

**A MULTI-PERSPECTIVE ANALYSIS OF COLLABORATIVE
FILTERING PERFORMANCE: METRICS, TOPOLOGIES, AND
ALGORITHMS**

(İŞBİRLİKÇİ FİLTRELEME PERFORMANSININ ÇOK BOYUTLU ANALİZİ:
METRİKLER, TOPOLOJİLER VE ALGORİTMALAR)

by

MERT ARDA ASAR, B.S.

Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

in the

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

JUNE 2025

ACKNOWLEDGMENTS

I am deeply thankful to the distinguished faculty members of the Computer Engineering Department at Galatasaray University for their continuous support, enriching courses, and academic mentorship during my graduate education. Their expertise and encouragement have played a vital role in developing my interest and knowledge in the field.

I am particularly grateful to research supervisor Assoc. Prof. Dr. Günce Keziban ORMAN for her patient guidance, insightful feedback, and continuous encouragement throughout the course of this thesis. Her academic support and thoughtful critiques were invaluable in shaping both the direction and quality of this work.

I would also like to thank my family for their unconditional love, patience, and constant encouragement throughout my academic journey. Their belief in me has been a continuous source of motivation and strength, especially during challenging times.

Mert Arda Asar

30 June 2025

TABLE OF CONTENTS

LIST OF FIGURES	v
LIST OF TABLES	ix
ABSTRACT	xi
RÉSUMÉ	xiii
ÖZET	xv
LIST OF ABBREVIATIONS	xvii
1 INTRODUCTION	1
1.1 Overview	1
1.2 Problem Statement	4
1.3 Contributions	6
2 LITERATURE REVIEW	8
2.1 Recommendation Systems	8
2.1.1 Content-Based Filtering (CBF)	9
2.1.2 Collaborative Filtering (CF)	10
2.1.3 Hybrid Models	11
2.1.4 Challenges of Recommendation Systems	12
2.2 Graph Neural Networks	15
2.2.1 Graph Construction Types for GNNs	16
2.2.2 Recommendation Systems with GNNs	19
2.3 Graph Topology and Recommendation Systems	22
3 REAL WORLD BIPARTITE GRAPHS	27
3.1 Graph Topological Features	27
3.2 Real World Datasets as Bipartite Graphs and Their Properties	32

4	METHOD	42
4.1	Personalized Recommendation Algorithms	42
4.2	Recommendation System Performance Evaluation	53
4.2.1	Quantitative Performance Evaluation : Accuracy Metrics	54
4.2.2	Qualitative Performance Evaluation Metrics	56
4.2.2.1	Fairness Metrics	56
4.2.2.2	Diversity Metrics	58
5	EXPERIMENTAL DESIGN AND RESULTS	60
5.1	Experiment Design	60
5.2	Train Datasets	67
5.3	Results	73
5.3.1	Models' Performance According to Evaluation Dimensions (RQ1)	73
5.3.2	Inter-Metric Relationships and Consistency (RQ2)	75
5.3.3	Model Dependency of Evaluation Metrics (RQ3)	77
5.3.4	Impact of Graph Topology on Model Performance (RQ4)	82
5.3.5	Complementary Statistical Analysis	88
6	CONCLUSION	91
	REFERENCES	96
	APPENDICES	110
	APPENDIX. A.	110
	APPENDIX. B.	114

LIST OF FIGURES

Figure 2.1 Illustration of individual-level diversity and system-level diversity (Gao et al., 2023)	13
Figure 2.2 An Illustration Of Main Graph Types. Different Colored Nodes Correspond To The Different Node Type. Figure a Represents The Homogenous Graphs And Figure b Represents The Heterogenous Graphs	17
Figure 2.3 An Illustration of Bipartite (Figure a) and Tripartite (Figure b) Graphs. Each Color Represents Different Node Type.	19
Figure 2.4 An Illustration of some common recommendation tasks. Figure a represents the social recommendation which includes both user-user and user-item edges. Figure b represents the sequential recommendation where the order of the interactions also considered. Figure c represents the bundle recommendation where a set of items are recommended at once and Figure d represents the multi-behavior recommendatiton where there are multiple user-item edge types	20
Figure 2.5 An Illustration of User-Item Recommendation	20
Figure 3.1 Degree Distributions of Entire Datasets	36
Figure 3.2 Degree Exponents of Entire Datasets	39
Figure 3.3 Degree Exponents of Only the User Nodes	40
Figure 3.4 Degree Exponents of Only the Item Nodes	41

Figure 4.1 A taxonomy diagram of the evaluated models in the thesis. Purple rectangles represent the main model categories, while blue rounded boxes indicate subcategories based on architectural or learning principles. Green ovals list the individual models evaluated under each category.	44
Figure 5.1 Degree Exponents of All Nodes in Train Sets	70
Figure 5.2 Degree Exponents of Only the User Nodes in Train Sets	71
Figure 5.3 Degree Exponents of Only the Item Nodes in Train Sets	72
Figure 5.4 Distribution of scaled model performances across evaluation metrics. Each subplot corresponds to a different metric, showing the distribution of performance scores after min-max normalization within each dataset-metric group. Boxes represent normalized score ranges with the median marked as a horizontal line, and the mean is highlighted with a black diamond. Colors indicate the architectural category of each model.	75
Figure 5.5 Pearson Correlation matrices displaying metric relationships within individual recommendation models. Each heatmap represents a different recommendation model, showing the correlations between different evaluation metrics. Red cells indicate positive correlations while blue cells represent negative correlations.	78
Figure 5.6 Correlation matrices illustrating the Pearson correlations between performance metrics across recommendation models for accuracy measures (NDCG, HR, Recall, Precision, and MRR). Darker red represents stronger positive correlations, while darker blue indicates stronger negative correlations.	79
Figure 5.7 Model correlations for diversity metrics across recommendation systems. The heatmaps illustrate Pearson correlation coefficients between different recommendation models for four diversity metrics : Intra-List Distance (ILD), Item Coverage (IC), Personalization (Pers), and Novelty (Nov). Red indicates positive correlations while blue represents negative correlations, with color intensity corresponding to correlation strength.	80

