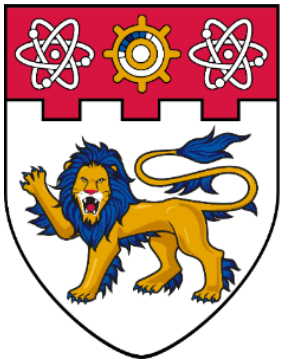

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

TOWARDS AN EFFECTIVE RECOMMENDATION ALGORITHM FOR E-COMMERCE

Tan Choon Wee

College of Computing and Data Science

2025

NANYANG TECHNOLOGICAL UNIVERSITY

SC4079

TOWARDS AN EFFECTIVE RECOMMENDATION ALGORITHM FOR E-COMMERCE

Submitted in Partial Fulfilment of the Requirements
for the Degree of Bachelor of Engineering in Computer Science
of the Nanyang Technological University

by

Tan Choon Wee

College of Computing and Data Science
2025

Abstract

E-commerce has revolutionized the way people shop, offering convenience and a vast array of products. However, with the sheer volume of items available, users often face difficulty in finding products that match their preferences.

Recommendation algorithms play a crucial role in enhancing user experience by suggesting products that align with individual tastes and needs. Over the years, various recommendation systems have been developed, ranging from collaborative filtering to content-based and hybrid approaches.

This project focuses on evaluating and comparing the effectiveness of four state-of-the-art recommendation algorithms: Matrix Factorization (MF), Bayesian Personalized Ranking (BPR), Neural Collaborative Filtering (NCF), and Neural Graph Collaborative Filtering (NGCF). Using the widely recognized MovieLens dataset, these algorithms are assessed through a comprehensive set of evaluation metrics, including Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG), Precision, and Recall, to determine their performance in an e-commerce context.

The project first compares the four algorithms to identify their relative strengths and weaknesses. Following this, a series of parameter tuning experiments will be conducted to examine the impact of various hyperparameters on performance. By systematically adjusting key parameters, the study aims to optimize each algorithm and assess how these modifications influence the evaluation metrics. Based on the results, the project recommends optimal parameter configurations to achieve the best performance, providing actionable insights for the implementation of effective recommendation systems in e-commerce platforms.

Acknowledgements

I would like to extend my deepest gratitude to my project supervisor, Professor Zhang Jie, for his invaluable guidance and unwavering support throughout this project. His professional advice and constant supervision have been instrumental in helping me navigate the challenges of this project and successfully bring it to completion.

I am also sincerely grateful to Dr. Du Yingpeng and Sun Zhu for their patient instruction on the algorithms used in this project. Their expertise and willingness to explain complex concepts have greatly enhanced my understanding and enabled me to overcome initial difficulties.

Lastly, I would like to express my heartfelt thanks to my friends and family for their constant encouragement and valuable feedback throughout this journey. Their support has been a source of motivation and strength, and I am deeply appreciative of their role in helping me achieve this milestone.

Contents

List of Tables

List of Figures

Chapter 1

1. Introduction

1.1 Background

E-commerce has transformed the global retail landscape, offering consumers convenience and access to an extensive selection of products. Online platforms such as Amazon, Alibaba, and eBay feature millions of items, making it increasingly challenging for users to find products that align with their preferences. To address this issue, recommendation systems have become an integral part of modern e-commerce platforms, enabling personalized shopping experiences, and enhancing customer engagement [1].

Over the years, various recommendation techniques have been developed, primarily categorized into collaborative filtering, content-based filtering, and hybrid approaches [2]. Collaborative filtering, one of the most widely used techniques, predicts user preferences based on historical interactions and the behaviours of similar users [3]. Matrix Factorization (MF) has been a key technique in collaborative filtering, effectively capturing latent factors in user-item interactions [4]. However, traditional MF-based models struggle with scalability and sparsity, particularly in large-scale e-commerce datasets.

To overcome some of these challenges, Bayesian Personalized Ranking (BPR) has been proposed to handle implicit feedback by optimizing ranking-based objectives rather than explicit ratings [5]. While BPR improves ranking performance, it may not effectively capture complex, nonlinear user-item relationships. Recent advancements in deep learning have led to the development of Neural Collaborative Filtering (NCF),

which replaces traditional matrix factorization with multi-layer perceptrons to enhance predictive accuracy [6]. Further extending deep learning approaches, Neural Graph Collaborative Filtering (NGCF) incorporates graph neural networks to model high-order user-item interactions, making recommendations more contextually aware [7].

Despite these advancements, several challenges persist in recommendation systems. Many existing models struggle with dynamically evolving user preferences, limiting their ability to provide real-time, personalized recommendations [8]. Additionally, scalability remains a pressing issue, as e-commerce platforms must process vast datasets efficiently while maintaining high recommendation accuracy [3]. Another critical limitation is the lack of effective integration of contextual information, such as browsing history, session-based interactions, and external factors, which significantly influence user decisions [2]. Addressing these challenges is essential for improving recommendation accuracy and enhancing the user experience in e-commerce platforms.

A well-optimized recommendation system not only improves customer satisfaction but also benefits businesses by increasing conversion rates and boosting sales performance [1]. By providing relevant and timely recommendations, e-commerce platforms can enhance user engagement, reduce information overload, and strengthen customer retention. Given the growing importance of recommendation systems in digital commerce, this project aims to evaluate and refine state-of-the-art recommendation algorithms, including MF, BPR, NCF, and NGCF. Through comprehensive experimentation and hyperparameter tuning, this study seeks to contribute to the advancement of recommendation technologies and offer practical solutions for modern e-commerce platforms.

1.2 Objectives

The primary objective of this project is to evaluate and refine state-of-the-art recommendation algorithms to enhance their effectiveness in e-commerce applications. Specifically, the project aims to achieve the following objectives:

1. Examine the current state of recommendation algorithms, including Matrix Factorization (MF), Bayesian Personalized Ranking (BPR), Neural Collaborative Filtering (NCF), and Neural Graph Collaborative Filtering (NGCF). This review will highlight their strengths and limitations in the context of e-commerce applications.

2. Perform an in-depth evaluation of MF, BPR, NCF, and NGCF using key evaluation metrics, including Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG), Precision, and Recall. Compare their performance on the MovieLens dataset to assess their effectiveness in generating accurate and relevant recommendations.

3. Perform systematic hyperparameter tuning for each algorithm to determine the optimal configurations that maximize performance. Analyse the impact of key parameters, such as top K items, number of epochs, and learning rate, on the evaluation metrics.

4. Based on the evaluation and tuning results, identify the most effective algorithm for e-commerce applications. Provide recommendations for optimal parameter configurations to achieve the best performance in real-world scenarios.

Chapter 2

2. Literature Review

Recommendation systems have evolved significantly over the years, with various methodologies designed to improve accuracy, scalability, and personalization. This section reviews four recommendation algorithms: Matrix Factorization (MF), Bayesian Personalized Ranking (BPR), Neural Collaborative Filtering (NCF), and Neural Graph Collaborative Filtering (NGCF). These techniques have been widely adopted in recommender systems and continue to shape the development of more advanced models.

2.1 Matrix Factorization Techniques for Recommender Systems

Matrix Factorization (MF) is a class of collaborative filtering algorithms widely used in recommender systems. The core idea is to decompose the user-item interaction matrix into the product of two lower-dimensional matrices, representing users and items in a shared latent space. This technique effectively captures latent factors in user-item interactions, making it a powerful tool for recommendation tasks [4]. MF techniques, such as Singular Value Decomposition (SVD) and Alternating Least Squares (ALS), enable the extraction of hidden patterns from sparse data, improving recommendation accuracy.

