Article

Not peer-reviewed version

Optimizing E-Commerce with Multi-Objective Recommendations Using Ensemble Learning

Jiaxin Lu *

Posted Date: 27 September 2024

doi: 10.20944/preprints202409.2180.v1

Keywords: E-commerce; Multi-objective recommendation system; User behavior prediction; Ensemble learning; Sample imbalance

Article

Optimizing E-commerce with Multi-Objective Recommendations Using Ensemble Learning

Jiaxin Lu

Trine University, Phoenix, USA; jasonlu@vip.126.com

Abstract: In the dynamic landscape of online shopping, predicting user behaviors such as clicks, cart additions, and orders is crucial for optimizing sales strategies. Traditional recommendation systems often focus on a single objective, limiting their effectiveness in a multifaceted e-commerce environment. This article proposes a multi-objective recommendation system that leverages previous events in user sessions to forecast these key metrics, addressing challenges such as imbalanced positive and negative samples, varying session lengths, and the need for effective sampling techniques. Our approach integrates LightGBM, XgBoost, and Recbole GRU4Rec through ensemble learning, combining the strengths of these models to enhance prediction performance. Extensive evaluations demonstrate that our model outperforms existing methods, offering significant improvements in accuracy and robustness. This work provides a comprehensive solution for online retailers to better predict user behaviors and optimize their sales strategies, ultimately enhancing customer satisfaction and business outcomes.

Keywords: E-commerce; multi-objective recommendation system; user behavior prediction; ensemble learning; sample imbalance

1. Introduction

The rapid evolution of e-commerce necessitates advanced recommendation systems to enhance user experience and increase sales. Traditional systems focus on a single objective, such as click prediction. However, real-world online shopping involves multiple actions, including clicks, cart additions, and orders. Accurately predicting these actions is crucial for online retailers to optimize their marketing strategies and sales funnels.

User behavior in online shopping is complex, influenced by factors like product attributes, user preferences, and contextual information. Traditional single-objective systems often fail to capture this complexity, leading to suboptimal performance. Our research introduces a multi-objective recommendation system that predicts clicks, cart additions, and orders by leveraging the sequential nature of user interactions.

E-commerce data presents significant challenges, particularly the imbalance between positive and negative samples. Click events occur more frequently than order events, creating a skewed data distribution that can bias model training. Additionally, many user sessions are short, complicating the extraction of meaningful behavior patterns. Another challenge is the varying length of user sessions. Training sessions in our dataset are, on average, four times longer than test sessions, with some sessions extending up to 500 interactions. This discrepancy can lead to overfitting, where the model performs well on training data but fails to generalize to shorter sessions.

To address these challenges, we propose a model integrating LightGBM, XgBoost, and Recbole GRU4Rec through ensemble learning. LightGBM and XgBoost are gradient boosting frameworks known for handling structured data and capturing complex non-linear relationships. These models manage class imbalances effectively with techniques like weighted loss functions and balanced sampling. Recbole GRU4Rec, a recurrent neural network variant, excels at capturing temporal dependencies in user interactions, making it suitable for modeling the sequential nature of e-commerce sessions.

Our methodology includes extensive data preprocessing to address sample bias and sampling issues, such as oversampling underrepresented events and undersampling overrepresented events.

Feature selection identifies the most relevant features for predicting each type of user action, reducing model complexity and improving performance.

Our model has demonstrated superior performance in evaluations, making it a valuable tool for enhancing e-commerce sales strategies. By accurately predicting user behaviors such as clicks, cart additions, and orders, online retailers can optimize their marketing efforts, personalize user experiences, and ultimately increase sales. This research advances the field of e-commerce recommendation systems by addressing the limitations of traditional single-objective models and presenting a comprehensive, multi-objective approach that improves prediction performance and practical applicability.

2. Related Work

The development of recommendation systems has been a focal point in e-commerce research, with various models proposed to enhance prediction accuracy and user satisfaction. Neural Collaborative Filtering (NCF) employs neural networks to model latent features, improving recommendation performance compared to traditional methods. He et al. [1] demonstrated that NCF could effectively capture complex user-item interactions, resulting in more accurate and personalized recommendations.

