ISSN 2393 - 9850
Pages :99-108

Evaluating Similarity Measures in Collaborative Filtering: Insights into Accuracy, Precision, and Computational Performance

Sumithmon KS
Research Scholar
APJ Abdul Kalam Technological University
Palai Kerala, India
sumithmarar@gmail.com

Dr. Rahul Shajan
Department of Computer Applications
St Joseph's College of Engineering and Technology
Kottayam, Kerala, India
rahulshajan@sjcetpalai.ac.in

Abstract

Recommendation systems assist users in making informed decisions by offering personalized suggestions tailored to their preferences and behaviours. The primary goal is to connect users with relevant items through the comprehensive analysis of input data. This study evaluates the performance of four widely used similarity measures—Cosine, MSD (Mean Squared Difference), Pearson, and Pearson Baseline—within a collaborative algorithm framework, specifically employing KNN Baseline, utilizing both user-based and item-based models. Key performance metrics, including RMSE, MAE, Precision@K, Recall@K, F1 Score@K, training time, and prediction time, are employed to assess the effectiveness of these measures.

The results reveal that both Pearson and Pearson Baseline yield the highest accuracy, characterized by low RMSE and MAE values, alongside exceptional recall rates. Conversely, while MSD demonstrates faster training and prediction times, its accuracy is slightly lower, making it more suitable for time-sensitive applications. Cosine similarity strikes a balance, offering consistent performance in terms of both accuracy and speed. Notably, item-based models typically require longer training and prediction durations compared to their user-based counterparts. This study offers valuable insights into selecting the most suitable similarity measures based on trade-offs between accuracy and efficiency, guiding developers in creating optimized recommendation systems tailored to specific use cases.

Keywords: Recommendation Systems, Similarity Measures, Collaborative Filtering, KNN Baseline, Performance Metrics

1 Introduction

Recently, recommendation systems have become essential components of digital platforms. They enhance user experiences by suggesting relevant content, products, and services tailored to individual preferences(Choi, Kang, & Jeon, 2006). These systems analyze user behaviour—such as age, location, gender, and social media activity—to generate personalized recommendations. Emerging systems also integrate Internet of Things (IoT) data, including GPS signals and real-time health metrics, further refining the quality of recommendations (Kumar & Thakur, 2018a).

The exponential growth of information online has made it challenging for users to find relevant content, leading to information overload(Yildirim & Krishnamoorthy, 2008). Recommendation engines play a crucial role in filtering large volumes of data and offering relevant suggestions. Initially popularized by e-commerce platforms, these systems are now applied across various domains, including real-time route optimization, intelligent chatbots, and medical diagnosis(Kulkarni & Rodd, 2020). Beyond enhancing user satisfaction, recommendation systems drive business success by promoting the right products to the right audiences(Bouraga, Jureta, Faulkner, & Herssens, 2014).

Recommendation systems are generally divided into four categories: content-based, collaborative filtering, hybrid, and knowledge-based models. Among these, collaborative filtering has become the most widely adopted and reliable technique(Silveira, Zhang, Lin, Liu, & Ma, 2019). In recent years, innovations have emerged: deep learning-based systems leverage neural networks to uncover intricate patterns, reinforcement learning-based systems adapt recommendations in real time to enhance user engagement, and context-aware systems offer personalized suggestions based on factors like time and location. Explainable systems are also gaining popularity by building user trust through transparency in recommendations(Han & Karypis, 2005).

◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇
Received : 10 October 2024
Revised : 18 November 2024
Accepted : 21 November 2024