One of the most notable applications of MF was during the Netflix Prize competition, where it demonstrated significant improvements in recommendation accuracy. The basic MF model can be represented as:

$$R \approx P \cdot Q^T$$

Figure 1: Basic MF model equation

where R is the user-item interaction matrix, P is the user latent feature matrix, and Q is the item latent feature matrix. The predicted rating $\hat{r}_{ui}$ for user u and the item i is given by the dot product of the corresponding latent vectors:

$$\hat{r}_{ui} = p_u \cdot q^T{}_i$$

Figure 2: MF predicted rating equation

Despite its effectiveness, MF suffers from limitations such as cold start issues and an inability to model complex, nonlinear relationships in user-item interactions.

2.2 BPR: Bayesian Personalized Ranking from Implicit Feedback

Bayesian Personalized Ranking (BPR) is an optimization criterion designed for personalized ranking tasks, particularly in scenarios with implicit feedback (e.g., clicks, purchases). Unlike traditional MF, which focuses on rating prediction, BPR directly optimizes for ranking by maximizing the posterior probability of the observed user-item interactions [5].

The BPR optimization criterion can be expressed as:

$$\text{BPR\_Opt} = \sum_{(u,i,j)\in D_s} \ln \sigma(\hat{x}_{uij}) - \lambda\|\Theta\|^2$$

Figure 3: BPP optimization criterion equation

where $\hat{x}_{uij} = \hat{x}_{ui} - \hat{x}_{uj}$ represents he difference in predicted scores between a preferred item i and a non-preferred item j for user u, σ is the sigmoid function, D_s is the set of training triples, and Θ denotes the model parameters.

BPR is highly relevant for e-commerce platforms, where implicit feedback (e.g., clicks, purchases) is abundant. BPR has been shown to outperform traditional MF and k-nearest neighbours (kNN) methods in ranking tasks, making it a valuable approach for e-commerce recommendations. Its ranking-based approach aligns well with the goal of recommending top-K items to users.

2.3 Neural Collaborative Filtering

Neural Collaborative Filtering (NCF) extends MF by leveraging deep learning to capture complex user-item relationships [6]. Unlike MF, which assumes a simple dot product interaction, NCF replaces it with a multi-layer perceptron (MLP) to learn more expressive features. The NCF framework consists of:

1. Generalized Matrix Factorization (GMF):
   A linear component that mimics traditional MF, captures latent interactions via element-wise product.

2. Multi-Layer Perceptron (MLP):
   A non-linear component that captures complex interactions.

3. Neural Matrix Factorization (NeuMF):
   A combination of GMF and MLP for enhanced performance.

The final prediction is computed as:

$$\hat{r}_{ui} = NeuMF(p_u, q_i) = GMF(p_u, q_i) + MLP(p_u, q_i)$$

Figure 4: NCF final prediction equation

where p_u and q_i are the latent vectors for user u and the item i, respectively. Extensive experiments have demonstrated that NCF outperforms traditional MF and other state-of-the-art methods on real-world datasets

NCF is highly relevant for e-commerce platforms, where capturing complex user-item interactions is critical for providing accurate recommendations. Its ability to model non-linear relationships makes it suitable for dynamic and diverse user behaviours.

2.4 Neural Graph Collaborative Filtering

Neural Graph Collaborative Filtering (NGCF) extends NCF by incorporating graph neural networks (GNNs) to model high-order user-item interactions [7]. The key idea is to construct a user-item interaction graph and use GNNs to propagate embeddings through the graph. The embedding for user u at layer l is updated as:

$$e^{(l)}{}_u = \sigma\left(W_1 e_u{}^{(l-1)} + \sum_{i \in N_u} \frac{1}{\sqrt{|N_u||N_i|}} W_2 e_i{}^{(l-1)} \right)$$

Figure 5: NGCF equation

where N_u and N_i represent the neighbours of user u and item i, and W_1 and W_2 are learnable parameters.

By leveraging graph neural networks (GNNs), NGCF captures complex structural dependencies, leading to enhanced recommendation accuracy. NGCF is highly relevant for e-commerce platforms, where understanding high-order user-item interactions can lead to more personalized and accurate recommendations. Its ability to model complex relationships makes it suitable for large-scale and dynamic e-commerce environments.

This literature review provides a comprehensive overview of the four algorithms, highlighting their methodologies, strengths, limitations, and relevance to e-commerce. The review highlights the evolution from traditional MF techniques to deep learning-based models such as NCF and NGCF. While MF and BPR offer robust and scalable solutions, deep learning approaches like NCF and NGCF provide greater flexibility in capturing complex user-item interactions. Understanding these models is crucial for designing more effective recommendation algorithms in e-commerce applications.

Algorithms	**Key Idea**	**Strengths**	**Limitations**
MF	Decomposes user-item matrix into latent factors.	Simple, scalable, interpretable.	Struggles with sparsity and cold-start; assumes linear relationships.
BPR	Optimizes personalized rankings using implicit feedback.	Effective for implicit feedback; ranking-based.	Requires careful tuning; limited in capturing complex interactions.
NCF	Uses neural networks to model non-linear user-item interactions.	Captures complex interactions; outperforms MF	Computationally expensive; requires hyperparameter tuning.
NGCF	Extends NCF with graph neural networks for high-order interactions.	Captures high-order relationships; contextually aware.	Computationally expensive; complex to implement.

Table 1: Summary table of algorithms

Chapter 3

3. Methodology

3.1 Code Implementation and Experimentation Setup

This project utilizes open-source implementations of recommendation algorithms from two GitHub repositories:

1. Cornac [9]
   A comparative framework for recommendation algorithms, particularly useful for evaluating collaborative filtering models. Cornac was chosen due to its comprehensive support for a variety of matrix factorization and deep learning-based recommendation methods, enabling seamless integration and benchmarking.

2. Recommenders [10]
   A collection of recommender system models, including Matrix Factorization (MF), Bayesian Personalized Ranking (BPR), Neural Collaborative Filtering (NCF), and Neural Graph Collaborative Filtering (NGCF). This repository provides well-documented implementations with customizable hyperparameters, allowing for precise experimentation.

The codes from these repositories were used for experiment testing and analysing the performance of various algorithms. The testing process involved running Jupyter Notebook scripts on **Visual Studio Code (VSCode) version 1.97.2**, with the **Python 3.9.6 kernel environment** configured for execution.

3.2 Dataset Utilization: MovieLens 100K

For consistency and reproducibility in evaluation, this project employs the **MovieLens 100K dataset**. This dataset contains **100,000 ratings (1-5 stars) from 943 users on 1,682 movies**, making it a widely used benchmark for recommendation systems. The dataset provides:

- **User-item interaction data:**
  A matrix indicating which user rated which movie and the rating given.

- **Movie metadata:**
  Information about each movie such as title, release year, and genre.

- **Temporal rating information:**
  Timestamps indicating when each rating was made.

The MovieLens dataset was selected due to its extensive adoption in research and its suitability for evaluating collaborative filtering and deep learning-based recommendation models. The dataset was pre-processed by normalizing ratings, splitting data into training (80%) and testing (20%) sets, and converting user-item interactions into a format compatible with the selected algorithms.

3.3 Experimentation Approach

3.3.1 Initial Parameter Settings

To ensure fair comparisons across different recommendation models, the initial parameter settings were kept uniform across experiments. The following hyperparameters were selected based on preliminary testing and literature review:

Parameter	Description	Value
Num_Factors	Dimension of the latent space	200
Num_Epochs	Number of iterations of Stochastic Gradient Descent (SGD)	100
Learning_Rate	Step size (α) in gradient update rules	0.01
Lambda_Reg	L2-Regularization (γ) in the objective function	0.001
Batch Size	Size of training batch	256

Table 2: Initial Parameter Settings

The selected hyperparameters balance computational efficiency and model accuracy, ensuring meaningful performance comparisons.

