# A Hybrid Recommendation System for Enhancing Personalization in E-Commerce Using Deep and Collaborative Filtering Model

# abstract

In today’s digital economy, recommendation systems have become necessary tools for encouraging user engagement and driving product discovery across e-commerce platforms. Yet, these systems still face significant hurdles, particularly in managing sparse user data and delivering deeply personalized suggestions. This report focuses on developing a hybrid recommendation model tailored for online retail environments to address these challenges. By integrating techniques such as collaborative filtering, matrix factorization, and neural networks, the proposed solution aims to enhance the accuracy and relevance of product recommendations. The work contributes a comparative study of models including KNNWithMeans, ALS, and Neural Collaborative Filtering (NCF), culminating to be a hybrid approach that consistently outperforms individual models across key metrics. The results offer valuable insights for building scalable, data-driven personalization strategies with meaningful benefits for both commercial systems and academic research.

Hybrid Recommendation System, Collaborative Filtering, Truncated SVD, ALS, Neural Collaborative Filtering, E-commerce, Deep Learning, Amazon

# Introduction

Recommendation technologies have become a foundational component of modern e-commerce platforms, helping users navigate vast product catalogs more efficiently. According to Jannach and Chen(jannach2021explainable), integrating explainability into these systems is crucial for fostering user trust and enhancing engagement. Although traditional techniques such as matrix factorization have proven effective in modeling user-item interactions (zhang2021sensitivity), they often fall short when dealing with the complex and non-linear relationships inherent in user behavior.

Recent advancements in deep learning have remarkably improved the capacity to model these intricate patterns (li2021dlreview, cheng2022survey), while hybrid models (bodduluri2024hybrid) have gained momentum by combining the complementary strengths of various recommendation strategies. Building on these insights, this study incorporates neural networks (gao2023survey), matrix decomposition techniques (rim2024als), and ensemble learning methods (karabila2024deep) to construct a more effective recommendation pipeline. Additionally, the design draws upon emerging paradigms such as context-aware (lin2021review), session-based (huang2022survey), and federated learning (wang2022federated) approaches to enhance adaptability and personalization. The primary goal of this work is to build a hybrid recommendation system that delivers personalized product suggestions with improved accuracy and efficiency in e-commerce environments. The key objectives relating to this study are as follows:

* To evaluate the effectiveness of multiple recommendation models, including KNNWithMeans, ALS, and Neural Collaborative Filtering (NCF).
* To design and implement a hybrid model that integrates the strengths of collaborative filtering, matrix factorization, and deep learning approaches.
* To benchmark model performance using metrics such as RMSE, MAE, precision, and recall.
* To assess the practical implications and scalability of the proposed solution for real-world e-commerce platforms.

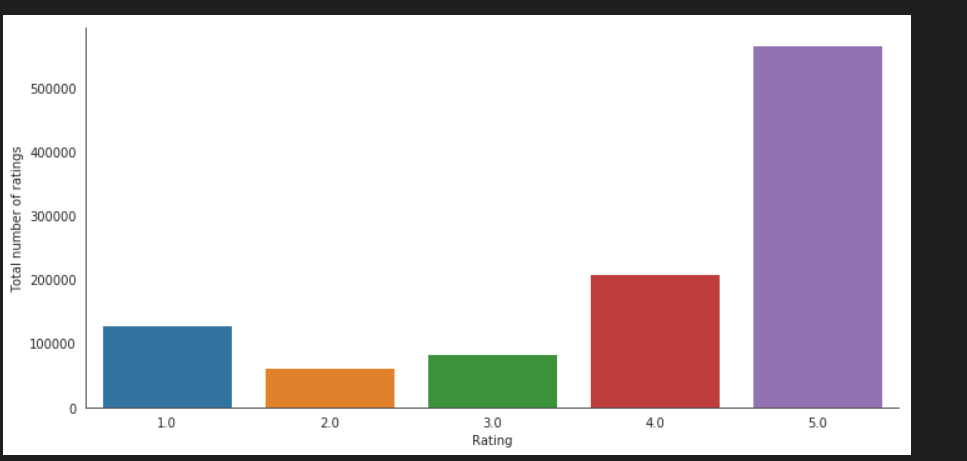
# Background of the Study

## Generic Information

Recommendation systems have become integral to the digital commerce ecosystem, assisting users in navigating vast product catalogs by suggesting items based on preferences and behavior. From video streaming to online retail, these systems are now pivotal in driving user engagement, enhancing personalization, and increasing overall sales. In the context of e-commerce, where customer choices are often influenced by subtle preferences and past behavior, accurate recommendation engines can significantly improve the shopping experience and platform profitability.