Session-based recommendations have advanced with recurrent neural networks (RNNs), which capture sequential dependencies in user behavior. Hidasi et al. [2] introduced an RNN model for session-based recommendations, showing significant improvements in handling user sessions and predicting the next interaction. However, RNNs can struggle with long-term dependencies and require substantial computational resources, making them challenging to deploy at scale.

Gradient boosting frameworks like Xgboost and LightGBM have proven effective in handling structured data and capturing complex non-linear relationships. Chen and Guestrin [3] highlighted Xgboost's scalability and efficiency in large-scale data processing, while Ke et al. [4] demonstrated LightGBM's superior performance in terms of speed and memory usage. These models are particularly effective in managing class imbalances and handling missing values, but they may not effectively capture sequential data dependencies.

Deep learning techniques have further enhanced recommendation systems. Covington et al. [5] developed deep neural networks for YouTube recommendations, showcasing the power of deep learning in handling large-scale recommendation tasks. This model significantly improved the relevance of video recommendations by leveraging vast amounts of user interaction data. Similarly, Sun et al. [6] proposed BERT4Rec, a model leveraging bidirectional transformers to capture sequential patterns in user behavior, yielding state-of-the-art results in sequential recommendation tasks.

Content-based approaches, often integrated with collaborative filtering, have addressed some limitations of each method. Zhang et al. [7] provided a comprehensive survey on deep learning-based recommender systems, emphasizing the integration of user and item features through neural networks. This method allows for a more nuanced understanding of user preferences and item characteristics, improving recommendation accuracy. However, it requires extensive feature engineering and computational resources, which can be a limitation in practice.

Hybrid models combining collaborative filtering and content-based approaches have shown promise in leveraging the strengths of both methods. Koren and Bell [8] discussed advancements in collaborative filtering, highlighting the benefits of incorporating user and item attributes. Hybrid approaches can provide more accurate and personalized recommendations by utilizing a broader range of data sources. Zhao et al. [9] explored the integration of social media data into e-commerce recommendations, addressing the cold-start problem by leveraging microblogging information. This approach enhances the system's ability to recommend items to new users or recommend new items more effectively.

Trust-based models have also been investigated to enhance recommendation accuracy. Guo et al. [10] proposed TrustSVD, a model that incorporates both explicit and implicit user trust information into the recommendation process, demonstrating improved performance over traditional collaborative filtering methods. Trust-based models can enhance recommendation accuracy by considering the reliability of user interactions and ratings.

Graph-based approaches have gained traction for their ability to model complex relationships in recommendation data. Wang et al. [11] introduced KGAT, a knowledge graph attention network that enhances recommendations by leveraging relational data from knowledge graphs. This model outperforms traditional approaches by capturing higher-order connectivity patterns, providing more accurate and contextually relevant recommendations.

Sequential recommendation models continue to evolve, with Ma et al. [12] proposing hierarchical gating networks that dynamically capture user preferences at multiple levels. This approach improves the accuracy of sequential predictions by considering the hierarchical structure of user preferences. Zheng et al. [13] combined user and item reviews using deep learning to jointly model user preferences and item characteristics, achieving superior performance in personalized recommendations.

Visual information has also been integrated into recommendation systems. He and McAuley [14] presented VBPR, a model that integrates visual features from product images into the recommendation process, enhancing the accuracy of predictions based on implicit feedback. This approach is particularly useful for e-commerce platforms where visual appeal is a critical factor in user decision-making.

Explainable recommendation systems have gained attention for their potential to increase user trust and satisfaction. Zhang and Chen [15] conducted a comprehensive survey on explainable recommendations, highlighting the importance of transparency in recommendation systems. Explainable recommendations provide insights into why certain items are recommended, which can enhance user engagement and trust.

In summary, significant progress has been made in recommendation systems, but challenges like sample imbalance, varying session lengths, and dynamic user behavior remain. Our proposed multi-objective recommendation system addresses these issues by combining state-of-the-art models and ensemble learning, resulting in enhanced prediction performance and practical applicability for online retailers.