3.3.2 Running Experiments

The experimentation involved executing pre-written scripts available in the GitHub repositories. The workflow consisted of:

1. **Data Preprocessing**
   Loading and splitting MovieLens 100K data into training and testing sets.

2. **Model Training**
   Running each algorithm (MF, BPR, NCF, NGCF) using the defined hyperparameters. The training processes leveraged gradient descent methods for optimization.

3. **Performance Evaluation**
   Assessing model effectiveness using evaluation metrics such as:

   - **Mean Average Precision (MAP)**:
     Evaluates the quality of ranking by calculating the mean of average precision scores for all users.

   - **Normalized Discounted Cumulative Gain (NDCG)**:
     Measures ranking quality by accounting for the position of relevant items in the recommendation list.

   - **Precision**:
     Calculates the proportion of relevant items among the top-K recommendations.

   - **Recall**:
     Measures the fraction of relevant items retrieved out of all relevant items available.

Each model underwent multiple runs with different random seeds to account for variance, and results were averaged to ensure robustness.

3.3.3 Hyperparameter Tuning

Following the initial testing, a **systematic hyperparameter tuning** process was conducted to refine the model configurations. This involved:

- Varying **Num_Factors**, **Learning_Rate**, **Lambda_Reg**, and **Num_Epochs**.
- Analysing the effect of different batch sizes.
- Adjusting ranking cut-off values for recommendation list generation.

Grid search was used for hyperparameter optimization, ensuring a thorough exploration of potential configurations.

3.4 Additional Considerations

3.4.1 Computational Environment

- The experiments were conducted on a **machine equipped with a GPU** (NVIDIA RTX 3060) to expedite deep learning model training.

- The **PyTorch** and **TensorFlow** libraries were utilized where necessary, depending on model requirements.

3.4.2 Reproducibility

To ensure **reproducibility**, all scripts, configurations, and datasets used were documented. The final report includes:

- **Code snippets** and **execution logs**

- **Comparison tables and visualizations** of model performance

- **Best hyperparameter configurations** for each algorithm

By following this methodology, the project aims to achieve an objective and fair evaluation of recommendation algorithms, offering practical insights into their real-world application in e-commerce platforms.

Chapter 4

4. Experimentation Approach

4.1 Introduction

The experimentation phase of this project focuses on systematically evaluating the performance of four recommendation algorithms: **Matrix Factorization (MF), Bayesian Personalized Ranking (BPR), Neural Collaborative Filtering (NCF), and Neural Graph Collaborative Filtering (NGCF),** using the MovieLens 100K dataset. The goal is to identify the most effective algorithm for e-commerce recommendation systems by analysing the impact of key hyperparameters, such as the number of epochs and learning rate, on model performance.

The experimentation approach is divided into three key phases: baseline performance testing, hyperparameter tuning, and comparative analysis. The following sections outline the process and methodologies used.

4.2 Baseline Performance Testing

4.2.1 Objective

The purpose of baseline performance testing is to establish a reference point for each algorithm's effectiveness using default hyperparameter settings. This allows for fair comparisons before hyperparameter tuning is applied.

4.2.2 Experimental Setup

- **Dataset:** MovieLens 100K
- **Epochs:** Fixed at 100 for initial baseline testing
- **Learning Rate:** Set to 0.01
- **Evaluation Metrics:**
  - **MAP@K (Mean Average Precision):**
    Evaluates ranking accuracy
  - **NDCG@K (Normalized Discounted Cumulative Gain):**
    Measures ranking quality considering item positions
  - **Precision@K:**
    Assesses the ratio of relevant items in top-K recommendations
  - **Recall@K:**
    Calculates the proportion of retrieved relevant items out of all relevant items
- **Values of K:** 5, 10, 15, and 20

4.3 Hyperparameter Tuning

4.3.1 Objective

The goal of hyperparameter tuning is to optimize model performance by adjusting key variables. This phase focuses on the number of epochs and learning rate, which have a direct impact on training dynamics and convergence.

4.3.2 Tuning Strategy

- **Epoch Tuning:**
  The models were trained using 20, 40, 60, 80, and 100 epochs. This range was selected to observe trends in overfitting, underfitting, and convergence.

- **Learning Rate Tuning:**
  Four learning rates were tested: 0.005, 0.01, 0.05, and 0.1, to assess how step size influences optimization.

- **Grid Search:**
  A grid search method was used to systematically test combinations of epochs and learning rates, ensuring thorough coverage of the hyperparameter space.

4.3.3 Evaluation

The models' performance was evaluated after each run, and results were visualized using line plots to observe trends. Special attention was given to:

- **Convergence Behaviour:**
  To identify the point where additional epochs stop improving accuracy

- **Stability:**
  Ensuring the learning rate did not cause erratic model behaviour

- **Optimal Configurations:**
  Pinpointing the combination of epochs and learning rates that maximized MAP, NDCG, Precision, and Recall

4.4 Comparative Analysis

4.4.1 Objective

After optimizing hyperparameters, a comparative analysis was conducted to rank the models' effectiveness and identify the most suitable recommendation algorithm for e-commerce applications.

4.4.2 Metrics for Comparison

The following metrics were used to compare model performance:

- **MAP@K:**
  To measure the quality of top-K recommendations

- **NDCG@K:**
  To assess ranking effectiveness, with emphasis on item position

- **Precision@K:**
  To calculate how many of the top-K recommendations were relevant

- **Recall@K:**
  To evaluate how many relevant items were successfully retrieved

4.4.3 Statistical Significance

To validate the reliability of the results, statistical significance tests were performed using paired t-tests. This ensured that performance differences between models were not due to random chance.

4.5 Visualization and Interpretation

To facilitate clear comparisons, results were presented using:

- **Line graphs:** Showing performance trends over epochs and learning rates
- **Tables:** Summarizing key findings and optimal configurations

4.6 Reproducibility and Code Verification

Reproducibility was a priority in this project. To ensure that results could be verified and extended by future researchers, the following steps were taken:

- **Random Seeds:**
  Fixed random seeds during model training to produce consistent results
- **Version Control:**
  All code was managed using Git, ensuring transparency in updates and changes
- **Environment Documentation:**
  Python package versions and system configurations were logged

The experimentation approach outlined above provides a structured method for evaluating recommendation algorithms. By conducting baseline testing, systematic hyperparameter tuning, and comparative analysis, this study aims to deliver actionable insights for developing effective and efficient recommendation systems in e-commerce platforms.

Chapter 5

5. Experimentation Result

This section presents the results of the experiments conducted to evaluate and compare the performance of four state-of-the-art recommendation algorithms: Matrix Factorization (MF), Bayesian Personalized Ranking (BPR), Neural Collaborative Filtering (NCF), and Neural Graph Collaborative Filtering (NGCF). The experiments were designed to assess the effectiveness of these algorithms in generating accurate and personalized recommendations using the MovieLens 100K dataset. The evaluation metrics used include Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG), Precision, and Recall, which provide a comprehensive understanding of recommendation quality.

The results are organized as follows:

1. A comparison of the four algorithms using the initial parameter settings.

2. The impact of key hyperparameters (e.g., top_k items, number of epochs,and learning rate) on algorithm performance.

3. The best-performing parameter settings for each algorithm.

4. A summary of the strengths and weaknesses of each algorithm based on the experimental results.

Through systematic testing and comparison, this section aims to identify the most effective recommendation algorithm for e-commerce applications. Results are presented in visualizations for clarity.