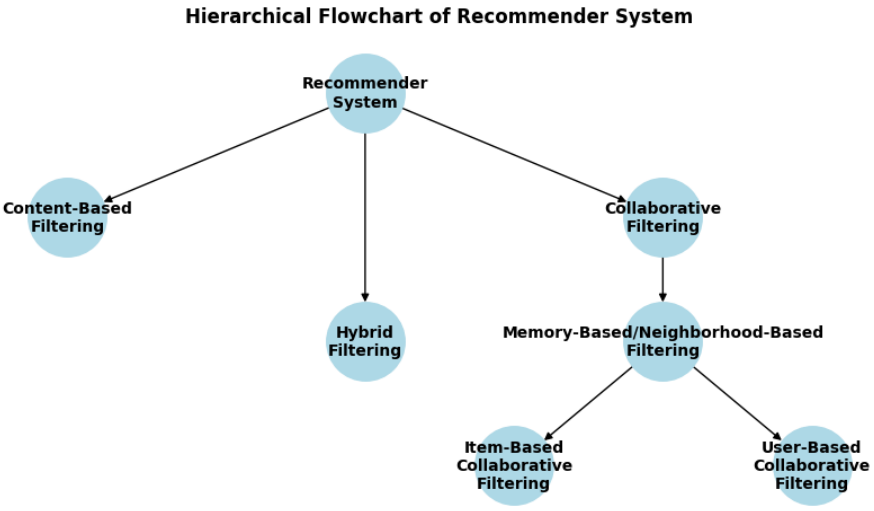

Figure 1: Hierarchical Flowchart of Recommender System

Despite their effectiveness, recommendation systems struggle with several challenges, such as data sparsity, cold start issues, and scalability. and the need to adapt to changing user behaviour. Addressing data sparsity—caused by limited user feedback—is critical, as it affects the accuracy of collaborative filtering models(Lekakos & Caravelas, 2008). To overcome these challenges, advanced machine learning and deep learning techniques are being employed, but there are still efficiency gaps to address, especially in scaling these systems and improving adaptability(Lops, De Gemmis, & Semeraro, 2011).

This paper evaluates and compares different similarity measures used with the KNN Baseline collaborative filtering algorithm. The measures analyzed include cosine similarity, mean squared difference (MSD), Pearson correlation, and Pearson baseline, across both user-based and item-based configurations. We assess these methods using various performance metrics, including accuracy (RMSE, MAE), ranking (Precision@K, Recall@K, F1 Score@K), and computational efficiency (training and prediction times). Additionally, this study proposes strategies to mitigate the impact of data sparsity by identifying users with similar preferences through enhanced similarity measures(Kumar & Thakur, 2018b).

By refining KNN Baseline collaborative filtering techniques with various similarity measures and addressing critical challenges, this study contributes to the ongoing development of recommendation systems. These enhancements aim to ensure that these systems remain accurate, efficient, and scalable, ultimately improving the relevance of recommendations and enhancing user experience across digital platforms(Isinkaye, 2021).

2 Literature Review

This literature review explores the collaborative filtering techniques employed in recommendation systems. Collaborative filtering is a widely-used method that leverages user interactions to recommend pertinent content. There are two primary types of collaborative filtering: user-based and item-based. User-based collaborative filtering identifies users with analogous preferences to suggest items they have enjoyed. This approach operates on the assumption that if two users have shared similar tastes in the past, they are likely to have similar preferences in the future. However, utilizing this method with large datasets can be challenging, as it necessitates extensive calculations to determine user similarities.(Pazzani & Billsus, 2007).

User-based collaborative filtering identifies users with similar preferences to recommend items that those users have enjoyed. The KNN Baseline algorithm recommends items to users by identifying the K nearest neighbours—users or items that are most similar based on their interactions(Shah, Gaudani, & Balani, 2016). It assumes that users who have similar preferences in the past will likely have similar preferences in the future. This method operates on the assumption that if two users have exhibited similar tastes in the past, they will likely share similar preferences in the future. However, the application of this approach to large datasets can pose challenges, as it necessitates extensive computations to determine user similarities(Duan, Jiang, & Jain, 2022).

Item-based collaborative filtering, on the other hand, puts more emphasis on the connections between things than on users. This method analyzes how similar different items are to each other based on user ratings. It tends to be more scalable than user-based filtering, making it suitable for situations where there are many users compared to the number of items available(Lapan, 2018).

To measure similarity in collaborative filtering, various methods are used. **Cosine similarity** calculates the angle between two vectors of user ratings, helping to find items that are similar based on user preferences. The formula for cosine similarity between two vectors A and B is given by:

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Mean Squared Difference (MSD) looks at the squared differences between corresponding ratings to assess similarity(Gomez-Uribe & Hunt, 2015). The formula for MSD between two items or users X and Y is expressed as:

$$\text{MSD}(X, Y) = \frac{1}{n} \sum_{i=1}^{n} (X_i - Y_i)^2$$

Pearson Correlation measures the linear relationship between two sets of ratings(Badaro, Hajj, El-Hajj, & Nachman, 2013). It is calculated using the formula:

$$\text{Pearson}(X, Y) = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n} (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^{n} (Y_i - \bar{Y})^2}}$$

where $\bar{X}$ and $\bar{Y}$ are the mean ratings of X and Y, respectively.

To evaluate the effectiveness of collaborative filtering algorithms, several metrics are used. **Accuracy metrics** like **RMSE** and **MAE** help determine how close the predicted ratings are to actual ratings(Yuyan, Xiayao, & Yong, 2019). The formulas for RMSE and MAE are:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (R_i - \hat{R}_i)^2}$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |R_i - \hat{R}_i|$$

where R_i is the actual rating and $\hat{R}_i$ is the predicted rating.

