DeepChoice: Enhancing E-Commerce with AI Recommendation Systems

Carolina Li
Northeastern University
Bellevue, WA, USA
li.yuhan5@northeastern.edu

Xuefeng Li
Northeastern University
Seattle, WA, USA
li.xuefen@northeastern.edu

Yaxin Yu
Northeastern University
Seattle, WA, USA
yu.yax@northeastern.edu

Abstract

Driven by advancements in artificial intelligence and machine learning, the digital marketplace is rapidly evolving. Traditional collaborative and content-based filtering methods face limitations when capturing the subtleties in user reviews. This project investigates the potential of transformer-based models to outperform classic methods using the Amazon Reviews dataset.

Keywords

Recommendation Systems, Transformers, Deep Learning, Collaborative Filtering, Content-Based Filtering

ACM Reference Format:
Carolina Li, Xuefeng Li, and Yaxin Yu. 2025. DeepChoice: Enhancing E-Commerce with AI Recommendation Systems. In . ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Tech giants like Amazon, TikTok Shop, and Google Shopping have pushed the frontier of personalization. Recommendation systems primarily rely on collaborative or content-based filtering, but often fail to capture contextual richness in user reviews. Transformer models, on the other hand, excel in textual understanding, offering a promising direction.

Collaborative filtering and content-based filtering are two common approaches used in recommendation systems. A content-based recommendation system uses information about the recommended item (item information from the specific user's past behavior), while a collaborative filtering-based system leverages user behavior data to find a group of similar users or items.

With the advancement of deep learning and large language models, recommendation systems have seen new opportunities. Transformer architectures, such as BERT, are capable of capturing meaningful textual information, enabling more sophisticated integration of text into recommendation strategies.

This project compares the performance of these models in a realistic shopping scenario. We evaluate how effectively traditional methods and deep learning models can predict user ratings for unseen items.

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 Dataset Description

2.1 Dataset Overview

We use the 2023 Amazon Reviews dataset by McAuley Lab, containing over 500 million reviews from 1996 to 2023 across 32 product categories.

2.2 Data Schema

Data fields for users/Reviews:

data fields	notes
rating	Rating of the product (from 1.0 to 5.0).
title	Title of the user review.
text	Text body of the user review.
images	Images that users post after they have received the product.
asin	ID of the product.
parent_asin	Parent ID of the product.
user_id	ID of the reviewer
timestamp	Time of the review (unix time)
verified_purchase	User purchase verification
helpful_vote	Helpful votes of the review

Figure 1: User data fields from the Amazon dataset.

2.3 Data Exploration

The dataset shows that most products have very high ratings (above 4.0), while only a small number get significant attention. The user-item matrix is highly sparse.

