Submission and Review Checklist for Authors and Reviewers of the *ABC* Conference / Journal (Dagstuhl Seminar 24211 Template, v1.0)

Author

Date

This document serves as a template for conference chairs and journal editors. They may use this template to populate a submission checklist to be filled and submitted by authors along with their manuscript and/or reviewers along with their review or simply as guidance. This template is published under the CC-BY 4 license[1]. Visit https://code.recommender-systems.com/Dagstuhl-24211-Checklist for the most recent version of this template. Inspired by the AutoML 2024 checklist[2], our template allows for the LaTeX commands **[TODO]**, [Yes], [No], and [N/A].

If you use this template or parts thereof, please give appropriate credit to or cite

> *Joeran Beel, Dietmar Jannach, Alan Said, Guy Shani, Tobias Vente, Lukas Wegmeth. 2024.* Best-Practices for Offline Evaluations of Recommender Systems. *In Report from Dagstuhl Seminar 24211 – Evaluation Perspectives of Recommender Systems: Driving Research and Education. Editors: Christine Bauer, Alan Said, Eva Zangerle.*

Version History:

- Version 1.0 (Mai 2024)

Author and Reviewer Checklist: Reproducibility We organize the proposed questions on reproducibility (and their corresponding explanations) in the following groups.

1. Code-related Aspects: Is the code of the full experimental pipeline publicly available?
   Sharing all artifacts needed or used to obtain the numerical results reported in a paper is essential for reproducibility. Appropriate documentation

[1] https://creativecommons.org/licenses/by/4.0/
[2] https://github.com/automl-conf/LatexTemplate/blob/main/instructions.tex

must also be provided so other researchers can re-execute the experiments. If possible, an execution-ready environment, e.g., in terms of a Docker container, should be made available.

1.1. Code of proposed algorithm/framework/method/model

1.2. Code of all baselines

1.3. Code for preprocessing and postprocessing

1.4. Code for hyperparameter tuning

1.5. Code for execution (training and testing)

1.6. Code for statistical analysis

1.7. Documentation and installation/execution instructions

2. Data-related Aspects: Is all relevant data publicly available?
Reproducibility is only possible if the data used as input to the models and the results are publicly available. It may be insufficient to provide pointers only to previously published datasets, e.g., because preprocessing steps have been applied or publicly shared datasets are sometimes updated.

2.1. Original datasets

2.2. Preprocessed datasets

2.3. Train/validation/test splits

2.4. Results (outcomes of measurements)

2.5. Trained models

3. Configuration Aspects: Are all relevant configuration parameters reported?
Besides code and data, the specifics of the execution of the experiment must be documented. This concerns how the models were tuned, the execution environment, and its configuration.

3.1. Hyperparameter search strategy, search space and search time for all models

3.2. Optimal hyperparameters per dataset and model

3.3. Train-test splitting configurations

3.4. Random seeds

3.5. Required external libraries and their versions

3.6. Used hardware (configuration)

4. Experiment specific aspects and other questions
Depending on the specifics of the experiment, information about various other aspects should be provided. These questions should help better to gauge the level of reproducibility of the experiment. Further, these questions may serve as a place for researchers to justify certain technical choices.

4.1. Has an existing evaluation framework been used? If not, why not?

4.2. Is "one-click" reproducibility supported?

4.3. Are any instructions provided to reproduce (at least parts of) the experiment with limited hardware resources?

4.4. Is an expected runtime to reproduce the results provided?

Author and Reviewer Checklist: Methodology For a mature research community, embracing the evaluation procedure used in previous papers can be considered good practice. However, we must acknowledge that several unjustified protocols, e.g., leave-one-out, have taken root in the recommendation system community. Hence, justifying a research protocol only by saying it was used in previous papers is perhaps unreasonable.

1. Research Questions and Hypothesis: Are the research questions and hypotheses expressed clearly and matching the method and the results?
   The research question should guide the development of the evaluation process. Therefore, it should be clearly stated, and the authors' choices throughout the method should correspond with the research question and conclusions.

   1.1. The research question is clearly stated.

   1.2. The hypothesis is derived from the research question.

   1.3. The experimental design is suited to address the stated research questions.

   1.4. The conclusion is based on the research question and the experimental design.

2. Baselines: Are baselines selected and tuned to ensure appropriate comparisons?
   While one should always compare to the latest best method for the particular task, it is also important to compare against earlier and probably simpler baselines to show the advantage of using the new, more complicated method.

   2.1. The chosen baselines are appropriate to the hypothesis and research question.

   2.2. One of the baselines is successful, e.g., state-of-the-art, for the given task.

   2.3. At least one simple baseline, e.g., kNN, popularity, or random, is included.

   2.4. The baselines are tuned. One must invest sufficient effort in properly training the baselines.

   2.5. There needs to be clarity about whether the baselines were rerun or whether the results were taken from a previous paper.

