Cornac-AB: An Open-Source Recommendation Framework with Native A/B Testing Integration

Darryl Ong
Singapore Management University
darrylong@smu.edu.sg

Quoc-Tuan Truong
Amazon, USA
truquoc@amazon.com

Hady W. Lauw
Singapore Management University
hadywlauw@smu.edu.sg

ABSTRACT

Recommender systems significantly impact user experience across diverse domains, yet existing frameworks often prioritize offline evaluation metrics, neglecting the crucial integration of A/B testing for forward-looking assessments. In response, this paper introduces a new framework seamlessly incorporating A/B testing into the Cornac recommendation library. Leveraging a diverse collection of model implementations in Cornac, our framework enables effortless A/B testing experiment setup from offline trained models. We introduce a carefully designed dashboard and a robust backend for efficient logging and analysis of user feedback. This not only streamlines the A/B testing process but also enhances the evaluation of recommendation models in an online environment. Demonstrating the simplicity of on-demand online model evaluations, our work contributes to advancing recommender system evaluation methodologies, underscoring the significance of A/B testing and providing a practical framework for implementation. The framework is open-sourced at *https://github.com/PreferredAI/cornac-ab*.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Collaborative filtering**.

KEYWORDS

recommendation library, A/B testing, open-source framework

ACM Reference Format:
Darryl Ong, Quoc-Tuan Truong, and Hady W. Lauw. 2024. Cornac-AB: An Open-Source Recommendation Framework with Native A/B Testing Integration. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3589335.3651241

1 INTRODUCTION

Recommender systems serve as crucial components that contribute to elevating user satisfaction across a diverse array of domains, ranging from e-commerce platforms to content streaming services. Over the years, numerous frameworks [1, 3, 4] have been developed to help build these systems, with a predominant emphasis on employing offline evaluation metrics to gauge recommendation accuracy through backward testing. Despite the extensive attention given to offline evaluation, there exists a noteworthy gap in the incorporation of A/B testing–a forward testing methodology–which is vital for assessing the online performance of recommender systems. Additionally, the seamless integration of these frameworks with existing systems and applications is an often-neglected yet critical aspect that warrants careful consideration for ensuring optimal functionality and user experience.

In this work, we address this gap by introducing a framework designed for the integration of A/B testing with the Cornac recommendation library [6]. Cornac stands out as a well-established library with a diverse collection of model implementations, making it a popular choice among both academic researchers and industry practitioners. Our objective is to develop a framework that seamlessly incorporates native A/B testing capability into Cornac, providing an efficient and user-friendly solution for deploying and evaluating recommender systems' performance.

Utilizing the offline trained models from Cornac, our framework simplifies the setup of A/B tests. We surpass mere integration by introducing a thoughtfully crafted dashboard and a robust backend designed for logging and scrutinizing user feedback through OpenSearch [1]. This not only streamlines the A/B testing process but also elevates the overall assessment of recommendation models in an online setting. Furthermore, we illustrate the ease of conducting on-demand evaluations of online models using the logged feedback data, presenting a comprehensive solution for assessing recommender systems in real-world scenarios.

2 OVERVIEW

2.1 Cornac Library

Within the evolving domain of recommender system development, Cornac stands as a noteworthy open-source Python library, offering a multifaceted approach to the advancement of recommendation algorithms. At its core, Cornac offers robust utilities that span the spectrum of recommender systems development. The distinctive focus of Cornac is on recommendation models harnessing multimodal auxiliary information, encompassing elements such as social networks, item textual descriptions, and product images. This targeted emphasis addresses the inherent sparsity prevalent in user-item interactions. The framework is complemented by an extensive support ecosystem, encompassing detailed documentation, tutorials, examples, and a diverse set of built-in benchmarking datasets. Cornac's compatibility with prevalent machine learning libraries like TensorFlow and PyTorch enhances its adaptability, providing users with a seamless integration path into their existing experimental workflows.

