Article

Not peer-reviewed version

A Multi-Objective Recommender System for Enhanced Consumer Behavior Prediction in E-Commerce

Yiping Sun *, Yijun Xiang, Dongxian Zou, Na Li, Haotian Chen

Posted Date: 14 November 2024

doi: 10.20944/preprints202411.1057.v1

Keywords: Multi-objective recommender system; online shopping; consumer behavior prediction; ensemble learning; LightGBM; XGBoost; CatBoost

Article

A Multi-Objective Recommender System for Enhanced Consumer Behavior Prediction in E-Commerce

Yiping Sun [1,*], Yijun Xiang [2], Dongxian Zou [3], Na Li [4] and Haotian Chen [5]

[1] Shanghai JiaoTong University, Shanghai, China
[2] Georgia Institute of Technology, Atlanta, USA
[3] Independent Researcher, Mill Creek, USA
[4] Independent Researcher, Jersey City, USA
[5] University of Illinois Urbana-Champaign, Urbana, USA
[*] Correspondence: sunacc@sjtu.edu.cn

Abstract: With the proliferation of online shopping, consumers are inundated with an overwhelming number of product choices, often leading to decision fatigue and abandoned shopping carts. To mitigate this issue, recommender systems have been employed to personalize the shopping experience and enhance consumer engagement. However, traditional recommender systems typically focus on single-objective optimization, such as predicting clicks or purchases, which limits their effectiveness. In this study, we propose a multi-objective recommender system designed to simultaneously optimize multiple consumer behaviors, including clicks, cart additions, and completed orders. Leveraging the OTTO Recommender Systems Dataset, our system employs a two-stage candidate reranker model that first generates a pool of potential items and then ranks them based on predicted relevance. Our experimental results demonstrate that an ensemble model combining LightGBM, XGBoost, and CatBoost Rankers outperforms individual models, achieving superior recall rates. This multi-objective approach provides a comprehensive solution for enhancing the accuracy and effectiveness of recommender systems in e-commerce.

Keywords: multi-objective recommender system; online shopping; consumer behavior prediction; ensemble learning; LightGBM; XGBoost; CatBoost

1. Introduction

The advent of e-commerce has revolutionized the retail industry, providing consumers with unprecedented access to a vast array of products from the convenience of their homes. However, this convenience comes with its own set of challenges. The sheer volume of available products can overwhelm consumers, leading to decision fatigue and a phenomenon known as "choice overload," where the abundance of options paradoxically results in decreased satisfaction and increased likelihood of abandoning shopping carts. This scenario not only frustrates consumers but also represents a significant loss in potential revenue for retailers.

This paper presents a novel multi-objective recommender system designed to enhance the shopping experience by simultaneously optimizing for multiple consumer behaviors, including clicks, cart additions, and completed orders. We leverage the OTTO Recommender Systems Dataset, a comprehensive dataset containing 12 million anonymized user sessions and over 220 million events, to develop and evaluate our system. The dataset's richness in real-world e-commerce interactions makes it an ideal foundation for building a robust and effective recommender system.

Our approach involves a two-stage candidate reranker model. In the first stage, we generate a pool of candidate items using a variety of methods, such as co-visitation, click-to-click patterns, and Word2Vec embeddings. In the second stage, we rank these candidate items based on their predicted relevance to the user, employing sophisticated feature engineering and negative sampling techniques to enhance model performance. We explore the use of ensemble learning, combining LightGBM, XGBoost, and CatBoost Rankers, to achieve superior performance across multiple objectives. Ensemble models are particularly effective in capturing the diverse patterns in user behavior by leveraging the strengths of each individual model.

2. Related Work

The field of recommender systems has seen significant advancements over the past two decades, with a variety of models and approaches developed to predict and enhance consumer behavior. Traditional methods have primarily focused on single-objective optimization, which has proven effective but limited in the context of multi-faceted consumer interactions in e-commerce. This section provides an overview of relevant literature, highlighting the evolution of recommender systems and recent innovations aimed at multi-objective optimization.