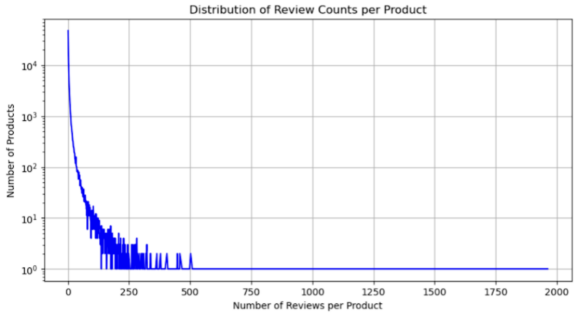

Figure 2: Review Count Distribution per Product

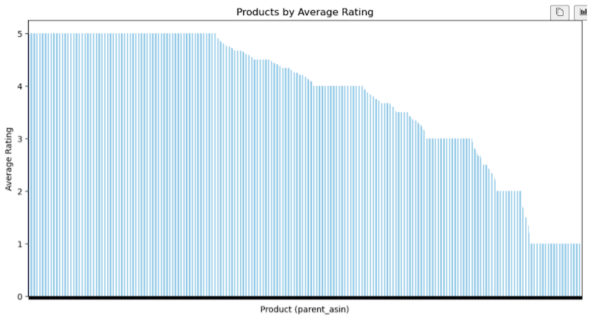

Figure 3: Average Ratings of Products

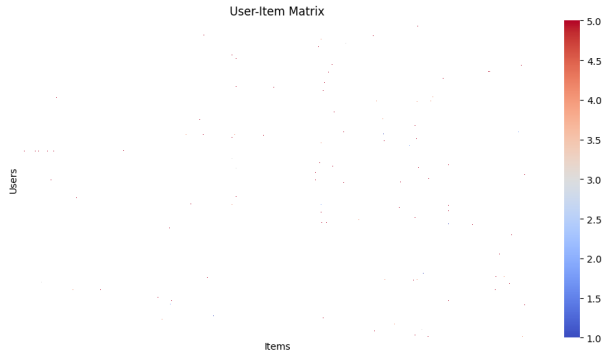

Figure 4: User-Item Rating Matrix

2.4 Data Preparation

We filtered reviews before 2020 and removed users with fewer than 5 reviews. We used the Leave-Last-Out strategy for train-test splitting to simulate real-world recommendation scenarios.

3 Method

In this section, we describe the models we use to build and evaluate our recommendation system. We view the problem as a regression task and explore both traditional collaborative filtering and deep learning models to understand their strengths and limitations.

3.1 Baseline Models

3.1.1 Collaborative Filtering. Collaborative filtering[3] is one of the common approaches in recommendation systems. Collaborative filtering-based system leverages user behavior data to find a group of similar users or items and predict future preferences of a user. In this work, we explore two primary kinds of collaborative filtering:

- **Latent Factor Model (Singular Value Decomposition - SVD)**: This method approximates the original user-item rating matrix by decomposing it into three smaller matrices and interactions (ratings) are represented by the dot product of user and item vectors. The model captures hidden patterns in user preference and item characteristics that explain the observed ratings. It is widely used due to its simplicity and strong performance on sparse data.
- **Neighborhood Model (k-Nearest Neighbors - KNN)**: This approach first calculate similarity between users. Then it predicts ratings and recommends items based on similar users. This approach relies on the assumption that similar users may give similar ratings to the same item. This method is intuitive and easy to interpret, though it struggles with scalability on large datasets.

3.1.2 LightFM. As discussed in prior work on implicit feedback [2], implicit interactions also contain valuable information that can be used in recommendation. LightFM[4] is such a hybrid recommendation model that combines the strengths of collaborative filtering and content-based filtering. LightFM learns embeddings for users and items using both implicit interaction and explicit ratings. Features like user profile or item descriptions can also be included in embedding process. These embeddings are optimized using matrix factorization objectives (e.g., WARP loss [5]) and gradient descending, allowing the model to generalize well even when interactions are limited. The final rating prediction is computed as the dot product of user and item embeddings. It is particularly useful when user-item interaction data is sparse and additional metadata is available. Moreover, because of its hybrid nature, LightFM is much mare flexible than pure collaborative models.

3.2 Neural Network Models

Deep Learning[6] has show its strength in Recommendation Systems. In this work, we also implement two deep learning models to explore whether neural networks can learn more complex features in user preferences.

3.2.1 Simple MLP with Text Embeddings. This model combines semantic information from item titles and user history to predict ratings:

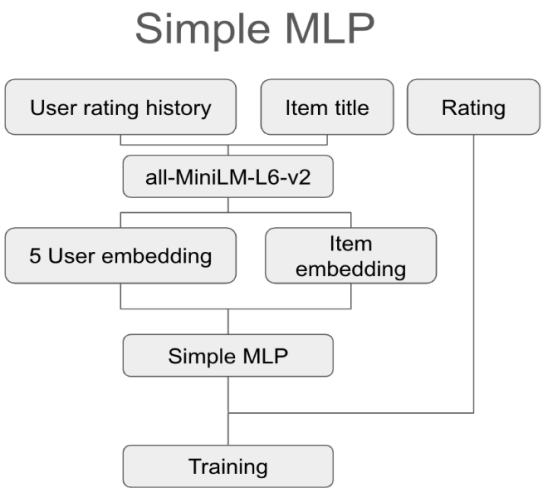

Figure 5: Model architecture for MLP neural network

- **Input**: The model takes two textual inputs—the user's rating history as user feature and the item title as item feature. It is noteworthy that one user have a unique embeddings for each rating value, allowing the model to capture distinct user behaviors corresponding to different rating levels.
- **Text Embedding**: We use `all-MiniLM-L6-v2`, a pretrained sentence-transformer language model, to convert these text inputs into 384 dimension dense vector representations (embeddings). These embeddings provides user features ans item features.
- **MLP Architecture**: The user and item embeddings are concatenated and passed through a multi-layer perceptron (MLP), to predict ratings.
- **Output**: The final layer outputs a single scalar value representing the predicted user rating.
- **Loss**: Here we use Root Mean Squared Error (RMSE) as loss function, the most common loss for rating prediction.