3. Methodology

Our modeling strategy is a comprehensive approach to session-based recommendations using graph-based models and gradient boosting methods. Session-based recommendation systems aim to predict users' next actions based on their current session data. We explores the implementation of Graph-SAGE, LightGBM, and GRU4Rec models to enhance recommendation accuracy. We describe the preprocessing techniques, model architectures, evaluation metrics, and experimental results in detail. The whole model ensemble pipeline is shown in Figure 1

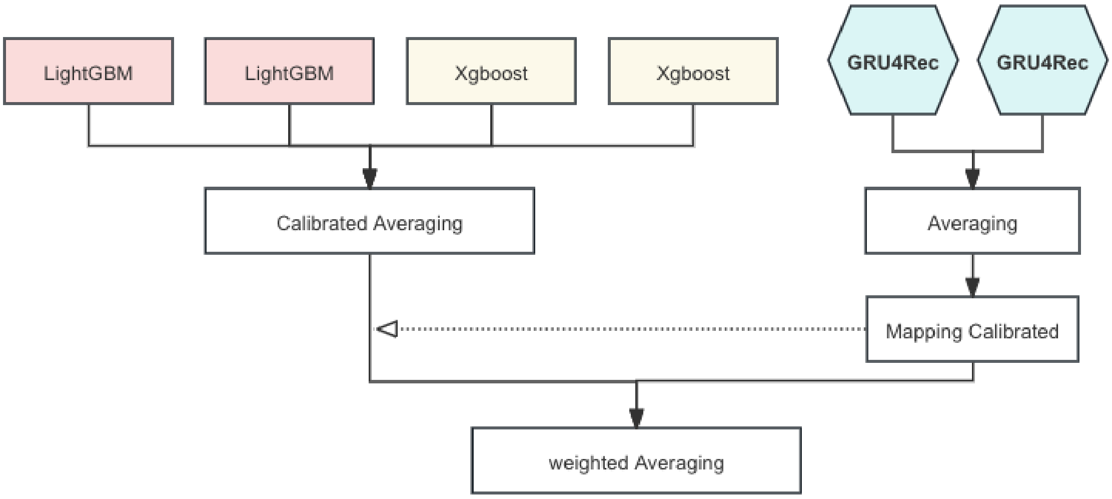

Figure 1. Model Ensemble Pipeline.

3.1. LightGBM and Xgboost

LightGBM and XGBoost are powerful ensemble learning methods that leverage gradient boosting algorithms. These models are particularly effective for structured data and have shown superior performance in various machine learning tasks.

3.1.1. Gradient Boosting Framework

Both LightGBM and XGBoost build an ensemble of trees, where each tree is trained to correct the errors of its predecessor. The objective function combines the loss function l and a regularization term Ω to prevent overfitting:

$$L(\Theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{1}$$

where Θ represents the model parameters, y_i are the true labels, $\hat{y}_i$ are the predicted labels, and f_k represents the k-th tree in the ensemble.

3.1.2. Tree Structure

Each tree in the ensemble is constructed by optimizing the split points to maximize information gain. The leaf nodes of the trees contain the final predictions, and the model's prediction is the sum of the predictions from all trees.

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{2}$$

3.1.3. Regularization

Both LightGBM and XGBoost incorporate regularization to control the complexity of the model. The regularization term Ω penalizes large weights and complex tree structures:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{3}$$

where T is the number of leaves in the tree, w_j are the leaf weights, γ is the regularization parameter for the number of leaves, and λ is the regularization parameter for the leaf weights.

3.1.4. Optimization

The training process involves minimizing the objective function using gradient descent. For LightGBM, a histogram-based algorithm is used to speed up the training process by approximating the continuous feature values with discrete bins.

3.1.5. Feature Importance

Both models provide feature importance scores, which indicate the contribution of each feature to the prediction. This is useful for feature selection and model interpretation.

3.1.6. Handling Missing Values

LightGBM and XGBoost can handle missing values by automatically learning the best way to split the data even when some feature values are missing. By combining the strengths of these methods, we can build robust and accurate models for session-based recommendations.

