Practical tips for the final exercise

Jakob Richter

3/16/2021

Your task will be to write a simplified AutoML System that only has to accept data from one data generating process (i.e. column names and types will be always the same). You will be given a skeleton code.

Hints for preparation:

- Read this document.
- Get familiar with mlr3 (or scikit-learn)
- Maybe have a look at this post: https://mlr3gallery.mlr-org.com/posts/2021-03-11-build-an-automated-machine-learning-system/
- You won't be forced to use mlr3 but it will make things easier for you.
- Already create a git project that everybody in your group has access to and create an empty report.

Your report should answer the following questions.

1. What is the problem you try to solve?
2. How does the data look like?
3. How did you come up with the steps in your AutoML System?
4. With which accuracy do you expect your AutoML System to perform on new data?
5. How does your method perform in comparison to a naive solution.

Guidelines

Recommended structure of your report

1. Introduction (0.5 pages)
2. Problem Description (0.5 pages)
3. Methods (2 pages)

- Each method should be referenced.
- Don't explain how the method works in detail.
- Name possible requirements for each method.
- Name when the method works good and possible problems.

4. Application (4 pages)

- Explain how you put all methods together.
- Show (intermediate) results.
- Bonus: Analyze which configurations your AutoML method chose.

5. Conclusion (1 page)

- Sum up your findings.
- Mention drawbacks of your solution.
- Mention possible improvements.

More hints

- The title page has to contain the names of the group members.
- You can communicate with other groups.
- You can ask us questions in moodle.
- You are allowed to use all packages that can be installed from CRAN, GitHub or imported via reticulate (if you really want that)
- You can write everything in python (please provide the virtual environment you worked in e.g.:`pip freeze > requirements.txt`) in this case just write a normal class instead of R6.
- Make sure that there is no form of data leakage in your process.
- Your solution should run on another system, so don't use absolute file paths.
- Your report can be written as an Rmd file, make sure the structure of the document is clean, but don't spend too much time on layout.
- The actual predictive performance of your approach matters less than an accurate write up and a correct validation.
- Comment your code
- Structure your working directory with the following subdirectories:
  - `./data` - the original datasets
  - `./automl_code` - the code that contains the automl methods
  - `./benchmark_code` - the code that uses the automl methods and the data to conduct the benchmark
  - `./benchmark_results` - store the benchmark results (e.g. as rds files) here
  - `./report` - store your (e.g. Rmd) report file here