Figure 5.8 Model correlations for fairness metrics across recommendation systems. The heatmaps display Pearson correlation coefficients between different recommendation models for three fairness metrics : Average Popularity (AP), Gini Index (GINI), and Item Exposure Fairness (IEF). Strong positive correlations appear in dark red, while strong negative correlations are shown in dark blue, illustrating how consistently different models rank datasets on these fairness dimensions. 81

Figure 5.9 Mean correlation analysis of evaluation metrics across recommendation models. Bar height indicates the average Pearson correlation coefficient for each metric when comparing between different models, with error bars showing minimum and maximum values. Metrics are color-coded by category (green for accuracy, blue for diversity, red for fairness). Higher correlation values suggest metrics that are more consistent across different model architectures, while lower values indicate metrics that better differentiate between models. 82

Figure 5.10 Scatter plot illustrating the relationship between a performance metric and a topological feature. Each color represents a different recommendation model. 83

Figure 5.11 MDI Feature importance heatmaps for accuracy metrics across different recommendation models. The heatmap shows the relative importance of various topological features (rows) for predicting NDCG, HR, Recall, Precision, and MRR performance for different recommendation algorithms (columns). Color intensity indicates feature importance, with darker red representing higher importance values. 84

Figure 5.12 MDI Feature importance heatmaps for diversity metrics across different recommendation models. The heatmap illustrates the importance of topological features (rows) for predicting ILD, IC, Personalization, and Novelty metrics for different recommendation algorithms (columns). Darker red cells indicate higher feature importance for the corresponding model-metric pair. 86

Figure 5.13MDI feature importance heatmaps for fairness metrics across different recommendation models. The heatmap displays the importance of topological features (rows) for predicting AP, GINI, and IEF metrics for different recommendation algorithms (columns). Higher importance values are represented by darker red coloration. 87

Figure 5.14Boxplots of normalized metric scores for each recommendation model. Each subplot represents a model, and the boxplots within each show the distribution of scores across datasets for 12 different evaluation metrics. Metrics are grouped by color to reflect evaluation dimensions : accuracy (red), diversity (green), and fairness (brown). 90



LIST OF TABLES

Table 2.1 Comparison of studies examining the impact of topological properties on model performance for various tasks	25
Table 3.1 Total number of users, items and interactions of experienced datasets.	34
Table 3.2 Degree statistics for experienced datasets. μ : Average Degree, σ : Standard Deviation, min : Minimum Degree, max : Maximum Degree. . . .	35
Table 3.3 α Values for Degree Exponents. A- α : alpha for the entire graph, U- α : alpha for only the user nodes' degree distribution, I- α : alpha for only the item nodes' degree distribution	37
Table 3.4 Graph properties for the experienced datasets. EC : Edge clustering coefficient, U-I Ratio : User-item ratio.	38
Table 4.1 Comparison of experienced recommendation models	43
Table 4.2 System Evaluation Metrics	54
Table 5.1 Total users, items and interactions of training datasets	67
Table 5.2 Degree statistics for the train sets. μ : Average Degree, σ : Standard Deviation, min : Minimum Degree, max : Maximum Degree.	68
Table 5.3 α Values for Degree Exponents of train sets. A- α : alpha for all nodes, U- α : alpha for only the user nodes' degree distribution, I- α : alpha for only the item nodes' degree distribution	68

Table 5.4 Graph properties for the training sets. Sparsity : Graph sparsity, Item Homophily : Tendency of items to be connected to similar items, L3 Similarity : Average 3-hop connectivity, EC : Edge clustering coefficient, U-I Ratio : User-item ratio.	69
Table APPENDIX A.1 Accuracy Performance for Each Model	111
Table APPENDIX A.2 Diversity Performance for Each Model	112
Table APPENDIX A.2 Fairness Performance	113
Table APPENDIX B.1 R^2 Regression Results Between Graph Topological Features and Accuracy Metrics	114
Table APPENDIX B.2 R^2 Regression Results Between Graph Topological Features and Diversity Metrics	114
Table APPENDIX B.3 R^2 Regression Results Between Graph Topological Features and Fairness Metrics	114

ABSTRACT

Recommendation systems play a central role in shaping user experiences across digital platforms by suggesting relevant items based on user behavior. Despite significant progress in model development, evaluating the performance of recommendation algorithms remains a complex challenge. Most prior work relies heavily on *accuracy*-oriented metrics, often overlooking other critical aspects such as *diversity* and *fairness*. Moreover, the effect of data characteristics—especially graph topological features—on model performance has not been fully explored. This gap creates uncertainty about whether high accuracy scores genuinely reflect model effectiveness in real-world scenarios or merely exploit structural biases in the data.