Resnick et al. [1] introduced GroupLens, an early collaborative filtering system that used user ratings to recommend articles. This method laid the groundwork for many subsequent recommendation algorithms.Pazzani and Billsus [2] discussed content-based methods that recommend items similar to those the user has liked in the past, using features derived from the items themselves. A Javanmardi et al. [3] Balancing proactive and reactive discussions in construction planning improves effectiveness and reduces risk.Burke [4] reviewed hybrid recommender systems, which combine collaborative and content-based filtering techniques to improve recommendation accuracy.G Li et al. [5] introduces IACO-IABC, combining ant colony optimization and artificial bee colony algorithms for improved mobile robot path planning.

Koren et al. [6] introduced matrix factorization techniques in recommender systems, significantly enhancing the scalability and accuracy of recommendations.Hu et al. [7] addressed the challenge of using implicit feedback (e.g., clicks, views) in collaborative filtering, which often contains less explicit user preferences.Zhang et al. [8] provided a comprehensive survey of deep learning techniques applied to recommender systems, highlighting their ability to capture complex user-item interactions.C He et al. [9] uses a graph neural network and ontology to improve bridge preservation planning and prediction accuracy.

Lacerda et al. [10] explored multi-objective optimization in recommender systems, emphasizing the need to balance multiple goals such as accuracy, diversity, and novelty.Z Lin et al. [11] develops an AI-based model for efficient and smooth robot path planning and navigation.C He et al. [12] identifies key collaborative scheduling practices that enhance construction project performance.B Zhang et al. [13] summarizes recent advancements in text sentiment analysis for improving enterprise decision-making and public opinion monitoring

Li et al. [14] proposed a contextual bandit algorithm for personalized news article recommendations, balancing exploration and exploitation to optimize recommendations.Ying et al. [15] developed Graph Convolutional Networks (GCNs) for recommender systems, demonstrating their ability to effectively capture user-item relationships.A Javanmardi et al. [16] shows that the Observe–Plan–Do–Check–React cycle enhances construction workflow reliability. Notably, the methodology for integrating generative AI, as discussed by [17] , provided critical insights into optimizing our ensemble model, particularly in the fine-tuning stages for improved model generalization and performance across multiple objectives.

Pan et al. [18] explored transfer learning techniques for cross-domain recommendation, addressing the challenge of sparsity in new domains by leveraging knowledge from related domains.Rokach [19] reviewed ensemble methods in machine learning, highlighting their potential to improve the robustness and accuracy of recommender systems by combining multiple models.S Liu et al. [20] introduces the Intelligent Bus Travel Service Model (IBTSM) to optimize bus service quality.Zhang and Chen [21] discussed the growing need for explainability in recommender systems, proposing methods to make recommendations more transparent and understandable to users.

V Nikolaenko et al. [22] introduced methods to ensure user privacy in recommender systems, using techniques like differential privacy to protect user data.H Yan et al. [23] explores how natural language processing enhances data mining and information retrieval in the big data era. Burke [24] discussed the need for fairness in recommender systems, proposing algorithms that ensure equitable treatment of all users and items.A multi-task learning approach for efficient modeling and monitoring of tool surface progression in ultrasonic metal welding [25]. Kunaver and Požrl [26] reviewed methods to enhance diversity in recommender systems, balancing relevance and variety to improve user satisfaction.

In summary, the literature on recommender systems has evolved significantly, with various methodologies being proposed to address different aspects of recommendation. While traditional systems have focused on single-objective optimization, there is a growing recognition of the need for multi-objective approaches to capture the complex nature of user behavior. This study contributes to this emerging field by proposing a multi-objective recommender system that optimizes for multiple consumer actions, demonstrating superior performance through ensemble learning techniques.

3. Dataset and Preprocessing

3.1. Dataset Description

The dataset utilized in this study is the *OTTO Recommender Systems Dataset* , which was prominently featured in the Kaggle competition 'OTTO-Multi-Objective Recommender System'. This comprehensive dataset includes 12 million anonymized user sessions, capturing over 220 million events that encompass clicks, cart additions, and orders. Furthermore, it contains a catalog of 1.8 million unique articles, making it an extensive resource for developing and testing recommender systems. The dataset is structured into sessions, where each session is uniquely identified by an ID, and events, which are time-ordered sequences associated with specific articles identified by their IDs (aids). Each event includes a Unix timestamp (ts) and is categorized by its type.