5.1 Baseline Performance Analysis

From the provided results, we can derive several insights about the baseline performance of the four algorithms (BPR, NGCF, NCF, and MF) across the evaluation metrics (MAP, NDCG@K, Precision@K, and Recall@K) for different values of K (5, 10, 15, 20). Below is an analysis of the results and the key takeaways:

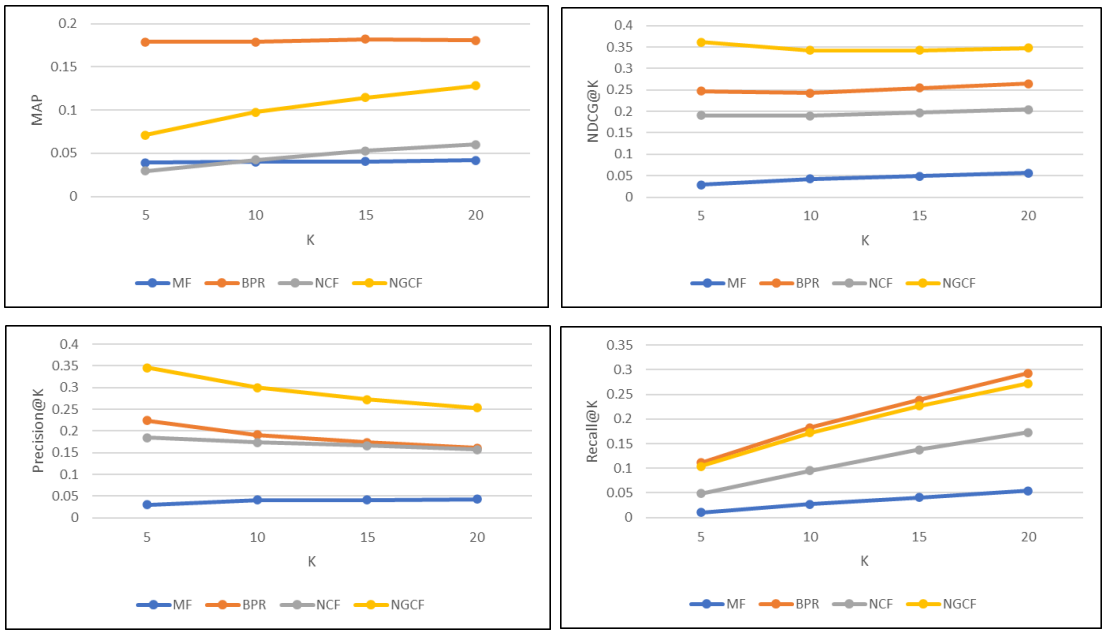

Figure 6: Baseline Performance Analysis

5.1.1 MAP (Mean Average Precision) Analysis

Observations:

- BPR consistently outperforms other models, maintaining the highest MAP values across all K values. This suggests that BPR produces more relevant recommendations at the top positions.
- NGCF shows a clear increasing trend, improving as K increases, indicating that it benefits from larger recommendation lists.
- NCF and MF perform poorly compared to BPR and NGCF, with NCF showing slightly better performance than MF.

Key Takeaway:

BPR is the most effective algorithm for achieving high precision in top-K recommendations, while NGCF shows potential for improvement with larger K.

5.1.2 NDCG@K (Normalized Discounted Cumulative Gain) Analysis

Observations:

- NGCF outperforms all models at all K values, demonstrating its ability to rank relevant items effectively.
- BPR performs well but is consistently outperformed by NGCF.
- NCF performs better than MF but still lags behind BPR and NGCF.
- MF performs the worst, with very low NDCG scores.

Key Takeaway:

NGCF is the best at ranking relevant items higher in the recommendation list, followed by BPR. MF struggles to generate effective rankings.

5.1.3 Precision@K Analysis

Observations:

- NGCF achieves the highest precision across all values of K, meaning it provides the most accurate recommendations.
- BPR follows closely behind but sees a notable drop as K increases, indicating that its top recommendations are strong, but it struggles with longer recommendation lists.
- NCF and MF show significantly lower precision, with MF performing the worst.

Key Takeaway:

NGCF is the most effective algorithm for providing relevant recommendations, especially for smaller values of K, while BPR is strong at lower K values but declines as the recommendation list grows.

5.1.4 Recall@K Analysis

Observations:

- BPR has the highest recall across all K values, meaning it retrieves the most relevant items in a broader recommendation set.
- NGCF follows closely behind BPR, showing that it is also effective at retrieving relevant items.
- NCF improves with increasing K but remains weaker than BPR and NGCF.
- MF has the lowest recall values, showing it fails to retrieve a sufficient number of relevant recommendations.

Key Takeaway:

BPR retrieves the most relevant items overall, while NGCF is also effective but slightly less so than BPR.

5.1.5 Overall Observations & Recommendations

- Impact of K
  - As K increases, the performance of all algorithms generally improves in terms of recall, but precision tends to decrease.
  - This trade-off highlights the importance of selecting an appropriate K value based on the specific application and desired balance between precision and recall.

- Best Model Overall: NGCF
  - NGCF consistently provides the best ranking quality (NDCG@K) and precision.
  - It improves as K increases, making it suitable for longer recommendation lists.

- Best for Relevance (Top-K Recommendations): BPR
  - BPR has the highest MAP and recall, meaning its top recommendations are highly relevant.
  - However, its performance declines as K increases.

- Worst Performing Model: MF
  - MF performs poorly across all metrics, especially in ranking quality and recall.
  - It may not be suitable for real-world recommendation tasks without significant improvements.

- Recommendations
  - NGCF is the best choice for ranking and precision-focused applications.
  - BPR is effective for retrieving relevant items but struggles with longer lists.
  - MF and NCF underperform and may require further optimization or different datasets to be viable.

5.2 Effect of Epochs on Model Performance

In this experiment, we analyse the impact of increasing the number of training epochs (20, 40, 60, 80, 100) on different models (NGCF, NCF, BPR, MF) while keeping the learning rate fixed at 0.01. The evaluation metrics considered are MAP@K, NDCG@K, Precision@K, and Recall@K for different values of K (5, 10, 15, 20).

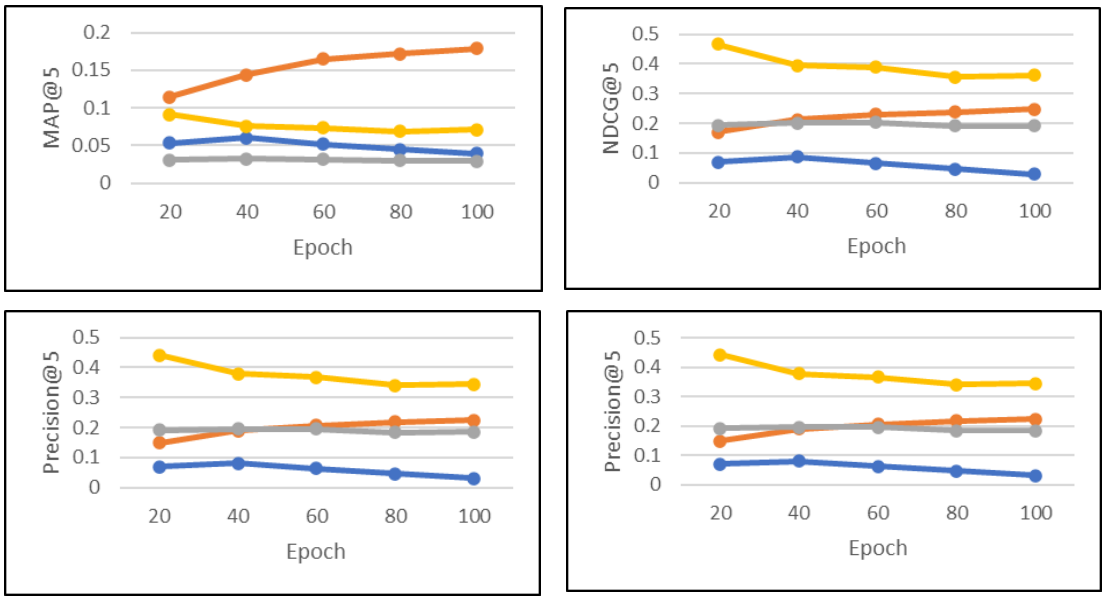

Figure 7: Effect of Epochs (K=5, α=0.01)