[1] https://opensearch.org

WWW '24 Companion, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0172-6/24/05
https://doi.org/10.1145/3589335.3651241

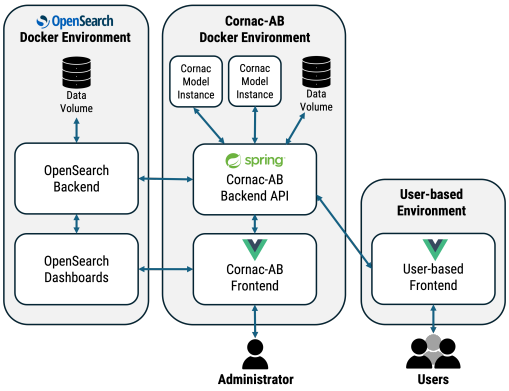


Figure 1: Architecture Overview

Cornac's recognition extends across industry practitioners and academia, evidenced by its adoption on GitHub [2] and multiple publications [6, 8, 9], including contribution to the Journal of Machine Learning Research. Notably, it has received an endorsement from ACM RecSys Conference for its role in systematic evaluation and reproducibility of recommendation algorithms.

2.2 Framework Design

The overall architecture, illustrated in Figure 1, is composed of three extensible and adaptable environments: Cornac-AB, OpenSearch, and the user-based application. Within the Cornac-AB environment, the core experiment logic resides, accompanied by a backend API serving as the orchestrator for the A/B testing solution.

The API plays a pivotal role in integrating each trained model with a Cornac instance and implementing the recommendation logic for user segmentation in A/B testing. Furthermore, the backend has the capability to leverage Cornac's evaluation functionalities on demand, facilitating the comparison and analysis of results across multiple model instances. In addition to its core functionalities, the Cornac-AB environment features an admin frontend. This frontend empowers administrators to visually set up experiments using Cornac-trained models. OpenSearch Dashboards are employed within the admin frontend to visualize and analyze recommendations and feedback data indexed by OpenSearch.

The user-based environment exemplifies how existing solutions can interact with Cornac-AB backend REST APIs to obtain recommendations and provide feedback. This separation of environments allows Cornac-AB to work alongside most existing solutions.

3 DEMONSTRATION

In this section, we demonstrate the usage of our framework through a focused exploration of a book recommendations application. Supposedly, we are interested in testing the performance of three prevalent recommendation models: BPR [5], BiVAECF [7], and LightGCN [2]. All of them have their implementations readily accessible within the Cornac library. To walk through the framework capabilities, we leverage the publicly available Goodbooks [3] dataset. The

[2]https://github.com/PreferredAI/cornac
[3]https://github.com/zygmuntz/goodbooks-10k

Table 1: Offline Evaluation Results

	AUC	NDCG@50	Recall@50
BPR	0.9033	0.1527	0.1975
BiVAECF	0.9555	0.2522	0.3295
LightGCN	0.9579	0.2468	0.3308

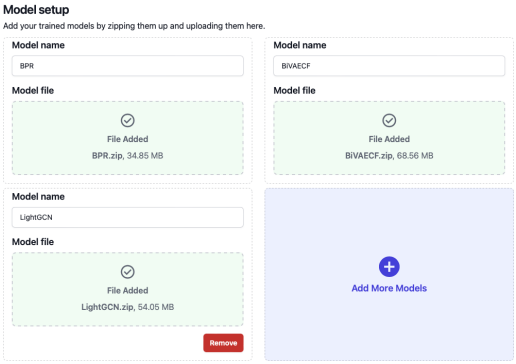


Figure 2: A/B Testing Experiment Setup

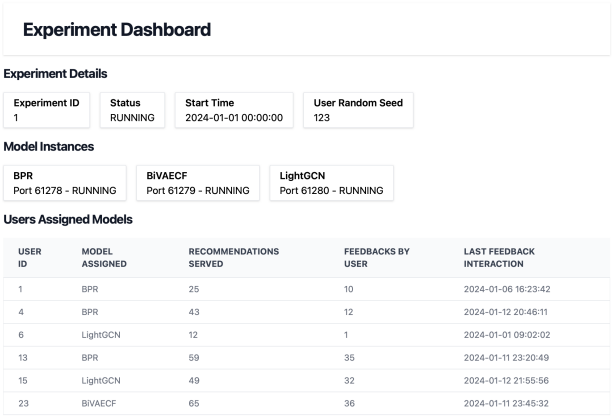
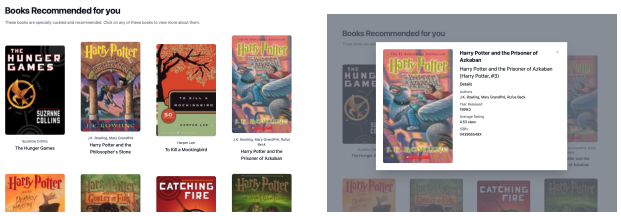


Figure 3: Experiment Monitoring Dashboard

experimentation process is structured and demonstrated through following key steps.

