Journal of Intelligent Computing, Systems, and Applications

e-ISSN: 3123-1705
Vol. 1 No. 2 (2026) 36-48
https://journal.del.ac.id/jicsa

JICSA

Comparative Analysis of Collaborative Filtering Models on Highly Sparse Book Datasets

Jeremy Nainggolan[1], Reinaldy Hutapea[1], Tennov Pakpahan[1], Ioka Purba[1], Mickael Sitompul[1]

[1] *Institut Teknologi Del, Faculty of Informatics and Electrical Engineering, Sitoluama, Laguboti, 22381, INDONESIA*

*Corresponding Author: iss22002@students.del.ac.id

Article Info
Received: 3 February 2026
Revised: 04 April 2026
Accepted: 04 May 2026
Available online: 05 May 2026

Keywords

Collaborative Filtering; Alternating Least Squares; Matrix Factorization; Data Sparsity; Implicit Feedback; Book Recommendation; MAP@10

Abstract

The rapid growth of digital information necessitates robust recommender systems to filter relevant content effectively. This study addresses the challenge of generating personalized book recommendations using a highly sparse dataset (>99% sparsity) derived from the Recommender System 2023 Challenge. The primary objective is to predict user interest based on implicit feedback and optimize the ranking quality of Top-N recommendations. We conduct a comparative analysis between a memory-based baseline, Item-Based Collaborative Filtering (IBCF), and a model-based approach, Alternating Least Squares (ALS). The methodology incorporates k-core decomposition for data preprocessing to mitigate sparsity issues, followed by rigorous hyperparameter optimization via grid search to tune latent factors, regularization parameters, and confidence weights. Experimental results demonstrate that the baseline IBCF model struggles significantly with data sparsity, yielding a Mean Average Precision at 10 (MAP@10) of only 0.0018. In contrast, the hyperparameter-tuned ALS model achieves a MAP@10 of 0.0348, representing a 19.3-fold improvement over the baseline and a 3.3-fold improvement over the default ALS configuration. These findings confirm that matrix factorization techniques, when systematically optimized, significantly outperform memory-based methods in handling sparse implicit feedback, providing a scalable and accurate solution for ranking tasks.

1. Introduction

The development of recommender systems has become one of the major research and application areas in machine learning and data mining. Their primary role is to help users cope with the rapidly growing volume of information and identify the most relevant items among numerous alternatives. The Recommender System 2023 Challenge – Polimi introduces a new challenge in this domain. Although the provided dataset contains explicit interactions (ratings), the main objective of the competition is to predict user interest (based on ratings ≥ 4), which effectively transforms the task into an implicit-feedback recommendation problem. Participants are required to generate a ranked Top-N recommendation list consisting of 10 items for each user, evaluated primarily using Mean Average Precision at 10 (MAP@10), a metric that is highly sensitive to the quality of the ranking order. The fundamental issues that arise include how the system can personalize item relevance, mitigate strong popularity bias, and handle data sparsity within the user–item interaction matrix. This context demands an approach that is not only predictively accurate but also conceptually sound.

As an initial step, implementing a baseline Item-Based Collaborative Filtering (IBCF) model is necessary. This model serves both as a baseline and as a means to validate that the evaluation pipeline functions correctly. However, memory-based approaches such as IBCF have inherent limitations: they struggle to handle sparsity effectively, tend to be biased toward popular items, and cannot model users' latent preferences. Therefore, to address these weaknesses, this project focuses on implementing a more advanced model-based approach, specifically Alternating Least Squares (ALS) for implicit data. This method is chosen for its strong capability in modeling implicit feedback. ALS learns latent user and item factors using a confidence-weighting scheme, which enables it to distinguish between negative interactions and missing values. This makes ALS more adaptable to uncertain preferences and allows it to directly optimize latent representations for ranking tasks, which is crucial for improving MAP@10.

The main objective of this project is to implement a recommender system that can produce an optimally ranked and relevant Top-10 list for each user. Success is validated through a significant improvement in MAP@10 compared to the baseline model (IBCF). Thus, the project not only contributes practically toward achieving the competition's targets but also offers academic value through comparative analysis and empirical testing of the effectiveness of matrix factorization (ALS) and its hyperparameter optimization in handling large-scale implicit feedback. The hypothesized outcome is a recommender system that is not only strong in predictive performance but also consistent and methodologically valid, making it defensible both in an academic competition setting and as a real-world development case study.

2. Literature Review

2.1 Recommendation System

A recommender system can be described as a set of software tools and methods that generate suggestions to support user decision-making (e.g., what items to buy, what movies to watch, or what news to read) [11]. Recommender systems are also positioned as a practical way to reduce information overload by extracting user preferences from large datasets and delivering more individualized services [11].