Ranking metrics such as **Precision@K**, **Recall@K**, and **F1 Score@K** evaluate the effectiveness of algorithms in recommending relevant items from the top suggestions. The formulas for these metrics are given as follows:

$$\text{Precision@K} = \frac{\text{Number of Relevant Items in the Top K}}{K}$$

$$\text{Recall@K} = \frac{\text{Number of Relevant Items in the Top K}}{\text{Total Number of Relevant Items}}$$

$$\text{F1 Score@K} = 2 \cdot \frac{\text{Precision@K} \cdot \text{Recall@K}}{\text{Precision@K} + \text{Recall@K}}$$

Collaborative filtering has found applications in various fields. Many studies have demonstrated its effectiveness in making personalized recommendations(Wang, Wang, Hsu, & Wang, 2014). User-based and item-based approaches have been shown to yield positive results, with item-based methods often outperforming user-based ones due to the stability of item data(Duan et al., 2022).

However, collaborative filtering also faces challenges. One significant issue is data sparsity, where users interact with only a small number of items, making it hard to find meaningful similarities. Scalability is another challenge, as large datasets can lead to performance issues(Stitini, Kaloun, & Bencharef, 2022). Additionally, there are trade-offs between achieving high accuracy and maintaining reasonable computational efficiency, especially for real-time recommendations(Kulkarni & Rodd, 2020).

Collaborative filtering techniques are essential for developing effective recommendation systems. Understanding the strengths and weaknesses of different methods helps in choosing the best approach for specific applications. The following sections will focus on experiments conducted with various collaborative filtering algorithms to identify the most suitable methods for enhancing user experiences in digital platforms(Jiang, Shang, & Liu, 2010).

3 Design

The dataset selected for this research is `goodbooks_10k.csv`, which serves as a comprehensive repository of user-book interaction data. After conducting a sampling process that reduced the dataset to a manageable size, we obtained a sample comprising **49,088** rows and **3** columns: `book_id`, `user_id`, and `rating`. This reduction to **5%** of the original dataset ensures that our analysis remains both representative and computationally efficient.

The dataset showcases a mean rating of approximately **3.86** with a standard deviation of **0.98**. This statistical analysis indicates a relatively consistent range of ratings provided by users, highlighting a shared understanding of the books within the dataset. The `book_id` and `user_id` columns are represented as integers, emphasizing the structured nature of the dataset. Furthermore, it includes a total of **22,717** unique users and **9,924** unique books, showcasing a rich diversity in user preferences and reading material.

Importantly, the dataset contains no missing values across any of the columns, which further ensures the integrity and reliability of the data for analysis. The robust diversity of user-book interactions captured in this dataset provides an extensive foundation for evaluating various collaborative filtering algorithms. This makes it particularly suitable for in-depth analysis and experimentation in the field of book recommendation systems, enabling us to derive meaningful insights into user preferences and enhance the accuracy of recommendations made by the system.

The design employed in this research centers around the development and evaluation of a KNN-based recommendation system, specifically designed to enhance book recommendations through the utilization of collaborative filtering techniques. The process initiates with the importation of necessary libraries, which include key components such as `Dataset`, `Reader`, `KNNBaseline`, `KFold`, and various accuracy metrics from the Surprise library, along with other relevant packages that facilitate data manipulation and analysis.

Following the library imports, the next step involves defining the file path for the dataset, exemplified by `sampled_datasets/5k.csv`. A `Reader` object is subsequently created to accurately parse the dataset, ensuring the correct interpretation of its structure. This is crucial for maintaining data integrity throughout the analysis.

Once the data is successfully loaded, various KNN algorithms are initialized. The focus here is on several similarity metrics, which are pivotal in determining user-book interactions. These metrics include Cosine Similarity, which can be applied in both user-based and item-based approaches; Mean Squared Difference (MSD), useful for evaluating collaborative relationships among users; Pearson Correlation, which assesses the linear relationship between users' ratings; and Pearson Baseline, which incorporates baseline adjustments to enhance the accuracy of recommendations.

To ensure a robust and reliable evaluation of the recommendation system, K-Fold cross-validation is implemented, with a specified number of splits set to five. This approach allows for comprehensive testing of the model's performance across different subsets of the data. During this phase, evaluation metrics, such as RMSE, MAE, precision, recall, F1 score, training times, and prediction times, are initialized and stored for subsequent analysis.

A critical aspect of this methodology is the definition of a relevant threshold for considering a recommendation as relevant. In this research, a threshold value of 3.5 is established, to retrieve the top 10 recommendations for each user. To evaluate the quality of these recommendations, a function named

`precision_recall_f1_at_10` is constructed. This function is designed to compute precision, recall, and F1 score for each user, based on the predicted ratings in comparison to the actual ratings.