This thesis aims to provide a comprehensive and systematic evaluation framework for personalized collaborative filtering models (PCFM) by examining how their performance is influenced by three key components: (i) evaluation metrics, (ii) graph topological properties of datasets, and (iii) algorithmic model characteristics. Each of these components involves multiple subcategories, such as different metric groups (accuracy, diversity, fairness), dozens of topological features, and a wide variety of model architectures (e.g. KNN-based, graph-based methods and autoencoder-based methods). Due to the high dimensionality and complexity of these interactions, simple or intuitive analysis is insufficient. Instead, we design a structured experimental setup guided by four central research questions (RQs), each addressing a specific dimension of the evaluation problem:

- RQ1: How do the recommendation models’ performance differ according to evaluation criteria?
- RQ2: Are the evaluation metrics coherent with each other?
- RQ3: Are performance metrics sensitive to the used recommendation model?
- RQ4: Can the recommendation model performance explained by topological characteristics?

To address these questions, we conduct large-scale experiments involving 13 recommen-

dation models, 10 real-world bipartite datasets, and 12 performance metrics spanning three evaluation dimensions. We further extract a rich set of topological features from each dataset and analyze the dependencies between data structure and model behavior.

Our findings reveal several key insights. First, model rankings are not stable across datasets or metrics. A model that performs well on one dataset may fall behind others on a different dataset, even within the same metric group. Similarly, a model may excel on a particular metric but perform poorly on different metrics. This variability highlights the importance of multidimensional evaluation and calls into question the validity of studies that rely solely on limited accuracy metrics. Second, evaluation metrics within the same group (e.g., diversity metrics) are not always consistent with each other. Metrics such as novelty and intra-list distance may favor different models, suggesting that each metric captures a different aspect of user experience. Therefore, relying on a single metric to represent an entire evaluation dimension may lead to incomplete or misleading conclusions. Third, our analysis suggests that graph topological feature have distinguishable impact on model performance, with some features appearing to be more relevant for certain model types. These effects are can be non-linear and can differ across model families.

In conclusion, this thesis emphasizes that model evaluation must go beyond simplistic comparisons. Accurate, diverse, and fair recommendations depend not only on algorithmic design but also on dataset properties and metric selection. Our proposed framework and findings serve as a foundational step toward more reliable, objective, and generalizable assessments of recommendation systems. We believe this approach will help researchers and practitioners make more informed choices in both academic benchmarking and real-world deployment scenarios.

Keywords : Recommendation Systems, Collaborative Filtering, Recommendation Performance Evaluation, Graph Topology, Evaluation Metrics, Fairness, Diversity, Accuracy

RÉSUMÉ

Les systèmes de recommandation jouent un rôle crucial dans les plateformes numériques en suggérant des éléments pertinents selon les comportements des utilisateurs. Malgré des avancées significatives dans le développement des modèles, l'évaluation de leur performance reste complexe. La majorité des études se concentrent principalement sur la précision, au détriment d'aspects essentiels comme la diversité ou l'équité. En outre, l'influence des propriétés topologiques des graphes sur les performances des modèles est encore peu explorée.

Ce mémoire propose un cadre d'évaluation systématique des modèles de filtrage collaboratif personnalisé (PCFM) en analysant trois dimensions principales : (i) les métriques d'évaluation (précision, diversité, équité), (ii) les propriétés topologiques des graphes dérivés des jeux de données, et (iii) les caractéristiques algorithmiques des modèles. Quatre questions de recherche structurent cette analyse:

- Question 1 : Comment les performances des modèles varient-elles en fonction des critères d'évaluation?
- Question 2 : Les différentes métriques d'évaluation sont-elles cohérentes entre elles?
- Question 3 : Dans quelle mesure les résultats sont-ils sensibles au choix du modèle utilisé?
- Question 4 : Les performances des modèles peuvent-elles être expliquées par les propriétés topologiques des graphes?

Pour y répondre, nous avons conduit une étude expérimentale à grande échelle sur 13 modèles, 10 jeux de données bipartis réels et 12 métriques couvrant trois dimensions. Nous avons aussi extrait de nombreux indicateurs topologiques pour étudier les liens entre structure des données et comportements des modèles.

Les résultats montrent que le classement des modèles varie fortement d'un jeu de données à l'autre et selon les métriques. Un même modèle peut exceller selon une métrique mais être moins performant selon d'autres. De plus, même au sein d'un groupe (comme

la diversité), les métriques peuvent fournir des résultats divergents. Par ailleurs, certaines caractéristiques topologiques semblent influencer de manière significative les performances, selon l'architecture du modèle.

En conclusion, l'évaluation des systèmes de recommandation nécessite une approche multidimensionnelle. Ce travail met en lumière les limites des approches traditionnelles et propose un cadre plus complet pour guider les chercheurs et praticiens dans la sélection et l'évaluation des modèles.

Mots Clés : Systèmes de Recommandation, Filtrage Collaboratif, Évaluation des Performances, Topologie de Graphe, Métriques d'Évaluation, Équité, Diversité, Justesse

ÖZET

Öneri sistemleri, kullanıcı davranışlarına dayalı olarak ilgili içerikleri önererek dijital platformlardaki kullanıcı deneyimlerini şekillendirmede merkezi bir rol oynamaktadır. Model geliştirme konusunda önemli ilerlemeler kaydedilmiş olmasına rağmen, öneri algoritmalarının performansının değerlendirilmesi hâlâ karmaşık bir problemdir. Önceki çalışmaların çoğu *doğruluk* odaklı metriklere aşırı derecede bağımlıdır ve *çeşitlilik* ile *adalet* gibi diğer önemli boyutları çoğunlukla göz ardı etmektedir. Ayrıca, özellikle grafların topolojik özellikleri gibi veri yapılarının model performansı üzerindeki etkisi yeterince araştırılmamıştır. Bu eksiklik, yüksek doğruluk puanlarının gerçekten modelin etkililiğini mi gösterdiği yoksa sadece veri setinin yapısal önyargılarından mı faydalandığını sorgulanabilir hâle getirmektedir.