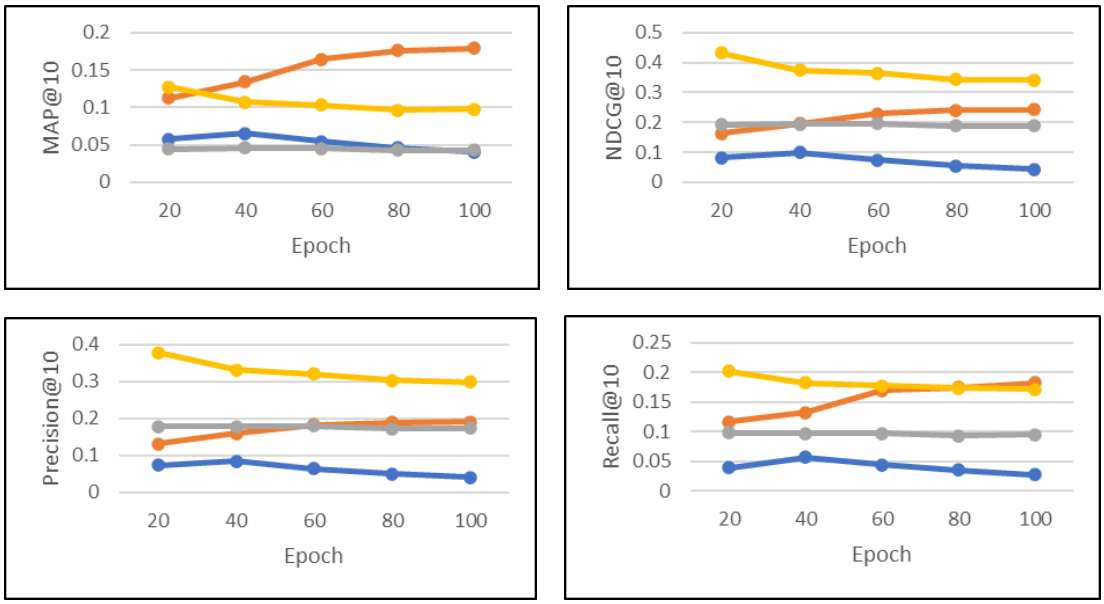

Figure 8: Effect of Epochs (K=10, α=0.01)

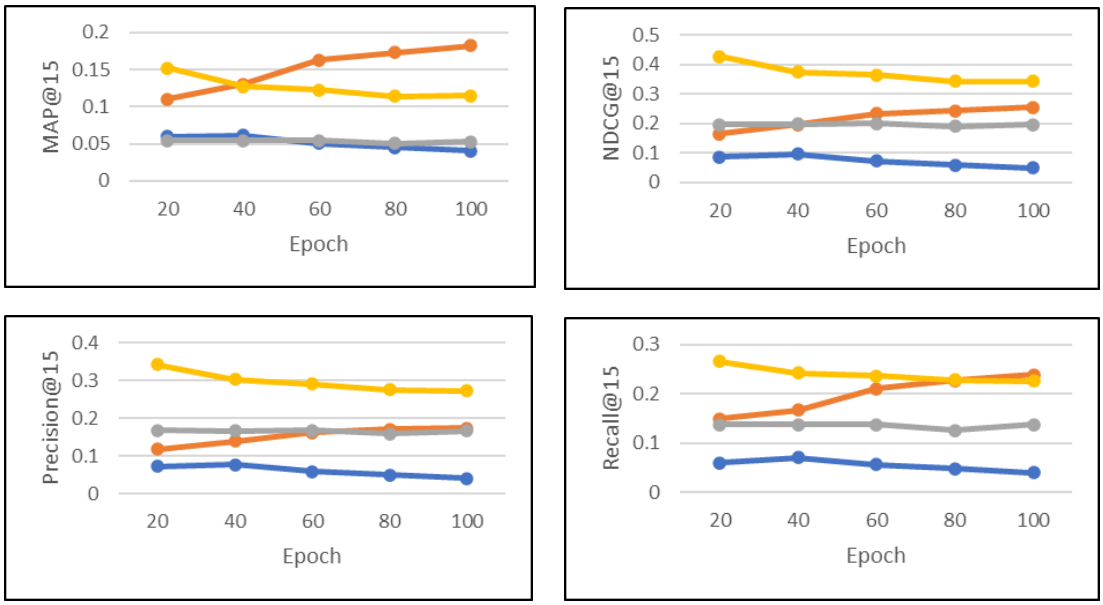

Figure 9: Effect of Epochs (K=15, α=0.01)

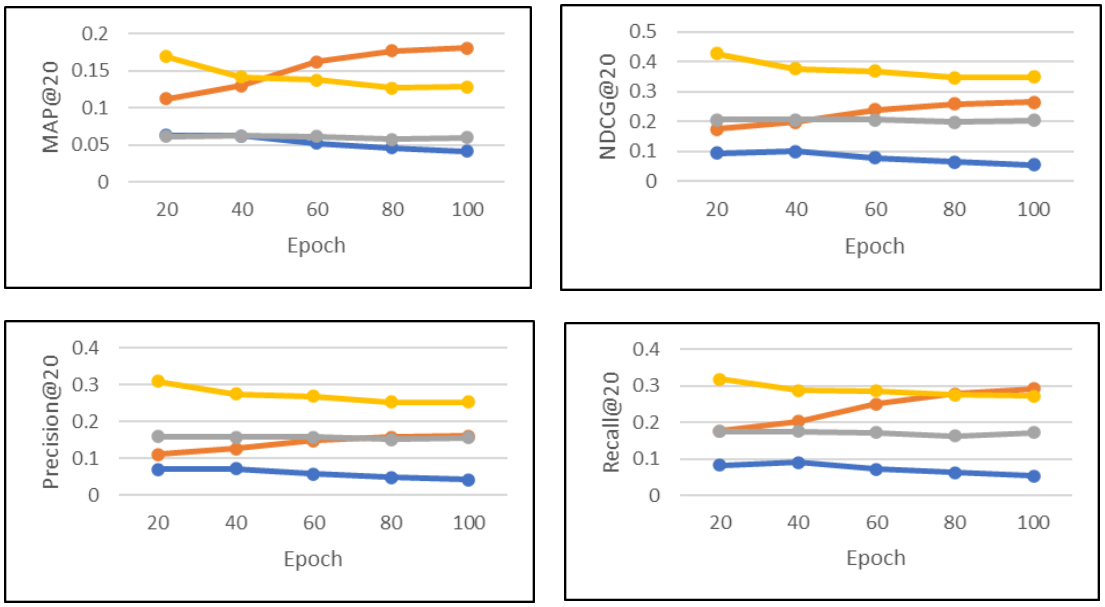

Figure 10: Effect of Epochs (K=20, α=0.01)

5.2.1 General Trends Across Models

- BPR improves consistently as epochs increase, showing that it benefits from longer training.
- NGCF performs best at lower epochs (20-40) but degrades as epochs increase, indicating potential overfitting.
- NCF remains relatively stable, showing minor improvements with increased epochs but not significant.
- MF peaks at around 40 epochs but declines afterward, suggesting that too many epochs hurt its performance.