The cross-validation process entails the division of the dataset into training and test sets for each fold. Within this framework, the training time is measured as the KNN algorithm is fitted to the training set. Subsequently, the prediction time is recorded as the trained algorithm predicts ratings on the test set. During this process, the evaluation metrics are computed using Surprise's built-in accuracy functions, and the previously defined `precision_recall_f1_at_10` function is invoked to calculate the precision, recall, and F1 score.

As each fold is processed, the computed metrics and times are meticulously stored in their respective lists. This systematic collection of data allows for a comprehensive analysis of the KNN-based recommendation system's performance. Finally, after all folds have been completed, the average of each metric across all folds is calculated and reported. This step not only summarizes the overall effectiveness of the recommendation system but also facilitates the identification of optimal book recommendations tailored to the diverse preferences of users, thereby enhancing the user experience in the domain of book recommendations.

The research employs a variety of tools and libraries that are pivotal in the development and implementation of the KNN-based recommendation system. The primary programming language utilized for this research is Python, which is renowned for its simplicity and versatility in data analysis and machine learning tasks.

To facilitate the recommendation system's design, the Surprise library is extensively used. Surprise is a

specialized library for building and analyzing recommender systems that offer powerful tools for collaborative filtering. It provides a comprehensive set of functionalities, including various algorithms and metrics that simplify the process of implementing and evaluating recommendation techniques.

In addition to Surprise, the research leverages `numpy`, a fundamental package for scientific computing in Python. `numpy` enhances the performance of numerical operations, enabling efficient manipulation and analysis of large datasets, which is essential for handling the extensive user-book interaction data utilized in this study.

Furthermore, other auxiliary libraries and tools may be incorporated as needed, such as `pandas` for data manipulation and analysis, `matplotlib` and `seaborn` for data visualization, and `scikit-learn` for additional machine learning functionalities. Combining these tools and libraries provides a robust framework for developing a sophisticated recommendation system, allowing for effective experimentation and analysis of various collaborative filtering algorithms.

4 Discussion

Evaluating various similarity metrics—such as cosine, mean squared difference (msd), Pearson, and Pearson baseline—for both user-based and item-based collaborative filtering models provides valuable insights into their behaviour and effectiveness across multiple performance indicators. Each of these metrics plays a distinct role in influencing the model's ability to predict user preferences accurately and rank relevant recommendations. Understanding their comparative performance in terms of prediction accuracy, precision, recall, F1 score, and computational efficiency is crucial for developing an optimized recommendation system.

Collaborative filtering models rely on these similarity metrics to quantify the relationships between users and items or among users themselves. The models analyzed in this study include both user-based filtering, which identifies similar users to generate recommendations, and item-based filtering, which finds similar items that align with a user's history. Evaluating these models under multiple metrics helps assess not only the quality of predictions but also their ranking performance in practical scenarios, such as recommending books or products.

Prediction accuracy is measured using RMSE (Root Mean Square Error) and MAE (Mean Absolute Error), which indicate how close the predicted ratings are to the actual values. Metrics like Precision@K, Recall@K, and F1 Score@K provide insights into how well the models perform in ranking items and retrieving relevant recommendations. Additionally, training and prediction times reflect the computational efficiency of these models, which is a critical factor in real-time applications.

In this study, the prediction accuracy of collaborative filtering models was evaluated using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), which measure how close the predicted ratings are to the actual ones. Lower values of RMSE and MAE indicate better performance, meaning the model makes fewer mistakes in predicting user preferences. Among the metrics tested, Pearson similarity, applied to both user-based and item-based models, demonstrated the best performance. It achieved the lowest RMSE of 0.9386 and MAE close to 0.7491, showing that Pearson-based models provide the most accurate predictions.

Although cosine similarity and mean squared difference (msd) also performed well, their accuracy was slightly lower than Pearson's. For example, both cosine and msd models had RMSE values of 0.944, which, while close to Pearson's results, indicate minor differences in performance. These small gaps in accuracy may seem negligible but could become important in contexts where precision is crucial. Additionally, the comparison between user-based and item-based models revealed that user-based configurations generally had a slight edge in terms of RMSE and MAE. However, the difference was not substantial enough to always favour one over the other, suggesting that both approaches can be effective depending on the situation.

The evaluation of Precision@K, Recall@K, and F1 Score@K reveals consistent patterns across the collaborative filtering models. Precision@K, which measures how accurately the top-K recommended items match relevant user preferences, shows similar results across all models, with values clustering between 0.682 and 0.684. The msd user-based model achieves the highest precision at 0.6847, indicating a slight edge in top-K recommendation accuracy. For Recall@K, which reflects the models' ability to retrieve all relevant items, Pearson-based configurations perform best, with the item-based Pearson Baseline model reaching the highest value of 0.9866. This suggests that Pearson models are more effective at capturing a larger proportion of relevant items. In terms of F1 Score@K, which provides a balanced measure of precision and recall, the Pearson Baseline user-based model delivers the highest score of 0.8068, demonstrating strong overall performance by effectively balancing accuracy with completeness. These results highlight that while certain models may excel in individual metrics, Pearson-based models, particularly the user-based and baseline versions, offer the most well-rounded performance across key evaluation measures.

