NOVA IMS Information Management School

Neural and Evolutionary Learning

Class 1 - Rules and foundations

Prof.: Karina Brotto Rebuli

krebuli@novaims.unl.pt

Purpose of Practical Classes

"Theory without practice is empty; practice without theory is blind."

Kant

- **Programming skills**: you definitely need to code well to be a good data scientist;
- Implement or use implemented methods studied in theoretical classes: **coding is debugging ideas**;
- **Teamwork**: we are not competing, we are cooperating: help and ask for help as soon as possible.

Feedback and Assistance

krebuli@novaims.unl.pt

(consider that I may not be available at the exact time you will ask me for help, so do it as soon as possible.)

Two-way feedback:

Agenda

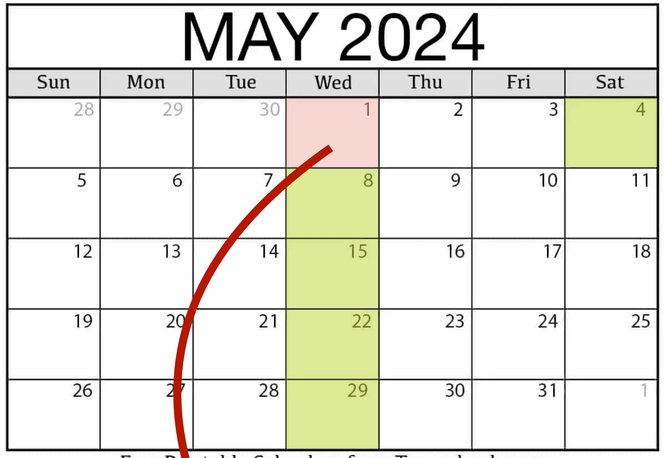
Practical classes:

- Wednesdays
  - Group 1: 10h to 11h30
  - Group 2: 15h to 16h30

APRIL 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

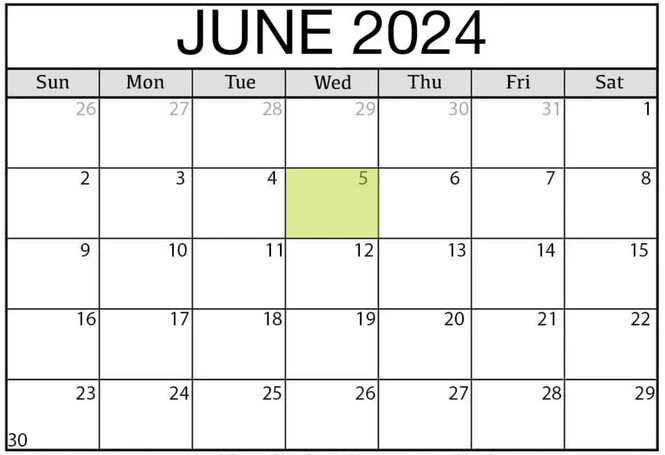
Free Printable Calendars from Typecalendar.com



MAY 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

Free Printable Calendars from Typecalendar.com



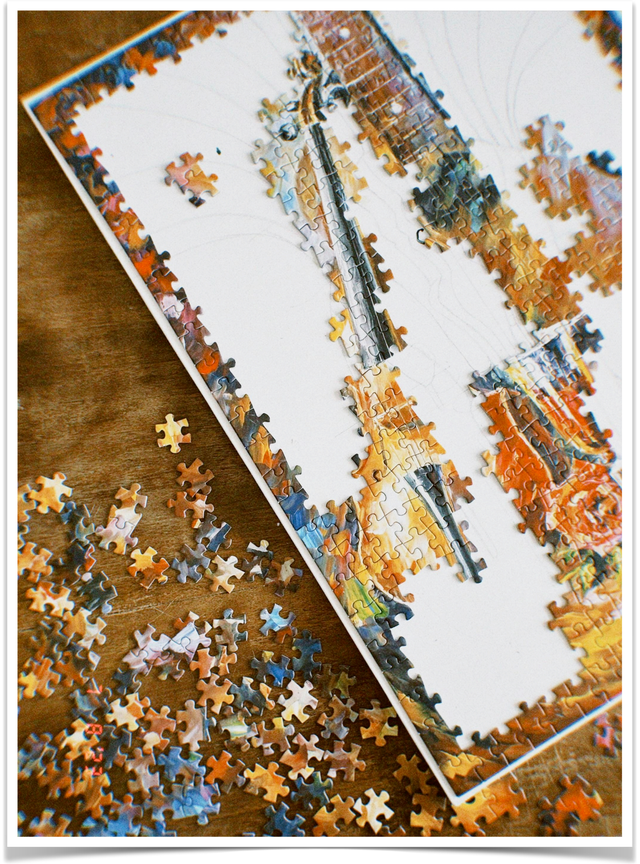
JUNE 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Free Printable Calendars from Typecalendar.com

Attention: There is a holiday on May 1st, so, according to the School guidelines, the class should be replaced on the following *Saturday, which will be on May 4th*.

Final Project



During the practical classes, we will be implementing and experimenting with the algorithms studied in theoretical classes. The project aims to **apply all this to a real-world problem, making sense of the pieces altogether**.

Thus, the better you **keep updated with practical exercises**, the easier the project will be for you and your group.

Overall, this will make you a better data scientist!

Final Project

- In the second practical class, you will receive the **dataset** with which you will work for the final project.
- The data is anonymised.
- **Part of the data will be held for the evaluation** of the generalization performance of the algorithms. Thus, **your code should be working plug-and-play in a Python script**.
- It will be done in **groups (3 or 4 people)**.
- A **report and the source code** should be delivered to my email (krebuli@novaims.unl.pt).
- The **report** should be no longer than **4 pages**, including plots and references.
- The **deadline** for the delivery is **at 23h59 on June 9th**.