Bu tez, kişiselleştirilmiş işbirlikçi filtreleme modellerinin performansını üç temel bileşen açısından inceleyerek kapsamlı ve sistematik bir değerlendirme çerçevesi sunmayı amaçlamaktadır: (i) değerlendirme metrikleri, (ii) veri setlerinin grafiksel-topolojik özellikleri ve (iii) algoritmik model yapıları. Bu bileşenlerin her biri, farklı metrik grupları (doğruluk, çeşitlilik, adalet), onlarca topolojik özellik ve geniş bir model yelpazesi (örneğin KNN tabanlı, grafik tabanlı ve otomatik kodlayıcı temelli yöntemler) gibi çeşitli alt kategoriler içermektedir. Bu etkileşimlerin yüksek boyutluluğu ve karmaşıklığı nedeniyle, basit ya da sezgisel analizler yeterli olmamaktadır. Bu nedenle, her biri değerlendirme probleminin farklı bir boyutuna odaklanan dört araştırma sorusu etrafında şekillenen yapısal bir deneysel düzen tasarlanmıştır:

- Araştırma Sorusu 1: Öneri modellerinin performansı farklı değerlendirme kriterlerine göre nasıl değişmektedir?
- Araştırma Sorusu 2: Değerlendirme metrikleri birbiriyle ne kadar tutarlıdır?
- Araştırma Sorusu 3: Performans metrikleri kullanılan modele karşı ne kadar duyarlıdır?
- Araştırma Sorusu 4: Model performansı, grafiksel-topolojik özelliklerle açıklanabilir mi?

Bu soruları yanıtlamak için, 13 öneri modeli, 10 gerçek dünya iki parçalı veri seti ve üç değerlendirme boyutunu kapsayan 12 performans metriğini içeren büyük ölçekli deneyler gerçekleştirilmiştir. Ayrıca her veri setinden zengin bir topolojik özellik kümesi çıkarılmış ve veri yapısı ile model davranışı arasındaki ilişkiler analiz edilmiştir.

Elde edilen bulgular bazı önemli içgörülerini ortaya koymaktadır. İlk olarak, model sıralamaları ne metrikler ne de veri setleri arasında sabit değildir. Bir veri setinde iyi performans gösteren bir model, başka bir veri setinde aynı metrik grubu içinde diğerlerinin gerisinde kalabilir. Benzer şekilde, bir model belirli bir metrik açısından öne çıkarken, farklı metrikler açısından zayıf performans gösterebilir. Bu değişkenlik, çok boyutlu bir değerlendirme yaklaşımının önemini vurgulamakta ve yalnızca doğruluk metriklerine dayanan çalışmaların geçerliliğini sorgulatmaktadır. İkinci olarak, aynı metrik grubu içinde yer alan değerlendirme kriterleri (örneğin çeşitlilik metrikleri) her zaman tutarlı sonuçlar vermez. Örneğin, “novelty” ve “intra-list distance” gibi metrikler farklı modelleri öne çıkarabilir; bu da her bir metrik türünün kullanıcı deneyiminin farklı bir yönünü yansıttığını göstermektedir. Bu nedenle, tüm bir değerlendirme boyutunu yalnızca tek bir metrik ile temsil etmeye çalışmak yetersiz veya yanıltıcı sonuçlara yol açabilir. Üçüncü olarak, analizlerimiz topolojik özelliklerin model performansı üzerinde ayırt edilebilir etkileri olabileceğini göstermektedir; bazı özellikler belirli model türleri için daha belirleyici görünmektedir. Bu etkiler doğrusal olmak zorunda değildir ve model aileleri arasında farklılık gösterebilir.

Sonuç olarak, bu tez, model değerlendirmelerinin yalnızca basit karşılaştırmalarla yapılmasının yetersiz olduğunu vurgulamaktadır. Doğru, çeşitli ve adil öneriler yalnızca algoritmik tasarıma değil, aynı zamanda kullanılan veri setlerinin yapısal özelliklerine ve seçilen değerlendirme metriklerine de bağlıdır. Önerilen çerçeve ve sunulan bulgular, öneri sistemlerinin daha güvenilir, nesnel ve genellenebilir şekilde değerlendirilmesi için temel bir adım sunmaktadır. Bu yaklaşımın hem akademik araştırmalarda hem de gerçek dünya uygulamalarında daha bilinçli model seçimlerine katkı sağlayacağına inanıyoruz.

Anahtar Kelimeler : Öneri Sistemleri, İşbirlikçi Filtreleme, Öneri Performans Değerlendirmesi, Grafik Topolojisi, Değerlendirme Metrikleri, Adillik, Çeşitlilik, Doğruluk

LIST OF ABBREVIATIONS

AP	Average Popularity
CF	Collaborative Filtering
GINI	Gini Coefficient
GNN	Graph Neural Network
HR	Hit Rate
IC	Item Coverage
IEF	Item Exposure Fairness
ILD	Intra List Distance
MLP	Multilayer Perceptron
MRR	Mean Reciprocal Rank
NDCG	Normalized Discounted Cumulative Gain
Nov	Novelty
PCFM	Personalized Collaborative Filtering Models
Pers	Personalization
RSPE	Recommendation System Performance Evaluation
VAE	Variational Autoencoder