In addition, recommendation services can be framed around personalization, where different users receive different suggestions, in contrast to non-personalized systems that provide the same suggestions to all users [6]. The core methodological families commonly discussed are content-based filtering, collaborative filtering, and hybrid approaches [11]. Content-based methods rely on item descriptions and a single user's profile to recommend items that are similar to what that user previously preferred [11]. Collaborative filtering, in contrast, leverages patterns across many users by recommending items based on similarities in behaviors or preferences among users [11].

Hybrid recommenders combine content-based and collaborative signals in multiple ways (e.g., integrating one method into another, running methods separately and merging outputs, or building a unified model) to reduce the limitations of single-technique systems [6]. Across the literature, several recurring challenges shape recommender-system design and evaluation, including changing user preferences, data sparsity, scalability constraints, synonymy issues, and privacy concerns [11]. The survey in agriculture-oriented recommendation research further highlights that content-based approaches may suffer from over-specialization, while collaborative filtering is often affected by cold-start, sparsity, and shilling attacks, which can degrade recommendation quality [12]. To mitigate such issues—especially sparsity—researchers frequently adopt hybrid strategies that mix collaborative filtering with content-based information [12].

To address the efficiency limitations of memory-based collaborative filtering (which can require full-matrix information and therefore higher time and memory costs), studies report a shift toward model-based approaches with reduced resource requirements [12]. In this direction, matrix factorization methods and other machine learning/data mining techniques are widely used alongside collaborative filtering to improve performance and better cope with large-scale data [12].

More recently, deep learning has been leveraged to enhance representation learning for user–item interactions, with evidence from a systematic review indicating that autoencoders are among the most widely adopted deep learning architectures for recommender systems, followed by CNNs and RNNs [4]. The same review notes that precision and RMSE are among the most commonly used evaluation metrics for deep learning-based recommender systems, underscoring that metric selection should align with the intended objective (e.g., predictive accuracy versus recommendation quality) [4].

2.2 Collaborative Filtering

Collaborative Filtering (CF) remains a predominant paradigm in recommender systems research, predicated on the assumption that users with similar historical preference patterns are likely to share similar choices in the future [10]. Unlike content-based approaches that rely on item attributes, CF algorithms operate directly on the user-item interaction matrix, attempting to predict missing entries to generate personalized recommendations (Torkashvand et al., 2023).

The taxonomy of CF algorithms is primarily divided into memory-based and model-based approaches [10]. Memory-based methods, also known as neighborhood-based techniques, compute recommendations by identifying similar users (User-based) or items (Item-based) using similarity metrics such as the Pearson Correlation Coefficient or Cosine Similarity [3]. While intuitive, these methods often struggle with scalability. Conversely, model-based approaches employ interaction data to train predictive models. Techniques such as Matrix Factorization (MF) decompose the sparse rating matrix into latent factors, while recent advancements have integrated Deep Learning architectures—including Autoencoders and Convolutional Neural Networks (CNNs)—to capture complex, non-linear user-item relationships.

Despite its efficacy, CF implementations face significant challenges, most notably data sparsity and the cold-start problem [5]. The cold-start phenomenon occurs when the system lacks sufficient interaction data for new users (new user problem) or new items (new item problem), rendering standard CF algorithms ineffective. To mitigate these issues, strategies such as Active Learning have been proposed to selectively elicit ratings that maximize information gain [5].

Evaluation of CF performance is typically conducted using predictive accuracy metrics, such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), which measure the deviation between predicted and actual ratings [8]. Furthermore, classification metrics like Precision and Recall are utilized to assess the relevance of Top-N recommendations. Contemporary evaluation frameworks have also expanded to include user-centric dimensions such as novelty and trust, providing a more holistic view of system quality [3].

2.3 IBCF

The evolution of Recommender Systems (RS) has seen a significant paradigm shift from User-Based Collaborative Filtering (UBCF) toward Item-Based Collaborative Filtering (IBCF) to address inherent scalability and performance limitations. Predicated on the assumption that a target user is likely to prefer items similar to those they have rated highly in the past, IBCF analyzes the relationships between items rather than the similarities between users. A critical theoretical advantage of this approach, as highlighted by [9], is the relative stability of item-item relationships compared to the volatile nature of user preferences. While user neighborhoods require frequent re-computation to reflect changing tastes, item similarities tend to be static, allowing for offline computation and improved real-time response rates in high-traffic environments [1]. Furthermore, IBCF has demonstrated superior robustness in handling the "grey sheep" phenomenon—users with eccentric preferences who effectively have no peers in a user-based cluster—by focusing on the attributes of the items themselves rather than social matching [9].