The analysis of training and prediction times reveals a clear trade-off between computational efficiency and accuracy in collaborative filtering models. Cosine and mean squared difference (msd) metrics, partic-

ularly in user-based configurations, demonstrate impressive computational efficiency, with the user-based cosine model averaging just 2.07 seconds for training, compared to 8.78 seconds for the item-based Pearson model. Prediction times also favour user-based configurations, with the cosine model achieving 0.089 seconds versus 0.129 seconds for item-based Pearson. This makes cosine and msd models preferable for real-time applications, despite their slightly lower accuracy compared to Pearson models, which, while offering higher accuracy, incur greater computational costs. Therefore, the choice of model should align with the specific requirements of the application, balancing the need for speed against the importance of prediction precision. For high-accuracy systems, such as personalized book recommendations, user-based Pearson Baseline models are recommended when prediction accuracy is paramount. In contrast, cosine or mean squared difference (msd) user-based models are ideal for real-time applications where low latency is essential, such as in product recommendations. Additionally, a hybrid approach that combines Pearson Baseline for batch recommendations with cosine or msd models for real-time predictions can provide an optimal balance between accuracy and speed, tailored to specific use cases.

5 Results

The following table and diagrams illustrate the performance of various collaborative filtering models using different similarity metrics, including cosine, mean squared difference (msd), and Pearson, both in user-based and item-based configurations. Each diagram provides insights into the models' performance across key metrics, allowing for a comparative analysis of prediction accuracy, precision, recall, F1 score, and computational efficiency.

Table 1: Performance Metrics for Different Similarity Measures

Metric	Cosine		MSD		Pearson		Pearson Baseline	
	user_based=TRUE	user_based=FALSE	user_based=TRUE	user_based=FALSE	user_based=TRUE	user_based=FALSE	user_based=TRUE	user_based=FALSE
RMSE	0.9443	0.9479	0.9441	0.9490	0.9390	0.9386	0.9387	0.9391
MAE	0.7501	0.7547	0.7503	0.7555	0.7497	0.7492	0.7491	0.7500
Precision@K	0.6832	0.6828	0.6847	0.6824	0.6816	0.6809	0.6826	0.6818
Recall@K	0.9811	0.9803	0.9812	0.9787	0.9860	0.9860	0.9865	0.9867
F1 Score@K	0.8055	0.8049	0.8065	0.8041	0.8060	0.8055	0.8068	0.8063
Training Time (s)	2.0685	7.3364	1.4996	6.2545	2.6433	8.7889	2.1185	7.5774
Prediction Time (s)	0.0897	0.0947	0.1042	0.1117	0.1037	0.1298	0.1042	0.1061

The table summarizes the performance metrics of various collaborative filtering models using different similarity measures, including cosine, mean squared difference (msd), and Pearson, across both user-based and item-based configurations. It presents key indicators such as Average RMSE and MAE, which reflect the models' prediction accuracy, with lower values indicating better performance. Additionally, Average Precision@K and Recall@K show how effectively the models retrieve relevant recommendations, while Average F1 Score@K balances precision and recall. The Average Training Time and Prediction Time highlight the computational efficiency of each model, with user-based cosine and msd models demonstrating faster training and prediction times compared to item-based Pearson models. Overall, this table provides a concise overview of each model's strengths and weaknesses, aiding in the selection of the most suitable approach for different applications.