1 INTRODUCTION

1.1 Overview

Recommender systems can be described as advanced computational tools that aid users in discovering relevant interests by examining their preferences and historical interactions with diverse types of content (Guan, 2024). These advanced systems are widely used across various digital platforms, including e-commerce sites (Albalawi and Alharbi, 2023; Ranjan et al., 2023), video streaming services (Dwivedi and Mahanty, 2025), and social networking sites (Fareed et al., 2023; Rao et al., 2024). A recommender system typically serves one of two purposes. First, it either suggests users to other users that might be of interest, such the social networks (Oubalahcen and El Ouadghiri, 2023). Second, it recommends items such as products, movies, or songs mostly in e-commerce or streaming platforms (Sharma et al., 2021). These systems aim to improve the user experience and reduce information overload by recommending items that match the user’s interests and preferences (Arora et al., 2017). An effectively designed and implemented recommender system plays a crucial role in augmenting user engagement by delivering personalized content tailored to the individual user’s behaviour (Rao et al., 2024).

In this thesis, we specifically concentrate on a particular category of recommender systems that is fundamentally grounded in the analysis of user-item interaction data. These systems are usually called *collaborative filtering (CF)* methods (Yurchak and Hryhlych, 2023). More specifically, we focus on *personalized collaborative filtering models (PCFM)*. Instead of relying on item or user attributes, these systems learn from the patterns of each user’s personal preferences across items to generate recommendations. We came across those type of recommendation in almost every digital platforms. For instance, consider a movie platform like Netflix. Suppose User A and User B have both watched and liked the same three movies. If User A later watches another movie, the system may recommend that movie to User B as well. The suggestion is made solely based on their similar interaction histories. In contrast, User C has interacted with different movies, so the system focuses on users whose histories are more similar to User C when generating recommendations. From the perspective of business, the success of

recommender system can be accounted by a certain value such as total purchases or subscriptions. However in the algorithmic perspective, evaluation of the PCFM is an important and open challenge (Castells and Moffat, 2022).

Most evaluation approaches focus on how much of the recommended items intersect with the user’s actual interactions, which is known as *accuracy* (Shani and Guna-wardana, 2011). However, accuracy does not capture all aspects of user satisfaction. Users may also benefit from discovering a variety of different items rather than receiving similar suggestions. This idea is measured through *diversity* metrics (Godinot and Tarissan, 2023). These metrics evaluate how different the recommended items are from one another. Diversity is especially important for e-commerce sites or streaming services. Recommending items from different categories or themes helps maintain the user’s curiosity and engagement (Sharma et al., 2024). It also increases the chance of introducing users to new products they might not have actively searched for. *Fairness* is also an important aspect, especially for platforms that serve multiple sellers or content providers. In real life, it is often appear that popular items dominate the market and reduce the visibility of other products (Klimashevskaja et al., 2024). Due to popular items have more interactions, PCFM see this popular items more than others during the learning phase. This situation may lead models to recommend the same items to many users (Prent and Mansoury, 2024). As a result, models often end up recommending the same popular items to many users. This reinforces the dominance of those items and further reduces the chances of less popular ones being seen. Fairness metrics are designed to capture how well a model balances the visibility of all items (Kecki and Said, 2024). For instance, on a movie streaming platform, a recommendation list with high diversity might include action, comedy, drama, and documentary films, offering the user a wide range of choices. In contrast, a list with low diversity may show only action movies, limiting the user’s exposure to other genres. In terms of fairness, a system with high item fairness might recommend not only the most popular blockbusters but also lesser-known independent films, giving smaller creators a chance to reach viewers. On the other hand, a system with low item fairness would mostly recommend top-grossing movies, making it harder for niche or new titles to appear in the recommendations.

In recent years, PCFM which are based on deep learning have yield more successful results (Metsai et al., 2022). However, these models are often seen as black boxes, making them difficult to interpret and understand (Chen et al., 2023). Although there

are three different metric categories, i.e. accuracy, diversity and fairness, model performances are mostly evaluated with only accuracy in the literature (Li, Liu, Jullien, Ariannezhad, Yates, Aliannejadi and de Rijke, 2024; Bhat et al., 2022). While accuracy remains the primary concern, assessing models by diversity and fairness explains other dimensions of the PCFM (Duricic et al., 2023). For instance, a recommendation model that exhibits low diversity or fairness in its suggestions reflects the underlying biases of the model. Moreover, a recommendation model with high accuracy may achieve this score by primarily recommending dominant popular items, rather than providing truly relevant or diverse content. While this can inflate the accuracy metric, it may not reflect the model’s actual usefulness or quality. Incorporating accuracy, diversity and fairness metrics in the evaluation can help reveal hidden biases and provide a more nuanced understanding of the model’s real-world performance. Therefore, using diversity and fairness metrics is not only important for detecting biases but also for assessing how objective and reliable the reported accuracy truly is.

Although the bias concerns are important in practice, there is still no standard framework for evaluating PCFM (Beattie et al., 2022). Besides, there is no comprehensive study that covers a wide spectrum of aspects and investigates the performance of state-of-the-art models. Each study typically uses different datasets, model types, and evaluation metrics while mostly focusing only on the accuracy. As a result of these limited evaluation approach variety, comparing findings across studies becomes difficult. Some recent research papers attempt to compare PCFM in different aspects, but they usually rely on a limited set of datasets, metrics or models. In addition, they do not use up-to-date deep learning models which have shown the most successful results.

Furthermore, even though accuracy, diversity, and fairness are treated as separate groups of evaluation, it is difficult to claim that each group is homogeneous, as the metrics within each group often focus on different types of information (Goethals et al., 2024). Thus, two metrics from the same group can lead to very different interpretations. It has not yet been investigated whether these metrics produce consistent results or whether they highlight different strengths and weaknesses of the models in practice.