The operational efficacy of the IBCF model is contingent upon the accuracy of the similarity measurement and the subsequent rating prediction mechanism. In a standard implementation, the algorithm constructs a similarity matrix by analyzing co-rated vectors from the user-item dataset, commonly utilizing metrics such as Cosine Similarity or Adjusted Cosine Similarity to normalize for user rating variances [7]. Once the k-nearest neighbors for a target item are identified, the system predicts the user's rating through a weighted aggregation method, where the similarity scores serve as weights [7]. However, the reliance on co-rated items introduces a significant dependency on the density of the dataset. [7] argue that while traditional metrics function adequately in dense matrices, their performance degrades precipitously in sparse data environments, where the overlap between item vectors is minimal or non-existent.

Consequently, recent scholarship has focused on mitigating the challenges of data sparsity and the cold-start problem through the development of novel similarity measures and hybrid architectures. Recognizing that standard cosine measures often yield unreliable results when rating information is scarce, [7] proposed a method utilizing vague sets and Kullback-Leibler (KL) divergence to capture user preference probabilities more accurately. Similarly, [1] introduced advanced similarity measures incorporating probability density functions to enhance prediction accuracy in sparse datasets. Beyond similarity metrics, researchers have also explored hybrid models; for instance, [1] demonstrated that optimizing weighting parameters between user-based and item-based approaches can balance positively and negatively correlated neighbors, thereby refining prediction baselines. Despite these advancements, the cold-start problem for entirely new items remains a persistent research gap, necessitating further investigation into algorithms that can relax the dependency on explicit ratings or integrate auxiliary information more effectively.

2.4 Alternating Least Square

The Alternating Least Squares (ALS) algorithm and its variants constitute a foundational pillar in the domain of dimensionality reduction and latent factor analysis, particularly within the context of Recommender Systems (RS) and Collaborative Filtering (CF). As a matrix factorization technique, ALS is widely recognized for its parallelizability and efficacy in handling large-scale, sparse datasets. [2] demonstrates the continued relevance of ALS in big data environments, specifically utilizing Apache Spark to process movie rating datasets. Their work highlights that despite the emergence of complex neural architectures, the ALS framework remains a robust standard for minimizing Root Mean Square Error (RMSE) in distributed computing scenarios, offering a favorable trade-off between computational scalability and prediction accuracy.

In recent years, the academic discourse has increasingly scrutinized the comparative performance of ALS against Deep Learning (DL) based recommendation models. While the prevailing narrative often positions neural networks as superior, recent rigorous re-evaluations challenge this assumption. [13] conducted an extensive benchmarking study on Implicit ALS (iALS), revealing that older matrix factorization methods are frequently underestimated due to suboptimal hyperparameter tuning in comparative studies. Their findings indicate that when embedding dimensions and regularization parameters are properly optimized, iALS achieves accuracy comparable to, or even exceeding, state-of-the-art variational autoencoders and other deep learning baselines. This suggests that the theoretical ceiling of linear factorization models has not yet been fully exploited and that the reported gains of deep models may stem partly from experimental inconsistencies rather than inherent architectural superiority.

Beyond standard recommendation tasks, the theoretical principles of alternating optimization have been adapted into deep learning architectures through techniques known as algorithm unrolling or unfolding. [8] exemplify this hybrid approach in the field of hyperspectral unmixing, proposing the SNMF-Net. By unfolding the iterative steps of Non-negative Matrix Factorization (NMF)—which shares the alternating update logic of ALS—into a deep neural network layers, they successfully combine the interpretability of model-based optimization with the learning capability of data-driven networks. This convergence indicates a trend where alternating least squares principles are no longer viewed merely as standalone solvers but as differentiable modules within larger, non-linear learning systems.

However, the application of least squares estimation extends beyond static matrix completion into dynamic parameter estimation for nonlinear systems. [9] explore recursive variations of least squares algorithms for identifying feedback nonlinear controlled autoregressive systems. While distinct from the batch-processing nature of standard ALS, their work on recursive least squares and stochastic gradient convergence underscores the mathematical versatility of least squares objectives in handling bilinear-in-parameter models. Collectively, these studies suggest that while ALS is a mature technique, significant research opportunities remain in refining its integration with deep learning frameworks and optimizing its application in high-sparsity, implicit-feedback environments.