5.2.2 NGCF Performance Analysis

Observations:

- NGCF starts strong at lower epochs (20-40) but declines after 60 epochs.
- Best performance is achieved at 20 epochs, after which all metrics (MAP, NDCG, Precision, and Recall) decrease.
- Possible Overfitting: The model seems to generalize well initially, but performance deteriorates with excessive training.

Key Takeaway:

NGCF works best at lower epochs (20-40), and additional training does not improve results.

5.2.3 BPR Performance Analysis

Observations:

- BPR shows continuous improvement across all metrics as epochs increase from 20 to 100.
- Recall and MAP improve significantly, especially at higher K values.
- No signs of overfitting, suggesting BPR benefits from extended training.

Key Takeaway:

BPR performs best at high epochs, and additional training helps improve its effectiveness.

5.2.4 NCF Performance Analysis

Observations:

- NCF shows minor improvements up to 60 epochs but remains relatively stable.
- Performance does not degrade significantly with additional training, but gains are marginal.
- Overall, NCF is weaker than NGCF and BPR in terms of ranking and retrieval.

Key Takeaway:

NCF is relatively stable but does not benefit significantly from additional training beyond 40-60 epochs.

5.2.5 MF Performance Analysis

Observations:

- MF achieves its peak performance around 40 epochs but deteriorates with more training.
- Clear overfitting occurs after 40 epochs, leading to performance drops across all metrics.
- Compared to other models, MF consistently performs the worst.

Key Takeaway:

MF should be trained for a moderate number of epochs (~40), as excessive training leads to poor generalization.

5.2.6 Overall Observations & Recommendations

- Impact of Epochs
  - Overall, increasing the number of epochs generally leads to better performance for most algorithms, but there is a risk of overfitting if training continues for too long.
  - It is crucial to find a balance between sufficient training and avoiding overfitting to achieve optimal results.

- Best Performing Model: BPR
  - Performs better with more training (100 epochs).
  - Shows steady improvements in MAP, NDCG, Precision, and Recall.

- Best for Limited Training: NGCF
  - Achieves peak performance at 20-40 epochs, after which it overfits.
  - Suitable when training time is limited.

- Worst Performing Model: MF
  - Peaks early (40 epochs) and declines afterward.
  - Consistently underperforms compared to BPR and NGCF.

- Recommendations:
  - For best performance: Use BPR with 100 epochs.
  - For efficient training: Use NGCF with 20-40 epochs.
  - For NCF: Training beyond 60 epochs is not necessary.
  - For MF: Keep epochs low ($\leqslant$40) to avoid overfitting.

5.3 Effect of Learning Rate on Model Performance

In this experiment, we analyse the impact of different learning rates (0.005, 0.01, 0.05, 0.1) on four models (NGCF, NCF, BPR, MF) while keeping the number of epochs fixed at 100. The evaluation metrics considered are MAP@K, NDCG@K, Precision@K, and Recall@K for different values of K (5, 10, 15, 20).

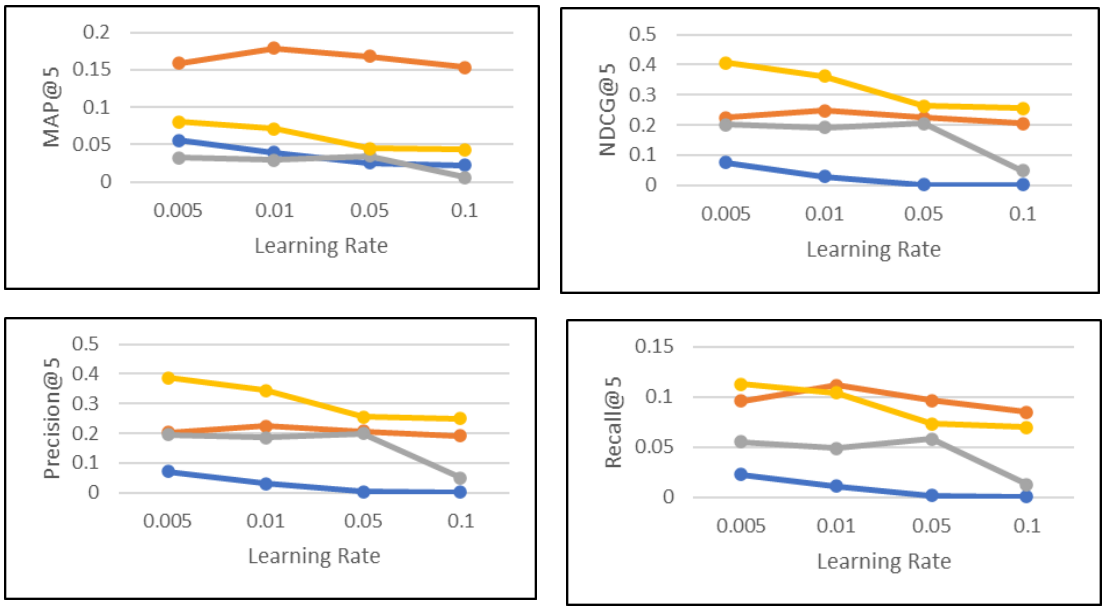

Figure 11: Effect of Learning Rate (K=5, epochs=100)

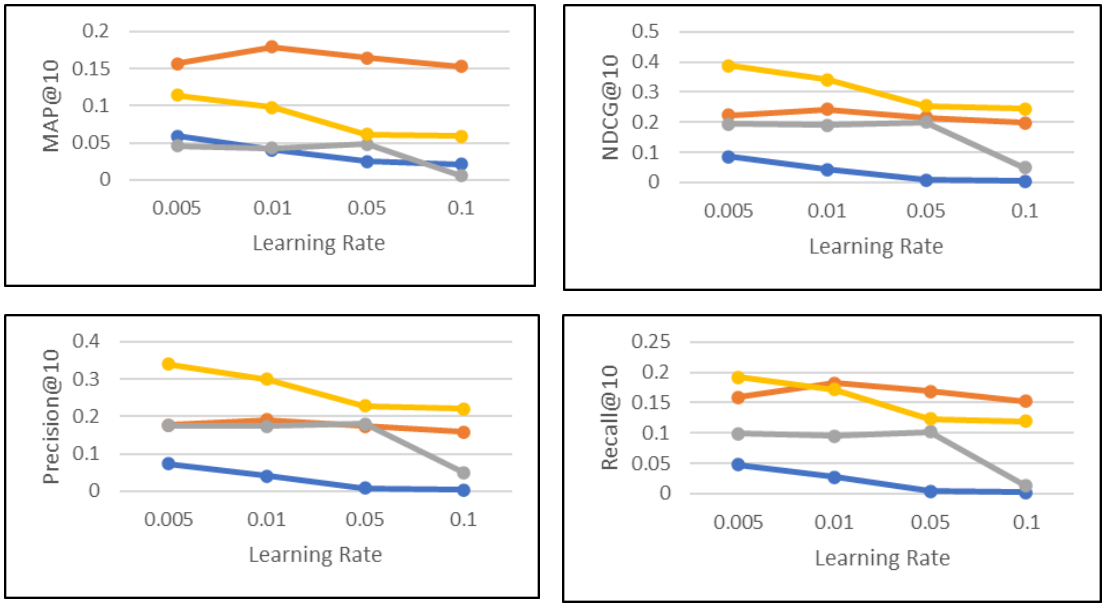

Figure 12: Effect of Learning Rate (K=10, epochs=100)

