Reflection and Traceability Report on TTE RecSys

Yinying Huo

This document encapsulates the development process of TTE RecSys, including changes made based on feedback from Dr. Smith and Yuanqi, the challenges I faced while implementing the project, how I addressed them, and what I learned from the experience.

1 Changes in Response to Feedback

[Summarize the changes made over the course of the project in response to feedback from TAs, the instructor, teammates, other teams, the project supervisor (if present), and from user testers. —TPLT]

[For those teams with an external supervisor, please highlight how the feedback from the supervisor shaped your project. In particular, you should highlight the supervisor's response to your Rev 0 demonstration to them. —TPLT]

[Version control can make the summary relatively easy, if you used issues and meaningful commits. If you feedback is in an issue, and you responded in the issue tracker, you can point to the issue as part of explaining your changes. If addressing the issue required changes to code or documentation, you can point to the specific commit that made the changes. Although the links are helpful for the details, you should include a label for each item of feedback so that the reader has an idea of what each item is about without the need to click on everything to find out. —TPLT]

[If you were not organized with your commits, traceability between feedback and commits will not be feasible to capture after the fact. You will instead need to spend time writing down a summary of the changes made in response to each item of feedback. —TPLT]

[You should address EVERY item of feedback. A table or itemized list is recommended. You should record every item of feedback, along with the source of that feedback and the change you made in response to that feedback. The response can be a change to your documentation, code, or development process. The response can also be the reason why no changes were made in response to the feedback. To make this information manageable, you will record the feedback and response separately for each deliverable in the sections that follow. —TPLT]

[If the feedback is general or incomplete, the TA (or instructor) will not be able to grade your response to feedback. In that case your grade on this document, and likely the Revision 1 versions of the other documents will be low. —TPLT]

This project has two reviewers: the course instructor, Dr. Smith, and a peer reviewer, Yuanqi. Every piece of feedback and the corresponding response has been tracked in GitHub issues. The following list summarizes the feedback and the changes made. The issue link provides more detailed feedback, and the commit link allows viewing the exact changes made based on the feedback.

1.1 SRS

Feedbacks from Dr. Smith

1. Issue 10: Typos, formatting issues, and lack of explanations for some terminologies.

   commit 46a581f: Fixed all typos and formatting issues and added more explanations for the terminologies used in the document.

2. Issue 11: TM is not abstract, lack of explanation of the symbols used in TM and IM.

   commit 9aa7710: Add explanations of the symbols used in TM and IM.

   commit 39bd533: Update TM

3. Issue 12: Split the functional requirements for the training and inference phases and add traceability to IM. Make the non-functional requirements unambiguous.

   Commit f60e042: Updated non-functional requirements and added traceability to IM.

   Commit 7cb3188: Updated functional requirements.

Feedbacks from Yuanqi

1. Issue 4: Add a subtitle.

   Commit 1dcf561: Added a subtitle.

2. Issue 5: Include a description of `argsort()` from IM2 and γ from IM3.

   Commit bd10f3f: Removed `argsort()` from IM2 since it may cause confusion and is unnecessary. Added missing mathematical notation.

3. Issue 6: A minor formatting issue.

   Commit bd10f3f: Fixed the formatting issue.

4. Issue 7: The image used in Section 3 does not correctly reflect the system description.

   Commit 3b09d2c: Updated the image. It now shows that the user's information is the input to the user embedding function, which outputs a user embedding for further processing.

5. Issue 8: It is more beneficial to use a BibTeX file.

   Commit 68bc19a: Switched to using a BibTeX file for all references.

1.2 Design and Design Documentation

Feedbacks from Yuanqi

1. Issue 17: In the first paragraph of the Introduction section of MIS, it may be helpful to include both the project name and a brief one or two sentence overview of your project.

   Commit 62c0f08: Added a brief project overview to the Introduction section of MIS.

2. Issue 18: Some notations are used in the document but not listed in notation section.

   Commit 62c0f08: Updated notation section, removed some unnecessary notations.

3. Issue 19: Formatting issue in section 7 (MIS of the Data Processing Module).

   Commit 51e9385: Update section7

4. Issue 20: 2 functions did not specific the input and output

   Commit 51e9385 Updated function definitions.

5. Issue 21: The Neural Network Architecture Module should not use Vector Operations Module.

   Commit 51e9385 Fixed this error.

1.3 VnV Plan and Report

Feedbacks from Dr. Smith

1. Issue 15: Typos and formatting issues

   Commit 387fdd7 Fixed typos and formatting issues. Added hyperlinks for documents and specified which tests will be automated.

2. Issue 16: Be specific about the system tests for functional requirements. Some tests may be challenging to implement. Section 3.7, Software Validation Plan, is incorrect—it is not a validation plan but a verification exercise.

   Commit b5f0d11, Commit 4ef7829, Commit ad87f5a: Rewrote Sections 4.1 and 4.2 to make each test explicitly define the input and output. Also simplified the system tests. Rewrote section 3.7.

   After these changes, while implementing the project, I found that some tests were still challenging to implement. As a result, I rewrote a significant portion of the VnV Plan. Commit 4ef7829

I haven't received the VnV Plan feedback from Yuanqi.

2 Challenge Level and Extras

2.1 Challenge Level

This is a non-research project.

2.2 Extras

The extra component of the project is a user guide.

3 Design Iteration (LO11 (PrototypeIterate))

When designing the first version, I attempted to incorporate too many features without having comprehensive experience in project development, particularly in system and unit testing, which made the project unfeasible. The most significant difference between the initial design and the final implementation was that the system and unit tests were redesigned to be simpler and more straightforward to implement.

Since I am following a widely used structure of a recommendation system, there is not a significant difference in the overall structure of different modules.

In my initial design, I failed to recognize that developers and users are distinct stakeholders with different needs. Users require a simple, intuitive interface for interaction, while developers need control over system parameters such as the number of layers in a deep neural network or the number of training epochs. In the final version, I incorporated a user interface to facilitate easier project usage.

4 Design Decisions (LO12)

A major limitation encountered during implementation was the poor quality of available datasets. I could only find public datasets with very limited personal information, and many items had received ratings from only a single user. This posed a significant challenge as personal information is crucial for recommendation systems, especially when employing deep neural network approaches which demand high-quality data.

To address this issue, I devoted time to feature engineering, calculating additional features not present in the original dataset, such as frequency metrics. While this resulted in some improvement, the impact was minimal. Given the time constraints, I was unable to replace the deep neural network approach, as doing so would have necessitated restructuring the entire project. Instead, I modified the Software Requirements Specification (SRS) to adjust the quality and performance requirements for the recommendation system. Therefore, the performance of this project will not be measured.

5 Reflection on Project Management (LO24)

5.1 What Went Well?

The Continuous Integration (CI) with GitHub Actions worked very well. I was really nervous about CI and GitHub Actions before I implemented the project. However, the automated unit tests and auto-training were easier than I expected. GitHub Actions has very easy-to-follow instructions and a lot of online resources.

5.2 What Went Wrong?

The performance and the quality of the recommended results were not good. This was primarily due to the dataset being limited and my choice of a deep learning approach which requires a good dataset.

5.3 What Would you Do Differently Next Time?

I would first find a dataset, and then decide the appropriate learning algorithm to use based on the dataset I have. I should not choose an algorithm then find a dataset. This is especially important for a project related to learning. I will not be able to get good results if the dataset is bad or if I am not using a suitable learning algorithm.