2.5 Data Sparsity

In modern recommendation system competitions, the phenomenon of data sparsity is a fundamental challenge that defines the problem space. This problem arises due to the extreme imbalance between the number of users and items compared to the observed interactions. [8] explain that this sparsity problem causes the system to fail to generate reasonable recommendations due to the lack of rating data available to connect user preferences. In industrial-scale datasets, interaction matrices are often dominated by empty values, where these information gaps make it difficult for algorithms to accurately capture user behavior patterns.

This situation becomes even more complex in the context of implicit feedback, which is the focus of this project. Unlike explicit feedback, implicit data (such as clicks or viewing history) is passive and contains a high degree of uncertainty. [8] highlight that although implicit data offers scalability, it is prone to ambiguity and noise; the absence of interaction does not always reflect dislike, but could be because the user is unaware of the item's existence. Furthermore, [8] emphasize that factors such as human error and uncertainty in user behavior introduce significant noise, which, if not handled properly, will degrade the accuracy and robustness of recommendations.

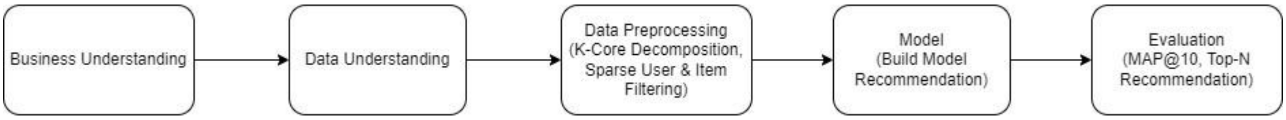
Extreme data sparsity creates a fundamental structural weakness for conventional approaches, most notably Item-Based Collaborative Filtering (IBCF), while IBCF is reliable when data is dense, its effectiveness collapses in sparse or cold-start scenarios. The root of this issue is IBCF’s dependence on item correlations; without enough rating data to establish valid similarities, the system is effectively blinded to new items or those with few interactions. This limitation is within collaborative filtering, simple neighbor-based algorithms like kNN often struggle to compete with more sophisticated classification methods when the dataset lacks ideal properties.

To address these shortcomings, the field has moved toward model-based techniques, particularly latent factor models, which offer far greater stability. [10] suggests that by modeling interactions in a latent space—and enriching them with pseudo-implicit feedback, it is possible to estimate preferences accurately even when sparsity is severe. This approach goes beyond simply filling in gaps; it has been shown to significantly boost precision in Top-K recommendations compared to other denoising strategies.

Adopting this model-based perspective aligns directly with the competition's focus on ranking quality. [6] argue that for implicit feedback datasets, the real challenge has shifted from predicting ratings to identifying algorithms that optimize specific ranking metrics like NDCG@10. They demonstrate that the right choice of algorithm and hyperparameter tuning is essential to mirror true user preferences in this context [6]. Therefore, leveraging matrix factorization specifically its ability to filter signal from noise is a critical strategy for improving the Mean Average Precision (MAP@10) in this project.

3. Methodology / Proposed Method

This study uses a structured framework adapted from the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology to guide the research process in an organized manner, from understanding the problem to evaluating the model. The research workflow includes data collection, data preprocessing, model development (both baseline and proposed models), and performance evaluation using ranking-based metrics.



3.1 Data Description and Acquisition

The experimental dataset was obtained from a book recommendation challenge, comprising implicit user-item interactions in the form of user reading histories. An initial exploratory data analysis revealed significant challenges inherent to the dataset. Specifically, the interaction matrix exhibited an extreme sparsity level exceeding 99%, wherein the vast majority of user-item pairs contained no observed interactions. This characteristic is commonly encountered in real-world recommendation scenarios, particularly in domains such as book recommendations where users typically interact with only a negligible fraction of the available item catalog. The dataset structure consisted of user identifiers, item identifiers (represented as ISBN codes), and implicit feedback signals indicating user engagement with particular items.

3.2 Data Preprocessing

To mitigate the adverse effects of extreme data sparsity on model performance, a rigorous preprocessing pipeline was implemented. The central component of this pipeline was the application of k-core decomposition, a graph-theoretic filtering technique designed to retain only the most densely connected subgraph of user-item interactions. In this approach, users and items that did not meet a minimum threshold of interactions were iteratively removed from the dataset. This procedure was applied recursively until convergence, ensuring that all remaining users and items possessed at least k interactions within the filtered dataset.