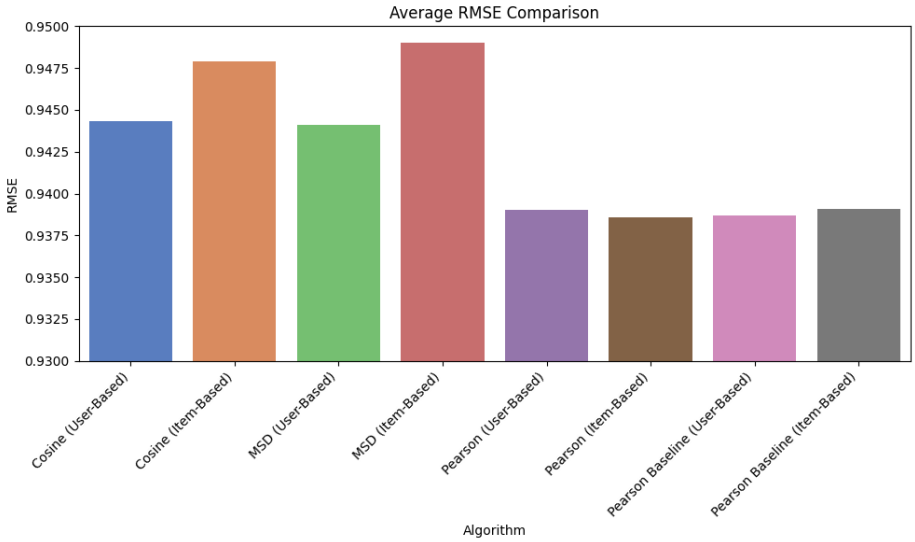

Figure 2: Average RMSE Comparison

This diagram displays the average Root Mean Square Error (RMSE) for each model configuration. Lower RMSE values indicate better predictive accuracy. The user-based Pearson model shows the lowest RMSE, suggesting it is the most accurate for predicting user preferences, while cosine and msd models also perform well but with slightly higher RMSE values.

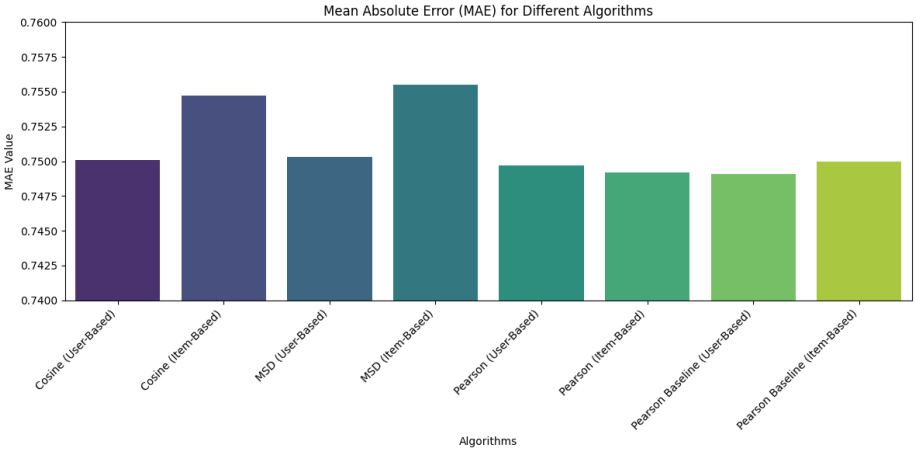

Figure 3: Average MAE Comparison

This diagram illustrates the average Mean Absolute Error (MAE) across different models. MAE provides a straightforward measure of prediction accuracy, where lower values signify better performance. The user-based Pearson model consistently achieves the lowest MAE, reinforcing its position as the top choice for accuracy in predictions, followed closely by cosine and msd configurations.

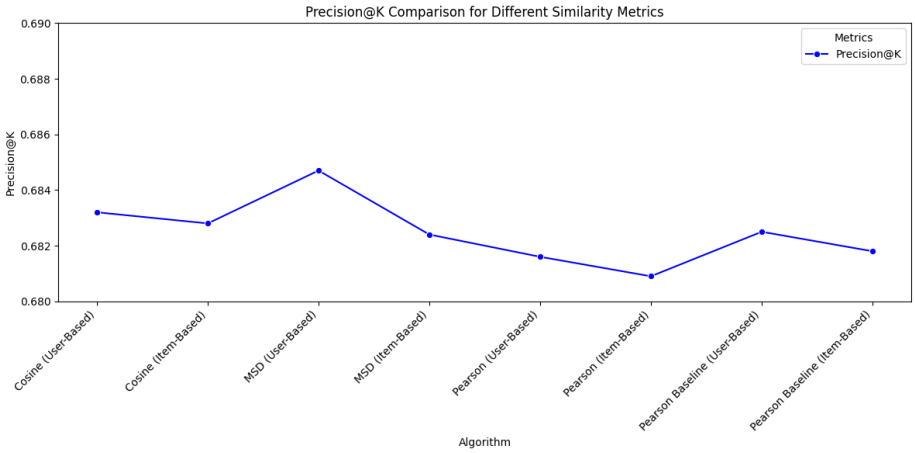

Figure 4: Average Precision@K

This diagram presents the average Precision@K values for each model. Precision@K indicates the proportion of relevant items in the top-K recommendations. The results show that the msd user-based model achieves the highest precision, highlighting its effectiveness in generating accurate top-K recommendations among the various configurations.

