drop file here
Limit 200MB per file
Browse files
Step 2: Select either from regression or classification from drop down menu
Select Regression Metric
Adjusted R2
Select Classification Metric
None
Calculate
Step 3: Select

Adjusted R2

Documentation: Adjusted R squared.

1.0



Open
Metrics Calculator
Effortlessly calculate a variety of metrics for both regression and classification.
Step 2: Choose the file to upload
Step 1: Click to browse file
Type/Paste the observed/true values or upload a file:
True Values
Type/Paste the predicted values or upload a file:
Predicted Values
OR
Upload CSV or Excel file with one column as predicted:
Drag and drop file here
Limit 200MB per file
Browse files
Book1.xlsx 7.9KB
Step 3: Select metrics from drop down menu
Select Regression Metric
Anomaly Correction Coefficient
Select Classification Metric
None
Calculate
Step 4: Select button

Anomaly Correction Coefficient

Documentation: Anomaly correction coefficient. See Langland_ et al., 2012; Miyakoda_ et al., 1972 and Murphy_ et al., 1989.

```
.. _Langland:
    https://doi.org/10.3402/tellusa.v64i0.17531

.. _Miyakoda:

.. _Murphy:
    https://doi.org/10.1080/02723646.1972.10642213
```

Testing and documentation

Following the 'unit test' protocol the library undergoes comprehensive testing of all regression and classification metrics. The library is tested for multiple scenarios especially for classification case which includes numerical and logits inputs, ensuring robustness in various classification contexts such as binary, multiclass, and multilabel. Such comprehensive testing ensures the robustness and accuracy of each metric, providing users with reliable results. Additionally, the library features extensive documentation, with detailed docstrings for each function and class, accompanied by fully executable examples. This thorough documentation enhances user understanding and facilitates efficient utilization of the library's capabilities in both regression and classification scenarios.

References

Botchkarev, A. (2018). Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology. *ArXiv*. https://doi.org/10.48550/arXiv.1809.03006

Chollet, F., & others. (2015). *Keras*. https://github.com/fchollet/keras; GitHub.

Detlefsen, N. S., Borovec, J., Schock, J., Jha, A. H., Koker, T., Di Liello, L., Stancl, D., Quan, C., Grechkin, M., & Falcon, W. (2022). Torchmetrics-measuring reproducibility in pytorch. *Journal of Open Source Software*, *7*(70), 4101. https://doi.org/10.21105/joss.041012

Gleckler, P. J., Taylor, K. E., & Doutriaux, C. (2008). Performance metrics for climate models. *Journal of Geophysical Research: Atmospheres*, *113*(D6). https://doi.org/10.1029/2007JD008972

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., & others. (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, *12*, 2825–2830.