The rationale underlying this preprocessing step is twofold. First, users with insufficient interaction histories provide inadequate signal for learning meaningful latent representations, potentially introducing noise into the collaborative filtering process. Second, items with minimal interactions lack sufficient collaborative evidence to establish reliable similarity relationships with other items. By enforcing a minimum interaction threshold through k-core decomposition, the resulting dataset achieves a substantially improved density while preserving the structural integrity of the user-item interaction graph. This preprocessing step is essential for enabling effective model training in scenarios characterized by implicit feedback and high sparsity.

3.3 Interaction Matrix Construction

Following the preprocessing phase, the filtered interaction data was transformed into a sparse user-item interaction matrix suitable for matrix factorization algorithms. Given the implicit nature of the feedback signals, all observed interactions were encoded as binary indicators, where a value of one denoted an observed interaction and zero indicated the absence of recorded engagement. The resulting matrix was stored in compressed sparse row (CSR) format to facilitate computational efficiency during model training, particularly given the large dimensionality of the user and item spaces.

3.4 Baseline Model Training

The Alternating Least Squares (ALS) algorithm was selected as the primary collaborative filtering approach due to its demonstrated effectiveness in handling implicit feedback data and its computational scalability to large-scale datasets. ALS operates by decomposing the user-item interaction matrix into two lower-dimensional latent factor matrices—one representing user preferences and another representing item characteristics—through an iterative optimization procedure that alternates between fixing user factors and optimizing item factors, and vice versa. An initial baseline model was trained using default hyperparameter configurations to establish a reference performance benchmark. This baseline served dual purposes: first, as a point of comparison for evaluating the impact of subsequent hyperparameter optimization; and second, as a source of initialized model weights for transfer learning in later experimental stages.

3.5 Hyperparameter Optimization

To enhance model performance beyond the baseline configuration, a systematic hyperparameter tuning procedure was conducted via grid search. The hyperparameter space explored encompassed three primary dimensions critical to ALS performance in implicit feedback settings:

1. Latent Factors: The dimensionality of the latent representation space, which determines the model's capacity to capture complex user-item interaction patterns. Higher dimensionality enables the representation of more nuanced preference structures but increases computational cost and the risk of overfitting.
2. Regularization Parameter (λ): A penalty term applied to prevent overfitting by constraining the magnitude of learned latent factors. Appropriate regularization is particularly critical in sparse data environments where overfitting to observed interactions poses a significant risk.
3. Confidence Weighting Parameter (α): A scaling factor that modulates the confidence assigned to observed interactions relative to unobserved entries. In implicit feedback scenarios, observed interactions are assumed to indicate positive preference with varying degrees of certainty, while unobserved interactions may reflect either negative preference or simply lack of exposure.

Each hyperparameter combination was evaluated using a held-out validation set, with Mean Average Precision at rank 10 (MAP@10) serving as the primary optimization criterion. The optimal configuration was selected based on maximal validation performance.

3.6 Transfer Learning and Model Fine-Tuning

A transfer learning strategy was employed to leverage knowledge acquired during baseline model training. Specifically, the latent factor matrices learned by the baseline ALS model were utilized to initialize the corresponding matrices in subsequent fine-tuning experiments. This warm-start initialization approach offers several advantages over random initialization: it accelerates convergence by providing a favorable starting point in the optimization landscape, reduces sensitivity to local minima, and enables more effective exploration of the hyperparameter space by building upon previously learned representations. During fine-tuning, the pre-initialized model was trained for additional iterations using the optimal hyperparameter configuration identified through grid search. Early stopping based on validation MAP@10 was implemented to prevent overfitting and ensure that training terminated at the point of maximal generalization performance.

3.7 Evaluation Metrics

Model performance was evaluated using Mean Average Precision at rank 10 (MAP@10), a ranking-oriented metric specifically designed for top-N recommendation tasks. MAP@10 quantifies the quality of the ranked recommendation lists by computing the average precision across all users, where precision at each rank position is weighted by relevance. Formally, for a given user, Average Precision is calculated as:

$$AP@K = \frac{1}{\min\,(m, K)} \sum_{k=1}^{K} P\,(k) x\, rel(k)$$

where P(k) denotes precision at cutoff k, rel(k) is a binary indicator of relevance at position k, and m represents the total number of relevant items for the user. The MAP@10 metric is subsequently obtained by averaging AP@10 across all users in the evaluation set. The selection of MAP@10 as the primary evaluation metric is motivated by its alignment with practical recommendation objectives, where the quality of top-ranked items is of paramount importance. Unlike pointwise metrics such as Root Mean Squared Error (RMSE), ranking metrics directly assess the model's ability to surface relevant items at prominent positions in the recommendation list, thereby providing a more meaningful measure of recommendation utility in implicit feedback scenarios.

