Harnessing Temporal Dynamics and Content: An Ensemble of Gradient Boosting Machines for News Recommendation

Tomomu Iwai*
PKSHA Technology Inc.,
tomomu_iwai@pkshatech.com, Japan

Akihiro Tomita*
PKSHA Technology Inc.,
akihiro_tomita@pkshatech.com,
Japan

Tomoyuki Arai*
PKSHA Technology Inc.,
tomoyuki_arai@pkshatech.com,
Japan

Hiroki Ogawa*
PKSHA Technology Inc.,
hiroki_ogawa@pkshatech.com, Japan

Takuma Saito*
PKSHA Technology Inc.,
takuma_saito@pkshatech.com, Japan

Abstract

This paper presents our approach for Team Tom3TK that achieved third place in the ACM RecSys Challenge 2024, organized by Ekstra Bladet. The challenge centered on large scale news recommendations with over 380 million impression logs and aimed to predict which article a user will click on from a series of articles displayed in a specific session. Our approach utilizes a LightGBM ensemble model, integrating article timeliness features with content-based recommendations from historical implicit feedback and employs advanced multilingual embedding models. Experiments revealed the significant impact of timeliness and content-based features, the importance of suitable embedding models, and the phenomenon of performance plateauing with the expansion of training samples.

CCS Concepts

• **Information systems**; • **Computing methodologies** → Machine learning; Machine learning algorithms;

Keywords

Recommender Systems, News Recommendation, Sentence Embeddings

ACM Reference Format:
Tomomu Iwai, Akihiro Tomita, Tomoyuki Arai, Hiroki Ogawa, and Takuma Saito. 2024. Harnessing Temporal Dynamics and Content: An Ensemble of Gradient Boosting Machines for News Recommendation. In *ACM RecSys Challenge 2024 (RecSys Challenge '24), October 14–18, 2024, Bari, Italy*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3687151.3687159

1 INTRODUCTION

Recommendation systems are crucial for filtering and suggesting contents to users. News recommendation systems, in particular, possess two distinctive characteristics. Firstly, they rely heavily on implicit feedback, such as click history, due to the unavailability of explicit feedback like rating data. Secondly, news articles have extremely short lifespans, making timely recommendations essential.

The ACM RecSys Challenge 2024 [1], organized by Ekstra Bladet, a prominent news publisher in Denmark, focused on large scale news recommendations. The challenge aimed to predict which article a user will click on from a series of articles displayed during a specific session. To benchmark performance for the challenge, the Ekstra Bladet News Recommendation Dataset (EB-NeRD) was released. This Danish news dataset contains over 2.3 million users, more than 120 thousand articles, and exceeds 380 million impression logs. Logs were collected from active users over six weeks in 2023, with the performance evaluated using the ROC-AUC metric.

In this paper, we present the third-place approach from team Tom3TK, illustrated in Figure 1. Our solution integrates timeliness and content-based features to enhance the relevance of news recommendations. A LightGBM ensemble effectively combines features that capture the temporal dynamics of news articles with advanced content similarity measures. This approach addresses the unique challenges of news recommendation, where both the freshness of content and its alignment with implicit user interests from historical click data are crucial. Our experiments aimed to understand the impact of different features, embedding models, and training dataset sizes on the model's performance. All code required for replication of this study is published on https://github.com/akihiro-tomita/recsys-2024-tom3tk.

2 FEATURE ENGINEERING STRATEGY

2.1 Dataset Overview

EB-NeRD consists of user logs collected over a six-week period from April 27 to June 8, 2023. The data is divided into three sets: training, validation, and test. Each set includes a. 7-day period of behavior logs and b. the users' click history 21 days prior to the behavior logs.

The behavior logs show how users interacted with news articles. The data includes information about articles that were inview, meaning the articles that were displayed to a specific user. Inview articles within a timeframe are grouped into impressions, each assigned a unique impression_id. The primary task of the competition is to predict which article a user clicked within each impression.

The history logs capture the users' click history for the 21 days before the behavior logs. The data includes details about when a

*Authors contributed equally to this work.

