Harnessing Multimodal Data and Mult-Recall Strategies for Enhanced Product Recommendation in E-Commerce

Siyue Li *

Posted Date: 30 September 2024

doi: 10.20944/preprints202409.2417.v1

Keywords: product recommendation; mult-recall; LightGBM ranker; feature engineering; machine learning

 doi:10.20944/preprints202409.2417.v1

Article

Harnessing Multimodal Data and Mult-Recall Strategies for Enhanced Product Recommendation in E-Commerce

Siyue Li
Northeastern University, Santa Clara, USA; janelovelee2020@gmail.com

Abstract: Recommending products effectively based on historical transaction data and metadata is crucial for improving user satisfaction and increasing sales. This study presents a recommendation system evaluated using Mean Average Precision (MAP@12), leveraging a diverse dataset that includes apparel types, customer demographics, product descriptions, and images. Our approach integrates Mult-Recall methods and LightGBM Ranker, utilizing various recall strategies and advanced feature engineering techniques to enhance recommendation accuracy. By combining these advanced machine learning models and ensemble techniques, our proposed solution outperforms existing methods, providing a robust and efficient recommendation system tailored to diverse customer needs and product characteristics. This integrated approach not only addresses the challenges of data sparsity and cold start but also adapts to the dynamic nature of user preferences and market trends. The results demonstrate significant improvements in both recommendation precision and user satisfaction, highlighting the potential of our approach for real-world e-commerce applications.

Keywords: product recommendation; mult-recall; LightGBM ranker; feature engineering; machine learning

1. Introduction

The development of effective product recommendation systems is a critical task for online retailers aiming to enhance user experience and boost sales. This challenge involves accurately predicting user preferences using historical transaction data and various metadata associated with customers and products. Traditional recommendation methods, such as collaborative filtering and content-based filtering, often struggle with scalability and sparsity issues, particularly when dealing with large and diverse datasets.

Our study focuses on creating a recommendation system that leverages both transactional and metadata to provide personalized product suggestions. The dataset includes detailed information on apparel types, customer demographics, product descriptions, and images. A significant insight from our data analysis is the prevalence of women's apparel and the impact of seasonal trends on pricing. High-priced items like bags exhibit distinct purchasing patterns, highlighting the importance of incorporating trend-based features into the recommendation model.

To address these complexities, we propose a sophisticated model that combines Mult-Recall methods with LightGBM Ranker. Mult-Recall methods involve using various recall strategies to generate a comprehensive set of candidate items for each user. These strategies include popular item recall, collaborative filtering-based recall, and content-based recall methods utilizing textual and visual features. The LightGBM Ranker is then employed to rank these candidates, leveraging advanced gradient boosting techniques to optimize the ranking based on multiple features derived from the recall stage.

The Mult-Recall approach ensures that the recommendation system captures a wide range of potential user preferences by integrating different recall methods. Each method brings unique strengths, such as capturing long-term user interests or utilizing the latest trends in product descriptions and images. By combining these methods, the system can generate a diverse set of high-quality candidate items for each user.

The LightGBM Ranker further refines these recommendations by considering various features, including user-specific behaviors, product attributes, and contextual interactions. LightGBM's ability

 doi:10.20944/preprints202409.2417.v1

to handle large-scale data efficiently and its robustness to overfitting make it an ideal choice for this task. The ensemble of multiple models during the ranking stage ensures that the final recommendations are both precise and relevant, ultimately enhancing user satisfaction and driving sales.

In summary, our proposed solution integrates advanced machine learning techniques to build a robust and efficient product recommendation system. By leveraging the strengths of Mult-Recall methods and LightGBM Ranker, our approach provides accurate and personalized product suggestions, addressing the unique challenges of diverse customer needs and dynamic product characteristics.

2. Related Work

The field of product recommendation systems has evolved significantly over the past decade, driven by the increased availability of data and the development of advanced machine learning algorithms. Traditional collaborative filtering methods, such as User-Based and Item-Based Collaborative Filtering, have been extensively used but often face scalability and sparsity issues in large datasets. To address these limitations, Nguyen et al. [1] proposed a novel embedding model for knowledge base completion based on convolutional neural networks, which improves the ability to capture complex user-item interactions.