3.2. Basic Statistics

The dataset is divided into training and test sets, with the training set comprising 12,899,779 sessions, 1,855,603 items, and 216,716,096 events. The test set includes 1,671,803 sessions, 1,019,357 items, and 13,851,293 events. The density of the dataset is notably low, at 0.0005% for both the training and test sets, indicating a sparse dataset. Detailed statistics on the number of events per session and per item are provided in Tables 1 and 2 , highlighting the distribution and variability within the dataset.

Table 1. Basic Statistics of Training and Test Sets: Sessions and Items

Dataset	Sessions	Items	Density (%)
Train	12,899,779	1,855,603	0.0005
Test	1,671,803	1,019,357	0.0005

Table 2. Basic Statistics of Training and Test Sets: Events, Clicks, Carts, and Orders

Dataset	Events	Clicks	Carts	Orders
Train	216,716,096	194,720,954	16,896,191	5,098,951
Test	13,851,293	12,340,303	1,155,698	355,292

Figures 1 and 2 provide visual representations of the distribution of events per session and events per item for the training and test datasets, respectively. These histograms illustrate the concentration of user interactions within specific ranges, with a notable peak in lower event counts.

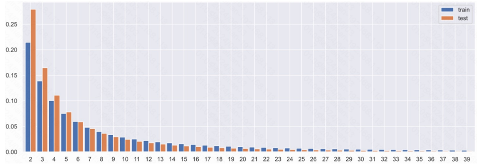

Figure 1. Events per Session Histogram (90th Percentile)

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 November 2024 doi:10.20944/preprints202411.1057.v1

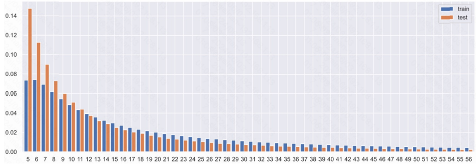

Figure 2. Events per Item Histogram (90th Percentile)

3.3. Data Processing

3.3.1. Handling Missing Values and Anomalies

Ensuring data integrity is crucial for accurate model training and evaluation. The initial assessment confirmed that there were no missing values or anomalies within the dataset. This allowed for a straightforward preprocessing phase without the need for data imputation or anomaly detection techniques.

3.3.2. Data Reformatting and Compression

The original format of the dataset, although comprehensive, presented challenges due to its multi-level structure and the memory-intensive nature of Unix timestamps. To address these issues, the timestamps were standardized to fall within a three-week period by subtracting a fixed number and converting them into numpy int32 data types. This reformatting reduced the dataset size significantly, from 11.3GB to 1.68GB, optimizing it for model training and analysis.

The reformatted dataset structure includes unique session IDs and time-ordered sequences of events within each session. Each event is associated with an article ID (aid) and includes a standardized timestamp (time) and event type (type).

3.3.3. Data Analysis

To gain insights into user behavior and item interactions, item matrix factorization was employed, transforming the anonymized item IDs into meaningful embeddings. These embeddings were visualized using t-SNE, a technique that projects high-dimensional data into a two-dimensional space, revealing clusters of related products. This analysis highlighted the inherent structure within the dataset, allowing for the identification of patterns in user interactions and preferences.

The visualizations in Figures 3 illustrates user interactions and item categories, respectively. It shows the clustering of items based on user interactions, where each dot represents an item, and proximity indicates similarity. It also depicts user activity over time, highlighting patterns in clicks, cart additions, and orders.

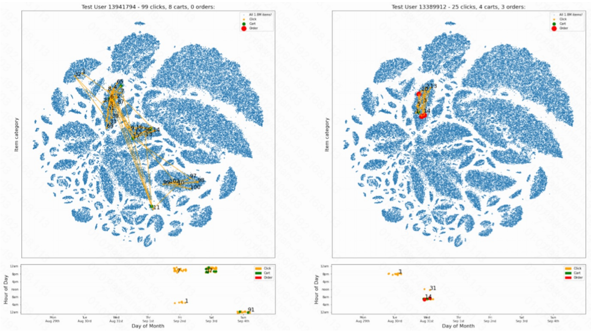

Figure 3. User Interaction and Item Category Visualization

Through detailed observation, several patterns in user behavior emerged:

- Users tend to interact with items within specific clusters, often browsing similar types of products.
- The sequence of actions, including clicks, adds to cart, and orders, follows identifiable patterns.

- Temporal factors, such as weekdays, weekends, and seasonal events, significantly influence user activity.

