Leveraging LightGBM Ranker for Efficient Large-Scale News Recommendation Systems

Tetsuro Sugiura
Starbucks Coffee Japan, Limited
Tokyo, Japan
TetsuroSugiura@starbucks.co.jp

Yosuke Yamagishi
The University of Tokyo
Tokyo, Japan
yamagishi-yosuke0115@g.ecc.u-tokyo.ac.jp

Yodai Kishimoto
InsightX, Inc.
Tokyo, Japan
yodai.kishimoto@insightx.tech

ABSTRACT

This study addresses the news recommendation task presented by Ekstra Bladet in the ACM RecSys Challenge 2024. The task aims to predict which articles users are likely to click on from a list of candidate articles, by leveraging users' browsing history, personal information, article details, and session information. Our approach is centered around a LightGBM Ranker model. Various features were used, including user, article, and session information, as well as their interactions. Additionally, embeddings were created from users' browsing history and news article texts, and their cosine similarities were used as additional features. Appropriate validating methods using time series were explored, and effective data sampling and ensemble methods were also proposed to fit the data within limited memory. Finally, the final model was created by performing a weighted ensemble using multiple periods and random seeds. This method achieved high performance with AUC of 0.8169. As a result, an 8th place finish was achieved among around 200 participating teams. The code is available at https://github.com/tetsuro731/RecSys-Challenge-2024-tetsuro731.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Recommender Systems, RecSys Challenge, LightGBM, Ranker Algorithm, News Recommendation

ACM Reference Format:
Tetsuro Sugiura, Yosuke Yamagishi, and Yodai Kishimoto. 2024. Leveraging LightGBM Ranker for Efficient Large-Scale News Recommendation Systems. In *ACM RecSys Challenge 2024 (RecSys Challenge '24), October 14–18, 2024, Bari, Italy*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3687151.3687156

1 INTRODUCTION

The number of Internet users has surpassed 5 billion, and the demand for web-based news is rapidly increasing [4]. For instance, BBC News, one of the world's largest news websites, reportedly attracts over 1 billion users per month [2]. Given that each user within this vast audience accesses news sites with distinct needs, developing personalized news recommendation systems has become a crucial technological challenge [6].

The ACM RecSys Challenge 2024 provided a platform for competing approaches in effective news recommendation, utilizing data from Ekstra Bladet, a major Danish media outlet. The task involved a dataset comprising user records of 2 million individuals over a six-week period, with over 120,000 news articles as potential recommendations. Leveraging this extensive data to propose optimal news articles for users from a vast pool of options presented not only a technically challenging task but also an imperative one, given its significant societal implications. The evaluation metrics adopted for this challenge included AUC, MRR, nDCG@5, and nDCG@10.

Our team participated in this competition, focusing on the efficient utilization of large-scale data to construct a high-accuracy machine learning model. While recent years have seen the publication of several GBDT-based ranking algorithms [7], we chose to employ the LightGBM Ranker model [5] among these options. This decision was motivated by LightGBM's widespread adoption among practitioners and researchers alike. We believed that validating its performance in this context would provide valuable insights to the research and industrial community, given its popularity and practical relevance. Our key contributions in this competition include:

- Generation of advanced features by combining users' browsing history and personal information, enabling the capture of temporal changes in user interests.
- Development of a method to embed news article text using deep learning language models and convert it into a numerical format compatible with LightGBM, facilitating recommendations that consider semantic similarities between articles.
- Proposal of a model construction method that efficiently utilizes massive data through a combination of sampling and incremental learning, achieving high-accuracy model training under computational resource constraints.

2 DATASET

Our system utilizes the Ekstra Bladet News Recommendation Dataset (EB-NeRD), a comprehensive Danish dataset specifically designed for advancing news recommendation research. This dataset, provided by Ekstra Bladet for the RecSys Challenge 2024, offers a rich source of information for developing and evaluating news recommendation systems.

RecSys Challenge '24, October 14–18, 2024, Bari, Italy

ACM ISBN 979-8-4007-1127-5/24/10
https://doi.org/10.1145/3687151.3687156

2.1 Dataset Overview

EB-NeRD comprises data from over 2.7 million users and includes more than 600 million impression logs from Ekstra Bladet's platform. The dataset also contains information on over 120,000 news articles, each enriched with textual content features such as titles, abstracts, and full body text. This wealth of information allows for the development of context-aware recommender systems in a low-resource language setting.

2.2 Data Collection and Preprocessing

The data was collected over a six-week period from April 27th to June 8th, 2023. This timeframe was carefully selected to avoid major events that could lead to atypical user behavior. Active users were defined as those who had between 5 and 1,000 news click records during a three-week period from May 18th to June 8th, 2023.

