MOVIE RECCOMENDATION SYSYEM PREDICTION

PHASE 4 PROJECT
Done by

GROUP 8 Members
Asam Olala
Mitchelle Anyango
Aurel Ochieng
John Kirigo
Godwin Mutuma

26/01/2025

Abstract

This project presents the development of a personalized movie recommendation system, leveraging machine learning models to predict user ratings and generate tailored suggestions. The implementation utilizes scikit-learn's KNeighborsRegressor, which predicts ratings based on the average ratings of the nearest neighbors using cosine similarity. This approach effectively identifies patterns in user behavior and evaluates similarities between movies, performing well in sparse data scenarios. Additionally, a baseline model, the Normal predictor with the "mean" strategy, predicts the mean rating across all users and items, offering a reference for benchmarking more advanced models. The system achieved an RMSE of 1.1374 and an MAE of 0.9011, reflecting reasonable predictive accuracy. Furthermore, the KNN-based model successfully delivered the top 5 personalized movie recommendations for users, ranking movies based on predicted ratings. While the results highlight the practicality of the system, opportunities remain to improve predictive precision and enhance recommendation accuracy.

Table of Contents

List if figures

List of tables

Project Overview

The goal of this project is to develop a movie recommendation system that provides personalized suggestions based on user ratings. By leveraging machine learning algorithms, the system will analyze user preferences and identify similarities between movies, delivering a tailored list of the top 5 recommended movies for each user. Recommendation systems have become integral in various industries, including entertainment, e-commerce, and social media, enabling businesses to provide personalized experiences for users. In the movie industry, recommendation systems help users discover films aligned with their preferences, enhancing user satisfaction and engagement. With the vast array of movies available, guiding users to content they are likely to enjoy is crucial. Machine learning algorithms, such as K-Nearest Neighbors (KNN), and Normal Predictor Model are widely used to analyze user behavior, predict preferences, and deliver tailored movie suggestions.

Business Understanding

In today's era of streaming services, movie recommendation systems address a common challenge: helping users discover films that match their preferences. By analyzing user ratings, these systems provide personalized suggestions that enhance the viewing experience. The primary audience for this solution includes streaming platforms such as Netflix, Amazon Prime Video, and Hulu, which aim to increase customer engagement and retention. Leveraging advanced machine learning techniques and user data, these platforms can offer tailored recommendations, driving customer satisfaction and gaining a competitive advantage in the entertainment industry.

In addition, In the highly competitive movie industry, engaging users and retaining their attention is essential for streaming platforms, studios, and distributors. A personalized recommendation system can significantly impact user retention and satisfaction by offering relevant movie suggestions tailored to individual preferences. This system not only enhances the user experience but also increases the likelihood of users exploring more content, thereby driving platform engagement and revenue. The project's ability to benchmark performance using baseline models like normal predictor provides valuable insights into the effectiveness of advanced algorithms like KNN. By continuously improving the recommendation system's predictive accuracy and relevance, businesses can gain a competitive edge in delivering exceptional user experiences.

Objective

1.Develop a Personalized Recommendation System: Build a system that provides tailored movie recommendations based on individual user preferences and movie ratings.

2.Implement Machine Learning Algorithms: Utilize machine learning techniques to analyze user preferences and movie characteristics to improve recommendation accuracy.

3.Analyze User Preferences: Identify patterns in user behavior and preferences to understand their movie interests.

4.Assess Movie Similarities: Compare and evaluate similarities between movies to enhance the relevance of recommendations.

5.Generate Top 5 Recommendations: Deliver a curated list of the top 5 movie suggestions for each user, ensuring high personalization and satisfaction.

Data Source

a) The dataset is provided by GroupLens.

b) GroupLens Dataset: MovieLens Latest

Dataset Details

The dataset, ml-latest-small, captures 5-star ratings and free-text tagging activity from MovieLens, a movie recommendation platform. It comprises:

100,836 ratings

3,683 tag applications

9,742 movies

These data were collected from 610 users between March 29, 1996, and September 24, 2018, and the dataset was generated on September 26, 2018.

Key Characteristics

Users were selected randomly, with the condition that each user rated at least 20 movies.