These insights were instrumental in developing the two-stage candidate reranker model, which aims to optimize recommendations across multiple user engagement metrics.

In summary, the extensive preprocessing and detailed analysis of the *OTTO Recommender Systems Dataset* provided a robust foundation for developing a sophisticated multi-objective recommender system. The insights gained from data analysis guided the model development process, ensuring that the system is tailored to the complex nature of e-commerce user interactions. The next section will delve into the methodology employed, detailing the candidate generation and ranking processes of our multi-objective recommender system.

4. Methodology

The methodology for developing our multi-objective recommender system involves comprehensive data processing, sophisticated feature engineering, and the implementation of a two-stage candidate reranker model (Figure 4). The system is designed to optimize multiple consumer behaviors, including clicks, cart additions, and completed orders. The most effective model combines LightGBM, XGBoost, and CatBoost rankers in an ensemble, achieving superior performance across these objectives.

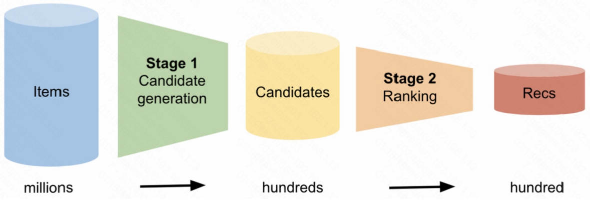

Figure 4. Flow of a two-stage candidate reranker recommendation system

4.1. Data Processing

The OTTO Recommender Systems Dataset comprises 12 million anonymized user sessions and over 220 million events, structured as follows:

- **Sessions** : Each session is uniquely identified by an ID.
- **Events** : Time-ordered sequences within each session.
- **aid** : Article ID associated with each event.
- **ts** : Unix timestamp of each event.
- **type** : Type of event (click, cart, order).

To optimize memory usage, we standardized timestamps by converting them to numpy int32 data types, reducing the training data size from 11.3GB to 1.68GB. Item embeddings were generated using the Word2Vec model, capturing semantic similarities between items. These embeddings were visualized using t-Distributed Stochastic Neighbor Embedding (t-SNE) to identify clusters of similar items.

4.2. Two-Stage Candidate Reranker

The two-stage candidate reranker model efficiently generates and ranks relevant items for each user session.

4.2.1. Stage 1: Candidate Generation

Candidate items were generated using various methods:

- **Co-visit (item co-visitation)** : Items frequently visited together, selecting the top 20 most common items associated with each prior selection.
- **Click-to-click** : Items viewed within a 12-hour timeframe.

- **Click-to-cart-or-buy** : Items clicked and then added to the cart or purchased within 24 hours.
- **Cart-to-cart** : Items added to the cart together within 24 hours.
- **Cart-to-buy** : Items where one was added to the cart and the other was purchased within 24 hours.
- **Buy-to-buy** : Items ordered together within 24 hours.
- **Similar items** : Items similar to those interacted with, identified using Word2Vec embeddings.
- **Popular items in session clusters** : Session embeddings calculated using Word2Vec embeddings, clustered using KMeans.
- **Revisit** : Reincorporate all previously examined items into the candidate pool for further consideration.
- **Click2click Graph and ProNE Application** : The hypothesis is that items clicked immediately after another are similar. To explore this, a two-column DataFrame was created, consisting of an item and the item clicked immediately after it. This DataFrame was processed using ProNE. We defined a score to measure the quality of the recall results:

$$\text{score} = 0.10 \cdot R_{\text{clicks}} + 0.30 \cdot R_{\text{carts}} + 0.60 \cdot R_{\text{orders}} \tag{1}$$

  The number of dimensions in ProNE was set to 1000, with the accuracy of candidate recall improving as the number of dimensions increased. Top-k aids were retrieved using ProNE embeddings, leveraging nearest neighbors with cosine metric. The results are shown in Table 3 .

Table 3. Recall@20 Scores for Different Actions

type	top10	top40	top60	top100
clicks	0.196203	0.5307	0.560093	0.6622
carts	0.152458	0.424199	0.50739	0.5113
orders	0.16003	0.481797	0.7059	0.713684
total	0.161375	0.469408	0.6317663	0.6478204

These methods generated a diverse pool of candidate items for each session.