Offline Model Training. We consider user-item interactions from rating data (*ratings.csv*) as the primary source of offline training data for our comparative models. To ensure comprehensive coverage, we employ a user stratification strategy for data splitting, guaranteeing the inclusion of all users in the training dataset. The data is partitioned for each user, allocating 80% for training, 10% for validation, and 10% for offline testing. The validation set is used for determining the optimal model performance through hyper-parameter tuning. All of the comparative models are defined to use the same size of 50 dimensions for user/item latent embedding. In this evaluation, we employ three widely-adopted metrics to measure recommendation accuracy, namely AUC, NDCG@50, and Recall@50. The results obtained from this model offline evaluation step, on the test set, are reported in Table 1.



(a) Browsing Recommendations (b) User Click Feedback

Figure 4: Front-end Interface for User Feedback Loop

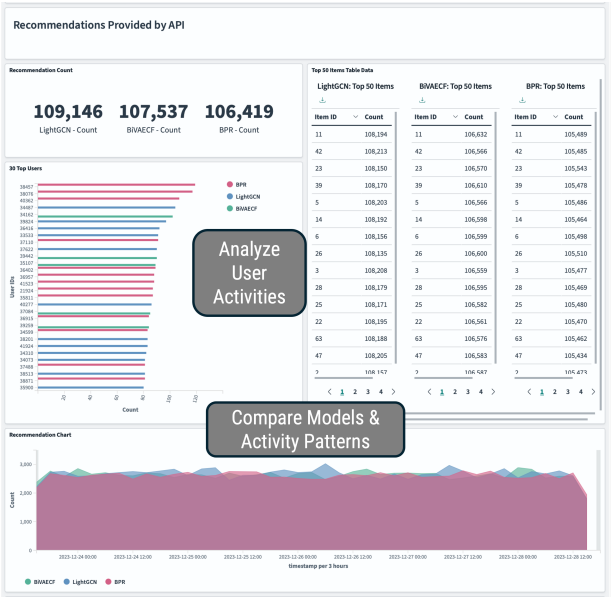


Figure 5: Recommendation Dashboard

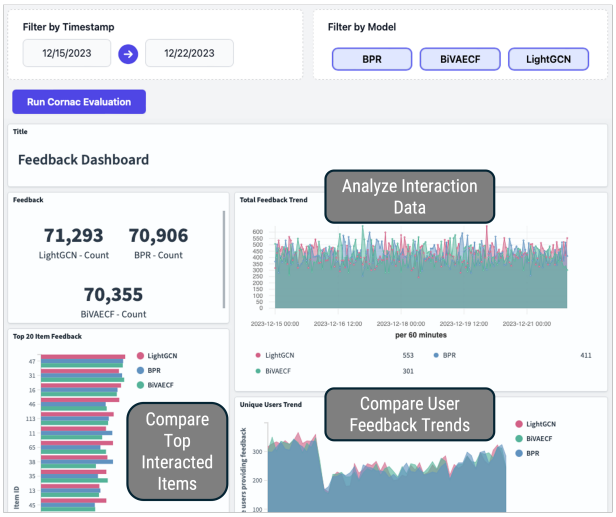


Figure 6: User Feedback Dashboard

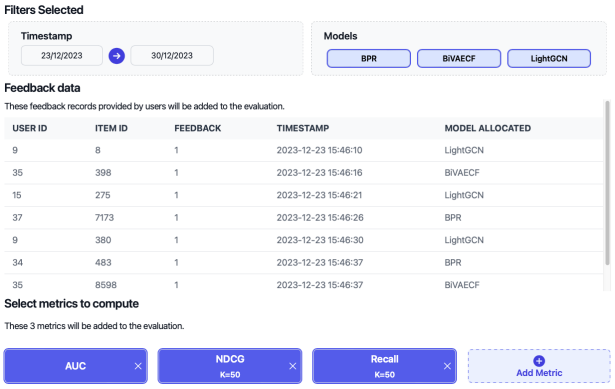


Figure 7: Online Evaluation Setting

Initializing A/B Test. Upon the completion of offline training and testing for all models, the initiation of an A/B testing experiment is straightforward. With the Cornac-AB backend operational, one navigates to the setup UI, as depicted in Figure 2. The interface facilitates uploading of pre-trained model files, initializing the experiment promptly. A random assignment of each user to one of the models constitutes their participation in the A/B test.