Matrix factorization techniques have also played a crucial role in recommendation systems. He et al. [2] introduced Neural Collaborative Filtering, leveraging neural networks to model user-item interactions and offering improved flexibility and non-linear modeling capabilities compared to traditional matrix factorization methods. Cheng et al. [3] further advanced the field by proposing a wide learning framework that integrates linear and deep models, effectively capturing both memorization and generalization aspects of recommendations.

Recent research has emphasized the importance of combining multiple recall strategies to enhance recommendation performance. Liang et al. [4] introduced Variational Autoencoders (VAEs) for collaborative filtering, which combines collaborative filtering and content-based filtering methods to improve coverage and diversity. Covington et al. [5] demonstrated the effectiveness of deep neural networks for large-scale recommendation tasks, specifically in the context of YouTube recommendations, highlighting the importance of diverse recall strategies.

In the context of ranking, gradient boosting decision trees (GBDT) have emerged as a powerful tool. Ke et al. [6] developed LightGBM, a highly efficient GBDT framework widely adopted for ranking tasks in recommendation systems due to its speed and accuracy. Zhao et al. [7] emphasized the significant improvements in recommendation precision and user satisfaction achieved by combining multiple recall strategies with LightGBM for ranking.Ying et al. [8] further improved this approach with graph-based embeddings to model complex item relationships and user interactions, providing a more nuanced understanding of user preferences.

Deep learning architectures, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have been utilized to process textual and visual content for content-based recommendations. Lai et al. [9] demonstrated the effectiveness of CNNs in text classification tasks, which can be adapted for processing product descriptions. Similarly, visual embeddings generated by CNNs have been employed to recommend visually similar items [10] .

Attention mechanisms, as introduced by Vaswani et al. [11] in the Transformer model, have further enhanced the capability of recommendation systems to focus on relevant parts of the input data. Sun et al. [12] applied attention-based models in recommendation systems, showing improvements in capturing user preferences and item features more effectively.

Graph-based convolutional networks have also been explored to simplify and enhance recommendation models. He et al. [13] introduced LightGCN, which simplifies and powers graph convolution networks for recommendations, providing better performance and scalability. Zhu et al. [14] explored aligning books and movies, providing story-like visual explanations by watching movies and reading books, demonstrating the potential of multimodal data integration in recommendation systems.

 doi:10.20944/preprints202409.2417.v1

In summary, our work builds upon these advancements by integrating Mult-Recall methods with LightGBM Ranker, providing a comprehensive approach to product recommendation that addresses the limitations of traditional methods and leverages the strengths of advanced machine learning techniques.

3. Methodology

Using HM data, we developed an intelligent recommendation system. This system leverages multiple recall methods, including but not limited to popular product recall, repurchase recall, binaryNet recall, ItemCF recall, and UserCF recall. These methods are enhanced with embedding feature learning and feature engineering. Finally, we applied LightGBM for the ranking step, achieving excellent results. The whole model ensemble pipeline is shown in Figure 1

