

Continuous Automated Reproducibility

Practical Tools and Workflows for Repeatable and Reproducible Research

Michael D. Ekstrand

mdekstrand@drexel.edu

Dept. of Information Science

Drexel University

Philadelphia, PA, USA

Abstract

While the RecSys community has been promoting and discussing the importance of reproducible research for many years, there remains a significant lack of practical resources to help RecSys researchers and practitioners, particularly those just getting started in the field, to actually build reproducible experiments and research workflows.

This hands-on tutorial fills that gap. Based on the idea of “continuous reproducibility”, I present open-source tools and practical experimental design and implementation techniques to build recommender systems experiments that are reproducible from Day 1. This tutorial will be useful for researchers at various stages who want to make their work more reproducible, as well as to industry practitioners and researchers who want to build reliable, adaptable, and reproducible pipelines to train and evaluate recommender systems for production applications.

Keywords

reproducibility, workflow automation

ACM Reference Format:

Michael D. Ekstrand. 2026. Continuous Automated Reproducibility: Practical Tools and Workflows for Repeatable and Reproducible Research. In *20th ACM Conference on Recommender Systems (RecSys '26)*, September 27–October 02, 2026, Minneapolis, MN, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3773078.3831854>

1 Motivation

Reproducibility is a hallmark of empirical science, and for at least 15 years the recommender systems community has been concerned with ensuring that research in the field is reproducible, and through that reproducibility, reliable. This has been discussed in papers [e.g., 3, 5], workshops [1], and more recently in the addition of a reproducibility track to the RecSys conference itself. Reproducibility-driven research has found difficulties arising from undocumented and differing metric implementations [6], incomparable experimental protocols [5], ill-tuned baselines [2, 4], and other problems.

Despite the importance of reproducibility to both the research effort and the recommender systems community, there are few practical resources available on *how* to achieve reproducibility. Papers, workshop reports, and similar provide some advice, such as

using established implementations of baselines and metrics, and documenting data processing steps, but a new researcher starting in the field is hard-pressed to find good, pedagogically-oriented resources to guide them to fully-reproducible recommender system experimental pipelines.

This tutorial aims to fill that gap. Drawing on my experience building fully-reproducible recommender systems experiments and creating a course for Drexel University’s M.S. in Data Science program on reproducible workflows (DSCI 641: *Data Workflow Automation*), I presented both a practical approach and workflow engineering techniques based on open-source tools to ensure an experiment is fully-reproducible, because the documented reproduction steps were used to produce the original results. I call this approach “continuous reproducibility”: a continuously-reproducible experiment is always in a fully-repeatable and fully-reproducible state, because the experiment is built out and extended by adding new steps to a reproducible workflow and then reproducing it. This process is significantly aided by workflow and pipeline automation tools such as Data Version Control¹ (DVC), which I will be teaching in the tutorial.

2 Instructor

Michael D. Ekstrand (mdekstrand@drexel.edu) is an assistant professor at Drexel University. He is also co-chair of the Reproducibility and Resource track at RecSys 2026 (although this tutorial is his personal guidance, not an official RecSys resource or opinion).

3 Tutorial Outline

The tutorial follows the following schedule:

1. Welcome and introduction (10m)
 - (a) Background and experience
 - (b) Goals of the tutorial
2. Principles of Reproducibility (15m)
 - (a) Definitions: repeatability, reproducibility, replicability
 - (b) Role of reproducibility in scientific progress
 - (c) Role of reproducibility in industrial application
 - (d) What is an ideal reproducible experiment?
 - (e) Roadblocks to reproducing research
 - (f) Roadblocks to achieving reproducibility
3. Philosophy: **Continuous Reproducibility** (10m)
 - (a) Key idea: run your experiment by reproducing it
 - (b) Building blocks to enable this:
 - Version control for code and artifacts



This work is licensed under a Creative Commons Attribution 4.0 International License. *RecSys '26, Minneapolis, MN, USA*