4.2.2. Stage 2: Ranking

The candidate items were ranked based on their predicted relevance, involving extensive feature engineering and model training:

- **Session Features** :
  - Type count per session (clicks, carts, orders).
  - Number of unique types per session.
  - Mean type per session (clicks=1, carts=2, orders=3).
- **Aid Features** :
  - Type count within all sessions and test sessions.
  - Click-to-cart rate, cart-to-order rate, click-to-order rate within sessions.
- **Session x Aid Features (Figure 5)** :
  - Co-visitation count, rate, time-weighted count, and rate.
  - Similarity features (cosine similarity between candidate items and session items using Word2Vec and ProNE embeddings).

These features were computed per batch on a 32GB 4*V100 GPU using RAPIDS for efficient processing.

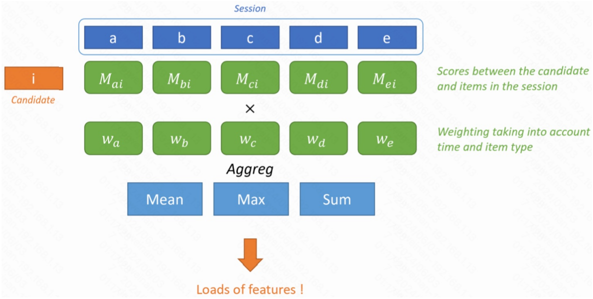

Figure 5. Session and candidate item features aggregation

4.3. *Model Training and Ensemble*

Three gradient boosting models were trained individually and then combined into an ensemble model:

4.3.1. LightGBM Ranker (Lambda Rank)

LightGBM is optimized for speed and efficiency, making it suitable for large-scale data. The lambda rank objective function is specifically designed for ranking tasks, aiming to optimize ranking performance by minimizing pairwise loss between examples:

$$\mathcal{L} = -\sum_{i,j} I_{ij} \log(\sigma(s_i - s_j)) \tag{2}$$

where I_{ij} is an indicator function that equals 1 if item i is correctly ranked higher than item j, and σ is the sigmoid function. This objective function effectively improves the ranking quality by focusing on the correct pairwise ordering of items.

4.3.2. XGBoost Ranker (Rank:pairwise)

XGBoost is a powerful machine learning algorithm that excels in ranking tasks. The rank:pairwise objective function in XGBoost is designed to improve the pairwise ranking of items by considering their relative importance, and it operates similarly to the lambda rank objective function in LightGBM. The objective function is defined as:

$$\mathcal{L} = -\sum_{i,j} I_{ij} \log(\sigma(s_i - s_j)) \tag{3}$$

where I_{ij} is an indicator function that equals 1 if item i is correctly ranked higher than item j, and σ is the sigmoid function. Both LightGBM and XGBoost utilize this pairwise loss minimization approach to enhance their ranking effectiveness.

4.3.3. CatBoost Ranker (PairLogit)

CatBoost is designed for handling categorical features efficiently. The PairLogit loss function is well-suited for ranking tasks:

$$\mathcal{L} = \sum_{i,j} I_{ij} \log(1 + \exp(-y_{ij}(s_i - s_j))) \tag{4}$$

where y_{ij} is the target value indicating the preference between items i and j.

4.4. *Negative Sampling*

Effective ranking of machine learning models requires a balanced mix of positive and negative samples. Positive samples represent real events in sessions, labeled as 1 (user clicks, adds to cart, or orders). Negative samples are labeled as 0, representing instances where a project was shown but not interacted with.

Different negative sample ratios were set for clicks (0.25), carts (0.5), and orders (0.2), ensuring a balanced dataset suitable for training. This approach allowed the ranking algorithms to prioritize projects with and without user engagement effectively.

4.5. Ensemble Model

The ensemble model combines the predictions of LightGBM, XGBoost, and CatBoost, each contributing to the final recommendation score. The weights for each model in the ensemble were determined using Optuna to maximize the recall@20 metric. The final prediction score for an item i in session s is calculated as follows:

$$\text{score}(s,i) = w_L \cdot \text{pred}_L(s,i) + w_X \cdot \text{pred}_X(s,i) + w_C \cdot \text{pred}_C(s,i) \tag{5}$$

where w_{L}, w_{X}, and w_{C} are the optimized weights for LightGBM, XGBoost, and CatBoost predictions, respectively.