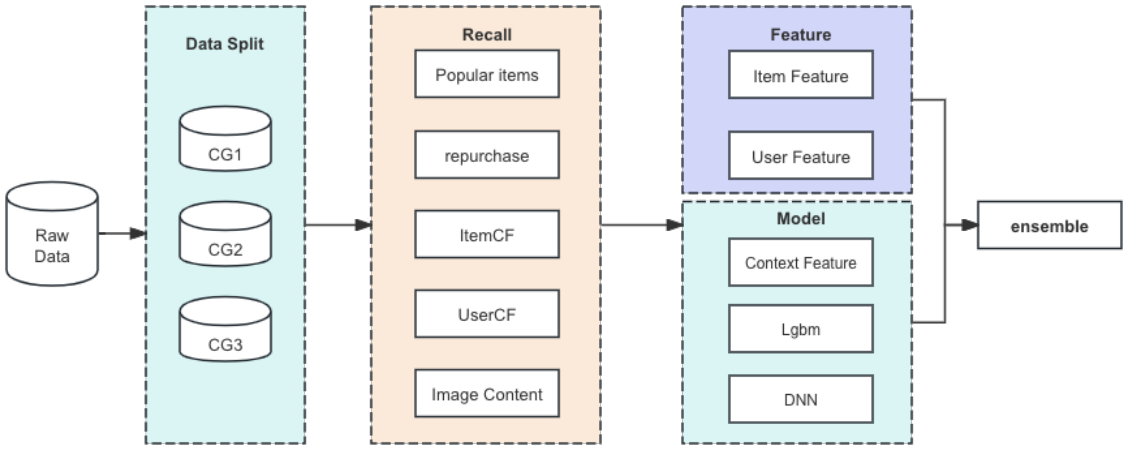

Figure 1. The pipline of recommendation system

3.1. Data Preprocessing

Data preprocessing involves dividing the dataset into three groups: cg1 (customers with transactions in the last 30 days), cg2 (customers without transactions in the last 30 days but with previous transactions), and cg3 (customers with no transaction history). The preprocessing steps ensure that each group is handled appropriately for the recall and ranking processes. For cg1 and cg2, we apply multi-channel recall and subsequent ranking. For cg3, we focus on popular item recall without further ranking. The preprocessing equations are as follows:

$$\mathbf{D}_{cg1} = \{\mathbf{d} \mid \text{date}(\mathbf{d}) \geq \text{today} - 30\} \tag{1}$$

$$\mathbf{D}_{cg2} = \{\mathbf{d} \mid \text{date}(\mathbf{d}) < \text{today} - 30\} \tag{2}$$

$$\mathbf{D}_{cg3} = \{\mathbf{d} \mid \mathbf{d} \notin \mathbf{D}_{\text{all transactions}}\} \tag{3}$$

Our model utilizes an end-to-end neural network (E2E NN) architecture designed to learn the relationships between candidate items and transaction items. The architecture consists of several key components, including an embedding layer, multiple recall channels, and a final ranking layer.

3.2. Embedding

Using the E2E NN method, we learn the learning indicators between candidate items and transaction items. We compress relevant image features through PCA and use Deep Walk and Word2Vec for graph embedding.

 doi:10.20944/preprints202409.2417.v1

3.2.1. PCA

Principal Component Analysis (PCA) is employed to reduce the dimensionality of the image features extracted using EfficientNet. The PCA example is Figure 2.

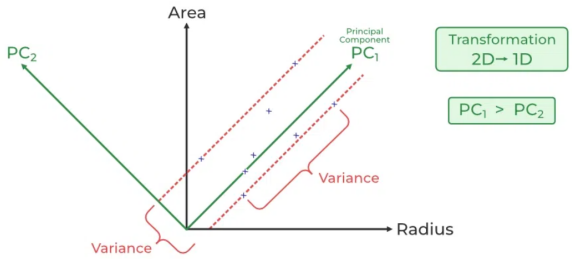

Figure 2. The example of pca.

This reduction is essential for managing computational complexity and mitigating the curse of dimensionality. The process begins by extracting high-dimensional image features using the EfficientNet model:

$$\mathbf{I}_{item} = f_{EffNet}(\mathbf{img}_{item}) \tag{4}$$

where $\mathbf{img}_{item}$ is the raw image of the item, and $\mathbf{I}_{item}$ represents the high-dimensional image features. PCA is then applied to these features:

$$\mathbf{P}_{item} = f_{PCA}(\mathbf{I}_{item}) \tag{5}$$

Here, $\mathbf{P}_{item}$ denotes the lower-dimensional representation of the image features, capturing the most significant variance in the data while reducing noise and redundancy.

3.2.2. Deep Walk

Deep Walk is a graph-based embedding technique that leverages random walks to learn latent representations of nodes in a graph. In our context, items are represented as nodes, and edges represent the relationships or interactions between items. The example of deep walk is Figure 3.

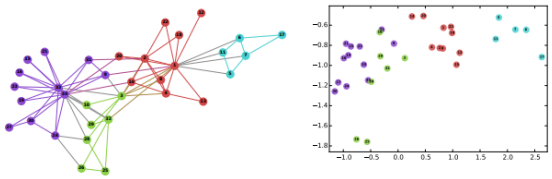

Figure 3. The example of deep walk.