Moreover, real-world user-item interaction systems differ significantly in their topological structures (Ji et al., 2024). The impact of these differences on model performance is still not well understood. In some platforms, a few items have very high number of interactions. Viral videos on social media are a good example. They attract millions of

views, while most other videos receive little attention. Other systems show a more homogeneous distribution. For example, in local library networks, books are borrowed in a more balanced way across different titles. These structural pattern variations shape the accuracy, fairness and diversity of recommendations. For instance, skewed distributions often make it harder for models to promote less popular items. Although some studies have examined the role of topological features on model performance, they often rely on linear models and fail to capture more complex, potentially non-linear relationships. As a result, it remains unclear which topological features are most influential and how they interact with different PCFM.

In this thesis, we address these gaps through a detailed and experimental analysis. Although our findings may have implications for real-world systems, our primary objective is to provide a comprehensive evaluation of recommender systems under three perspectives : *(i)* Model performance evaluation per different types of metrics, *(ii)* The coherence and conflict between evaluation metrics, *(iii)* The effect of topological features on the recommender systems.

1.2 Problem Statement

Although there are studies in the literature targeting some of the objectives of this thesis, these studies have limited research scope and experimental setup. For our first research perspective, there are limited studies which investigates the performance differences of PCFM on different datasets. For instance, Anelli et al. (Anelli et al., 2023) showed that, on different datasets, existing PCFM can perform worse than the models they were originally reported to outperform. Although their study is one of the most recent studies which includes deep learning models, they have some critical shortcoming. The primary shortcoming is the low experimental scale. They focused only on graph-based PCFM as deep learning models, worked with a limited set of datasets and evaluated models only with 2 accuracy metrics. The second shortcoming of their study is the evaluation method itself. Although they state that these performance differences may be due to differences in dataset features, a statistical approach is not established here either. Third, they did not finalize their findings by giving an overall model rankings.

For our second perspective, there are also limited studies investigating the coherency

between accuracy, diversity and fairness. As example, Zhao et al. (Zhao et al., 2025) investigated the relationships between fairness and diversity while Isufi et al. (Isufi et al., 2021) focusing on the accuracy and diversity. Their results suggest for a positive relationship between fairness and diversity and a negative relationship between diversity and accuracy respectively. However, both studies are open to discussion due to the limited number of data sets and model diversity.

For our third research perspective, several studies have examined the relationship between dataset structure and recommendation performance. Most of these studies focused only on basic accuracy metrics such as NDCG@k or Recall@k. They usually evaluated a small number of models and datasets, and mainly relied on classical methods like KNN or matrix factorization instead of deep learning-based models. Some studies, like (Adomavicius and Zhang, 2012), analyzed the effect of data characteristics using simple regression models. However, they mainly considered accuracy and only a few fairness metrics. Although some recent studies, such as (Dietz et al., 2025), started to include diversity and fairness metrics, there is still no study that systematically covers all three main evaluation dimensions : accuracy, diversity, and fairness.

A common limitation in previous work is the small experimental scale. Many studies tested only a few datasets and used a narrow set of evaluation metrics. They often focused on a limited number of topological features, e.g. sparsity etc. However, possible important topological features like degree distributions or high order connectivities have been overlooked.

In this thesis, we design a large-scale experimental setup to fill these gaps. We combine a wide variety of topological features, datasets with different characteristics, and PCFM from different architectural families. The challenge is the high number of experiment variables : *datasets*, *topological features*, *performance evaluation metrics* and *personalized recommendation models* in the experiments. These variables can each be divided into several subcategories as well. Because of this complexity, trivial analysis is not feasible. That’s why we address this challenge by constructing a non-trivial analysis methodology by proposing a specified experimental design. We support our findings with statistical tests as well. Our main focus is to investigate the dependencies between mentioned experiment variables. We study whether dataset features affect model performance, whether model results are biased toward certain evaluation metrics, and whether there is a strong coherence among different evaluation metrics. Ultimately, the

objective of this work is to provide useful insights for researchers and practitioners who want to design more reliable and balanced recommender systems.

To provide a clearer structure for our investigation, this thesis is built around a set of key research questions, each addressing a distinct aspect of the evaluation problem. These questions not only guide the experimental design but also help to frame our analyses within a coherent and comprehensive framework. Throughout this thesis we will consider and answer the research questions listed below :

- **RQ1** : How do the recommendation models' performance differ according to evaluation criteria ?
- **RQ2** : Are the evaluation metrics coherent with each other ?
- **RQ3** : Are performance metrics sensitive to the used recommendation model ?
- **RQ4** : Can the recommendation model performance explained by topological characteristics ?

1.3 Contributions

In this thesis, we present a detailed and large-scale experimental setup to better understand how PCFM perform under different conditions. The major contribution of this work is proposing a systematic experimental design shaped around four different research questions. Each of these research questions focuses on different type of relationships between metrics, PCFM and topological features. Our other contributions are listed as follows :

- We propose a novel taxonomy for PCFM. Instead of the traditional separation between personalized and non-personalized models or between content-based and collaborative filtering, we classify models based on their methodological strategies. This taxonomy focuses only on models that do not use any content information. We split the models into the traditional and deep learning based ones. Deep-learning based models also split into their data modeling approaches such as graph-based models, auto-encoders and simply linear models. We have also sub-categorised graph-based models according to their learning methods (such as constractive learning or augmentation). Our experimental results support the taxonomy strategy we propose.
- Unlike many recent studies that test their proposed recommendation methods

only in favorable environments using limited metrics, we compare a wide range of models using a large and diverse set of datasets and evaluation metrics. With the large set of experiments we have created, we have demonstrated the position of popular PCFM on many data sets and evaluation criteria. We used 13 PCFM, 12 evaluation metrics and 10 diverse datasets. This allows us to provide a more comprehensive and fair comparison of models.