3. Evaluation Metrics: Is the chosen evaluation metric appropriate to answer the research question?
   Choosing the appropriate evaluation metric for the task is critical. Reporting a large number of unrelated metrics is not good practice.

3.1. The selected metrics are derived from the hypothesis, e.g., RMSE for rating prediction or precision@N for top-N recommendation.

3.2. The reported metrics are not redundant, e.g., RMSE and MAE or DCG and NDCG.

3.3. Tradeoffs between the metrics are explained and evaluated.

4. Data collection: Is the data collection process reasonable and well explained?
This is appropriate when a new dataset is presented. This dataset may be collected from an already running system or using a particular user study.

4.1. The data collection process is clear.

4.2. The study participants' recruitment, introduction, and participation incentives are explained.

4.3. Biases that exist in the data or arise from the data collection process are explained.

4.4. The used datasets are publicly available.

5. Datasets: Are the chosen datasets appropriate for the task?
In offline evaluation, choosing appropriate datasets is highly important. Using a diverse set of datasets supports claims for generalization. In cases where a particular domain is targeted, the datasets must be focused on the task.

5.1. The chosen datasets are appropriate to the task at hand.

5.2. It should be clear whether the datasets were chosen to demonstrate generalization.

5.3. In the generalization case, it is desirable to experiment with a sufficient number of datasets.

5.4. If showing the general applicability of a model is the goal, a diverse set of datasets is used.

5.5. The origins of public datasets are specified.

6. Data preprocessing: Is the data preprocessing well justified and explained?
It is often the case that researchers preprocess, prune, and filter the original dataset before training and testing. In general, preprocessing should be discouraged, especially the dataset's filtering and pruning. Such preprocessing should be kept to a minimum and should be well explained.

6.1. If users or items were pruned from the dataset, the pruning is well justified.

6.2. When pruning is done because the evaluated method works better on a subset of the data, this is made clear.

6.3. : This process is clearly explained and justified if the data was converted, e.g., from numeric ratings to binary like/dislike.

7. Data-splitting: Does the train-test split fit the structure of the dataset and the task? *Most machine learning methods require a training phase. It is, hence, standard practice to split the data into training and test sets, where the test set is used only once to evaluate the algorithm once the training phase is done. The train-test split is designed to simulate the behavior at run time, when the system is aware of all information to date and must make future recommendations. Hence, the split procedure should correspond to the task at hand.*

    7.1. Typically, user-item interactions are split on time, where the training data contains the earlier interactions, and the test data contains the newer ones. When other types of splits are used, this is justified.

    7.2. All algorithms are run on the same train-test split.

    7.3. Cross-validation is applied when possible.

8. Hyperparameter Optimization (HPO): Is the hyperparameter optimization procedure justified and appropriate for the task?
For many machine learning methods, it is well known that HPO is a critical factor for performance. ML algorithms may underperform significantly when their parameters are not tuned to the dataset. Using an organized HPO process for all evaluated algorithms is highly important.

    8.1. The optimization strategy is clearly stated.

    8.2. The hyperparameter configuration space (parameter range) is sufficiently large and clearly defined.

    8.3. The optimization time or number of tested configurations is clearly stated.

    8.4. It is stated in case some algorithms are optimized differently.

9. Experiment execution: Was the experiment executed such that the comparison results are fair and statistically sound?
When running the experiments, all algorithms should receive equal treatment. Statistical significance should be computed to test the likelihood that the observed differences between the algorithms are real.

    9.1. The boundaries between train and test data are respected (i.e., test data not used for checking convergence).

    9.2. There is equal treatment of all compared algorithms (with respect, e.g., to HPO, runtime, hardware).

    9.3. The statistical significance testing method is appropriate for the task.

    9.4. The p-values are properly computed and reported.

    9.5. Confidence intervals are provided whenever possible.

    9.6. The hardware used in the experiment (e.g., memory, processor speed, GPU) is properly described.

10. Sensitivity analysis: Did the authors conduct and report a sensitivity analysis concerning the method parameters and the dataset properties? *Many algorithms have some parameters that must be tuned. It is important to analyze how different values for these parameters influence the performance. In many cases, an algorithm may also be sensitive to the dataset's properties (e.g., sparsity).*

    10.1. The method is executed with different parameter values.

    10.2. The values of all parameters are fixed except for the tested one.

    10.3. The effect of the parameters on the method is reported and discussed.

    10.4. If there are trade-offs between the parameters, they are made clear.

    10.5. Sensitivity to dataset parameters is done similarly to the method parameters.