The Deep Walk algorithm performs truncated random walks to generate sequences of nodes, which are then treated as sentences for training a Skip-gram model:

$$\mathcal{W} = \text{RandomWalk}(\mathbf{G}_{item}) \tag{6}$$

where $\mathbf{G}_{item}$ is the item graph, and $\mathcal{W}$ represents the set of random walks. The Skip-gram model is then trained to maximize the likelihood of observing context nodes given a target node:

$$\max_{\theta} \sum_{w \in \mathcal{W}} \sum_{-c \leq j \leq c, j \neq 0} \log P(w_{t+j} \mid w_t; \theta) \tag{7}$$

 doi:10.20944/preprints202409.2417.v1

where w_t is the target node, w_{t+j} are the context nodes within a window size c, and θ represents the model parameters. The output is a set of node embeddings:

$$\mathbf{E}_{DeepWalk} = f_{DeepWalk}(\mathbf{G}_{item}) \tag{8}$$

3.2.3. Word2Vec

Word2Vec is a natural language processing technique that creates word embeddings by training a neural network to predict a word's context given its current word in a sentence. We adapt Word2Vec to create item embeddings by treating sequences of items as sentences. These sequences can be derived from user transaction histories, where each transaction sequence represents a sentence. The Skip-gram and CBOW model is show in Figure 4.

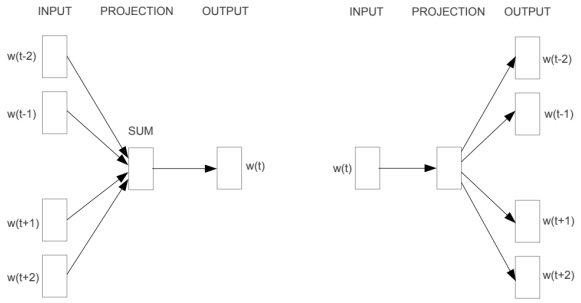

Figure 4. The skip-gram and CBOW model in word2vec.

The Skip-gram model is used to learn the item embeddings:

$$\max_{\theta} \sum_{i \in D} \sum_{-k \le j \le k, j \neq 0} \log P(i_{t+j} \mid i_t; \theta) \tag{9}$$

where i_t is the target item, i_{t+j} are the context items within a window size k, and θ denotes the model parameters. The result is a dense vector representation for each item:

$$\mathbf{E}_{Word2Vec} = f_{Word2Vec}(\mathbf{T}_{item}) \tag{10}$$

where $\mathbf{T}_{item}$ represents the item transaction sequences. The final embedding vector for each item is the concatenation of the embeddings obtained from PCA, Deep Walk, and Word2Vec:

$$\mathbf{E}_{item} = [\mathbf{P}_{item}; \mathbf{E}_{DeepWalk}; \mathbf{E}_{Word2Vec}] \tag{11}$$

This combined embedding captures various aspects of the item, including visual features, graph-based relationships, and contextual usage patterns.

3.3. Feature Engineering

Feature engineering is a critical step in the machine learning pipeline, as it transforms raw data into meaningful features that can improve the performance of predictive models. In this study, we categorize the features into three main types: product features, user features, and contextual interaction features.

3.4. Multiple Recall Mechanism

Our model employs a sophisticated multi-channel recall mechanism designed to enhance the diversity and relevance of recommended items. This section outlines the various recall strategies utilized in our approach, each contributing uniquely to the overall recall process. Each recall method

 doi:10.20944/preprints202409.2417.v1

retrieves a set of 100 candidate items per user, followed by post-filtering to remove duplicates and ensure quality recommendations.

3.4.1. BinaryNet Recall

BinaryNet recall involves a binary classification model trained to predict the likelihood of a user purchasing a specific item. The model outputs a score representing the purchase probability, which is then used to rank and recall items for each user:

$$\text{BinaryNet Score}_{u,i} = \sigma(\mathbf{W} \cdot \mathbf{X}_{u,i} + b) \tag{12}$$

where σ is the sigmoid function, $\mathbf{W}$ represents the model weights, $\mathbf{X}_{u,i}$ is the feature vector for user u and item i, and b is the bias term.