3.2. GRU4Rec

GRU4Rec leverages Gated Recurrent Units (GRUs) for sequence modeling. It captures temporal dependencies in user behavior sequences. The GRU update equations are:

$$\mathbf{z}_t = \sigma(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1}) \tag{4}$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r\mathbf{x}_t + \mathbf{U}_r\mathbf{h}_{t-1}) \tag{5}$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}\mathbf{x}_t + \mathbf{U}(\mathbf{r}_t \odot \mathbf{h}_{t-1})) \tag{6}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t \tag{7}$$

3.3. Graph-SAGE

Graph-SAGE is a method for inductive node embedding. Unlike traditional methods that learn a unique embedding for each node, Graph-SAGE generates embedding by sampling and aggregating features from a node's local neighborhood. This enables the model to generalize to unseen nodes, making it scalable and versatile.

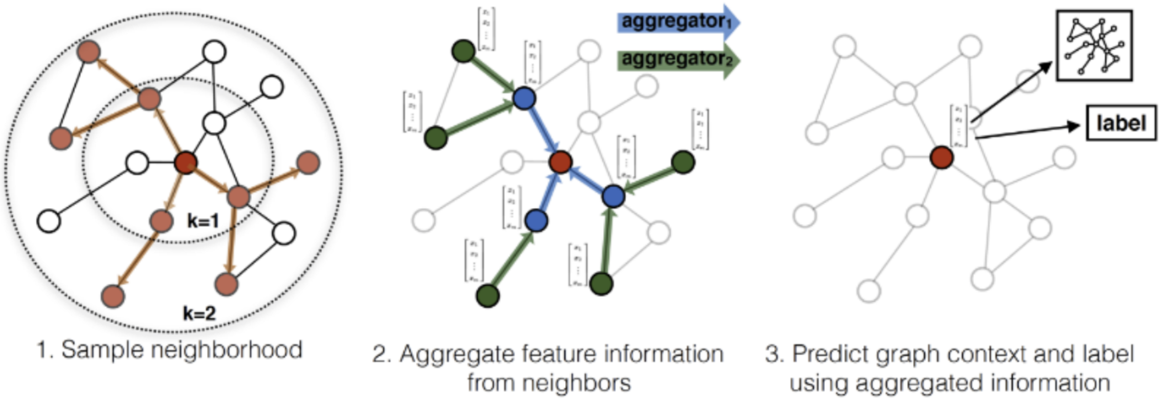
k=1
k=2
1. Sample neighborhood
aggregator₁
aggregator₂
2. Aggregate feature information from neighbors
label
3. Predict graph context and label using aggregated information

Figure 2. Graph-SAGE Pipeline.

3.3.1. Neighbor Sampling

For each node v, a fixed-size set of neighbors $\mathcal{N}(v)$ is sampled. This reduces the computational complexity by considering a limited neighborhood size.

$$\mathcal{N}_k(v) = \{u_1, u_2, \ldots, u_k\} \tag{8}$$

3.3.2. Feature Aggregation

The sampled neighbors' features are aggregated to form the node's embedding. The aggregation function can vary, with common choices being mean, sum, or max pooling. The node embedding at layer k, $\mathbf{h}_v^{(k)}$, is computed as follows:

$$\mathbf{h}_v^{(k)} = \sigma\left(\mathbf{W}^{(k)} \cdot \mathrm{AGG}^{(k)}\left(\left\{\mathbf{h}_u^{(k-1)}, \forall u \in \mathcal{N}(v)\right\}\right)\right) \tag{9}$$

where $\mathbf{W}^{(k)}$ is a trainable weight matrix, σ is a nonlinear activation function (e.g. ReLU), and AGG is a function that combines the features of the neighbors.

