Verification and Validation Report: TTE RecSys

Yinying Huo

April 2, 2025

1 Revision History

Date	Version	Notes
Apr. 2 2025	1.0	First draft

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test

[symbols, abbreviations or acronyms – you can reference the SRS tables if needed —SS]

Contents

List of Tables

List of Figures

This document includes the results of VnV plan.

3 Functional Requirements Evaluation

The functional requirements are tested using both system tests and unit tests.

3.1 Dataset

To ensure the project receives a valid dataset for training:

Manually, a Jupyter Notebook (data_preprocessing.ipynb) is used to merge the raw data (three CSV files) into a single CSV file.

Then, the automated system test (test_data_validation.py) runs each time the dataset is updated before training starts to ensure that the input CSV file contains all the required columns.

Additionally, an automated unit test (test_data_processing.py) runs on each push to verify the correctness of the data processor, which generates new features for users and items.

All tests have passed successfully.

3.2 Model Training Convergence

The system test (test_model_convergence.py) checks whether the training loss decreases as training progresses.

This test has passed successfully.

3.3 Model storage

The system test (test_model_storage.py) runs each time model training is completed. The test ensures that the trained model and computed embeddings are correctly stored in the 'output' folder.

This test has passed successfully.

3.4 Embedding generation

The unit test 'test_embedding_generation.py' will check the correctness of embedding generation. This test has passed successfully.

4 Nonfunctional Requirements Evaluation

4.1 Usability

4.2 Reliability

4.3 Protability

5 Unit Testing

6 Changes Due to Testing

[This section should highlight how feedback from the users and from the supervisor (when one exists) shaped the final product. In particular the feedback from the Rev 0 demo to the supervisor (or to potential users) should be highlighted. —SS]

7 Automated Testing

8 Trace to Requirements

9 Trace to Modules

10 Code Coverage Metrics

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Reflection.

1. What went well while writing this deliverable?
2. What pain points did you experience during this deliverable, and how did you resolve them?
3. Which parts of this document stemmed from speaking to your client(s) or a proxy (e.g. your peers)? Which ones were not, and why?
4. In what ways was the Verification and Validation (VnV) Plan different from the activities that were actually conducted for VnV? If there were differences, what changes required the modification in the plan? Why did these changes occur? Would you be able to anticipate these changes in future projects? If there weren't any differences, how was your team able to clearly predict a feasible amount of effort and the right tasks needed to build the evidence that demonstrates the required quality? (It is expected that most teams will have had to deviate from their original VnV Plan.)