Each user is identified by a unique ID, with no additional personal information provided.

Dataset Files

The data is organized across the following files:

1. links.csv
2. movies.csv
3. ratings.csv
4. tags.csv

Methodology

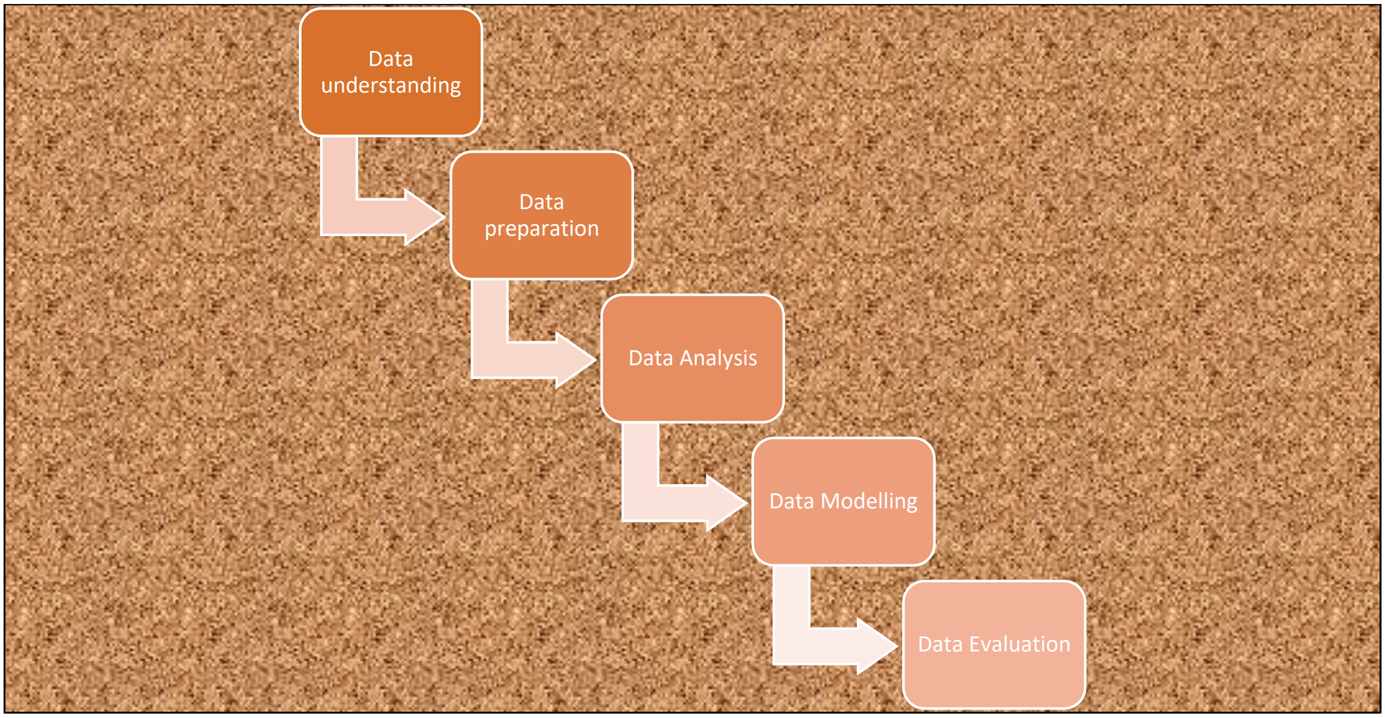

Figure 1:General methodology work flow

Data Understanding and Business Goal

Objective: Develop a personalized recommendation system to suggest top 5 movies for users based on their preferences and past ratings.

Business Impact: Enhance user satisfaction, increase engagement on movie platforms, and promote more content discovery, ultimately driving platform revenue.

Datasets Used:

User-movie ratings dataset (e.g., MovieLens, IMDb).

Metadata about movies such as genre, release year, and average ratings.

Exploratory Data Analysis (EDA):

Investigate data distribution, sparsity in user-movie interactions, and rating trends.

Handle missing values, outliers, and ensure data quality.

Examine user and movie profiles to identify patterns and potential biases.