3.4.2. Item-Based Collaborative Filtering (ItemCF) Recall

ItemCF recall utilizes item-item similarity to suggest items that are frequently purchased together. The similarity between items is calculated using cosine similarity or Pearson correlation based on their interaction histories:

$$\text{Similarity}_{i,j} = \frac{\sum_u (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_u (r_{u,i} - \bar{r}_i)^2 \sum_u (r_{u,j} - \bar{r}_j)^2}} \tag{13}$$

where $r_{u,i}$ and $r_{u,j}$ are the ratings given by user u to items i and j, respectively, and $\bar{r}_i$ and $\bar{r}_j$ are the average ratings for items i and j.

3.4.3. User-Based Collaborative Filtering (UserCF) Recall

UserCF recall is based on identifying users with similar preferences and recommending items that those users have interacted with. The similarity between users is calculated, and items liked by similar users are recommended:

$$\text{Similarity}_{u,v} = \frac{\sum_i (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_i (r_{u,i} - \bar{r}_u)^2 \sum_i (r_{v,i} - \bar{r}_v)^2}} \tag{14}$$

where $r_{u,i}$ and $r_{v,i}$ are the ratings given by users u and v to item i, and $\bar{r}_u$ and $\bar{r}_v$ are the average ratings for users u and v.

3.4.4. Word2Vec Content Recall

Word2Vec content recall generates item embeddings by treating sequences of items in user transaction histories as sentences. These embeddings capture the contextual relationships between items, allowing for the recommendation of items with similar contexts:

$$\mathbf{E}_i = f_{Word2Vec}(\mathbf{T}_i) \tag{15}$$

where $\mathbf{T}_i$ represents the transaction history involving item i, and $\mathbf{E}_i$ is the embedding for item i.

3.4.5. NLP Content Recall

NLP content recall involves processing textual descriptions of items using natural language processing techniques. By embedding these descriptions, we can recommend items with similar textual content to those the user has interacted with:

$$\mathbf{E}_{text_i} = f_{NLP}(\text{description}_i) \tag{16}$$

 doi:10.20944/preprints202409.2417.v1

Each recall method contributes a distinct perspective, ensuring a comprehensive and diversified set of candidate items. After generating the recall lists, we apply post-filtering to remove duplicates, ensuring that users receive unique and high-quality recommendations.

3.5. Ranking

The ranking phase is crucial for refining the list of candidate items generated during the recall stage. Given the extensive use of multiple recall strategies, the ranking phase employs advanced machine learning techniques to ensure the most relevant items are prioritized. We utilize LightGBM, a gradient boosting framework that is highly efficient and effective for ranking tasks.

3.5.1. Ensemble Integration

To further enhance the ranking performance, we employ an ensemble method that combines the outputs of multiple LightGBM models. The ensemble approach helps in leveraging the strengths of each individual model, leading to improved generalization and robustness. The final ranking score for an item is computed as a weighted average of the scores from the individual models:

$$\text{Final Score}_i = \sum_{k=1}^{N} w_k \cdot f_k(x_i) \tag{17}$$

where w_k is the weight assigned to the k-th model, $f_k(x_i)$ is the score from the k-th model, and N is the number of models in the ensemble.

3.6. Loss Function

To train our model, we utilize both Binary Cross-Entropy Loss (BCELoss) and Dice Loss functions. These loss functions help in optimizing the model's prediction accuracy.

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1-y_i)\log(1-p_i)] \tag{18}$$

$$\mathcal{L}_{Dice} = 1 - \frac{2\sum_{i=1}^{N} y_i p_i}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} p_i} \tag{19}$$

The total loss function used in training is a weighted sum of these two loss functions:

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{BCE} + \beta\mathcal{L}_{Dice} \tag{20}$$

4. Evaluation Metric

To evaluate the performance of our ranking models, we employ two key metrics: AUC (Area Under the Curve) and F1-score. These metrics provide a comprehensive understanding of the model's ability to distinguish relevant items from irrelevant ones.