- We present a systematic analysis of the coherency and interdependencies between Recommendation System Performance Evaluation (RSPE) metrics. Evaluating recommendation systems involves considering multiple factors, including accuracy, diversity, and fairness. In this contribution, we move beyond treating these as separate evaluation dimensions and instead explore how they relate to one another.
- We further investigate how model performance varies across different datasets and RSPE metrics. Evaluating recommendation systems involves not only comparing metrics but also understanding how model characteristics influence these metrics. In this contribution, we move beyond reporting metric scores and instead explore whether the observed behaviors of these metrics are consistent or model-dependent.
- We expand the analysis of graph topological features beyond what previous studies have done. We use a much larger set of features, including degree distributions and other important network properties. We also aim to capture potential non-linear relationships between graph features, model architectures, and evaluation outcomes, which were overlooked in earlier works.

2 LITERATURE REVIEW

This thesis investigates the quantitative and qualitative performance of recommendation systems. We delve into the relationship between recommendation performance and graph topology through a series of experiments. Given the multifaceted nature of recommendation systems, our literature review is structured around three main themes.

In this chapter, we first provide an overview of traditional and the most common recommendation approaches followed by the current challenges of recommendation systems. Second, since we focus on the effect of graph topology, we detailed the Graph Neural Network (GNN) based algorithms. Finally, we conduct a comparative analysis of studies examining the influence of graph structures on recommendation performance.

2.1 Recommendation Systems

Recommender systems are algorithms designed to suggest relevant items to users by analyzing their preferences, behaviors, or interactions with similar users or items (Raza et al., 2024). These systems have evolved into a diverse field, offering various approaches to address the growing complexity and demands of modern applications. These systems are generally categorized into three main types : content-based systems, which rely on item attributes and user preferences (Nath and Ahmad, 2022) ; collaborative filtering systems, which leverage user-user or item-item similarities solely based on interaction data (Papadakis et al., 2022) ; and hybrid systems, which combine multiple methods to overcome the limitations of individual approaches (Seth and Sharaff, 2022). Each category is tailored to leverage specific data types, user behaviors, and application contexts, reflecting the unique challenges and goals of the domain in which they are applied. By understanding the foundational principles and methodologies of these categories, we can better appreciate their strengths, limitations, and suitability for particular tasks. This section provides an overview of these primary categories of recommender systems, highlighting their core characteristics and the contexts in which they excel.

2.1.1 Content-Based Filtering (CBF)

Content-based filtering (CBF) systems recommend items by analyzing their inherent features or attributes and matching them to user preferences. These features can include textual descriptions, metadata, user profiles, or contextual information, such as product specifications, keywords, or tags (Singhal et al., 2017). The core idea behind CBF is to identify patterns in a user’s historical interactions, such as items they have rated highly, purchased, or viewed, and recommend new items that share similar characteristics. For example, a content-based system might suggest books with similar genres, authors, or topics based on a user’s reading history.

One of the primary advantages of CBF is its ability to handle personalized recommendations effectively, as it focuses on individual user preferences without requiring data from other users (Anand and Nath, 2020). This property makes CBF systems especially effective in scenarios where user behavior is highly specific or when cross-user similarities are sparse (Gunathilaka et al., 2025). Additionally, content-based approaches are well-suited to address the cold-start problem for items (Chia and Najafabadi, 2022). For instance, when a new product is added to the system, its attributes can be used immediately to generate recommendations, even in the absence of user interaction data.

A significant limitation of content-based filtering systems is their reliance on the availability and quality of feature data. Poorly defined or sparse item attributes can hinder the system’s ability to make accurate recommendations. For instance, when using Vector Space Models (VSM) with techniques like TF-IDF and BM25, insufficient or irrelevant item metadata can result in poorly weighted feature representations, reducing the effectiveness of recommendations (Thorat et al., 2023). Additionally, these systems often suffer from a lack of diversity in recommendations, as they tend to suggest items too similar to those the user already knows, leading to a phenomenon known as the serendipity problem. Despite such advancements, the narrow focus of traditional content-based approaches continues to limit their ability to surprise users or expose them to novel content, underlining the need for hybrid solutions that combine the strengths of different methodologies.

2.1.2 Collaborative Filtering (CF)

Collaborative filtering (CF) systems rely on the idea that users who have shown similar preferences in the past are likely to have similar preferences in the future (Bux et al., 2022). These systems use user-item interaction data such as ratings, purchases or clicks to identify patterns for recommendation generation without needing explicit item attributes. The two main categories of CF systems : (i) user-based collaborative filtering which groups users according to their interaction patterns (Jia et al., 2015), (ii) item-based collaborative filtering which identifies items that users with similar preferences tend to interact with together (Sarwar et al., 2001). For example, in a movie recommendation system, user-based systems look at similar users. If Alice and Bob both like action movies and Bob watched “Mad Max,” the system may recommend “Mad Max” to Alice. In item-based systems, the recommendation is based on similar items. If Alice liked “Inception,” the system may recommend “Interstellar” because many users who liked “Inception” also liked “Interstellar.”