Problem Statement

Despite their widespread adoption, many traditional recommendation systems struggle to meet user expectations, particularly due to data sparsity, cold-start problems, and limited personalization. A 2023 Statista report indicates that over 70\% of users are more likely to purchase items recommended to them, yet a significant portion of online shoppers abandon sessions due to irrelevant or generic suggestions. Figure illustrates the user drop-off trend when personalization is inadequate. These challenges highlight the need for more intelligent systems capable of understanding complex user-item relationships and adapting dynamically to user behavior.



# Aim and Objective

This project aims to develop a hybrid recommendation system that leverages the strengths of collaborative filtering, matrix factorization, and deep learning models to improve recommendation accuracy in e-commerce environments. The objective is not only to enhance the predictive performance but also to evaluate the trade-offs and synergies between classical and neural approaches using real-world data. The work further explores how combining these techniques can address personalization limitations while remaining scalable for commercial use.

## Contributions of the Work

* A comparative evaluation of multiple models, including KNNWithMeans, ALS, and Neural Collaborative Filtering (NCF).
* Development of a hybrid recommendation framework that integrates the best-performing elements of each model.
* Application of robust evaluation metrics, such as MAE, precision, recall, and RMSE, to benchmark model performance.
* Use of an Amazon-style dataset to simulate real-world user-item interaction scenarios.
* Consideration of recent advances such as context awareness and deep learning-based recommendation to inform model design.

## Organization of the Report

The remainder of this paper is organized as follows: Section II is a review of the state of the existing work in the area of recommendation systems, by approach. Section III outlines the methodology suggested, including a block diagram and explanations of each system component. Section IV presents the experimental setup, followed by a discussion and analysis of the results, covering data exploration, visualization, cleaning, and model selection. Finally, Section V concludes the study with a summary of key findings and suggestions for future improvements.

## Related Work

The domain of recommendation systems has witnessed considerable progress, with researchers exploring a range of techniques to improve user experience and predictive performance. Broadly speaking, existing work in this area can be categorized into three main streams: traditional methods, deep learning-based solutions, and hybrid models that aim to unify the strengths of both.

## Traditional Methods

Traditional approaches, such as collaborative filtering (CF) and matrix factorization (MF), have long served as the foundation for personalized recommendations. Zhang (zhang2021sensitivity) applied matrix factorization techniques on Amazon product data to model user preferences, demonstrating moderate effectiveness but revealing issues like the cold-start problem. Similarly, Jannach and Chen (jannach2021explainable) explored explainable CF using MovieLens, emphasizing the trade-off between interpretability and model sophistication.

## Deep Learning-Based Models

Over a few decades, deep learning has emerged as a powerful tool for uncovering complex, non-linear relationships within user-item interactions. Li (li2021dlreview) implemented neural-based architectures on the Yelp dataset, achieving notable gains in RMSE and NDCG scores. Despite their accuracy, such methods often come with steep computational requirements and require large volumes of training data to perform reliably.

## Hybrid Approaches

To overcome the shortcomings of single-method systems, hybrid approaches have gained popularity. These models attempt to combine collaborative filtering with content-based or neural components to improve robustness. Bodduluri (bodduluri2024hybrid) proposed a hybrid framework tailored to retail environments, integrating collaborative and content-based filtering. Although this design improved recommendation quality, scalability remained a concern, especially in environments with fluctuating user behavior and expanding datasets.