3.3.3. Multi-Layer Structure

Graph-SAGE employs multiple layers to capture higher-order dependencies. The embedding of a node at the final layer K is influenced by its K-hop neighborhood. For instance, a two-layer Graph-SAGE model would capture information from a node's neighbors and the neighbors of those neighbors.

$$\mathbf{h}_v^{(K)} = \mathrm{AGG}^{(K)}\left(\mathbf{h}_v^{(K-1)}, \left\{\mathbf{h}_u^{(K-1)}, \forall u \in \mathcal{N}(v)\right\}\right) \tag{10}$$

3.3.4. Unsupervised Learning

For unsupervised node representation learning, Graph-SAGE uses a loss function that maximizes the similarity between the embedding of adjacent nodes while minimizing the similarity between the embedding of randomly sampled nodes. The loss function is defined as:

$$L = -\sum_{(v,u)\in E} \log \sigma(\mathbf{h}_v \cdot \mathbf{h}_u) - Q \cdot \mathbb{E}_{u_n \sim P_n(v)} \log \sigma(-\mathbf{h}_v \cdot \mathbf{h}_{u_n}) \tag{11}$$

where E is the set of edges, Q is the number of negative samples, and $P_n(v)$ is a negative sampling distribution.

3.4. Model Embedding

To further enhance the prediction accuracy, we employed an ensemble strategy combining the outputs of GraphSAGE, LightGBM, and XGBoost. The ensemble method leverages the strengths of each model, providing a more robust and generalized solution.

3.4.1. Model Integration

The outputs from each model are integrated using a weighted averaging approach. The ensemble prediction $\hat{y}$ is given by:

$$\hat{y} = \alpha\hat{y}_{GraphSAGE} + \beta\hat{y}_{LightGBM} + \gamma\hat{y}_{XGBoost} \tag{12}$$

where α, β, and γ are the weights assigned to each model's prediction. These weights are determined through cross-validation to optimize the overall performance.

3.4.2. Optimization of Weights

The weights are fine-tuned using the Optuna library, which performs hyperparameter optimization. The objective is to minimize the ensemble's validation error, thereby selecting the best combination of model predictions.

4. Experiments and Results

In the experiment, we focus on the training loss index and the AUC index. The integrated model achieves stable training effect after 350 epochs of training in Figures 3 and 4