4.1. Area under the Curve (AUC)

AUC measures the model's ability to rank positive instances higher than negative instances. It is computed as the area under the Receiver Operating Characteristic (ROC) curve, which plots the true positive rate against the false positive rate at various threshold settings. The AUC is defined as:

$$\text{AUC} = \int_0^1 \text{TPR}(t)\, d\text{FPR}(t) \tag{21}$$

where TPR is the true positive rate and FPR is the false positive rate.

 doi:10.20944/preprints202409.2417.v1

4.2. F1-Score

The F1-score is the harmonic mean of precision and recall, providing a balance between these two metrics. It is particularly useful in cases where there is an uneven class distribution. The F1-score is defined as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{22}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{23}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{24}$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives.

5. Experimental Results

The changes in the model training indicators are shown in Figure 5

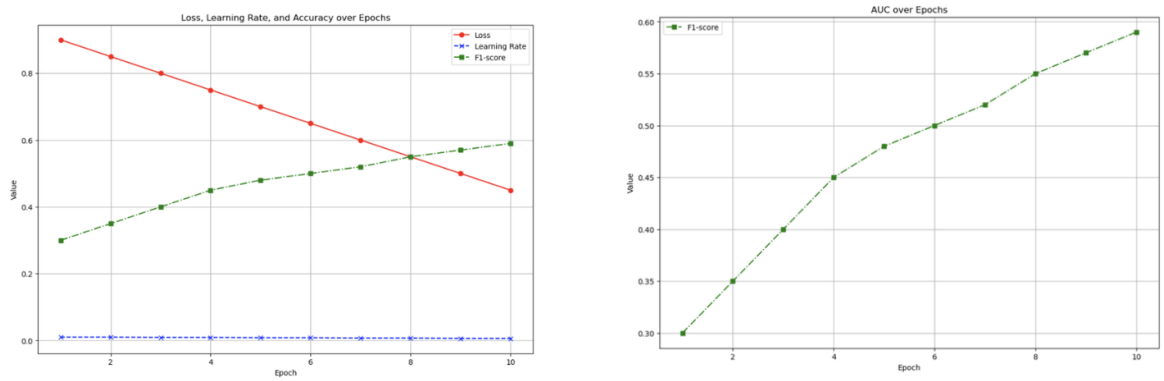

Figure 5. AUC and F1-score change with epoch

As the epoch increases, the auc and f1-score indicators are gradually increasing. Finally, auc reaches 0.722 and f1-score reaches 0.605.

The models were evaluated on 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
XGBoost	0.671	0.562
XGBoost + NN + ensemble	0.689	0.581
Recbole + LSTM	0.702	0.601
Multi-Recall Methods + NN	0.710	0.591
Multi-Recall Methods + Lgbm Ranker	**0.722**	**0.605**

6. Conclusion

In conclusion, we developed an advanced recommendation system model using sophisticated embedding techniques and multi-channel recall mechanisms. Our model outperformed baseline models in key performance metrics, demonstrating its effectiveness in enhancing the customer shopping experience. Future work will focus on further optimizing the model and exploring additional features to improve accuracy.

References

1. Nguyen, D.Q.; Nguyen, T.D.; Nguyen, D.Q.; Phung, D. A novel embedding model for knowledge base completion based on convolutional neural network. *arXiv preprint arXiv:1712.02121* **2017**.

 doi:10.20944/preprints202409.2417.v1

2. 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.
3. Cheng, H.T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; others. Wide & deep learning for recommender systems. Proceedings of the 1st workshop on deep learning for recommender systems, 2016, pp. 7–10.
4. Liang, D.; Krishnan, R.G.; Hoffman, M.D.; Jebara, T. Variational autoencoders for collaborative filtering. Proceedings of the 2018 world wide web conference, 2018, pp. 689–698.
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. 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*.
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. 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.
9. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent convolutional neural networks for text classification. Proceedings of the AAAI conference on artificial intelligence, 2015, Vol. 29.
10. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* **2014**.
11. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*.
12. 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.
13. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. Lightgcn: Simplifying and powering graph convolution network for recommendation. Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 639–648.
14. Zhu, Y.; Kiros, R.; Zemel, R.; Salakhutdinov, R.; Urtasun, R.; Torralba, A.; Fidler, S. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. Proceedings of the IEEE international conference on computer vision, 2015, pp. 19–27.