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

ACM ISBN 979-8-4007-1127-5/24/10
https://doi.org/10.1145/3687151.3687159

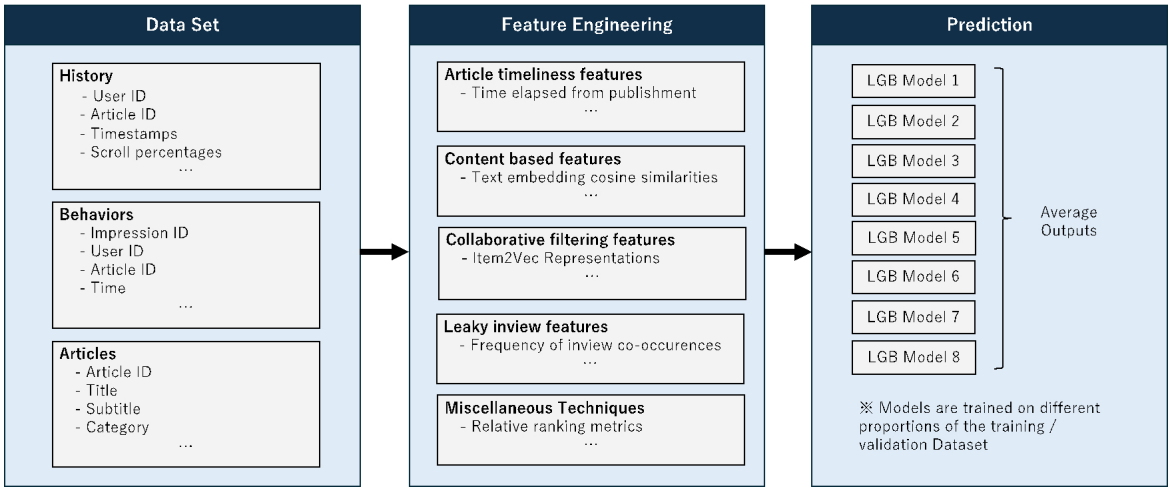

Figure 1: Overall prediction diagram of the constructed recommendation system

user clicked on an article, the duration they spent reading it, and the extent to which they scrolled through the article.

In addition to these main data types, basic article information such as publication dates and article contents, and precalculated embedding data from text-base and image-base models were provided for the articles listed across the datasets.

2.2 Article Timeliness

In news recommendation systems, the freshness of articles is crucial. We engineered features to capture this timeliness, such as the elapsed time from publication to inview_time. To understand the relative freshness of each article within an impression, we used two methods: ranking the articles by elapsed time and normalizing the elapsed times within the impression. These approaches help gauge an article's likelihood of being clicked, as newer content tends to attract more user attention. Incorporating these timeliness features enhances the relevance of our recommendations.

2.3 Content-based Features

Content-based recommendation is a method used in recommendation systems to suggest items to users based on the attributes of the items and the user's past interactions or preferences.

We constructed latent representations of user preferences based on their article reading history. These representations were then used to compute cosine similarity with the latent representations of target articles, serving as features in our model.

Rather than using the pre-computed embeddings provided by the competition organizers, we opted to recalculate article embeddings ourselves. For this task, we employed the multilingual-e5-large-instruct model [2]. At the time of our research, this model was considered state-of-the-art (SOTA) according to the Danish language results in the Scandinavian Embedding Benchmark [3].

We explored multiple strategies for computing user latent representations based on their article reading history: a. Utilizing the entire historical record of articles b. Focusing on the most recent reading history c. Applying weights based on metrics such as scroll percentage and read time d. Constructing category-specific representations using only articles from the same category. Among the strategies explored, our results indicated that applying weights based on engagement metrics, particularly scroll percentage, was the most effective approach.

In addition to average-based user embeddings, we implemented a more computationally intensive approach. We calculated cosine similarities between each article in a user's history and the articles in the target impression_id. From these similarity scores, we derived additional features such as the maximum similarity, minimum similarity, and the 90th percentile of similarities.

2.4 Collaborative Filtering

Collaborative filtering is a technique used in recommendation systems to suggest items to users based on the preferences and behaviors of other users.

While collaborative filtering techniques have proven effective in recent global competitions, such as the H&M Personalized Fashion Recommendations [4] and the OTTO Multi-Objective Recommender System [5], their efficacy is significantly diminished in the domain of news recommendation. This reduced effectiveness can be attributed to the inherent factor to news consumption patterns. The rapid obsolescence and the continuous influx of news articles results in a much shorter content lifespan. Consequently, traditional collaborative filtering approaches, which rely heavily on historical user-item interactions, face substantial limitations in capturing the dynamics of news recommendation scenarios.

We explored a wide range of collaborative filtering techniques but found that most failed to significantly improve predictive accuracy. The sole exception was the Item2Vec approach [6], which we adapted to our context by treating each article as a word and a user's reading history as a sentence. This method allowed us to derive latent representations for individual articles. However, even this approach yielded only marginal performance improvements. The limited efficacy of collaborative filtering techniques can be

largely attributed to the pervasive cold-start problem in news recommendation: there is minimal overlap between articles in a user's reading history and those presented in current recommendations.

2.5 Leaky Inview Features

In this competition, we can exploit all information recommended by the current recommendation system, including future inview data that would typically be unavailable in real-time production environments. Although such future information would be inaccessible in a real-world scenario, its use was permitted within the competition framework.