4.6. Evaluation Metric

The effectiveness of the proposed multi-objective recommender system is assessed using the Recall@20 metric for each action type: clicks, carts, and orders. To obtain a comprehensive evaluation score that reflects the relative importance of each action type, the recall values are combined in a weighted average.

The evaluation score score is formulated as follows:

$$\text{score} = 0.10 \cdot R_{\text{clicks}} + 0.30 \cdot R_{\text{carts}} + 0.60 \cdot R_{\text{orders}} \tag{6}$$

In this equation, R_{clicks}, R_{carts}, and R_{orders} represent the Recall@20 values for clicks, carts, and orders, respectively.

The recall for each action type R_{type} is rigorously defined by the following expression:

$$R_{\text{type}} = \frac{1}{N} \sum_{i=1}^{N} \frac{\left|\{\text{pred\_aids}\}_{i,\text{type}} \cap \{\text{ground\_truth\_aids}\}_{i,\text{type}}\right|}{\min(20, \left|\{\text{ground\_truth\_aids}\}_{i,\text{type}}\right|)} \tag{7}$$

where:

- N denotes the total number of sessions in the test set.
- $\{\text{predicted\_aids}\}_{i,\text{type}}$ represents the set of predicted article IDs (aids) for session i and action type type.
- $\{\text{ground\_truth\_aids}\}_{i,\text{type}}$ corresponds to the set of actual article IDs (ground truth) for session i and action type type.
- The denominator $\min(20, \left|\{\text{ground\_truth\_aids}\}_{i,\text{type}}\right|)$ normalizes the recall, ensuring that sessions with fewer than 20 ground truth article IDs are appropriately accounted for.

To compute the Recall@20 metric, the predicted article IDs for each session-type (e.g., each row in the submission file) are truncated after the first 20 predictions. This truncation ensures that the evaluation remains focused on the top 20 recommendations, which is a realistic representation of a user's interaction with a recommendation system. The weighted average of recall values across different action types ensures a holistic evaluation of the recommender system, reflecting its ability to perform well across various user behaviors.

By employing this evaluation metric, we provide a robust and comprehensive assessment of the multi-objective recommender system, ensuring its efficacy in optimizing multiple consumer behaviors concurrently.

5. Experiments Results

To evaluate the performance of our multi-objective recommender system, we conducted experiments using the OTTO Recommender Systems Dataset. We measured the effectiveness of the individual models (LightGBM Ranker, XGBoost Ranker, and CatBoost Ranker) as well as their ensemble combination. The results are summarized in the following Table 4 .

Table 4. Performance of Different Models

Model	CV Score	Public Score	Private Score
LightGBM (Lambda)	0.584491	0.59325	0.59449
XGBoost (Pairwise)	0.583787	0.59467	0.5956
CatBoost (PairLogit)	0.58476	0.59552	0.59661
Ensemble	**0.585176**	**0.59572**	**0.59678**

The ensemble model combining LightGBM, XGBoost, and CatBoost rankers demonstrated superior performance across all metrics, achieving the highest CV, public, and private scores. This confirms the effectiveness of the ensemble approach in optimizing multiple consumer behaviors simultaneously, enhancing the overall recommendation system performance.

6. Conclusions

In conclusion, this study presents a significant advancement in e-commerce by developing a multi-objective recommender system that simultaneously optimizes multiple consumer behaviors, including clicks, cart additions, and completed orders. Using the extensive OTTO Recommender Systems Dataset, the proposed two-stage candidate reranker model efficiently generates and ranks potential items. The ensemble approach, combining LightGBM, XGBoost, and CatBoost Rankers, outperformed individual models, achieving higher recall rates and demonstrating the effectiveness of the multi-objective strategy. This system enhances user satisfaction and engagement while increasing conversion rates for retailers. Future work will focus on refining the model, exploring additional features, and ensuring fairness, transparency, and user privacy in recommendations.

