Second peer review report (FES)

Polina Pilyugina, Alexey Voskoboinikov

May 13, 2021

1 Code review summary

Structure and reproducibility The repository is structured using Kedro framework. In our opinion, it is a huge advantage of the project. The structure of the project is inherited from the Kedro framework, which allows those familiar with typical project structure of Kedro to easily inspect the code and its parts. Overall, the code is well structured and easy to follow. The team implemented the synthetic data creation, data preprocessing and data science pipelines.

As for data creation, it is implemented as a class inherited from the abstract Kedro dataset class. It contains generation of sparse synthetic data function, and the generation of grouped dataset is work in progress. The dataset structure of Kedro is exceptionally useful, when a user has a perspective on how it works. Explanation of the dataset preparation in the code is thorough and easy to follow. It might be useful also to add more explanation and links to Kedro docs on how to add custom datasets and use custom data. In status report it is already stated, that one need to create a custom Kedro dataset class, but a small example on some real-data dataset will be useful. We look forward to see such an example with Boston Housing dataset, as stated per report.

The data preprocessing pipelines part consist of several nodes and corresponding pipelines for synthetic datasets creation with different parameters, stated in the configuration file. Currently it contains 4 configurations of synthetic datasets: the one with default parameters; with 3d degree polynomial features; with increased signal-to-noise ratio and with increased redundancy rate. The report contains full rationale behind the choice of these configurations in terms of performance evaluation of permutation method.

The data science pipelines part currently contains a permutation importance feature selection evaluation pipeline, baseline method. This pipeline contains of the model fitting and evaluation of performance nodes. The future implementations of additional feature selection algorithms is also

expected to be contained in this pipelines part and in methods pipelines section. There are already placeholders for the two future methods.

Overall the code is clean, nice structured and contains explanations of all arguments and outputs.

Possible improvements Currently, the parameter section contains parameters for permutation importance feature selection algorithm. Although we see the rationale behind creation of 4 separate pipelines for 4 different sets of configurations of synthetic datasets for evaluation process, these pipelines are similar to each other, except for the input/output names. Also there are 4 datasets in catalog, with only several parameters changed. It might be more convenient to move the synthetic dataset configuration from the catalog to parameter section. In this way, there will be only one pipeline for synthetic dataset evaluation, which can be run with changes in various parameters through the same Kedro run pipeline.

We do not insist on this implementation, while we provide an example of how it can be changed and used further in previously described manner. Proposed refactoring is available in the following pull request `https://github.com/adasegroup/FES-feature-selector/pull/1`. We changed the data processing pipeline, so that it is now responsible for synthetic dataset creation. Configuration of synthetic parameters is now moved to the parameters.yaml file. Therefore there is only one pipeline for synthetic dataset, instead of four, which can be altered using altering of parameters in terminal run code line. In our opinion, it can be more convenient, as it allows to vary parameters of synthetic datasets more easily. In our refactored code, there is only one dataset and one synthetic dataset evaluation pipeline, with configuration specified in the parameters section. Also, our refactored code allows to set parameters of evaluation in command line as well.

This implementation, though, has some limitation. In order to use command line configuration of parameters, the parameters required for change must not be in a nested constructions. Therefore, under current Kedro parsing parameters option, it adds a parameter from terminal code line to the whole parameters dictionary, without possibility to change nested parameters. We propose a workaround in defining the lists of parameters names for each function. It could be worked around in some other ways.

In our pull request we changed several code files, for consistency. And also updated README.MD file with the changed commands for running the specified in status report experiments. However we believe, that the current implementation of colleagues is also perfectly suitable. And we see that variability of synthetic dataset creation parameters might not be the necessary part.

2 Project readiness

We were perfectly able to tun the code, after installing the requirements and the project package from src. All tests work and produce the same results under the same seed, as described in paper.

Currently, the project contains the baseline method and the report contains its evaluation on synthetic dataset. Further steps will include the following implementations:

- Implementation of two new methods for feature selection
- Addition of preprocessing for Boston Housing dataset
- Evaluation on Boston Housing dataset

Also we suggest colleagues to add the graph from kedro-viz functional with the structure of pipelines, as it can be useful to understand various steps of the project.

Overall, this repository leaves exceptionally good impression. The code is structured nicely, the steps are explained and the report contains all the necessary info. And the fact that the team used Kedro is a great advantage, in our opinion