Leveraging this data, we engineered several potent features such as; the frequency of inview occurrences for each article id across various time buckets (e.g., 5 minutes, 30 minutes, 24 hours), the count of inview article for specific article ids per individual user; the co-occurrence frequency of pairs of article ids within the same impression.

2.6 Miscellaneous Techniques

Aside from our main strategies, we employed additional feature engineering techniques.

For key features, we augmented their absolute values with relative metrics within each impression_id, such as the ranking within the impression, and the ratio to the maximum value in the impression. In addition, we computed various statistical measures across each impression_id, including mean, min, max, skew. We then calculated the deviation of individual feature values from these statistical measures, creating new features that captured relative positioning of the target article id within each impression context.

This approach to feature engineering enhanced our model's ability to discern subtle differences between articles within the same impression, leading to higher AUC. The final number of features used in our approach is 432.

3 MODEL AND TRAINING STRATEGY

3.1 Model

For our model, we utilized LightGBM [7] with the lambda rank loss objective [8]. Our approach involved eight models, forming an ensemble to enhance robustness and predictive accuracy. We simply averaged the outputs of these models to achieve a final prediction. Each of the eight LightGBM models were trained with the same features, but on different subsets of the data to improve generalization and reduce overfitting. Detailed hyperparameter configurations can be found in the accompanying solution code.

3.2 Training Strategy

Due to memory and computational constraints, we decided to partition the training and validation datasets into 100 chunks each. For training, we randomly selected different 20 chunks (around 2.6M samples) from the 200 chunks for each of the eight models. This number was experimentally determined; using more than 20 chunks did not improve performance but linearly increased training time. The training and validation dataset covered different dates, and mixing chunks from the two datasets improved the performance on the test dataset. To further enhance robustness, we varied the epochs and random states for each model.

Table 1: Impact of Leaky Features

Setting	ROC-AUC	Num of Features
Leaky	0.8525	436
Non-Leaky	0.7775	107

3.3 Result

Our methodology achieved third place in final leaderboards of the challenge. The primary evaluation metric, ROC-AUC, reached 0.8707, demonstrating the effectiveness of our model.

4 EXPERIMENTS

4.1 Experiments Setting

We conducted experiments to better understand our model's components and impacts. Due to computational constraints, we used a small dataset (2% of full data) for most experiments. The reported ROC-AUC scores are based on the validation dataset using a single trained LightGBM model. While these scores are lower than our final competition results, they provided valuable insights. Importantly, performance trends from this small dataset consistently correlated with leaderboard standings, validating our experimental approach and guiding our optimization efforts.

4.2 Leaky Features

In this challenge, future inview information was available, which is not realistic in a practical scenario. To address this issue, we calculated the model's accuracy after removing all features that could potentially cause data leakage, specifically those derived from future inview data, total_inviews, total_pageviews, and total_read_time. This reduction in features decreased the total number from 436 to 107.

After removing the potentially leaky features, the ROC-AUC score decreased from 0.8525 to 0.7775, a reduction of approximately 0.075 points, as shown in Table 1. This significant drop in performance indicates that the features derived from future information had a substantial impact on the model's predictive power.

Table 2 presents the importance weights of the top features in the Leaky Model, categorized for convenience. The importance was calculated based on the gain from LightGBM. The category with the highest weight is "Leaky," confirming that the model heavily relies on future information that has leaked into the training data. Timeliness and Content-based has a substantial weight as well, indicating their significant impact on the model's predictions. As anticipated, the Collaborative Filtering category shows minimal contribution to the model's predictions.

4.3 Comparison of Article Embedding Models

For content-based feature engineering, it is crucial to represent articles with high-quality latent representations. While the competition host provided pre-computed latent representations for the articles,

Table 2: Importance Weights

Feature Category	Importance Weight
Timeliness	29.7%
Content-based	23.1%
Collaborative Filtering	<0.1%
Leaky	37.3%
Basic*	9.8%

*'Basic' category includes features such as the number of articles displayed.

Table 3: Embedding Model Comparison

Embedding Model	ROC-AUC
multilingual-e5-large-instruct	0.8525
bge-m3 / dense	0.8515
Word2Vec*	0.8476
BERT*	0.8427
Contrastive*	0.8507
xlm*	0.8458
Image*	0.8430

* Provided by the competition host

we conducted additional experiments using latest models' embeddings. We report results of experiments using the multilingual-e5-large-instruct model and the bge-m3 model [9].

As demonstrated in Table 3, latest models such as multilingual-e5-large-instruct and bge-m3 outperform other embeddings models. The multilingual-e5-large-instruct model achieved the highest ROC-AUC score of 0.8525, closely followed by the bge-m3 / dense model at 0.8515. These results surpass the best-performing host-provided embedding (Contrastive) by 0.0018 and 0.0008 points, respectively.