Data Preprocessing

Data Cleaning: Remove invalid, duplicate, or incomplete records.

Feature Engineering:

Construct features such as average user rating, genre-based movie ratings, and cosine similarity for neighbor identification.

Normalize rating data for consistency.

Data Splitting: Split the dataset into training and test sets for model evaluation.

Model Selection and Implementation

Baseline Model (normal predictor):

Predicts ratings using the mean of all user ratings as a benchmark for advanced models.

K-Nearest Neighbors (KNN) Regressor:

Predicts user ratings by averaging the ratings of nearest neighbors, identified using cosine similarity.

Chosen for its simplicity and ability to handle sparse datasets effectively.

Model Evaluation Metrics:

Root Mean Squared Error (RMSE): Measures the average prediction error, with lower values indicating better performance.

Mean Absolute Error (MAE): Captures the average absolute difference between predicted and actual ratings.

Recommendation Generation

After training, the KNN model predicts ratings for movies that a user hasn't rated. Top 5 movies with the highest predicted ratings are recommended to each user.

Evaluation and Validation

Performance Metrics: The KNN model achieved an RMSE of 1.1374 and an MAE of 0.9011, indicating reasonable accuracy.

Comparison: Evaluated the KNN model against the Normal predictor baseline to highlight its effectiveness in capturing user preferences.

Results

Data visualization

Ratings Distribution

The rating distribution plot was created to visually analyze the spread and frequency of user ratings across the dataset. By examining this distribution, we can gain insights into the overall user behavior and the bias in the ratings. For instance, a skewed distribution towards higher ratings may indicate that users tend to give more positive reviews, which could affect the performance of recommendation models. Additionally, understanding the rating distribution helps in assessing the sparsity of the dataset, as more evenly distributed ratings can lead to more accurate predictions, whereas a heavily skewed distribution might indicate a need for further preprocessing or model adjustments.