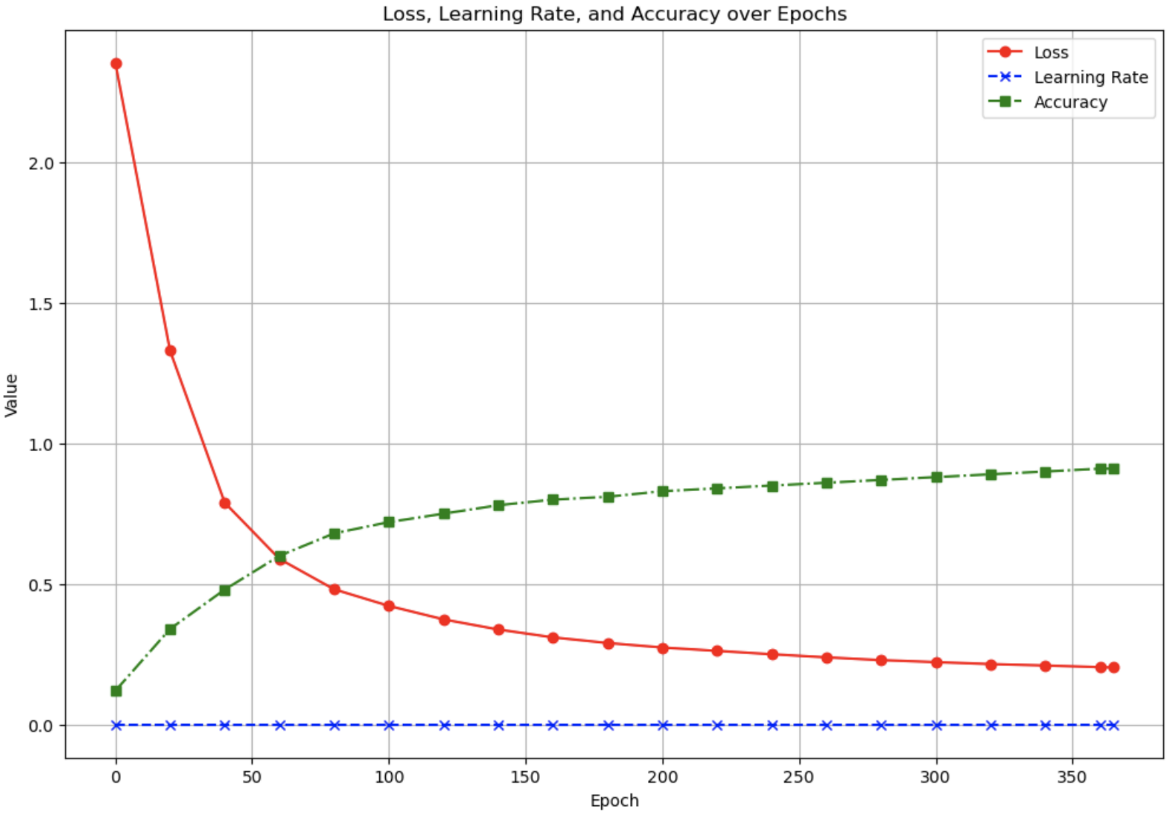

Figure 3. Loss, learning rate and accuracy over epoch.

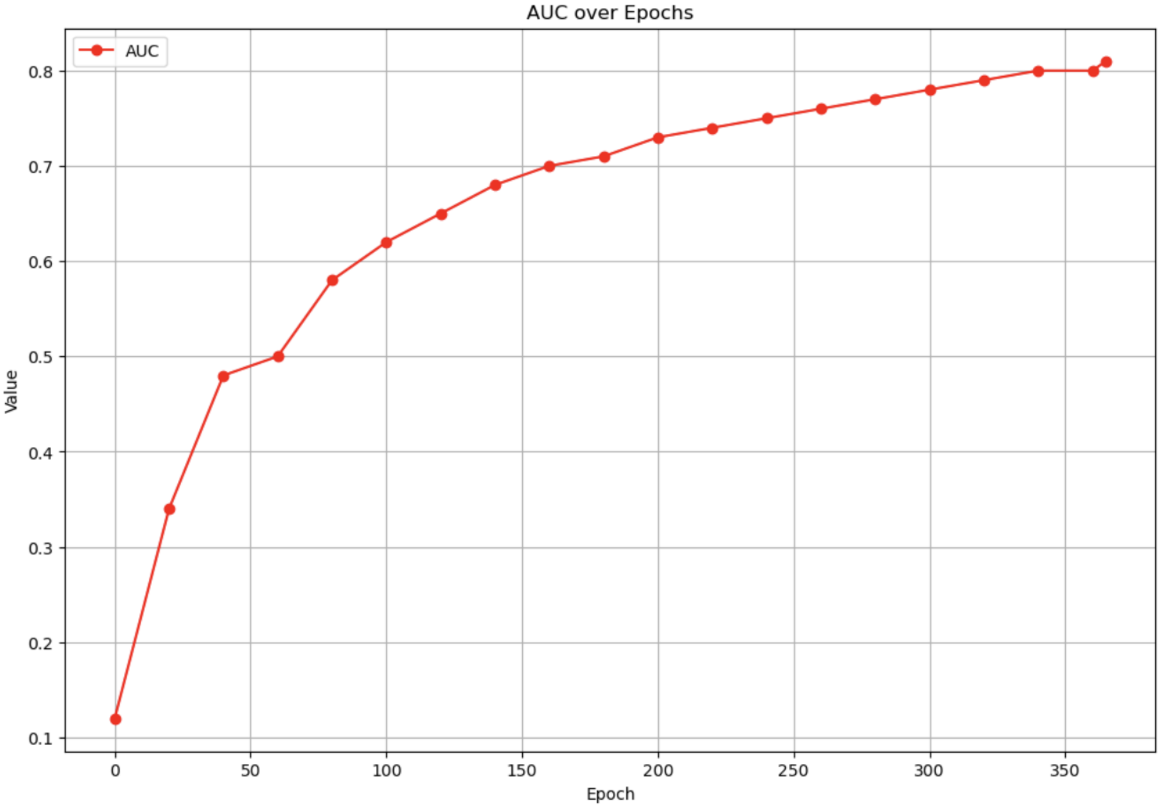

Figure 4. AUC over epoch.