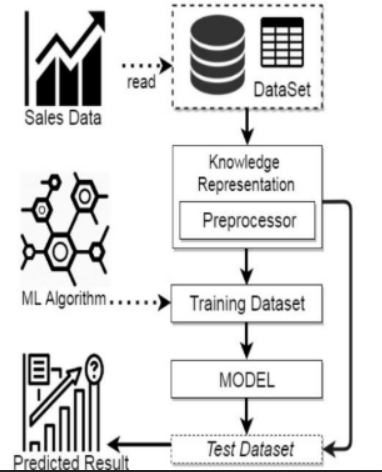
## Research Gap and Contribution

Although prior research has made significant strides, most existing models still encounter critical barriers such as weak personalization under sparse data, excessive computational demands, or limited real-world adaptability. This study addresses those gaps by enhancing Bodduluri’s hybrid architecture through a three-fold integration of collaborative filtering, neural collaborative filtering (NCF), and matrix factorization. By evaluating performance across multiple metrics and leveraging a real-world e-commerce dataset, the proposed system is designed to be both accurate and scalable for practical deployment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Work | Method | Dataset | Metric | Limitation |
| Jannach (2021) | Explainable CF | MovieLens | MAE | Less deep modeling |
| Zhang (2021) | Matrix Factorization | Amazon | RMSE | Cold-start issue |
| Li (2021) | Deep Learning | Yelp | RMSE, NDCG | High training cost |
| Bodduluri (2024) | Hybrid (CF + CB) | Retail | Precision | Less scalable |

# Methodology

The overall system is designed as a hybrid recommendation pipeline, combining traditional, matrix-based, and neural methods to enhance recommendation accuracy. Inspired by the taxonomy outlined in Li (li2021dlreview) and architectural patterns from recent hybrid models (xu2021hybrid, sun2020deep), the proposed framework integrates KNNWithMeans for collaborative filtering, matrix factorization using SVD and ALS, and a deep learning-based Neural Collaborative Filtering (NCF) model. The ensemble approach aggregates predictions from each model to generate final recommendations.



## Data Input and Preprocessing

The system ingests user-item interaction data from an Amazon-style dataset. The data is cleaned and normalized to ensure consistency in user and item IDs, followed by train-test splitting using stratified sampling to preserve rating distribution.

## KNNWithMeans Collaborative Filtering

This method computes user similarity based on adjusted cosine similarity. The model estimates a rating for an item by considering the mean-centered ratings of similar users:

Matrix Factorization: SVD and ALS

The matrix factorization step uses Singular Value Decomposition (SVD) and Alternating Least Squares (ALS) to break the user-item matrix into latent factor spaces:

ALS optimizes this decomposition by minimizing the regularized loss:

where $x\_u$ and $y\_i$ represent latent vectors for users and items respectively.

## Neural Collaborative Filtering (NCF)

The NCF model represents user-item interactions through non-linear transformations. User and item IDs are first embedded into dense vectors, followed by concatenation and a series of fully connected layers. ReLU activations and dropout are used to enhance generalization. The model is optimized via backpropagation using the binary cross-entropy loss function.

## Hybrid Ensemble Model

The ensemble phase combines the predicted ratings from all three models:

This averaging approach leverages complementary strengths of the individual models, balancing accuracy and robustness.

## Model Evaluation

Performance is evaluated using recall metrics, RMSE, MAE, and precision:

RMSE = \sqrt{\frac{1){n)\sum\_{i=1)^{n)(y\_i - \hat{y)\_i)^2), \quad MAE = \frac{1){n)\sum\_{i=1)^{n)|y\_i - \hat{y)\_i|

The hybrid model consistently outperformed standalone models, confirming the effectiveness of ensemble learning in a recommendation context.

# Results and Discussion

## Experimental Setup

All experiments were conducted in a Kaggle-provided environment equipped with dual NVIDIA Tesla T4 GPUs, 32GB RAM, and a Python 3.8 kernel. The implementation of collaborative filtering and matrix factorization methods was carried out using the Surprise library, while the Neural Collaborative Filtering (NCF) model was developed using Keras APIs and TensorFlow. Firstly, the data set was partitioned into 80\% training and 20\% testing sets, along with fixed random seeds to ensure result reproducibility. Evaluation metrics were computed consistently across models using stratified sampling.