3.8 Model Comparison Framework

To comprehensively assess the effectiveness of the proposed methodology, a comparative evaluation framework was established incorporating multiple baseline approaches:

1. Item-Based Collaborative Filtering (IBCF): A memory-based approach that computes item-item similarity based on co-occurrence patterns and generates recommendations by aggregating the similarity-weighted ratings of items previously consumed by the target user.
2. Default ALS: The baseline matrix factorization model trained with standard hyperparameter settings, serving as a reference point for evaluating the impact of hyperparameter tuning.
3. Tuned ALS with Transfer Learning: The optimized model incorporating hyperparameter tuning and transfer learning from baseline weights, representing the proposed methodology.

This comparative framework enables systematic assessment of the incremental contributions of each methodological component to overall recommendation performance.

4 Result and Discussion

4.1 Experimental Result

This section presents the experimental results obtained from the evaluation of collaborative filtering models on highly sparse implicit feedback data. Three models were compared: Item-Based Collaborative Filtering (IBCF) as the baseline, Alternating Least Squares (ALS) with default parameters, and ALS with optimized hyperparameters obtained through grid search. The evaluation metric employed was Mean Average Precision at 10 (MAP@10), which measures the quality of top-10 recommendations by considering both relevance and ranking position.

Table 1 Performance comparison of collaborative filtering models based on MAP@10

Model	Model Type	Key Configuration	MAP@10	Improvement over IBCF
IBCF	Memory-based CF	Item-item similarity	0.0018	1.0×
Default ALS	Model-based CF	Default parameters	0.0106	5.9×
Tuned ALS	Model-based CF	256 factors, $\alpha = 30$, $\lambda = 0.05$	0.0348	19.3×

The results demonstrate a clear progression in model performance. The baseline IBCF model achieved a MAP@10 of 0.0018, indicating severe limitations in handling highly sparse data. The default ALS model yielded a substantially higher MAP@10 of 0.0106, representing a 5.9-fold improvement over the baseline. The hyperparameter-tuned ALS model achieved the highest performance with a MAP@10 of 0.0348, corresponding to a 19.3-fold improvement over IBCF and a 3.3-fold improvement over the default ALS configuration.

4.2 Analysis of Baseline IBCF Performance

The Item-Based Collaborative Filtering model exhibited the lowest performance among all evaluated approaches. This outcome can be attributed to several factors inherent to memory-based methods when applied to highly sparse datasets. IBCF relies on computing item-item similarity matrices based on co-occurrence patterns in user interactions. In datasets with sparsity levels exceeding 99%, the probability of finding meaningful co-occurrence patterns diminishes significantly, resulting in unreliable similarity estimates. Furthermore, IBCF suffers from the cold-start problem for items with limited interaction history, as the similarity computation requires sufficient overlapping users between item pairs. The sparse nature of the dataset exacerbates this issue, leading to similarity matrices dominated by zero or near-zero values. Consequently, the recommendation quality degrades substantially, as reflected in the low MAP@10 score of 0.0018.

4.3 Impact of Matrix Factorization

The transition from memory-based IBCF to model-based ALS demonstrates the superiority of matrix factorization techniques in handling sparse implicit feedback data. Even with default parameters, the ALS model achieved a MAP@10 of 0.0106, representing a significant improvement over the baseline. This enhancement can be attributed to the fundamental differences in how these approaches model user-item interactions. Matrix factorization methods decompose the sparse user-item interaction matrix into lower-dimensional latent factor representations. These latent factors capture underlying patterns and preferences that are not explicitly observable in the raw interaction data. Unlike IBCF, which requires direct co-occurrence for similarity computation, ALS can infer relationships between users and items through their shared latent space representations. This property enables the model to make meaningful predictions even when direct interaction overlap is minimal. The implicit feedback formulation of ALS, which treats interactions as confidence indicators rather than explicit ratings, aligns well with the nature of the dataset. The confidence weighting mechanism allows the model to differentiate between varying levels of user engagement, thereby providing more nuanced preference modeling.

4.4 Effect of Hyperparameter Tuning

The most substantial performance improvement was observed following hyperparameter optimization. The tuned ALS model, configured with 256 latent factors, a confidence weight (α) of 30, and a regularization parameter (λ) of 0.05, achieved a MAP@10 of 0.0348. This represents a 3.3-fold improvement over the default ALS configuration, underscoring the critical importance of hyperparameter tuning in matrix factorization models.

4.2.1. Latent Factors.

The selection of 256 latent factors provides sufficient representational capacity to capture complex user-item interaction patterns. A higher number of factors enables the model to encode more nuanced preferences and item characteristics. However, this must be balanced against the risk of overfitting, which is mitigated through appropriate regularization.