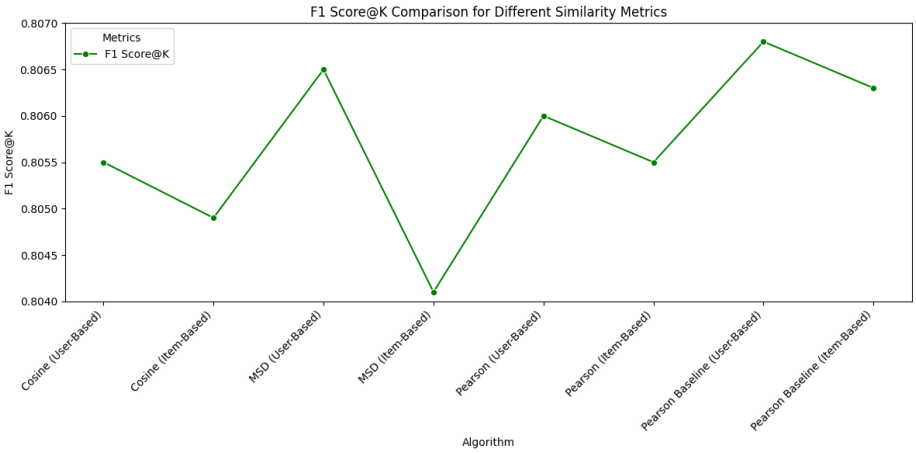

Figure 5: Average F1 Score@K

This diagram illustrates the average F1 Score@K, which balances precision and recall. The F1 Score provides insight into the models' overall effectiveness in generating relevant recommendations. The results indicate that the user-based Pearson Baseline achieves the highest F1 score, reflecting a well-rounded performance in both retrieving relevant items and maintaining precision.

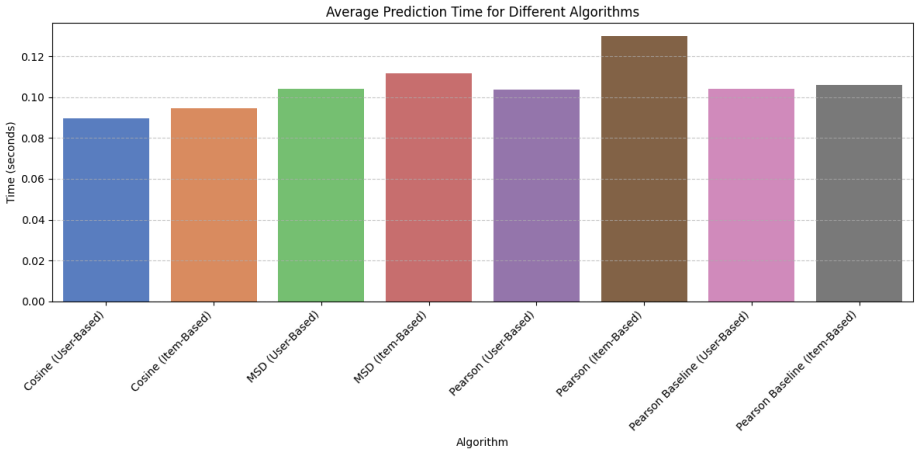

Figure 6: Average Training Time

This diagram shows the average training time for each model configuration, measured in seconds. The results demonstrate significant differences in training efficiency, with user-based cosine and msd models exhibiting the fastest training times, making them suitable for applications requiring rapid model updates. Conversely, the item-based Pearson models require notably longer training times.

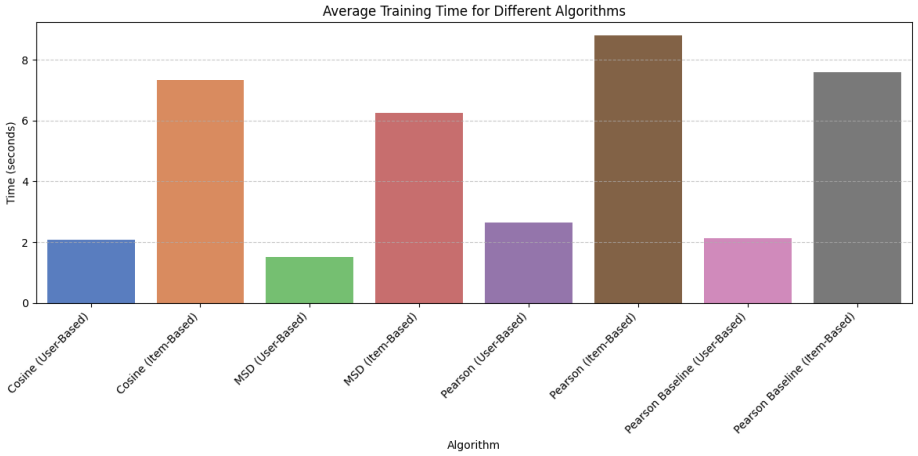

Figure 7: Average Prediction Time

This diagram presents the average prediction time taken by each model to generate recommendations. The user-based configurations generally show quicker prediction times, particularly the cosine model, which indicates that these models are well-suited for real-time applications. In contrast, item-based models, particularly Pearson, take longer to predict, which may impact their usability in time-sensitive contexts.