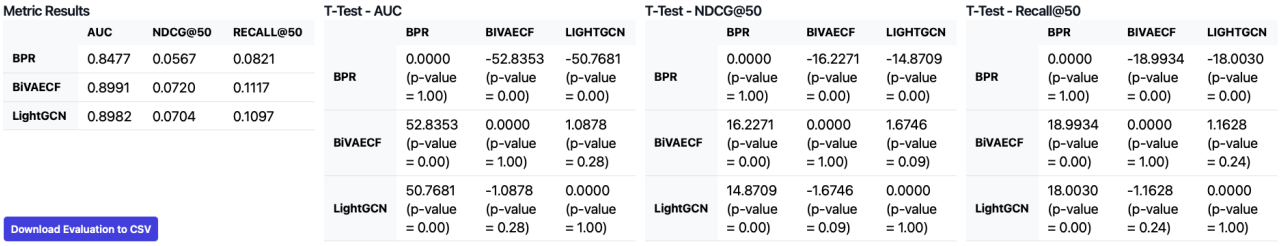
The experiment monitoring dashboard which includes all settings, also allows one to track important statistics, as illustrated in Figure 3. This not only provides transparency but also ensures a comprehensive view of the experiment's dynamics. Furthermore, one can easily identify the user-model assignments within the experiment, streamlining subsequent observations and analyses.

Logging User Feedback. So the demonstration is self-contained, we develop a dedicated front-end to showcase user recommendations and facilitate the collection of user feedback, as depicted in Figure 4. When a user accesses this interface, they are presented with personalized recommendations generated by the model assigned to them during the earlier random assignment phase.

During the user's interaction with the recommendations, click feedback is systematically logged into the backend data repository for subsequent analysis and online evaluation. This logging is pivotal in closing the feedback loop for our application users, ensuring that their interactions contribute to the iterative improvement of the recommendation system. In fact, the logged feedback is the main source used for model online evaluation during the A/B test.

Analyzing Logged Feedback. The recommendation dashboard as shown in Figure 5 provide valuable insights into the displayed recommendations by each model. This is instrumental for gaining a comprehensive understanding of the inner workings of the models.

Importantly, our focus extends beyond mere observation, with an interest in the logged user feedback obtained during the dynamic interactions between users and our recommendation front-end interface. Figure 6 showcases the feedback dashboard, allowing researchers to specify a designated time period for log analysis and choose models for visualization. This functionality serves the dual purpose of facilitating preliminary model comparison and identifying specific subsets of the log data that goes into subsequent model evaluations. To exemplify the next step in our model evaluation process, we leverage bookmark data (*to_read.csv*) provided in the



Metric Results

	AUC	NDCG@50	RECALL@50
BPR	0.8477	0.0567	0.0821
BiVAECF	0.8991	0.0720	0.1117
LightGCN	0.8982	0.0704	0.1097

Download Evaluation to CSV

T-Test - AUC

	BPR	BIVAECF	LIGHTGCN
BPR	0.0000 (p-value = 1.00)	-52.8353 (p-value = 0.00)	-50.7681 (p-value = 0.00)
BiVAECF	52.8353 (p-value = 0.00)	0.0000 (p-value = 1.00)	1.0878 (p-value = 0.28)
LightGCN	50.7681 (p-value = 0.00)	-1.0878 (p-value = 0.28)	0.0000 (p-value = 1.00)

T-Test - NDCG@50

	BPR	BIVAECF	LIGHTGCN
BPR	0.0000 (p-value = 1.00)	-16.2271 (p-value = 0.00)	-14.8709 (p-value = 0.00)
BiVAECF	16.2271 (p-value = 0.00)	0.0000 (p-value = 1.00)	1.6746 (p-value = 0.09)
LightGCN	14.8709 (p-value = 0.00)	-1.6746 (p-value = 0.09)	0.0000 (p-value = 1.00)

T-Test - Recall@50

	BPR	BIVAECF	LIGHTGCN
BPR	0.0000 (p-value = 1.00)	-18.9934 (p-value = 0.00)	-18.0030 (p-value = 0.00)
BiVAECF	18.9934 (p-value = 0.00)	0.0000 (p-value = 1.00)	1.1628 (p-value = 0.24)
LightGCN	18.0030 (p-value = 0.00)	-1.1628 (p-value = 0.24)	0.0000 (p-value = 1.00)

Figure 8: Online Evaluation Results

Goodreads dataset, simulating logged feedback from users engaged while browsing the model recommendations.