4.2.2. Confidence Weight (α).

The confidence weight parameter of 30 amplifies the influence of observed interactions relative to unobserved entries. In implicit feedback scenarios, this parameter controls the degree to which the model trusts positive interactions. A higher α value assigns greater importance to actual user engagements, allowing the model to more strongly distinguish between interacted and non-interacted items during optimization.

4.2.3 Regularization (λ).

The regularization parameter of 0.05 provides a moderate constraint on model complexity. This value prevents overfitting by penalizing large factor values while still allowing the model sufficient flexibility to capture meaningful patterns. The balance achieved through this regularization setting contributes to improved generalization performance on the validation set.

4.5 Discussion

The experimental results validate the hypothesis that matrix factorization with optimized hyperparameters significantly outperforms traditional memory-based collaborative filtering approaches in highly sparse implicit feedback environments. The progression from IBCF (MAP@10 = 0.0018) to default ALS (MAP@10 = 0.0106) to tuned ALS (MAP@10 = 0.0348) illustrates the cumulative benefits of adopting advanced modeling techniques and systematic optimization procedures. The relatively low absolute MAP@10 values across all models reflect the inherent difficulty of the recommendation task under extreme sparsity conditions. With dataset sparsity exceeding 99%, even well-optimized models face fundamental limitations in capturing user preferences from minimal interaction signals. Nevertheless, the relative improvements achieved through hyperparameter tuning demonstrate that substantial gains remain possible through careful model configuration. The findings also highlight the importance of evaluation metrics aligned with the recommendation task. MAP@10 provides a ranking-focused assessment that prioritizes the quality of top recommendations, which directly corresponds to the user experience in practical recommendation scenarios. The observed improvements in MAP@10 indicate that the tuned ALS model produces more relevant items at higher positions in the recommendation list. From a practical perspective, these results suggest that practitioners working with sparse implicit feedback data should prioritize matrix factorization approaches over memory-based methods. Furthermore, investment in hyperparameter tuning yields substantial returns in recommendation quality, with the tuned model achieving nearly 20 times the performance of the baseline IBCF approach.

4.6 Limitations

Despite the improvements achieved, several limitations should be acknowledged. First, the absolute MAP@10 values remain relatively low, indicating room for further enhancement through alternative modeling approaches or feature engineering. Second, the grid search optimization, while effective, explores a limited hyperparameter space and may not identify the global optimum. More sophisticated optimization techniques, such as Bayesian optimization, could potentially yield further improvements. Third, the evaluation was conducted on a single dataset, and the generalizability of the optimal hyperparameter configuration to other domains requires further investigation.

5 Conclusion

This study presented a comparative analysis of collaborative filtering approaches for book recommendation systems operating on highly sparse implicit feedback data. Three models were evaluated: Item-Based Collaborative Filtering (IBCF) as the baseline, Alternating Least Squares (ALS) with default parameters, and ALS with optimized hyperparameters. The experimental results demonstrated that the baseline IBCF model achieved a MAP@10 of 0.0018, the default ALS model improved performance to 0.0106, and the hyperparameter-tuned ALS model—configured with 256 latent factors, confidence weight (α) of 30, and regularization (λ) of 0.05 achieved the highest performance with a MAP@10 of 0.0348. These results represent a 19.3-fold improvement over IBCF and a 3.3-fold improvement over the default ALS configuration, validating the superiority of matrix factorization techniques in sparse implicit feedback environments.

The findings of this research contribute to the field of recommender systems by providing empirical evidence that matrix factorization approaches significantly outperform memory-based methods under extreme data sparsity conditions. The substantial performance improvement achieved through systematic hyperparameter tuning underscores the critical importance of model optimization in practical recommendation system deployment. Furthermore, the methodology employed in this study, including k-core decomposition for preprocessing and transfer learning for model initialization, presents a replicable framework for developing recommendation systems under challenging data conditions characterized by sparsity levels exceeding 99%.

From a practical perspective, organizations dealing with sparse user interaction data should prioritize matrix factorization approaches over traditional memory-based methods, as the investment in hyperparameter tuning yields substantial returns in recommendation quality. However, several limitations should be acknowledged: the absolute MAP@10 values remain relatively low due to inherent data sparsity challenges, the grid search optimization explores a limited hyperparameter space, and the generalizability of the optimal configuration to other domains requires further validation. These limitations notwithstanding, the relative improvements achieved demonstrate that meaningful performance gains remain attainable through careful model selection and configuration.