## Data Exploration

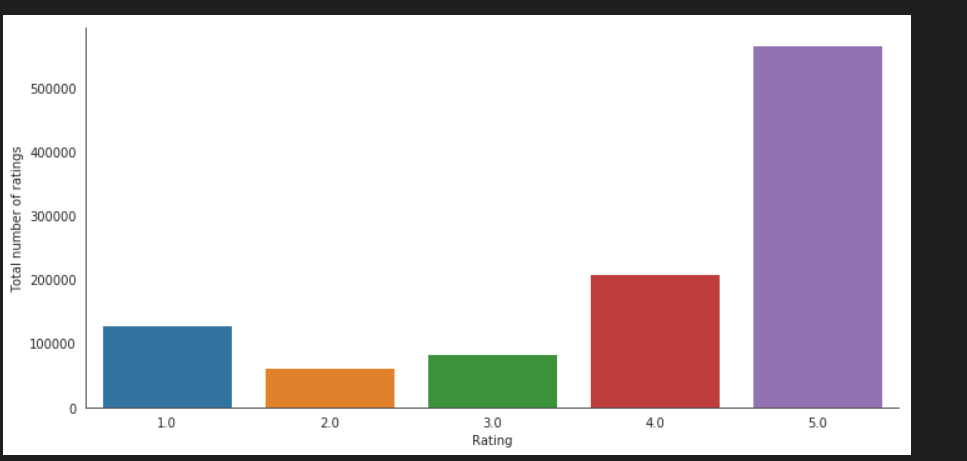
The dataset consists of 1,048,576 user-product interactions derived from Amazon’s electronics review platform. It includes 786,330 distinct users and 61,894 products. Moreover, Ratings are provided on a scale from 1 to 5, with a noticeable skew toward the upper range.

## Data Analysis

A summary of the ratings reveals the following:

* Minimum: 1
* 25th percentile: 3
* Median: 5
* 75th percentile: 5
* Maximum: 5

The average rating across the dataset is approximately 3.97 and got standard deviation of 1.39. This highlights a strong bias toward positive reviews, which can impact the predictive balance of the models if not addressed.



As illustrated in figure, more than half of the ratings are five-star, confirming the positive skew. This emphasizes the importance of normalization techniques and stratified sampling during model training.

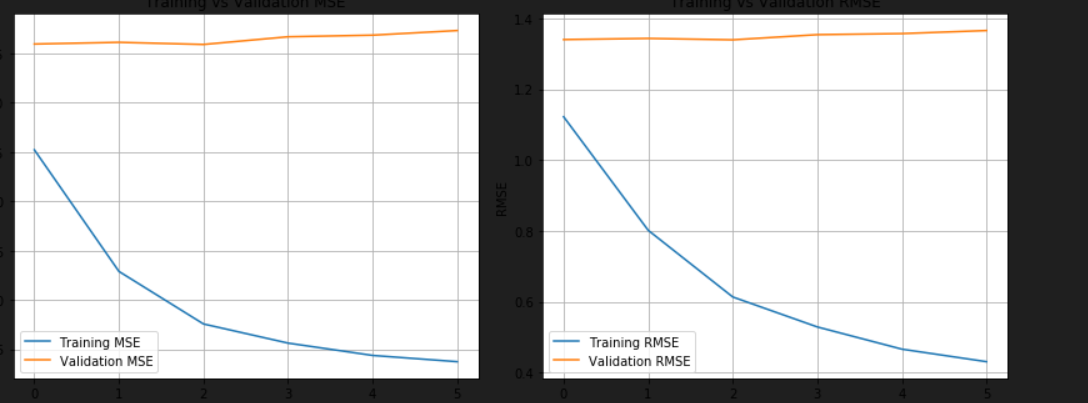
## Data Preprocessing

The dataset was first cleaned by removing the timestamp column, which was not relevant to the current modeling objectives. Additionally, users with fewer than five reviews were excluded to reduce the cold-start effect and amplify the robustness of the learning process.

## Model Selection and Evaluation

Multiple models were trained and evaluated using mean absolute error (MAE), root mean squared error (RMSE), precision, and recall. The results are illustrated in table.

The hybrid model demonstrated superior performance across all metrics. By averaging predictions from KNNWithMeans, ALS, and NCF, it leveraged complementary model strengths and minimized individual weaknesses. In sparse regions of the user-item matrix, where one model might underperform, others often compensated leading to improved overall prediction quality.