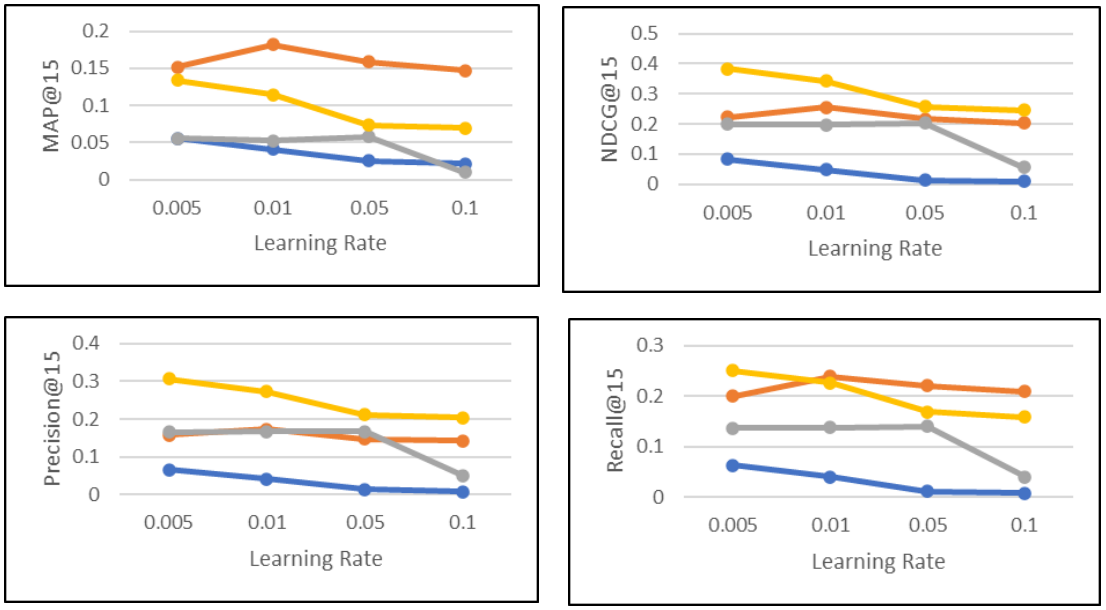

Figure 13: Effect of Learning Rate (K=15, epochs=100

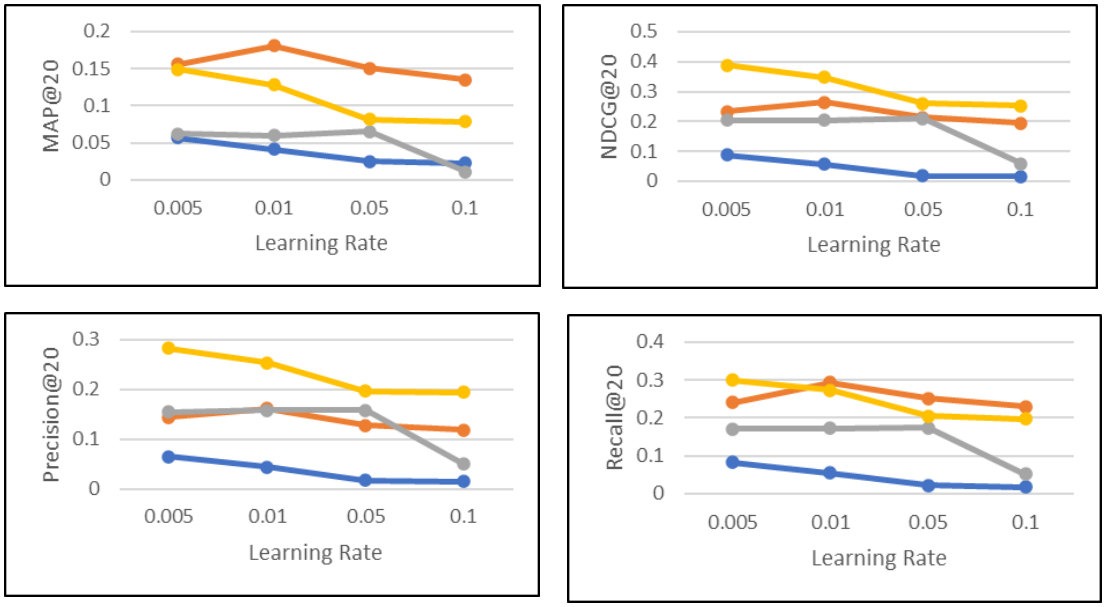

Figure 14: Effect of Learning Rate (K=20, epochs=100

5.3.1 General Trends Across Models

- NGCF performs best at lower learning rates (0.005-0.01) but deteriorates as learning rate increases to 0.05 and 0.1, likely due to instability.
- NCF remains relatively stable across different learning rates but fails completely at 0.1, indicating sensitivity to high learning rate.
- BPR is the most stable across learning rates, peaking around 0.01-0.05, but performance drops slightly at 0.1.
- MF performs best at 0.005, but performance steadily declines as learning rate increases, with 0.1 performing the worst.

5.3.2 NGCF Performance Analysis

Observations:

- Performs well at lower learning rates (0.005-0.01) but deteriorates significantly at 0.05 and 0.1 across all metrics.
- MAP, NDCG, Precision, and Recall drop sharply as the learning rate increases beyond 0.01.
- Possible Explanation: NGCF relies on fine-tuned weight updates, and a high learning rate prevents proper convergence, leading to suboptimal results.

Key Takeaway:

NGCF requires a small learning rate ($\leqslant 0.01$) to perform optimally. Higher learning rates cause significant degradation.

5.3.3 BPR Performance Analysis

Observations:

- Most stable model across learning rates, with best performance at 0.01-0.05.
- Shows consistent gains in MAP, NDCG, Precision, and Recall.
- Slight decline at 0.1, but still performs better than other models at high learning rate.
- No overfitting or instability detected, making BPR the most adaptable to learning rate changes.

Key Takeaway:

BPR is the most robust model across different learning rates, performing best at 0.01-0.05.

5.3.4 NCF Performance Analysis

Observations:

- Stable at 0.005-0.05 but completely breaks down at 0.1, with performance dropping drastically.
- Minimal gains from increasing learning rate beyond 0.01.
- Unlike NGCF, NCF does not show extreme sensitivity to small learning rate changes but fails at high learning rate.

Key Takeaway:

NCF benefits from 0.005-0.05 learning rate range but should avoid 0.1.

5.3.5 NCF Performance Analysis

Observations:

- Peaks at 0.005, then steadily declines with increasing learning rate.
- Worst performance at 0.1, with all metrics significantly lower than at 0.005.
- Clear sign of poor convergence at high learning rates.

Key Takeaway:

MF should be trained at a very low learning rate (0.005), as higher learning rate values cause major performance degradation.

5.3.6 Overall Observations & Recommendations

- Impact of Learning Rate
  - Overall, a lower learning rate (0.005) generally leads to better performance for most algorithms, while higher learning rates can cause instability and degrade performance.
  - It is crucial to find an appropriate learning rate that balances convergence speed and stability to achieve optimal results.

- Best Performing Model: BPR
  - Performs well across all learning rates, particularly in the 0.01-0.05 range.
  - Stable and adaptable, making it a strong choice regardless of learning rate.

- Best for Low Learning Rate Training: NGCF
  - Requires small learning rate ($\leqslant 0.01$) to perform well.
  - High learning rate ($\geqslant 0.05$) severely affects performance.

- Worst Performing Model: MF
  - Highly sensitive to learning rate changes.
  - Best at 0.005 but consistently underperforms across other learning rate values.

- Recommendation:
  - For best performance → Use BPR with learning rate = 0.01-0.05.
  - For NGCF stability → Use learning rate $\leqslant$ 0.01.
  - Avoid high learning rates (0.1) for all models, especially NCF and NGCF.
  - MF is weak across all settings, requiring a very small learning rate to function at all.