Online Model Evaluation. Having determined the subset of feedback and selected models for online evaluation, one can seamlessly transition to the definition of desired metrics, as depicted in Figure 7. In this case, we adopt the same metrics employed in our offline experiment for continuity and comparability across the evaluation process. Upon completion of the online evaluation, the results will be presented in a structured table format, as illustrated in Figure 8. This presentation of data allows for a comprehensive comparison across the selected models, providing insights into their respective performances. The table facilitates a nuanced contrast with the offline results, as shown in Table 1, thereby offering a holistic perspective on the models' efficacy in both offline and online settings. Moreover, T-tests are also performed and supplemented for all the metrics. This concludes an A/B testing cycle.

4 INTEGRATION

For developers aiming to implement a recommender system in their applications, the framework stands out for its seamless integration.

Modular Design. By separating the Cornac-AB environment from the the application environment, every component can be developed and interfaced independently. This modular architecture not only ensures scalability to handle growing datasets and user bases but also facilitates the incorporation of new features and improvements without disrupting existing systems.

Data-Intensive Focus. With OpenSearch as the indexing solution, the framework is efficient for data-intensive operations. Its flexibility and scalability are well-suited for handling the diverse data sources associated with recommendation systems. Developers can benefit from OpenSearch's open-source nature, ensuring adaptability and transparency in managing and indexing large datasets.

Large Model Collection. At the heart of our framework lies the Cornac library, which offers a diverse set of models. This diversity empowers developers with a wide range of choices to build recommender systems tailored to the specific needs of their applications. Whether the focus is on collaborative filtering, content-based filtering, or other types of models, Cornac provides the flexibility needed to align recommendations with unique user preferences.

Our framework unleashes the power of recommendation A/B testing, allowing developers to systematically compare different algorithms, configurations, or strategies. Whether building a recommender system for an e-commerce platform, a content streaming service, or any user-based application, the framework provides the tools and flexibility needed for implementation and integration.

5 CONCLUSION

We address a significant gap in existing recommender system frameworks by introducing a new platform that integrates A/B testing seamlessly with the Cornac library. Unlike traditional evaluation approaches that primarily focus on offline metrics, our framework unlocks the capability of A/B testing–a forward testing methodology that is often overlooked. Leveraging Cornac's extensive collection of model implementations, our framework enables the easy setup of A/B tests using offline trained models. We also introduce a meticulously designed dashboard and a robust backend for efficient logging and analysis of user feedback.

By showcasing the simplicity of conducting on-demand online model evaluations using logged feedback data, our framework offers a comprehensive solution for real-world recommender system assessment. In doing so, we contribute significantly to the advancement of recommender system evaluation methodologies, emphasizing the crucial role of A/B testing integration. This work is poised to empower both academic researchers and industry practitioners in enhancing the performance evaluation of recommender systems, thereby contributing to the continual improvement of user experiences across diverse application domains.

ACKNOWLEDGMENTS

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2021-020).

REFERENCES

[1] Scott Graham, Jun-Ki Min, and Tao Wu. 2019. Microsoft recommenders: tools to accelerate developing recommender systems. In *ACM RecSys*. 542–543.
[2] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. 639–648.
[3] Nicolas Hug. 2020. Surprise: A Python library for recommender systems. *Journal of Open Source Software* 5, 52 (2020), 2174. https://doi.org/10.21105/joss.02174
[4] Maciej Kula, James Chen, Xinyang Yi, Tiansheng Yao, Maheswaran Sathiamoorthy, Lichan Hong, and Ed Chi. 2020. *TensorFlow Recommenders*. https://github.com/tensorflow/recommenders
[5] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. 452–461.
[6] Aghiles Salah, Quoc-Tuan Truong, and Hady W Lauw. 2020. Cornac: A comparative framework for multimodal recommender systems. *The Journal of Machine Learning Research* 21, 1 (2020), 3803–3807.
[7] Quoc-Tuan Truong, Aghiles Salah, and Hady W Lauw. 2021. Bilateral variational autoencoder for collaborative filtering. In *WSDM*. 292–300.
[8] Quoc-Tuan Truong, Aghiles Salah, and Hady W Lauw. 2021. Multi-modal recommender systems: Hands-on exploration. In *ACM RecSys*. 834–837.
[9] Quoc-Tuan Truong, Aghiles Salah, Thanh-Binh Tran, Jingyao Guo, and Hady W Lauw. 2021. Exploring cross-modality utilization in recommender systems. *IEEE Internet Computing* 25, 4 (2021), 50–57.