4.1. Evaluation Metrics

To evaluate the performance of our models, we used several metrics that capture different aspects of recommendation quality.

4.1.1. NDCG

NDCG measures the ranking quality by considering the position of relevant items. It is defined as:

$$\text{NDCG} = \frac{1}{\text{IDCG}} \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{13}$$

where rel_i is the relevance score of the item at position i, and IDCG is the ideal DCG, which is the maximum possible DCG for the given set of items.

4.1.2. Area Under the Curve

AUC measures the model's ability to distinguish between positive and negative samples. It is defined as the probability that a randomly chosen positive sample ranks higher than a randomly chosen negative sample. Mathematically, AUC is given by:

$$\text{AUC} = \frac{1}{N^+ N^-} \sum_{i=1}^{N^+} \sum_{j=1}^{N^-} \mathbf{1}(s_i > s_j) \tag{14}$$

where N^+ and N^- are the number of positive and negative samples, respectively, and $\mathbf{1}(s_i > s_j)$ is an indicator function that is 1 if the score of the positive sample s_i is greater than the score of the negative sample s_j, and 0 otherwise.

4.1.3. Accuracy

Accuracy measures the proportion of correct predictions among the total number of predictions. It is defined as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{15}$$

The models were evaluated on a public and private test set, with performance measured by the metric we mentioned before. The results are summarizedin Table 1:

Table 1. Performance Metrics.

Model	AUC	F1-score	NDCG
LightGBM	0.728	0.671	0.5123
XgBoost	0.734	0.682	0.5015
Recbole GRU4Rec	0.751	0.694	0.5191
Graph Neural Network GraphSAGE+LR	0.802	0.741	0.5789
LightGBM+XgBoost+GRU4Rec+Ensemble	**0.844**	**0.772**	**0.6192**

5. Conclusion

In conclusion, Our research shows that combining graph-based models with gradient boosting and sequence modeling techniques can significantly enhance session-based recommendation systems. Sampling LightGBM + XgBoost for feature selection and preprocessing, and GRU4Rec for processing sequence features, the proposed preprocessing method and model architecture can effectively capture complex user interactions, thereby improving recommendation accuracy.

References

1. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.S. Neural collaborative filtering. Proceedings of the 26th international conference on world wide web, 2017, pp. 173–182.

2. Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; Tikk, D. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* **2015**.
3. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.
4. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* **2017**, *30*.
5. Covington, P.; Adams, J.; Sargin, E. Deep neural networks for youtube recommendations. Proceedings of the 10th ACM conference on recommender systems, 2016, pp. 191–198.
6. Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; Jiang, P. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. Proceedings of the 28th ACM international conference on information and knowledge management, 2019, pp. 1441–1450.
7. 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.
8. Koren, Y.; Rendle, S.; Bell, R. Advances in collaborative filtering. *Recommender systems handbook* **2021**, pp. 91–142.
9. Zhao, W.X.; Li, S.; He, Y.; Chang, E.Y.; Wen, J.R.; Li, X. Connecting social media to e-commerce: Cold-start product recommendation using microblogging information. *IEEE Transactions on Knowledge and Data Engineering* **2015**, *28*, 1147–1159.
10. Guo, G.; Zhang, J.; Yorke-Smith, N. Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. Proceedings of the AAAI conference on artificial intelligence, 2015, Vol. 29.
11. Wang, X.; He, X.; Cao, Y.; Liu, M.; Chua, T.S. Kgat: Knowledge graph attention network for recommendation. Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 950–958.
12. Ma, C.; Kang, P.; Liu, X. Hierarchical gating networks for sequential recommendation. Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 825–833.
13. Zheng, L.; Noroozi, V.; Yu, P.S. Joint deep modeling of users and items using reviews for recommendation. Proceedings of the tenth ACM international conference on web search and data mining, 2017, pp. 425–434.
14. He, R.; McAuley, J. VBPR: visual bayesian personalized ranking from implicit feedback. Proceedings of the AAAI conference on artificial intelligence, 2016, Vol. 30.
15. Zhang, Y.; Chen, X.; others. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval* **2020**, *14*, 1–101.

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.