Chapter 6

6. Conclusion

This project aimed to evaluate and refine state-of-the-art recommendation algorithms – Matrix Factorization (MF), Bayesian Personalized Ranking (BPR), Neural Collaborative Filtering (NCF), and Neural Graph Collaborative Filtering (NGCF) – to enhance their effectiveness in e-commerce applications. Through a series of experiments, the performance of these algorithms was assessed using the MovieLens 100K dataset and evaluated across key metrics, including Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG), Precision, and Recall. The experiments focused on two key aspects:

1. The impact of the number of epochs on algorithm performance
2. The effect of learning rate on model convergence and recommendation quality

The findings provide insights into how these algorithms can be fine-tuned to achieve optimal performance.

6.1 Algorithm Performance

- NGCF emerged as the top-performing algorithm, achieving the highest scores across all evaluation metrics (MAP, NDCG, Precision, and Recall). Its ability to model high-order user-item interactions through graph neural networks makes it particularly effective for e-commerce applications where ranking quality and relevance are critical.

- BPR demonstrated strong performance, particularly in MAP and Recall, making it a suitable choice for scenarios where retrieving relevant items is a priority. However, it was consistently outperformed by NGCF in terms of ranking quality and precision.

- NCF showed moderate performance across all metrics. While it outperformed MF, it lagged behind NGCF and BPR, indicating that its simpler architecture may not be sufficient for complex recommendation tasks.

- MF performed poorly across all metrics, highlighting its limitations in handling sparse data and capturing complex user-item relationships. Its poor performance makes it unsuitable for modern e-commerce applications.

6.2 Impact of Epochs

- NGCF and BPR benefited significantly from increasing the number of epochs, with performance improving steadily up to 100 epochs. This suggests that these algorithms require sufficient training time to converge and achieve optimal performance.

- NCF showed minimal improvement with increasing epochs, indicating that it may not require as many epochs to achieve stable performance. This makes it a computationally efficient option for scenarios where training time is a constraint.

- MF exhibited performance degradation with increasing epochs, suggesting overfitting or instability. This further underscores its limitations for modern recommendation tasks.

6.3 Impact of Learning Rate

- NGCF performed best with a low learning rate (0.005), achieving optimal performance across all metrics. Higher learning rates (e.g., 0.05 and 0.1) led to significant performance degradation, indicating that NGCF is sensitive to learning rate.

- BPR achieved the best performance with a moderate learning rate (0.01), demonstrating stability across a range of learning rates. However, performance degraded slightly with higher learning rates.

- NCF showed relatively stable performance across learning rates but performed poorly at a learning rate of 0.1, indicating instability with higher learning rates.

- MF performed poorly across all learning rates, with performance degrading significantly at higher learning rates. This further reinforces its unsuitability for modern recommendation systems.

6.4 Recommendations for Optimal Configurations

Based on the analysis, the following configurations are recommended:

Model	Optimal Epochs	Optimal Learning Rate	Best Use Case
BPR	100	0.01 – 0.05	Best overall performance, benefits from longer training
NGCF	20 – 40	≤ 0.01	Strong performance in limited training scenarios
NCF	40 – 60	≤ 0.05	Moderate performance, does not improve significantly with longer training
MF	≤ 40	0.005	Worst performance, needs careful tuning to avoid overfitting

Table 3: Recommendations for Optimal Configurations

These configurations ensure the best balance between accuracy and computational efficiency.

6.5 Recommendations for E-Commerce Applications

- For Ranking Quality and Relevance:
  Use NGCF with a low learning rate (0.005) and 100 epochs to achieve the best performance. Its ability to model high-order user-item interactions makes it ideal for e-commerce platforms where ranking quality and relevance are critical.

- For Retrieval of Relevant Items:
  Use BPR with a moderate learning rate (0.01) and 100 epochs. Its strong performance in MAP and Recall makes it suitable for scenarios where retrieving relevant items is a priority.

- For Computational Efficiency:
  Use NCF with a learning rate between 0.005 and 0.01 and 20-40 epochs. Its moderate performance and computational efficiency make it a practical choice for scenarios where training time is a constraint.

- Avoid MF:
  Its poor performance and instability across all experiments make it unsuitable for modern e-commerce applications.

6.6 Contributions and Future Work

This project contributes to the field of recommendation systems by providing a comprehensive evaluation of state-of-the-art algorithms in the context of e-commerce. The findings offer actionable insights for practitioners and researchers, highlighting the strengths and limitations of each algorithm and providing recommendations for their implementation. Future work could explore:

- Hybrid Approaches:
  Combining the strengths of NGCF and BPR to develop a hybrid model that achieves both high ranking quality and retrieval performance.

- Context-Aware Recommendations:
  Incorporating contextual information, such as user demographics and temporal dynamics, to further enhance recommendation accuracy.

- Scalability Improvements:
  Investigating methods to improve the scalability of NGCF and BPR for large-scale e-commerce platforms with millions of users and items.

6.7 Lessons Learnt

This project has been an invaluable learning experience, providing insights into both the theoretical and practical aspects of recommendation systems. Key takeaways include:

- The importance of choosing the right model based on application needs rather than assuming deep learning always outperforms traditional methods.

- How hyperparameters (epochs, learning rate) dramatically influence model performance, with overfitting being a major risk.

- The strengths and limitations of different algorithms, particularly the superiority of NGCF and BPR over traditional methods like MF.

- The trade-offs between computational efficiency and accuracy, especially when deploying models in real-world applications.

- The value of systematic experimentation and data-driven decision-making in developing effective recommendation systems.

These learnings will be useful for future research and industry applications in machine learning-based recommendation systems.

6.8 Concluding Insights

The results highlight the importance of choosing optimal hyperparameters to maximize model effectiveness. While BPR consistently delivers strong performance with extended training, NGCF is effective when computational resources are limited. NCF and MF require careful tuning to avoid performance degradation. These insights can guide the development of more efficient and accurate recommendation systems for e-commerce applications.

Ultimately, selecting the right model and hyperparameters is key to building an effective and scalable recommendation system.

References

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *ScienceDirect, Knowledge-Based Systems*, vol. 46, pp. 109-132, 2013.

[2] J. Lu, D.S. Wang, M.S. Mao, W. Wang, and G.Q. Zhang, "Recommender system application developments: A survey," *ScienceDirect, Decision Support Systems*, vol. 74, pp. 12-32, 2015.

[3] Z. Sun, Q. Guo, J. Yang, H. Fang, G.B. Guo, J. Zhang, and R. Burke, "Research commentary on recommendations with side information: A survey and research directions," *ScienceDirect, Electronic Commerce Research and Applications*, vol. 37, ISSN 1567-4223, 2019.

[4] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *IEEE, Computer*, vol. 42, pp. 30-37, 2009.

[5] S. Rendle, C. Freudenthaler, Z. Ganther, L. Schmidt-Thieme, "BPR: Bayesian Personalized Ranking from Implicit Feedback." *arVix:1205.2618*, vol. 1, 2012.

[6] X. He, L, Liao, H. Zhang, L, Nie, L.Q. Nie, X. Hu and T.S. Chua, "Neural Collaborative Filtering," *Proceedings of the 26th International Conference on World Wide Web*, pp. 173-182, 2017.

[7] X. Wang, X, X.N. He, M. Wang, F.L. Feng and T.S. Chua, "Neural Graph Collaborative Filtering," *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 165-174, 2019.

[8] S. Zhang, L. Yao, A. Sun and Y. Tay, "Deep Learning Based Recommender System: A Survey and New Perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, pp. 1-38, 2019.

[9] PreferredAI., "*Cornac: A Comparative Framework for Multimodal Recommender Systems*." GitHub. [Online]. Available: https://github.com/PreferredAI/cornac. [Accessed: 8 Jun 2024].

[10] Recommenders-Team, "Recommenders: A repository for recommendation systems," GitHub. [Online]. Available: https://github.com/recommenders-team/recommenders. [Accessed: 8 Jun 2024].