2.3 Dataset Structure

The dataset consists of a training set and a validation set, along with the relevant articles. The official test set is provided separately. Each data split contains two main files:

(1) `behaviors.parquet`: Contains the impression logs for a 7-day period.
(2) `history.parquet`: Includes users' click histories for the 28 days preceding the behavior logs.

Additionally, two supplementary files are provided:

(3) `articles.parquet`: Contains detailed information about the news articles.
(4) `artifacts.parquet`: Includes embeddings of the articles' textual information.

3 METHODOLOGY

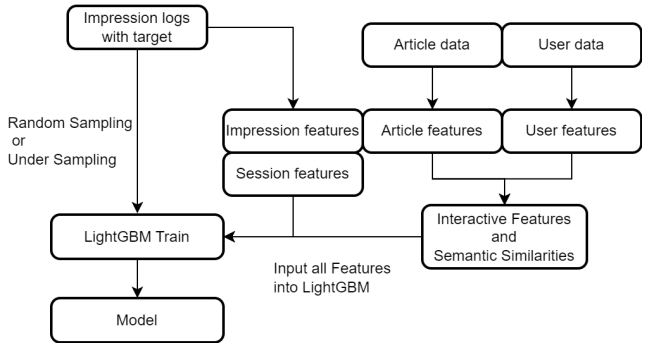

Figure 1: Feature Generation and Training Process

Our approach to the RecSys Challenge 2024 combines sophisticated feature engineering with LightGBM's ranking model, which are visualized in Figure 1. We used the 'lambdarank' objective, which is designed for ranking tasks, and nDCG@k as the evaluation metric. The evaluation metric will be discussed in more detail later. This section details our methodology, including the feature engineering process, model architecture, and training procedure.

3.1 Feature Engineering

Our feature engineering process involved creating a diverse set of features to capture various aspects of user behavior, article characteristics, and their interactions. The following categories of features were developed:

3.1.1 Article and User Features. We extracted several features directly from the article and the user metadata, respectively. In addition, we created features that captured the interaction between articles and users like:

- Average age of users who viewed each article, compared to the actual user's age
- User's preferred articles (based on grouping by `article_id` for each user) compared to actual article values

3.1.2 Text Embedding Features. Embedding-based features were used in our approach:

- Embeddings provided by the organizers, obtained by embedding the text incorporating title, subtitle, and body using FacebookAI's xlm-roberta-base model [3].
- Our original embeddings, extracted using the sentence transformers library and the pre-trained multilingual model "distiluse-base-multilingual-cased-v2," focusing on title [8, 9]. This approach considers the human decision-making process where users often select articles based on the title alone.

The final feature used was the cosine similarity between user and article embeddings. This was determined by calculating the cosine similarity between the mean embeddings of articles in the user's history and the embeddings of the target article.

3.1.3 Impression and Session Features. We engineered features to capture impression and session-level information like:

- Time elapsed since the article was published at the time of the impression
- Number of articles displayed as a list in the impression
- Frequency of article appearances per session

In addition, impression time distribution features like statistical measures (mean, standard deviation, count, etc.) of when users impressed articles were also used.

We integrated all our engineered features as input to the LightGBM ranker. In addition, instead of using Pandas for dataframe operations in feature engineering, Polars was employed in order to manage data more quickly and with greater memory efficiency.

3.2 Data Split Strategy

The dataset provided for the challenge consisted of consecutive weekly data for training, validation, and testing periods. Given that the ultimate goal was to predict user behavior for the 7-day test period, we structured our local data subsets to mirror this setup, using 7 days of training data to predict the subsequent 7 days of validation data. We implemented the following strategy:

- We used only 7 days of the training data for our local validation process. This allowed us to fine-tune our model and determine optimal parameters within our computational constraints.
- Using the parameters obtained from the local validation, we then trained on the validation dataset

Both models were used in our final ensemble for making predictions on the test dataset. This approach was particularly effective because the validation data immediately preceded the test data chronologically, making it more representative of the conditions and patterns we needed to predict.

In addition, due to limited memory, random sampling or undersampling was conducted on the impression logs. We conducted several experiments and finally adopted 60% random sampling.

3.3 Detailed Training Process

In recommender systems, a common approach is usually what is called "two-phase approach":

(1) Narrowing down from a vast number of articles to a few dozen to a few hundred candidates (**Candidate Generation Phase**).
(2) Sorting the articles by score (**Re-Ranking Phase**).

In this competition, since a list of candidate articles for each impression is already provided, the key lies in how accurately we can perform the second phase, which is re-ranking. For our model architecture, we employed LightGBM's ranking model "LightGBM Ranker", which is specifically designed for learning-to-rank tasks. LightGBM is an efficient implementation of gradient boosting decision trees (GBDT). In the context of the ongoing development of novel methodologies, particularly those based on neural networks like Transformer [10], LightGBM consistently demonstrates superior performance, especially in handling tabular data.