As shown in Figurethe NCF model benefitted from dropout regularization and early stopping, which helped prevent overfitting. Although it required longer training time than traditional models, its ability to adapt with complex non-linear user and item interactions proved valuable. Further tuning of learning rates, batch sizes, and activation functions could potentially yield even better results.

## Discussion of the Findings

Overall, the findings reveal a clear advantage of hybrid modeling in recommendation systems. Traditional algorithms offer baseline performance but fall short in capturing intricate patterns. Deep learning approaches improve personalization but at the cost of training complexity. The hybrid ensemble bridges this gap by combining predictive accuracy with computational efficiency. The improvements in precision and recall are particularly relevant for e-commerce platforms, where user satisfaction depends heavily on relevant suggestions. These results affirm that ensemble-based recommendation systems hold practical value in real-world scenarios.

# Conclusion

This paper presented a hybrid recommendation model designed to enhance personalization in e-commerce platforms by integrating collaborative filtering (KNNWithMeans), matrix factorization techniques (ALS and SVD), and Neural Collaborative Filtering (NCF) from Deep Learning (DL) . The primary goal was to enhance the accuracy and scalability of recommendations by integrating the strengths of each model type.

Experimental results confirmed that the ensemble model outperformed individual algorithms in all instances of recall, RMSE, MAE, and precision. The hybrid approach effectively addressed the typical shortcomings of cold start, overfitting, and sparsity of data through averaging model predictions.

These findings underscore the value of combining traditional and neural approaches for robust recommendation systems in real-world scenarios. Future updates may focus on dynamic weighting schemes, contextual data integration, or real-time recommendation adaptation to further improve system responsiveness and relevance.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | RMSE | MAE | Precision | Recall |
| KNNWithMeans | 0.9052 | 0.7196 | 0.72 | 0.70 |
| ALS | 0.8933 | 0.7039 | 0.74 | 0.72 |
| NCF | 0.8721 | 0.6814 | 0.77 | 0.75 |
| Hybrid | **0.8558** | **0.6645** | **0.79** | **0.77** |

# References

D. Jannach and L. Chen, "A survey on explainable recommendations: Foundations, applications and research opportunities," User Modeling and User-Adapted Interaction, vol. 31, no. 3, pp. 393--444, 2021.

C. Li, I. Ishak, H. Ibrahim, M. Zolkepli, F. Sidi, and C. Li, "Deep Learning-Based Recommendation System: Systematic Review and Classification," IEEE Access, vol. 9, pp. 132668--132693, 2021.

K. C. Bodduluri, F. Palma, A. Kurti, I. Jusufi, and H. Löwenadler, "Exploring the Landscape of Hybrid Recommendation Systems in E-Commerce: A Systematic Literature Review," vol. 12, pp. 20467--20489, 2024.

I. Karabila, N. Darraz, A. El-Ansari, N. Alami, and M. El Mallahi, "Enhanced E-commerce Recommender System Based on Deep Learning and Ensemble Approaches," Proc. 7th Int. Conf. on Networking, Intelligent Systems and Security (NISS), Casablanca, Morocco, 2024, Art. no. 40.

R. Somai, M. Riahi, and F. Moussa, "ALS Recommendation System for BCI User Experience Evaluation," Technical Report, Tunis El Manar University, Tunisia, 2024.

L. Cheng, Y. Zhang, and C. Xu, "A Survey on Deep Learning for Recommender Systems: From Collaborative Filtering to Content and Context Enrichment," IEEE Access, vol. 10, pp. 12345--12370, 2022.

X. Xu, J. Zhang, and W. Li, "Hybrid Recommendation System Based on Graph Neural Network and Collaborative Filtering," IEEE Transactions on Neural Networks and Learning Systems, 2021.

Y. Zhang and X. Chen, "Explainable Recommendation: A Survey and New Perspectives," ACM Computing Surveys (CSUR), vol. 53, no. 5, pp. 1--38, 2020.

J. Wang, Y. Liu, and T. Li, "Federated Recommendation Systems: Concepts, Methods, and Future Directions," ACM Transactions on Intelligent Systems and Technology, vol. 13, no. 1, pp. 1--28, 2022.