6 Conclusion

Conclusion This study thoroughly evaluated the performance of various similarity measures—Cosine, MSD (Mean Squared Difference), Pearson, and Pearson Baseline—within a KNN Baseline collaborative filtering algorithm. The results demonstrated that the Pearson similarity measure, both user-based and with a baseline adjustment, consistently produced the lowest RMSE and MAE values, indicating superior accuracy in predicting user preferences.

In terms of ranking performance, all similarity measures exhibited high recall values, with Pearson achieving the highest recall at 0.986, suggesting its effectiveness in retrieving relevant recommendations. The F1 Score further reinforced this, with Pearson Baseline achieving a score of 0.806, highlighting its balanced performance in precision and recall.

While the MSD measure provided faster training and prediction times, it showed slightly reduced accuracy compared to Pearson, making it suitable for applications requiring quick recommendations. Cosine similarity demonstrated a balanced trade-off between accuracy and processing speed. A hybrid approach combining both types may also be effective in balancing accuracy and speed.

References

Badaro, G., Hajj, H., El-Hajj, W., & Nachman, L. (2013). A hybrid approach with collaborative filtering for recommender systems. In *2013 9th international wireless communications and mobile computing conference (iwcmc)* (pp. 349–354).

Bouraga, S., Jureta, I., Faulkner, S., & Herssens, C. (2014). Knowledge-based recommendation systems: A survey. *International Journal of Intelligent Information Technologies (IJIIT)*, *10*(2), 1–19.

Choi, S. H., Kang, S., & Jeon, Y. J. (2006). Personalized recommendation system based on product specification values. *Expert Systems with Applications*, *31*(3), 607–616.

Duan, R., Jiang, C., & Jain, H. K. (2022). Combining review-based collaborative filtering and matrix factorization: A solution to rating's sparsity problem. *Decision Support Systems*, *156*, 113748.

Gomez-Uribe, C. A., & Hunt, N. (2015). The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, *6*(4), 1–19.

Han, E.-H., & Karypis, G. (2005). Feature-based recommendation system. In *Proceedings of the 14th acm international conference on information and knowledge management* (pp. 446–452).

Isinkaye, F. O. (2021). Matrix factorization in recommender systems: algorithms, applications, and peculiar challenges. *IETE Journal of Research*, 1–14.

Jiang, Y., Shang, J., & Liu, Y. (2010). Maximizing customer satisfaction through an online recommendation system: A novel associative classification model. *Decision Support Systems*, *48*(3), 470–479.

Kulkarni, S., & Rodd, S. F. (2020). Context aware recommendation systems: A review of the state of the art techniques. *Computer Science Review*, *37*, 100255.

Kumar, P., & Thakur, R. S. (2018a). Recommendation system techniques and related issues: a survey. *International Journal of Information Technology*, *10*, 495–501.

Kumar, P., & Thakur, R. S. (2018b). Recommendation system techniques and related issues: a survey. *International Journal of Information Technology*, *10*(4), 495–501.

Lapan, M. (2018). *Deep reinforcement learning hands-on: Apply modern rl methods, with deep q-networks, value iteration, policy gradients, trpo, alphago zero and more*. Packt Publishing Ltd.

Lekakos, G., & Caravelas, P. (2008). A hybrid approach for movie recommendation. *Multimedia tools and applications*, *36*, 55–70.

Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. *Recommender systems handbook*, 73–105.

Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In *The adaptive web: methods and strategies of web personalization* (pp. 325–341). Springer.

Shah, L., Gaudani, H., & Balani, P. (2016). Survey on recommendation system. *International Journal of Computer Applications*, *137*(7), 43–49.

Silveira, T., Zhang, M., Lin, X., Liu, Y., & Ma, S. (2019). How good your recommender system is? a survey on evaluations in recommendation. *International Journal of Machine Learning and Cybernetics*, *10*, 813–831.

Stitini, O., Kaloun, S., & Bencharef, O. (2022). An improved recommender system solution to mitigate the over-specialization problem using genetic algorithms. *Electronics*, *11*(2), 242.

Wang, X., Wang, Y., Hsu, D., & Wang, Y. (2014). Exploration in interactive personalized music recommendation: a reinforcement learning approach. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, *11*(1), 1–22.

Yildirim, H., & Krishnamoorthy, M. S. (2008). A random walk method for alleviating the sparsity problem in collaborative filtering. In *Proceedings of the 2008 acm conference on recommender systems* (pp. 131–138).

Yuyan, Z., Xiayao, S., & Yong, L. (2019). A novel movie recommendation system based on deep reinforcement learning with prioritized experience replay. In *2019 ieee 19th international conference on communication technology (icct)* (pp. 1496–1500).