Final Project

- Evaluation criteria

Category	Subcategory	Points	Total
Source code	Code organization	1	8
	Code efficiency	1	
	Classes basic implementation (to be informed in each class)	2	
	Classes advanced implementation (to be informed in each class)	4	
	Extra implementation	*	
Report	Writting (maximum 4 pages, including references)	1	4
	Plots	1	
	Statistical methods	1	
	Discussion of results	1	
	Extra page **	-2	
Oral presentation	Working group distribution	1	8
	Code implementation discussion	2	
	Algorithms implementation discussion	3	
	Individual alogorithms comparison discussion	2	
		Final grade	**20**

Obs: * It depends on the complexity of the implementation.

** Maximum of 2 extra pages. Reports longer than 6 pages will not be accepted.

Resources

Python should be used for coding the algorithms and experiments for the final project.

Algorithm	Framework
GP	GPOL[1]-based GP
GSGP	GPOL[1]-based GP
NN	Tensorflow[2]
NEAT	NEAT-Python[3]

For writing the final report, and for producing plots or other resources, you can use any tool you want.

1: Bakurov, I., Buzzelli, M., Castelli, M., Vanneschi, L., & Schettini, R. (2021). General purpose optimization library (Gpol): A flexible and efficient multi-purpose optimization library in python. Applied Sciences (Switzerland), 11(11), 1-34. [4774]. https://doi.org/10.3390/app11114774
2: Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
3: McIntyre, A., Kallada, M., Miguel, C. G., Feher de Silva, C., & Netto, M. L. neat-python [Computer software]

Python Enviroment

It is highly recommended to work in an environment dedicated to the course.

Install torch: https://pytorch.org/

PyTorch is a robust Python library that handles GPU acceleration and has extensive community support. The GP and GSGP codes I will provide you are based on GPOL, which, in turn, is based on PyTorch. Thus, let's start with a review of PyTorch!

Algorithms comparision

1. Dataset split
2. Metrics
3. Statistical tests
4. Plots

Algorithms comparision

1. Dataset split

   1.1. For evaluating the generalization ability of the algorithm:

   - It is always necessary to evaluate the performance of the algorithms on unseen data. Thus, the data should be split into the following partitions:

     •**Train**: used for the algorithm learning phase;

     •**Validation**: used for tuning hyperparameters w.r.t. its generalization ability;

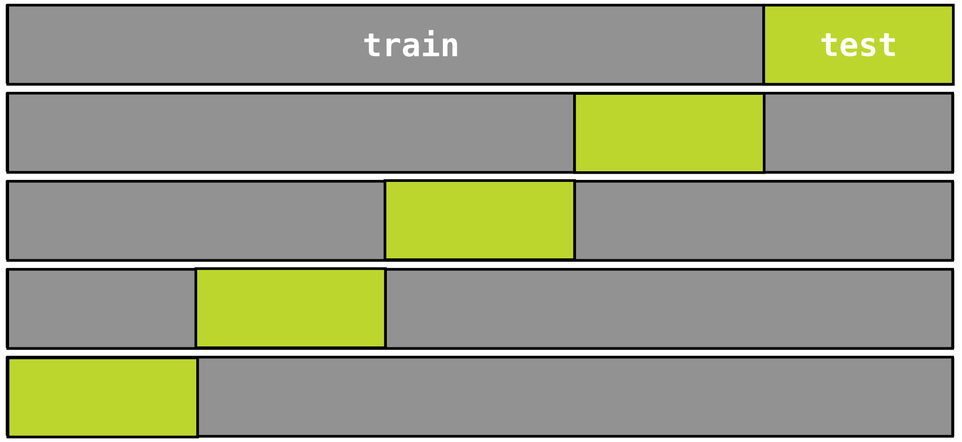
     •**Test**: **unseen** data, is used to assess the generalisation ability of the algorithm; therefore, it only should be used after the training phase, when model architecture and hyperparameters are defined.

Algorithms comparision

1. Dataset split

   1.2. For statistical evaluation of the performance of the algorithm:

   - K-fold Cross-validation ensures that all data is used both for train and test partitions, reducing the probability of biases in the results. So, whenever possible, **use it**.



5-fold cross-validation diagram.

   - It is possible to repeat the initial fold split to have more runs of the experiment.

Algorithms comparision

2. Metrics for regression problems

- **Error**: MAE (mean absolute error), MSE (mean squared error), RMSE (root mean squared error);
- **Good of fitness**: Pearson correlation coefficient, Spearman correlation coefficient, R^2.

Algorithms comparision

3. Statistical tests

- **Frequentist Tests**:

  The Null Hypothesis is that there is no *real* difference among models' performances, and the Alternative Hypothesis is that there is a *real* difference among models' performances;

  The *p*-value is the probability of having, by chance, a value of the statistic of the test that is equal to or more extreme than the observed value; thus, small p-values mean a low probability of rejecting H_0 when H_0 is true.

  The same training and test sets should be used for all the models to be compared;

  Non-parametric tests for regression tasks:

  - Wilcoxon Signed-rank Test for comparing between two models;

  - Friedman Test for comparing between two models;

Algorithms comparision

3. Statistical tests

Rainio, O., Teuho, J. & Klén, R. Evaluation metrics and statistical tests for machine learning. Sci Rep 14, 6086 (2024). https://doi.org/10.1038/s41598-024-56706-x

Algorithms comparision

4. Plots

- The cleaner, the better;
- Each characteristic of the plot should be used to give the reader a new information;
- Take care with the scale of the axis;
- As results refer to experiments, there is variability in the data. Include this in the plots.
- Example: https://cavalab.org/srbench/results/

NOVA IMS Information Management School

Questions?