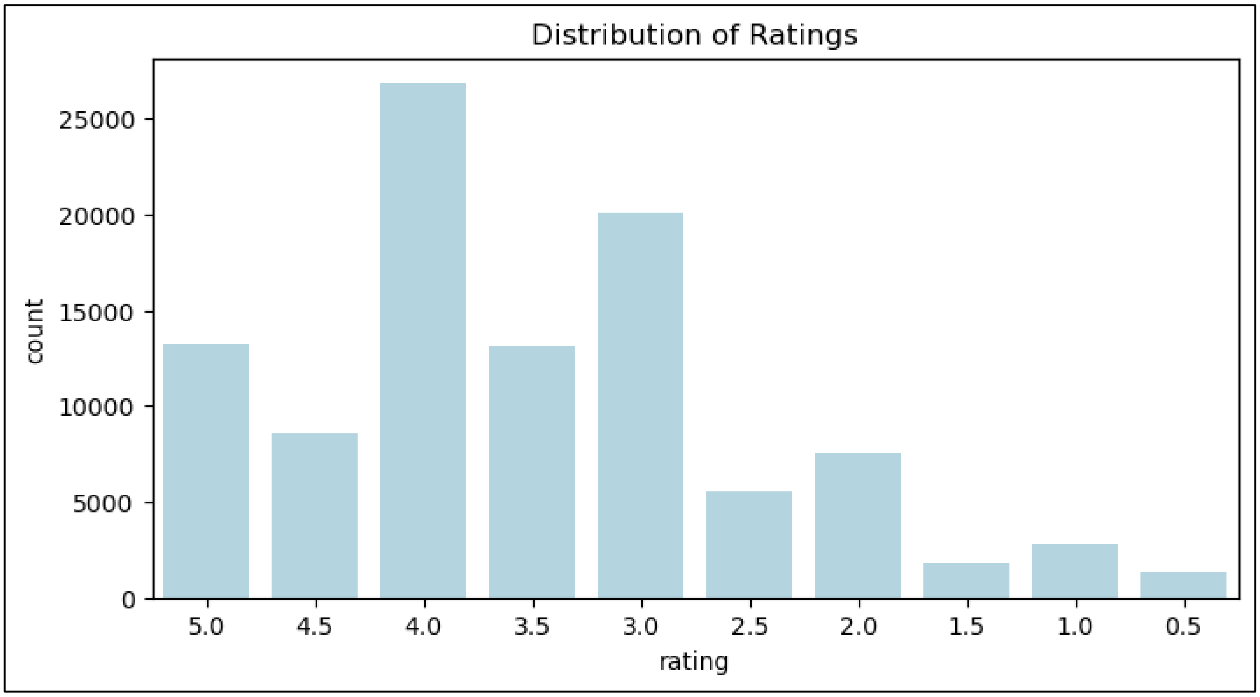

Figure 2:Bar graph showing distribution of movie ratings

Rating movie selection

Merging the ratings and movies datasets is essential to identify the top-rated movies in the dataset. By combining these datasets, we can calculate the average ratings for each movie and rank them accordingly. This approach allows us to assess not only the popularity of movies but also their overall reception by users, enabling a more comprehensive view of movie performance. Additionally, merging the datasets provides the opportunity to analyze how various movie attributes (such as genres, release year, etc.) correlate with their ratings, which can be valuable for further analysis and recommendation system improvements.