In terms of hyperparameter optimization, Optuna was used [1], focusing on maximizing nDCG@k. This approach allowed us to efficiently explore the hyperparameter space and identify the most effective configuration for our model. For more detailed settings of these parameters, please refer to our GitHub repository.

3.4 Ensemble Inference

Model construction was performed using three distinct seed values. During inference, a total of six models, trained on both the training and validation datasets, were utilized to compute scores. The final prediction was obtained by averaging these scores as an aggregated score.

3.5 Hardware Setup

For our computational needs, we utilized Google Colaboratory's TPU v2 node. It's important to note that our primary motivation for choosing this platform was not the TPU itself, but rather the access to approximately 330GB of CPU RAM that it provided. This substantial memory capacity was crucial for handling our large dataset and complex feature engineering processes efficiently. We had access to this hardware setup from May 2024 to June 2024.

4 EVALUATION METRICS

In this competition, several standard metrics in the recommendation field were measured at the same time after submissions, including the area under the ROC curve (AUC), mean reciprocal rank (MRR), and normalized discounted cumulative gain (nDCG@k) for k shown recommendations. The primary metric which was used to determine the final ranking for the challenge was AUC.

Table 1: The final submission scores for the primary metric, AUC, and other metrics, which are MRR, nDCG@5, and nDCG@10 for each dataset.

	Provisional	Final
AUC	0.8164	0.8169
MRR	0.6106	0.6102
nDCG@5	0.6790	0.6787
nDCG@10	0.7002	0.6998

4.1 Correlation among Evaluation Metrics

Although AUC was adopted as the primary metric, it was more commonly used as a metric for classification tasks, and nDCG@k was easier to implement as an evaluation metric when training with the LightGBM Ranker. Therefore, assuming that there is a strong positive correlation between AUC and nDCG@k, we adopted nDCG@k as a metrics in our experiments.

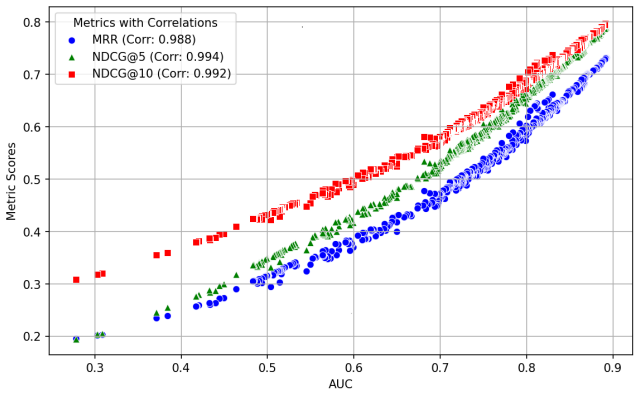

Figure 2: Correlation between AUC and Ranking Metrics.

Figure 2 shows a plot of the correlation between AUC and other metrics measured from all participants' submissions during the competition period in the final dataset. It is evident that there is a very strong positive correlation, with a Pearson correlation coefficient of 0.99, across all metrics including MRR, nDCG@5, and nDCG@10. This result seems consistent with our hypothesis.

5 RESULTS AND DISCUSSIONS

5.1 Performance on Evaluation Metrics

The test dataset was partitioned into two distinct subsets: one for updating the provisional leaderboard and another for determining the final rankings. The final submission scores for the primary metric, AUC, and other metrics in each subset are summarized in Table 1.

In this competition, the variation in scores between these subsets was minimal. Our final model achieved a score of 0.8169 on the final dataset for AUC.

5.2 Correlations between Validation Results and Leaderboard Scores

Figure 3 illustrates the correlation between our local validation nDCG@10 scores and the provisional scores on the leaderboard.

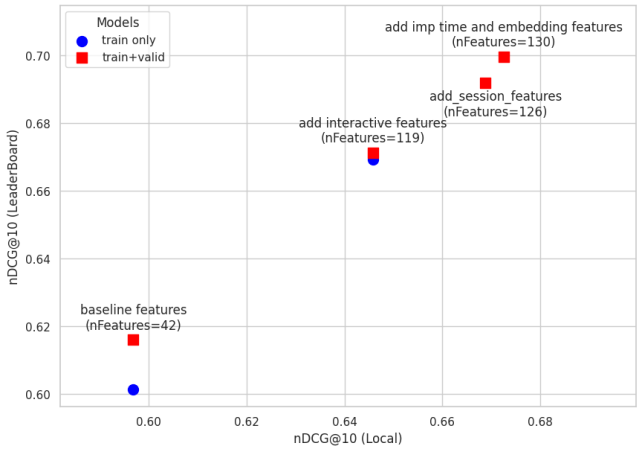

Figure 3: Comparison of nDCG@10 Score between Local Validation Score and Leaderboard Score

The blue circular markers in the figure represent models created using data from the training period, while the red square markers indicate ensembles of models trained separately during the training and validation periods. The "nFeatures" in the figure represents the number of features used to create the models; our final model utilized 130 features. The figure shows that there is a strong positive correlation between the validation scores and the leaderboard scores. We repeatedly conducted experiments based on local validation scores rather than leaderboard scores for the following two reasons:

(1) There is a limit to the number of submissions per day in the system.
(2) There is a risk of overfitting to the leaderboard.