References

1. Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; Riedl, J. Grouplens: An open architecture for collaborative filtering of netnews. Proceedings of the 1994 ACM conference on Computer supported cooperative work, 1994, pp. 175–186.
2. Pazzani, M.J.; Billsus, D. Content-based recommendation systems. In *The adaptive web: methods and strategies of web personalization* ; Springer, 2007; pp. 325–341.
3. Javanmardi, A.; Liu, M.; He, C.; Hsiang, S.M.; Abbasian-Hosseini, A. Improving Construction Meeting Effectiveness: Trade-Offs between Reactive and Proactive Site-Level Planning Discussions. *Journal of Management in Engineering* **2024** , *40* , 04024029.
4. Burke, R. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction* **2002** , *12* , 331–370.
5. Li, G.; Liu, C.; Wu, L.; Xiao, W. A mixing algorithm of ACO and ABC for solving path planning of mobile robot. *Applied Soft Computing* **2023** , *148* , 110868.
6. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009** , *42* , 30–37.
7. Hu, Y.; Koren, Y.; Volinsky, C. Collaborative filtering for implicit feedback datasets. 2008 Eighth IEEE international conference on data mining. Ieee, 2008, pp. 263–272.
8. Zhang, S.; Yao, L.; Sun, A.; Tay, Y. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)* **2019** , *52* , 1–38.
9. He, C.; Liu, M.; Hsiang, S.M.; Pierce, N. Synthesizing Ontology and Graph Neural Network to Unveil the Implicit Rules for US Bridge Preservation Decisions. *Journal of Management in Engineering* **2024** , *40* , 04024007.
10. Svore, K.M.; Volkovs, M.N.; Burges, C.J. Learning to rank with multiple objective functions. Proceedings of the 20th international conference on World wide web, 2011, pp. 367–376.

11. Lin, Z.; Xu, F. Simulation of Robot Automatic Control Model Based on Artificial Intelligence Algorithm. 2023 2nd International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS). IEEE, 2023, pp. 535–539.
12. He, C.; Liu, M.; Alves, T.d.C.; Scala, N.M.; Hsiang, S.M. Prioritizing collaborative scheduling practices based on their impact on project performance. *Construction management and economics* **2022** , *40* , 618–637.
13. Zhang, B.; Xiao, J.; Yan, H.; Yang, L.; Qu, P. Review of NLP Applications in the Field of Text Sentiment Analysis. *Journal of Industrial Engineering and Applied Science* **2024** , *2* , 28–34.
14. Li, L.; Chu, W.; Langford, J.; Schapire, R. Contextual-bandit approach to personalized news article recommendation, 2012. US Patent App. 12/836,188.
15. Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W.L.; Leskovec, J. Graph convolutional neural networks for web-scale recommender systems. Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 974–983.
16. Javanmardi, A.; He, C.; Hsiang, S.M.; Abbasian-Hosseini, S.A.; Liu, M. Enhancing construction project workflow reliability through observe–plan–do–check–react cycle: A bridge project case study. *Buildings* **2023** , *13* , 2379.
17. Sun, Y.; Ortiz, J. Rapid Review of Generative AI in Smart Medical Applications. *arXiv preprint arXiv:2406.06627* **2024** .
18. Pan, W.; Xiang, E.; Liu, N.; Yang, Q. Transfer learning in collaborative filtering for sparsity reduction. Proceedings of the AAAI conference on artificial intelligence, 2010, Vol. 24, pp. 230–235.
19. Rokach, L. Ensemble-based classifiers. *Artificial intelligence review* **2010** , *33* , 1–39.
20. Liu, S.; Li, X.; He, C. Study on dynamic influence of passenger flow on intelligent bus travel service model. *Transport* **2021** , *36* , 25–37.
21. Zhang, Y.; Chen, X.; others. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval* **2020** , *14* , 1–101.
22. Nikolaenko, V.; Ioannidis, S.; Weinsberg, U.; Joye, M.; Taft, N.; Boneh, D. Privacy-preserving matrix factorization. Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 2013, pp. 801–812.
23. Yan, H.; Xiao, J.; Zhang, B.; Yang, L.; Qu, P. The Application of Natural Language Processing Technology in the Era of Big Data. *Journal of Industrial Engineering and Applied Science* **2024** , *2* , 20–27.
24. Burke, R. Multisided fairness for recommendation. *arXiv preprint arXiv:1707.00093* **2017** .
25. Chen, H.; Yang, Y.; Shao, C. Multi-task learning for data-efficient spatiotemporal modeling of tool surface progression in ultrasonic metal welding. *Journal of Manufacturing Systems* **2021** , *58* , 306–315.
26. Kunaver, M.; Pož rl, T. Diversity in recommender systems–A survey. *Knowledge-based systems* **2017** , *123* , 154–162.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.