3.2.2 Category-Based Neural Network. We also design a simplified neural model that learns directly from categorical user and item identifiers. This model is much more lightweight and interpretable:

- **Input**: Each user and item is identified by a unique ID and represented using one-hot encoding.
- **Embedding Layer**: These encoding are passed through an embedding layer, which embeds them to dense, trainable vectors. These embeddings are learned during training to capture user preferences and item characteristics.
- **Concatenation and Dense Layer**: The user and item embeddings are concatenated, flattened, and passed through dense layers to produce a rating prediction. We also apply dropout during training to prevent overfitting.
- **Output**: The final layer outputs a single scalar value representing the predicted user rating.
- **Loss**: Here we use Root Mean Squared Error (RMSE) as loss function, the most common loss for rating prediction.

4 Result Analysis

We evaluated five models: SVD, KNN with Means, a One-Hot encoded Neural Network, Multi-Layer Perceptron (MLP), and LightFM across five product categories from the Amazon dataset. The primary evaluation metric was validation Root Mean Squared Error (RMSE), supported by loss curves and Top-5 recommendation examples for qualitative analysis.

4.1 Model Performance Overview

SVD achieved the best overall performance, particularly in structured and denser categories such as *Office Products*, where it reached the lowest RMSE of 1.14. This highlights the strength of collaborative filtering in environments with consistent user-item interaction patterns.

The One-Hot Neural Network model closely followed SVD, demonstrating strong generalization and stable loss convergence, particularly in *Office Products* and *Baby Products*.

LightFM yielded higher RMSEs and underperformed in sparse categories. However, it generated more diverse recommendations in

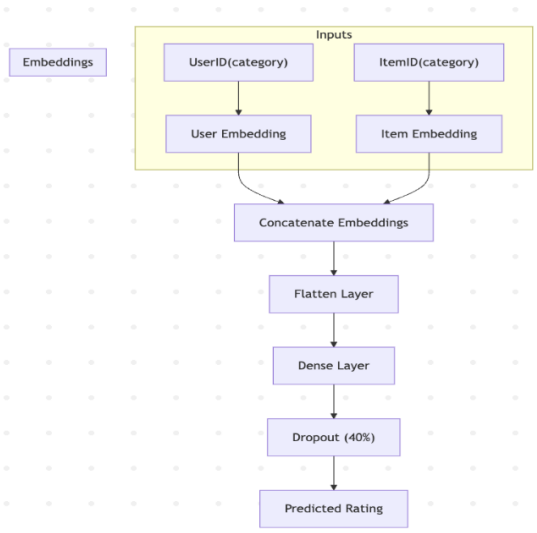

Figure 6: Model architecture for category-based neural network

Video Games, likely due to its hybrid architecture that incorporates content features.

KNN with Means performed well in *Office Products* and *Arts, Crafts & Sewing*, but struggled to maintain diversity in more complex, content-rich categories, leading to lower overall performance.

The MLP model was competitive but consistently trailed SVD and the One-Hot Neural Network. Its dependence on one-hot encoding and lack of rich item metadata likely limited its ability to capture nuanced preferences.

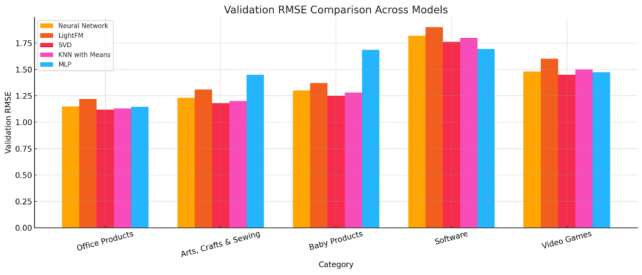

Figure 7: Validation RMSE comparison across five models and product categories.