Regarding the top right point in the figure, some of the impression time distribution features are not available in a live setup, but these features do not rank high in feature importance. Therefore, they do not largely affect our strategy or results.

5.3 Feature Importance Analysis

We analyzed the feature importance calculated using the built-in feature importance method of LightGBM, which measures the total gain of a feature when it is used as a split point in all trees in the model.

5.3.1 Top Three Features. The three most important features were:

- Time since article publication (1.464×10^7)
- Difference between the length of the article list in the impression and the average length of the article list. (6.185×10^6)
- View count of the article in the last 10 minutes (5.359×10^6)

These features emphasize the critical role of timing and popularity metrics in predicting user engagement. The dominance of time since publication suggests that content recency is a primary driver of user interest. This could indicate a user preference for up-to-date information or reflect the platform's content promotion strategies. The high importance of view counts in 10 minutes highlights that popular content tends to attract more engagement, possibly due to social proof or content quality indicators.

5.3.2 Semantic Similarity Features. Two semantic similarity features showed moderate importance in our model:

- Cosine similarity based on XLM-RoBERTa embeddings (6.706×10^5)
- Cosine similarity based on title embeddings (1.392×10^5)

The XLM-RoBERTa-based feature, which ranked 27th overall in importance, was derived from embeddings extracted using the XLM-RoBERTa base model, subsequently dimensionally reduced using PCA. This feature captures semantic similarity between the current article and the user's reading history, considering the full article content.

The title-based similarity feature, which ranked 67th overall in importance, was calculated using a similar method but focused solely on article titles. While direct comparison between these features is challenging due to the use of different base models, it is noteworthy that the feature incorporating full article content demonstrated higher importance than the title-only feature.

This suggests that semantic similarity, particularly when considering the entire article content, plays a significant role in predicting user engagement. This insight could be valuable for content recommendation strategies and understanding user preferences at a deeper semantic level.

6 CONCLUSION

The RecSys Challenge 2024 provided a platform to compete in developing effective web news recommendation methods using vast amounts of user history and article data. Our solution was built upon the LightGBM Ranker model, a standard approach, and incorporated the following innovative elements:

- Advanced feature generation utilizing user history and textual information
- An efficient training process that validates on a subset of data and readjusts the model using the most recent data
- Large-scale data processing methods under computational resource constraints

By integrating these techniques, we achieved a high ranking of 8th place among all participants, demonstrating excellent performance across various metrics.

The main contribution of this study lies not only in proposing basic feature generation methods but also in demonstrating that efficient and high-accuracy model construction is possible even in situations where using the entire dataset is challenging due to its enormous size. These findings suggest the potential for direct application in implementing real-time recommendation systems on actual large-scale news platforms.

ACKNOWLEDGMENTS

We acknowledge the support of the Data Analytics Team at Starbucks Coffee Japan and GCoE team at Starbucks Corporation for reviewing this paper.

REFERENCES

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2623–2631.

[2] BBC News. 2021. New data shows BBC is the world's most visited news site. https://www.bbc.com/mediacentre/worldnews/2021/new-data-shows-bbc-is-the-worlds-most-visited-news-site Accessed: 2024-06-29.

[3] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 8440–8451.

[4] DataReportal. 2024. Global Digital Overview. https://datareportal.com/global-digital-overview Accessed: 2024-06-29.

[5] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[6] Cornelius A. Ludmann. 2017. Recommending News Articles in the CLEF News Recommendation Evaluation Lab with the Data Stream Management System Odysseus. In *Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum*. CEUR-WS.org. https://ceur-ws.org/Vol-1866/paper_111.pdf

[7] Ivan Lyzhin, Aleksei Ustimenko, Andrey Gulin, and Liudmila Prokhorenkova. 2023. Which tricks are important for learning to rank?. In *International Conference on Machine Learning*. PMLR, 23264–23278.

[8] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[9] Nils Reimers and Iryna Gurevych. 2020. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/2004.09813

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st Neural Information Processing Systems*. Neural Information Processing Systems Foundation. https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

Received 15 July 2024