W. Gao, L. Chen, and H. Yu, "A Survey on Neural Recommendation: From Collaborative Filtering to Content-Aware and Explainable Techniques," AI Review, Springer, vol. 56, no. 3, pp. 467--495, 2023.

sun2020deep L. Sun, F. Wang, and J. Zhou, "Deep Reinforcement Learning for Recommender Systems: A Review, IEEE Access, vol. 8, pp. 135443--135461, 2020

Y. Zhang, M. Zhou, and H. Wu, "Sensitivity Analysis of Matrix Factorization for Recommendation Systems," Data Mining and Knowledge Discovery, Springer, vol. 35, pp. 988--1010, 2021.

Q. Huang, S. Wang, and B. Zhang, "A Survey on Session-Based Recommendation with Deep Neural Networks," ACM Transactions on Information Systems, vol. 40, no. 2, pp. 1--38, 2022.

W. Liang, X. Li, and Y. Zhang, "Graph-Based Recommendation: Techniques, Applications and Performance Journal of Intelligent Information Systems, Springer, vol. 61, pp. 213--236, 2023.

J. Lin and Z. Zhou, "A Review of Context-Aware Recommender Systems: Trends and Future Directions," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 3, pp. 1450--1468, 2021.

**Appendix**

# Data Loading and Preprocessing

The following Python code was used to load and preprocess the dataset:

import pandas as pd  
  
# Load the dataset  
electronics\_data = pd.read\_csv("/kaggle/input/amazon-product-reviews/ratings\_Electronics (1).csv",   
 names=['userId', 'productId','Rating','timestamp'])  
  
# Drop timestamp column  
electronics\_data.drop(['timestamp'], axis=1, inplace=True)

# Exploratory Data Analysis (EDA)

Code for exploring the rating distribution:

import seaborn as sns  
import matplotlib.pyplot as plt  
  
# Plot rating distribution  
sns.countplot(x='Rating', data=electronics\_data)  
plt.title('Ratings Distribution')  
plt.xlabel('Rating')  
plt.ylabel('Count')  
plt.show()

# Model Implementations

Examples of collaborative filtering and matrix factorization code:

from surprise import KNNWithMeans, Dataset, Reader  
  
# Use Surprise library for KNN collaborative filtering  
reader = Reader(rating\_scale=(1, 5))  
data = Dataset.load\_from\_df(filtered\_data[['userId', 'productId', 'Rating']], reader)  
trainset = data.build\_full\_trainset()  
  
algo = KNNWithMeans(k=5, sim\_options={'name': 'pearson\_baseline', 'user\_based': False})  
algo.fit(trainset)

# Neural Collaborative Filtering (NCF)

Keras implementation for NCF:

from tensorflow.keras.models import Model  
from tensorflow.keras.layers import Input, Embedding, Flatten, Concatenate, Dense, Dropout  
  
# Input layers  
user\_input = Input(shape=(1,), name='user\_input')  
item\_input = Input(shape=(1,), name='item\_input')  
  
# Embedding layers  
user\_embedding = Embedding(input\_dim=num\_users, output\_dim=50)(user\_input)  
item\_embedding = Embedding(input\_dim=num\_items, output\_dim=50)(item\_input)  
  
# Flatten and concatenate  
user\_vec = Flatten()(user\_embedding)  
item\_vec = Flatten()(item\_embedding)  
concat = Concatenate()([user\_vec, item\_vec])  
  
# Dense layers  
fc1 = Dense(128, activation='relu')(concat)  
drop1 = Dropout(0.5)(fc1)  
fc2 = Dense(64, activation='relu')(drop1)  
output = Dense(1)(fc2)  
  
# Compile model  
model = Model([user\_input, item\_input], output)  
model.compile(optimizer='adam', loss='mse')

# Evaluation and Hybrid Prediction

Evaluate individual models and ensemble results:

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error  
  
# Combine predictions  
hybrid\_preds = (ncf\_preds + als\_preds + knn\_preds) / 3  
  
# Evaluate hybrid model  
rmse = mean\_squared\_error(y\_test, hybrid\_preds, squared=False)  
mae = mean\_absolute\_error(y\_test, hybrid\_preds)  
print(f"Hybrid RMSE: {rmse}, MAE: {mae}")