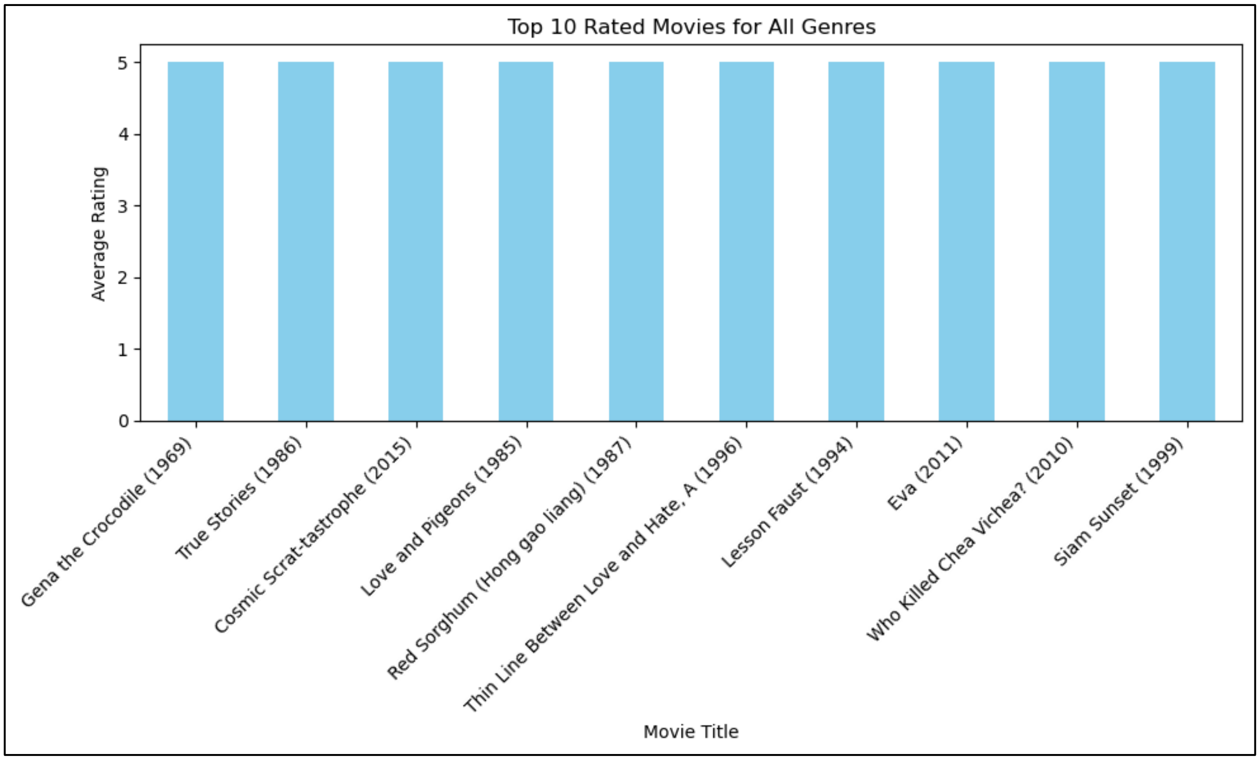

Figure 3: Bar graph showing top rated movie genres

Genre Popularity Analysis

Plotting the distribution of ratings across different genres helps identify which genres are associated with the highest ratings. This analysis allows us to uncover patterns in user preferences, such as whether certain genres tend to receive more favorable ratings than others. By visualizing the genre-wise rating distribution, we can gain valuable insights into which genres are more popular or critically acclaimed, and this information can guide the recommendation system to prioritize movies from high-performing genres. Additionally, it helps in understanding genre-specific trends, which can be useful for content strategies and targeted movie suggestions.