Future research directions include the exploration of deep learning-based approaches such as neural collaborative filtering, the incorporation of side information through hybrid recommendation methods, the application of advanced optimization techniques such as Bayesian optimization, and the extension of evaluation frameworks to include additional metrics such as NDCG and diversity measures. In conclusion, this study demonstrates that systematic hyperparameter optimization of matrix factorization models can substantially improve recommendation quality in sparse data environments, providing practical guidance for practitioners seeking to develop effective implicit feedback recommendation systems.

Conflict of Interest Statement

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Author Contributions

J.N. (Jeremy Nainggolan): Methodology, Software, Validation.

R.H. (Reinaldy Hutapea): Writing – Original Draft, Resources.

T.P. (Tennov Pakpahan): Software, Investigation, Validation.

I.P. (Ioka Purba): Formal Analysis, Visualization, Writing – Original Draft.

M.S. (Mickael Sitompul): Formal Analysis, Writing – Review & Editing.

All authors have read and agreed to the published version of the manuscript

References

[1] Abdalla, H. I., Amer, A. A., Amer, Y. A., Nguyen, L., & Al-Maqaleh, B. (2023). Boosting the Item-Based Collaborative Filtering Model with Novel Similarity Measures. *International Journal of Computational Intelligence Systems*, *16*(1), 1–16. https://doi.org/10.1007/s44196-023-00299-2

[2] Awan, M. J., Khan, R. A., Nobanee, H., Yasin, A., Anwar, S. M., Naseem, U., & Singh, V. P. (2021). A recommendation engine for predicting movie ratings using a big data approach. *Electronics (Switzerland)*, *10*(10). https://doi.org/10.3390/electronics10101215

[3] Bobadilla, J., Hernando, A., Ortega, F., & Bernal, J. (2011). A framework for collaborative filtering recommender systems. *Expert Systems with Applications*, *38*(12), 14609–14623. https://doi.org/10.1016/j.eswa.2011.05.021

[4] Da'u, A., & Salim, N. (2020). Recommendation system based on deep learning methods: a systematic review and new directions. In *Artificial Intelligence Review* (Vol. 53, Number 4). Springer Netherlands. https://doi.org/10.1007/s10462-019-09744-1

[5] Elahi, M., Ricci, F., & Rubens, N. (2016). A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*, *20*, 29–50. https://doi.org/10.1016/j.cosrev.2016.05.002

[6] Gupta, S. (2020). A Literature Review on Recommendation Systems sept 2020. *Int. Res. J. Eng. Technol*, 3600–3605. https://www.academia.edu/download/64789612/IRJET_V7I9633.pdf

[7] Hartatik, H., Nurhayati, S. D., & Widayani, W. (2021). Sistem Rekomendasi Wisata Kuliner di Yogyakarta dengan Metode Item-Based Collaborative Filtering. *Journal Automation Computer Information System*, *1*(2), 55–63. https://doi.org/10.47134/jacis.v1i2.8

[8] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, *22*(1), 5–53. https://doi.org/10.1145/963770.963772

[9] Iwendi, C., Ibeke, E., Eggoni, H., Velagala, S., & Srivastava, G. (2022). Pointer-Based Item-to-Item Collaborative Filtering Recommendation System Using a Machine Learning Model. *International Journal of Information Technology and Decision Making*, *21*(1), 463–484. https://doi.org/10.1142/S0219622021500619

[10] Papadakis, H., Papagrigoriou, A., Panagiotakis, C., Kosmas, E., & Fragopoulou, P. (2022). Collaborative filtering recommender systems taxonomy. *Knowledge and Information Systems*, *64*(1), 35–74. https://doi.org/10.1007/s10115-021-01628-7

[11] Patel, B., Desai, P., & Panchal, U. (2017). Methods of recommender system: A review. *Proceedings of 2017 International Conference on Innovations in Information, Embedded and Communication Systems, ICIIECS 2017*, *2018-Janua*, 1–4. https://doi.org/10.1109/ICIIECS.2017.8275856

[12] Patel, K., & Patel, H. B. (2020). A state-of-the-art survey on recommendation system and prospective extensions. *Computers and Electronics in Agriculture*, *178*(June), 105779. https://doi.org/10.1016/j.compag.2020.105779

[13] Rendle, S., Krichene, W., Zhang, L., & Koren, Y. (2022). Revisiting the Performance of iALS on Item Recommendation Benchmarks. *RecSys 2022 - Proceedings of the 16th ACM Conference on Recommender Systems*, 427–435. https://doi.org/10.1145/3523227.3548486