4.2 Category-Level Insights

- **Office Products**: Achieved the lowest RMSE (1.14) with a smooth, stable loss curve—indicating consistent user behavior and strong agreement across models.
- **Arts, Crafts & Sewing** and **Baby Products**: Showed moderate RMSEs (1.22 and 1.30 respectively) and early convergence, suggesting predictable patterns in user preferences.

- **Software**: Had the highest RMSE (1.81) and the steepest loss curves, reflecting heterogeneous user feedback and greater rating variance.
- **Video Games**: Showed moderate performance (1.49 RMSE). LightFM performed notably well here in generating diverse, content-aware recommendations.

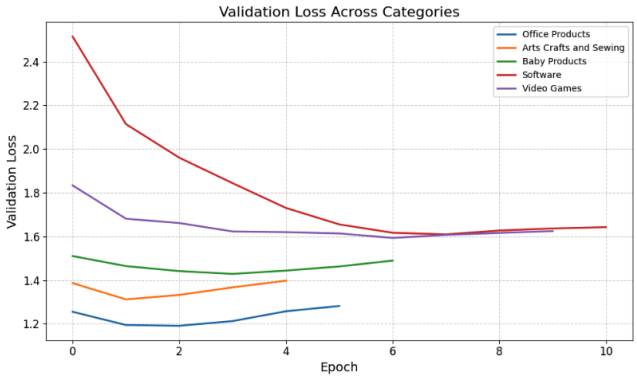

Figure 8: Validation loss across product categories over training epochs.

Model Performance Comparison Across Product Categories

Product Category	NN Top-5 Recommendations	LightFM Top-5 Recommendations	Validation RMSE
Office Products	16787, 8701, 52393, 5799, 43476	562, 4274, 41, 105, 47	1.1397
Arts, Crafts & Sewing	68183, 52697, 46901, 26864, 69824	707, 4945, 7757, 1314, 6072	1.2205
Baby Products	7006, 9609, 14078, 20273, 600	768, 2794, 682, 28, 570	1.295
Software	4693, 5258, 777, 2842, 2625	51, 73, 239, 41, 69	1.8148
Video Games	10908, 10256, 10727, 94, 4961	195, 291, 1667, 1737, 1428	1.4877

Figure 9: Top-5 product recommendations by NN and LightFM per category, including RMSE.

4.3 Summary of Findings

Overall, our results indicate that no single model performs best across all categories. Collaborative filtering approaches such as SVD and KNN are effective in structured, sparse domains [3], where user-item interaction patterns are relatively consistent and interpretable. In contrast, neural and hybrid models (e.g., One-Hot NN, LightFM) offer greater adaptability and recommendation diversity, particularly in content-rich or behaviorally complex environments [1, 6].

5 Conclusion

Our findings demonstrate that traditional collaborative filtering methods, such as SVD and KNN, consistently outperformed deep learning models in terms of Mean Squared Error (MSE). This reinforces the strength of collaborative filtering in capturing user-item interactions with minimal feature engineering. In contrast, deep learning approaches—including neural networks and BERT-based transformers—offer enhanced flexibility through the integration of rich content features such as product descriptions and metadata. However, their effectiveness depends on larger datasets and more extensive hyperparameter tuning.

This reflects a fundamental trade-off: collaborative filtering offers simplicity and interpretability, making it well-suited for structured and sparse datasets, while deep learning provides scalability and adaptability in content-rich environments.

Future work will focus on refining deep learning architectures to enhance generalization and robustness. We also plan to explore reinforcement learning techniques to model sequential and evolving user behaviors, such as shifting preferences over time. Hybrid frameworks that integrate collaborative filtering, deep learning, and reinforcement learning may offer a more holistic approach to recommendation, capable of capturing both structural and contextual signals. Finally, extending evaluations to a broader range of product categories will be crucial for assessing the scalability and personalization quality of these models in real-world settings.

References

[1] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys)*. ACM, 191–198.
[2] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM)*. IEEE, Pisa, Italy, 263–272. doi:10.1109/ICDM.2008.22
[3] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.
[4] Maciej Kula. 2015. Metadata embeddings for user and item cold-start recommendations. In *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems co-located with RecSys*.
[5] Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. WSABIE: Scaling up to large vocabulary image annotation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*. 2764–2770.
[6] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 1–38.