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2284-4/2026/09

<https://doi.org/10.1145/3773078.3831854>

¹<https://dvc.org>

- Complete, declarative software manifests
 - Automated experiment workflow implemented in a data pipeline tool
 - Configurable experimental scripts
- (c) Additional benefits
- Versioned models, evaluation results, etc.
 - History of experimental artifacts at multiple stages of development or data
 - Tooling can inspect dependencies between experiment stages and components
4. Overview of tools we will use in this tutorial (10m)
- (a) uv
 - (b) Git
 - (c) Data Version Control
 - (d) Jupyter or Marimo
5. Hands-On 1: reproducible data processing and description (30m)
- Working from an example, participants will build a small reproducible pipeline that imports a commonly-used RecSys dataset, prepares it for analysis or evaluation, and computes descriptive statistics into a notebook output.
6. Hands-On 2: adding a recommender (15m)
- We will add a small recommender model training and evaluation to the example in the preceding step.
7. Break
8. Questions so far (10m)
9. Sharing Artifacts (15m)
- (a) Remote storage for versioned artifacts
 - (b) Pushing and pulling within a team
 - (c) Publishing intermediate results
 - (d) Connecting to deployment and other projects
10. Building Up and Out: design patterns for continuously-reproducible RecSys experimental workflows (15m)
11. Special considerations (15m)
- (a) Integrating and automating notebooks
 - (b) Tracking metrics
 - (c) Tracking progress of experiments
12. Hands-On 3: comparing metrics across branches (20m)
13. Documentation and publication (10m)
14. Closing remarks (5m)

4 Audience and Resources

This tutorial is introductory with respect to its primary subject matter (automated reproducibility), and is also appropriate for people just starting with recommender systems research and development. The tutorial is written to be valuable to several audiences:

- **Junior researchers** will learn how to build reproducible experiments from the ground up, helping them start out their research careers on firm reproducible footing.
- **Intermediate and experienced researchers** will learn useful tools and techniques to improve the reproducibility and documentation of their experimental designs and workflows.
- **Industry practitioners**, especially junior engineers or people at smaller firms with less-established data infrastructure,

will learn techniques that they can apply to make their internal model training, evaluation, and update processes reliable and automatically repeatable.

5 Tutorial Resources

The tutorial web site² provides links to the following:

- Lecture slides
- Additional documentation and resources for further reading
- A working example of a fully-reproducible recommender systems experiment using Data Version Control and LensKit³.

A segment of the tutorial will include an interactive example, and attendees are encouraged to follow along on their laptops.

The instructor needs to be able to project from his own laptop in order to lead the examples.

Acknowledgments

This tutorial is based on work supported by the National Science Foundation under Grants 17-51278 and 24-09199.

References

- [1] Alejandro Bellogín, Pablo Castells, Alan Said, and Domonkos Tikk. 2014. Report on the Workshop on Reproducibility and Replication in Recommender Systems Evaluation (RepSys). *SIGIR Forum* 48, 1 (June 2014), 29–35. doi:10.1145/2641383.2641389
- [2] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. 2021. A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research. *ACM Trans. Inf. Syst. Secur.* 39, 2 (Jan. 2021), 1–49. doi:10.1145/3434185
- [3] Michael D Ekstrand, Michael Ludwig, Joseph A Konstan, and John T Riedl. 2011. Rethinking the Recommender Research Ecosystem: Reproducibility, Openness, and LensKit. In *RecSys '11*. ACM, 133–140. doi:10.1145/2043932.2043958
- [4] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19)*. Association for Computing Machinery, New York, NY, USA, 101–109. doi:10.1145/3298689.3347058
- [5] Alan Said and Alejandro Bellogín. 2014. Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks. In *RecSys '14*. ACM Press, New York, NY, USA, 129–136. doi:10.1145/2645710.2645746
- [6] Yan-Martin Tamm, Rinchin Damdinov, and Alexey Vasilev. 2021. Quality Metrics in Recommender Systems: Do We Calculate Metrics Consistently?. In *RecSys '21*. Association for Computing Machinery, New York, NY, USA, 708–713. doi:10.1145/3460231.3478848

²<https://md.ekstrand.net/tutorials/reproducibility/>

³The tutorial will make some reference to LensKit and use it in at least some of the examples, but this tutorial is primarily about reproducible workflows and Data Version Control, not about LensKit.