# Scientific Programming with Python
# Assignment: The Performance of Numpy Versus Regular Python Lists when Computing a Loss Function

**Karl N. Kirschner**

Department of Computer Science, University of Applied Sciences Bonn-Rhein-Sieg,
Sankt Augustin, Germany

November 27, 2023

---

**Goal** The goal of this assignment is to understand how Numpy usage [1] affects the performance of numerical calculations. You will also strengthen your usage of Pandas[2, 3]. You will also learn about the idea and usage of a loss function. All conceptual aspects learned in the course so far should be considered in developing the solution, unless otherwise stated (e.g. allowed libraries, significant figures).

**Problem and Input Data** Weather researchers created a machine learning model that predicts the rainfall and evaporation on different days at different location in Australia [4, 5]. The experimental and model observables collected are shown in Table 1. The observables are described by the following

- Date - The observation date.
- Location - The weather station location.
- MinTemp - The minimum temperature ($^o$C).
- MaxTemp - The maximum temperature ($^o$C).
- Rainfall - The rainfall amount in 24 hours (mm).
- Evaporation - The evaporation amount in 24 hours (mm).
- Sunshine - The sunshine amount in 24 hours (h).
- WindGustSpeed - The maximum wind gust speed in 24 hours (h).
- RainToday - Did it rain on that day? yes: if precipitation >= 1 mm, no:if precipitation < 1 mm.
- RainTomorrow - Did it rain in the following day? yes: if precipitation >= 1 mm, no: if precipitation < 1 mm.

**Table 1.** Australian experimental and model prediction data for weather features.

| | | Experimental | | | | | | | | Model Prediction | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Date** | **Location** | **Min. Temp ($^o$C)** | **Max. Temp ($^o$C)** | **Rainfall (mm)** | **Evapor. (mm)** | **Sunshine (h)** | **Wind Speed (km/h)** | **Rain Today** | **Rain Tomorrow** | **Rainfall (mm)** | **Evapor. (mm)** |
| 2009-01-02 | Cobar | 18.4 | 28.9 | 0.0 | 14.8 | 13.0 | 37.0 | No | No | 1.164527 | 7.564111 |
| 2009-01-04 | Cobar | 19.4 | 37.6 | 0.0 | 10.8 | 10.6 | 46.0 | No | No | 1.077602 | 2.872613 |
| 2009-01-05 | Cobar | 21.9 | 38.4 | 0.0 | 11.4 | 12.2 | 31.0 | No | No | 2.082352 | 8.060459 |
| 2009-01-06 | Cobar | 24.2 | 41.0 | 0.0 | 11.2 | 8.4 | 35.0 | No | No | 7.453461 | 7.468973 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

The performance of their model to predict the rainfall and evaporation for each date and location was done using the following loss function:

$$\text{Loss} = \alpha * \left|R^{\text{Pred.}} - R^{\text{Exp.}}\right| + \beta * \left|E^{\text{Pred.}} - E^{\text{Exp.}}\right| \tag{1}$$

where $\alpha$ is the rainfall weighting factor, $\beta$ is the evaporation weighting factor, $R^{\text{Pred.}}$ and $R^{\text{Exp.}}$ are the predicted and experimental rainfall values, while $E^{\text{Pred.}}$ and $E^{\text{Exp.}}$ are the corresponding evaporation values.

A loss function is simply a mathematical scheme to evaluate how well a model predicts observables with regard to some target data [6]. Loss functions have gained more usage due to the raise in machine learning's popularity and the need to optimize parameters to improve modeling results [7].

Note: The weighting factors allow one to simply adjust the relative importance of each observable with respect to the overall computed loss value, and they should sum together to a value of 1.0.

---

**Assignment Tasks** In the following tasks you will create functions with the goal of computing the loss function for each corresponding date and location using a) Numpy and b) regular Python lists. You will then examine the performance difference when using Numpy versus regular lists.

**Task 1** Read in the data contained in weather_experiment.csv and weather_prediction.csv files.

**Task 2** Create user-defined functions that encodes and computes the loss function (Equation 1), which:

1. performs the calculation using regular Python lists (i.e. **do not use Numpy or ndarrays**), and
2. performs the calculation using Numpy (i.e. **maximizing the use of Numpy's library and performance**).

(Hint: the functions should return the proper "container" that holds several thousand loss function values as computed for the input data values.)

**Task 3** Evaluate the speed performance between your Task 2 functions by computing the loss value for when $\alpha = \beta = 0.5$. You will use the timeit library (i.e. timeit.timeit) for this, and assign its "number" parameter to 100.

**Allowed Python3 [8, 9] functions & libraries/modules**

- all built-in functions
- Numpy [1], Pandas [2, 3], and timeit libraries
- typing library (optional)

**Assignment Due** Turn in your solution as a Jupyter-notebook to **LEA** by **December 4th, 2023 at 09:00**.

**Note 1** : Please include your **full name** at the **top** of your notebook.

**Note 2** : You **do not** need to consider **significant figures**.

# References


[1] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature, 585 (2020) 357–362 (DOI: 10.1038/s41586-020-2649-2)

[2] The Pandas Development Team pandas-dev/pandas: Pandas Zenodo, 2020 (https://pandas.pydata.org)

[3] Pandas user guide, https://pandas.pydata.org/docs/user_guide/index.html#user-guide, visited on April 30, 2022.

[4] Oswal, N. Predicting Rainfall using Machine Learning Techniques. arXiv, 2019 (https://arxiv.org/abs/1910.13827).

[5] Joe Young and Adamyoung. Rain in Australia, Kaggle https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package?resource=download&select=weatherAUS.csv. Online; accessed on November 27, 2022.

[6] Wikipedia contributors, Loss Function, https://en.wikipedia.org/wiki/Loss_function. Online; accessed on November 27, 2022.

[7] Datarobot, Introduction to Loss Functions, updated on March 26, 2021 https://www.datarobot.com/blog/introduction-to-loss-functions/. Online; accessed on November 27, 2022.

[8] Python Software Foundation. Python Language Reference, version 3.8. Available at http://www.python.org.

[9] van Rossum, G. Python tutorial, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, 1995.