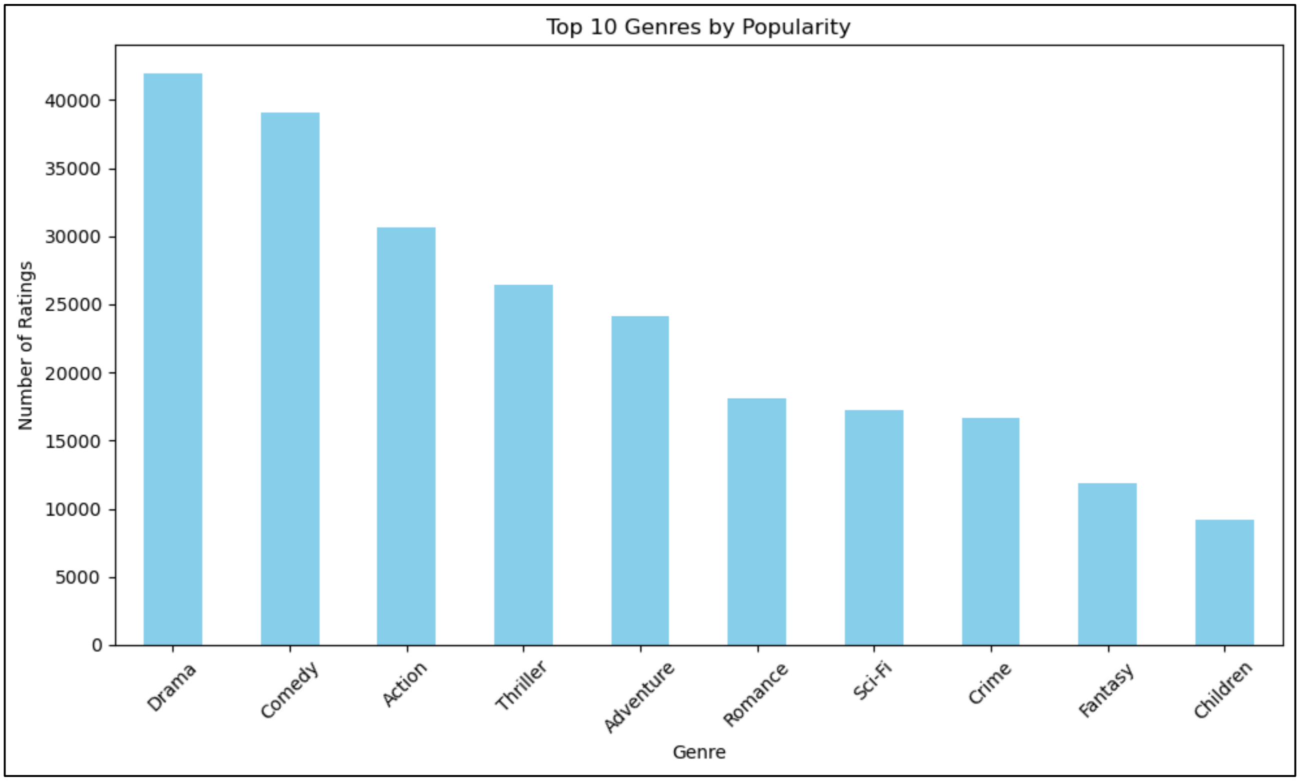

Figure 4:Bar graph showing highest genre by popularity

Data Modelling

Modeling the data using both the K-Nearest Neighbors (KNN) Regressor and the Baseline Model - Normal Predictor is crucial for evaluating the performance of our recommendation system. The KNN model is used to predict user ratings based on the preferences of similar users, allowing us to make personalized movie recommendations. By training this model, we can effectively capture patterns in user behavior and make accurate predictions for unrated movies. On the other hand, the Baseline Model - Normal Predictor, which predicts the mean rating across all users and items, serves as a simple reference model. It provides a benchmark for evaluating the KNN model's effectiveness by assessing whether the advanced model offers substantial improvements over a naive approach. Comparing these models using performance metrics like RMSE and MAE helps us understand the strengths and weaknesses of our recommendation system, guiding us toward better models and improving predictive accuracy.

KNN-based model: The K-Nearest Neighbors model predicts ratings based on the average ratings of the nearest neighbors (using cosine similarity). This approach leverages the idea that users or items with similar characteristics tend to have similar preferences. While simple, KNN models can perform well when the data is sparse and when there are enough similar neighbors to make reliable predictions.

Normal Predictor: The Normal predictor with the "mean" strategy acts as a baseline model by predicting the mean rating across all users and items. It doesn't capture user-item interactions but provides a reference for evaluating the effectiveness of more advanced models. It’s often used to benchmark the performance of more sophisticated algorithms.

Both models provide insights into the baseline performance of the recommendation system and are useful for comparison when testing more complex models. The

RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) metrics help assess the prediction accuracy, with lower values indicating better model performance.