Models like Word2Vec and BERT, which were not primarily intended for sentence embeddings, demonstrate markedly lower performance, with ROC-AUC scores of 0.8476 and 0.8427, respectively. This performance gap underscores the importance of using embedding models specifically tailored for sentence-level representations in recommendation tasks.

4.4 Training Dataset Size

We investigated the impact of training dataset size to the performance. It is generally assumed that larger training datasets result in better performance. However, increased memory usage and computation time can obstruct efficient experimentation. Therefore, identifying the dataset size at which improvements in accuracy begin to saturate is valuable. According to the results depicted in Figure 2, accuracy improves gradually up to 10 million samples, beyond which only marginal improvements are observed. Furthermore, in this competition, it was found that a dataset size ranging from 0.2M to 2M was sufficient to make a reasonable assessment of the experimental results.

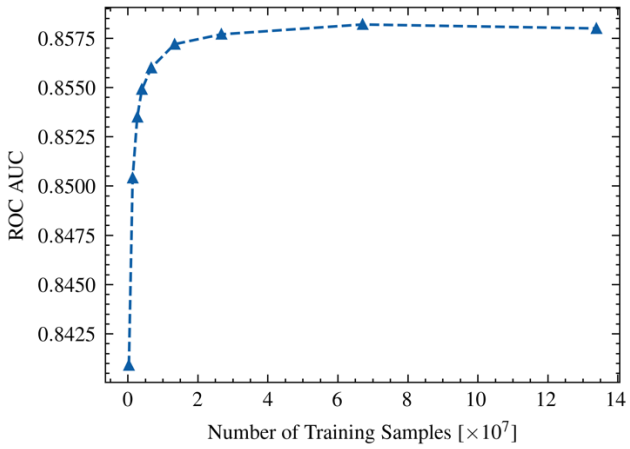

Figure 2: Training data size impact to model performance

5 CONCLUSION

In this paper, we present our third place approach in the 2024 ACM RecSys Challenge organized by Ekstra Bladet. Our solution focuses on integrating timeliness and content-based features to enhance the relevance of news recommendations. In many fields where large models such as LLM play a dominant role, it has been confirmed that the GBDT (Gradient Boosting Decision Tree) approach remains effective in recommendation systems.

Additionally, our experiments revealed three key findings: first, both timeliness and content relevance significantly impact prediction results. Second, selecting an appropriate embedding model enhances performance. Third, performance saturation occurs beyond a certain number of training samples. These insights provide valuable direction for future improvements in recommendation system design and data utilization strategies.

Lastly, we highlight critical limitations in current recommendation system competitions, which primarily rely on logs generated by pre-existing algorithms. This structure risks merely replicating existing systems rather than fostering genuine advancements. To address this, we propose a paradigm shift towards using data logs gathered through random or probabilistic sampling. This would mitigate algorithmic biases and provide a more accurate measure of a model's real-world efficacy. By redesigning competition frameworks to incorporate such unbiased data collection, we can better evaluate and enhance the performance of recommendation systems in practical scenarios.

References

[1] 2024. RecSys Challenge 2024. https://www.recsyschallenge.com/2024/
[2] Wang, Liang, *et al.* 2024. Multilingual e5 text embeddings: A technical report. arXiv preprint arXiv:2402.05672 (2024).
[3] Enevoldsen, Kenneth. 2023. Scandinavian Embedding Benchmark. Retrieved July 7, 2024, from https://kennethenevoldsen.github.io/scandinavian-embedding-benchmark/.
[4] Carlos García Ling, ElizabethHMGroup, FridaRim, inversion, Jaime Ferrando, Maggie, neuraloverflow, xlsrln. 2022. H&M Personalized Fashion Recommendations. Kaggle. https://kaggle.com/competitions/h-and-m-personalized-fashion-recommendations
[5] Andreas Wand, Philipp Normann, Sophie Baumeister, Timo Wilm, Walter Reade, Maggie Demkin. 2022. OTTO – Multi-Objective Recommender System. Kaggle. https://kaggle.com/competitions/otto-recommender-system

[6] Barkan, Oren, and Noam Koenigstein. 2016. Item2vec: neural item embedding for collaborative filtering. In 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP).
[7] 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. Advances in neural information processing systems 30 (2017).
[8] C.J.C. Burges, R. Ragno and Q.V. Le. 2006. Learning to Rank with Non-Smooth Cost Functions. Advances in Neural Information Processing Systems, 2006.
[9] Chen, Jianlv, *et al.* 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. arXiv preprint arXiv:2402.03216 (2024).