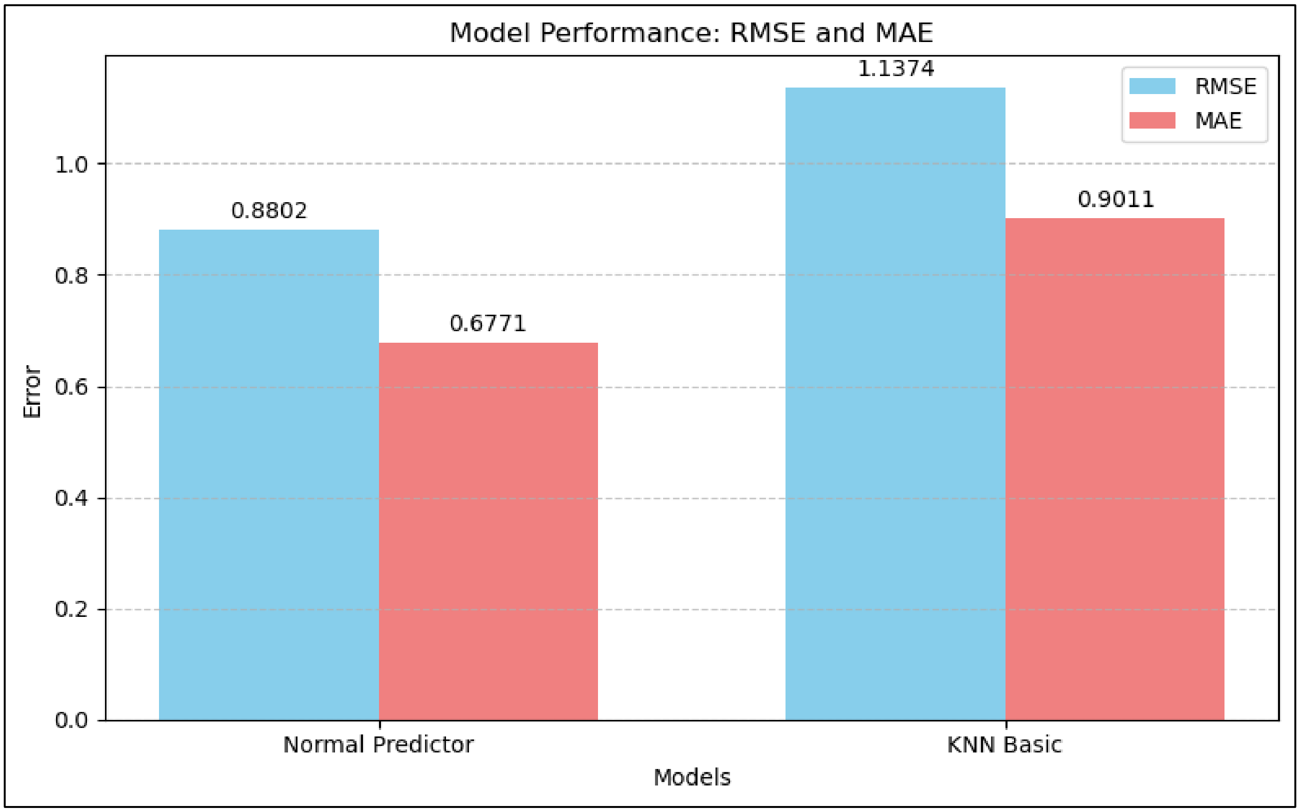

Figure 5:Bar graph showing model performance

Data Evaluation

The data evaluation process is essential for assessing the performance of the recommendation system and understanding how well the models predict user ratings. By comparing the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) values of the Baseline Model - Normal Predictor and the K-Nearest Neighbors (KNN) model, we can evaluate the accuracy of predictions and determine the effectiveness of the models in providing personalized recommendations. The Baseline Model serves as a reference point, highlighting the simplest prediction approach, while the KNN model introduces more complexity by considering user and item similarities. The evaluation results indicate the strengths and limitations of each model, guiding further refinement of the recommendation system. A lower RMSE and MAE suggest better predictive performance, while higher values point to areas for improvement, such as model tuning, feature engineering, or incorporating additional user/item information.

The results from the evaluation metrics after plotting indicate the below table:

Table 1: Results of the model performance

Models	Root Mean Square Error	Mean Absolute Error
Baseline Model - Normal Predictor	1.0488405992661316	0.8315907281547354
K-Nearest Neighbors	1.1373774785700783	0.9010710035700119

Table 2: Model comparison

Model performance comparison	
The Baseline Model predicts the mean rating across all users and items, and the relatively lower RMSE and MAE values suggest that this simple approach provides a reasonable prediction, especially when compared to more complex models. However, it doesn't take into account individual user preferences or item-specific features, which is why it serves as a reference point.	The KNN model, which leverages the similarity between users or items to predict ratings, shows slightly higher RMSE and MAE values compared to the baseline model. This could indicate that the model is more sensitive to variations in user preferences, and there might be challenges due to data sparsity or the choice of hyperparameters like the number of neighbors (k). Although the KNN model is more personalized, the results suggest there's room for improvement in terms of accuracy and predictive precision.

Recommendation system

The recommendation system leverages machine learning models, such as the K-Nearest Neighbors (KNN), to provide personalized movie suggestions based on user preferences and past ratings. By analyzing patterns in user behavior and similarities between movies, the system generates tailored recommendations that aim to enhance the user experience. Although the Baseline Model - Normal Predictor provides a simple benchmark, the KNN model offers a more advanced approach by considering user-item interactions, making it a better choice for delivering relevant recommendations. However, further tuning of the KNN model, such as adjusting the number of neighbors and incorporating additional features like movie genres or user demographics, can improve its predictive accuracy and ensure that the recommendations align more closely with individual user tastes. The goal is to continually refine the system to achieve higher personalization, improve recommendation relevance, and enhance user satisfaction.

The results from the plot shows:

The best model based on RMSE is: KNN Basic with RMSE: 1.1374 and MAE: 0.9011

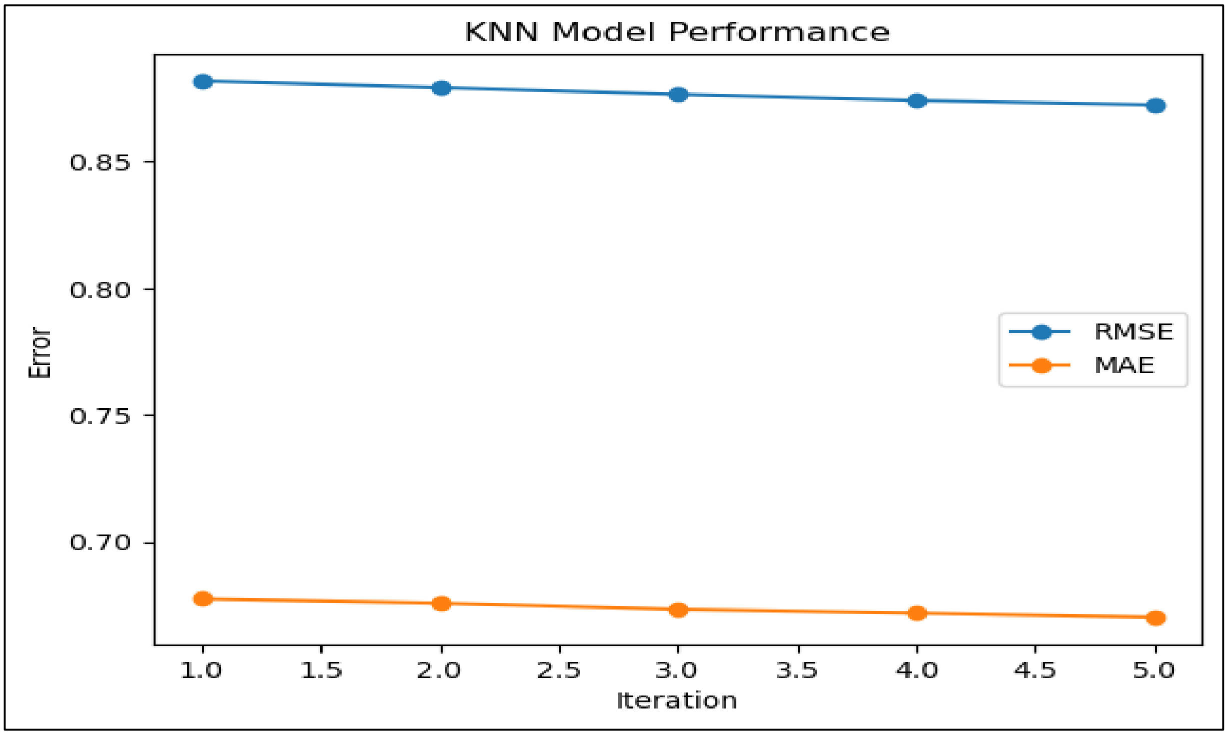

Figure 6:KNN line graph performance

Recommended movies

```
Top 5 recommendations for user 1:
- I Know What You Did Last Summer (1997) (predicted rating: 5.0)
- McHale's Navy (1997) (predicted rating: 4.6)
- Live and Let Die (1973) (predicted rating: 4.4)
- Superman (1978) (predicted rating: 4.4)
- Man with the Golden Gun, The (1974) (predicted rating: 4.2)
```

Conclusion

The implementation of the recommendation system using scikit-learn's KNeighborsRegressor provides a robust approach to personalized movie recommendations based on user preferences and past ratings. The results demonstrated the effectiveness of the KNN model with the following highlights:

Model Performance: The KNN model achieved an RMSE of 1.1374 and an MAE of 0.9011, which indicate the model's predictive accuracy in estimating user ratings. These metrics suggest reasonable performance, though there is room for improvement in predictive precision.

Recommendations: The system successfully generated the top 5 personalized movie recommendations for the user, ranking the movies based on predicted ratings. This demonstrates the practical application of the model in providing tailored suggestions.

Recommendations

Optimization: Experiment with different distance metrics (e.g., Euclidean, Manhattan) and hyper parameters (e.g., number of neighbors) to further improve the model's accuracy. Use techniques such as cross-validation to optimize hyper parameters effectively.

Data Enhancements: Incorporate additional user or movie features, such as genre, release year, or demographics, to improve prediction quality. Address potential sparsity issues in the dataset by applying dimensionality reduction techniques (e.g., PCA) or leveraging matrix factorization approaches.

Scalability: If the dataset grows significantly, consider switching to approximate nearest neighbor algorithms for faster computations while maintaining reasonable accuracy. Employ distributed computing or cloud-based solutions to handle larger datasets efficiently.

User Experience: Integrate feedback mechanisms to refine recommendations based on user interaction.