On one hand, the main benefit of CF lies in its ability to produce recommendations directly from interaction data, without needing additional user or item information (Vhatkar, 2024). This makes CF a flexible approach that can work across many different domains. On the other hand, the user-item interaction matrix in CF systems often has a major challenge : most users interact with only a small portion of the available items, leading to very sparse data. This sparsity makes it difficult for the CF systems to discover meaningful patterns or similarities between users and items (Althabiti et al., 2021). As a result, CF systems often struggle to make good recommendations for inactive or newly arriving users, a challenge known in the research community as the *cold-start problem* (Nguyen et al., 2024).

Researchers have proposed various methods to address these challenges. The traditional CF systems uses Singular Value Decomposition (SVD) as a matrix factorization technique to lower interaction matrix dimensions so it can discover hidden user and item preference factors. Memory-based approaches, like neighborhood-based methods, use similarity measures such as cosine similarity or Pearson correlation to identify relationships between users or items. UserKNN (Resnick et al., 1994) finds similar users based on their interaction histories and recommends items preferred by those users. ItemKNN (Deshpande and Karypis, 2004) identifies similar items based on user interactions and suggests items similar to those a user has engaged with.

More recent approaches rely on neural network-based PCFM due to their ability to capture complex relationships between users and items. Deep Matrix Factorization (Xue et al., 2017) enhances traditional matrix factorization by using deep learning layers to uncover complex, high-order patterns in user-item interactions. SimpleX (Mao et al., 2021) uses cosine contrastive loss (CCL) to increase the similarity of positive interactions while separating them from negative ones.

Variational Autoencoders (VAEs) learn a latent representation for each user and reconstruct their interaction history from it (Kingma et al., 2019). This makes VAEs generative models that can produce new recommendations by sampling from the learned distribution. MultiVAE (Liang et al., 2018) adapts this approach for collaborative filtering by compressing user interactions into a probabilistic latent space and reconstructing them using a multinomial likelihood, which better captures implicit feedback patterns. RecVAE (Shenbin et al., 2020) improves MultiVAE and provides a more powerful learning mechanism. By improving the model’s regularization methods, it applies different levels of regularization based on user interaction intensity and improves recommendation quality. In addition, it provides a more efficient learning process by alternately updating the encoder and decoder networks. DiffRec (Wang et al., 2023) adapts diffusion models to recommendation by treating interaction prediction as a denoising process. Unlike previous generative methods (GANs, VAEs), it gradually adds noise to user interaction histories and learns to recover them step by step.

Despite these advancements, collaborative filtering systems are often biased toward popular items, as they rely heavily on historical interaction data (Shi et al., 2023). This popularity bias can limit the diversity of recommendations and reduce the exposure of less frequently interacted items (Carnovalini et al., 2025). Furthermore, CF systems may struggle to adapt to dynamic changes in user preferences or item availability (Gasmi et al., 2015).

2.1.3 Hybrid Models

Hybrid recommendation systems combine multiple recommendation techniques, such as content-based filtering (CBF) and collaborative filtering (CF), to overcome the limitations of individual methods while leveraging their strengths (Çano and Morisio, 2017). These systems aim to address challenges like sparsity, cold-start issues, and lack of diversity by integrating features or outputs from different approaches. Hybrid

systems are typically categorized into weighted, mixed, and switching strategies, each offering unique advantages in integrating diverse data and methods (Morales Murillo et al., 2022).

2.1.4 Challenges of Recommendation Systems

Developing effective and robust recommendation systems comes with many challenges. These challenges arise from the inherent complexity of user behavior, the large scale of modern datasets, and the evolving nature of user preferences and item popularity. In this section, we will examine some of the major challenges facing recommendation systems today, explore their underlying causes, and discuss potential solutions.

Diversity and Fairness

Recommender systems often struggle to balance relevance with diversity. While personalized recommendations aim to provide users with items they are likely to interact with, they can inadvertently limit user exposure to a narrow range of content. When users are primarily exposed to items similar to those they have previously interacted with, it restricts their exploration of potentially interesting or novel items. Diversity in recommendations aims to expose users to a wider range of items, even if these items are not strictly within their established preferences (Zhang et al., 2024). This can lead to increased user satisfaction by helping users discover new interests and broaden their horizons. As shown in Fig. 2.1, diversity can be considered at two levels (Gao et al., 2023) : individual and system-level diversity. Individual-level diversity focuses on the recommendations received by a single user (Yang et al., 2023). It aims to maximize the variety of items the user is exposed to in a single recommendation list. System-level diversity ensures different users receive different recommendations, promoting overall catalog exploration (Antikacioglu et al., 2019). This exploration is particularly important for *long-tail* items. Long-tail items are those that have relatively low popularity compared to the most popular items in a catalog.

Fairness, on the other hand, focuses on ensuring that users and items are treated equally. In the context of user fairness, the system should not discriminate against certain user groups based on like demographics (Greenwood et al., 2024). For item fairness, the system should not exhibit bias towards items with a high number of interactions, potentially overshadowing less popular but equally relevant items (Li, Ren, Sanderson

and Deng, 2024). This bias can arise due to the *class imbalance* problem, where some items naturally have significantly more interactions than others. Favoring highly interacted items can create a self-reinforcing loop within the recommender system. This loop further amplifies the popularity of those items, while potentially valuable items with fewer interactions are neglected. This can disadvantage both users who might be interested in these less popular items and the creators of those items. Therefore, a system can be diverse but unfair, or fair but not very diverse. The goal is to achieve both, offering users a wide range of relevant and high-quality items in a way that is equitable and unbiased, mitigating the impact of interaction imbalances.

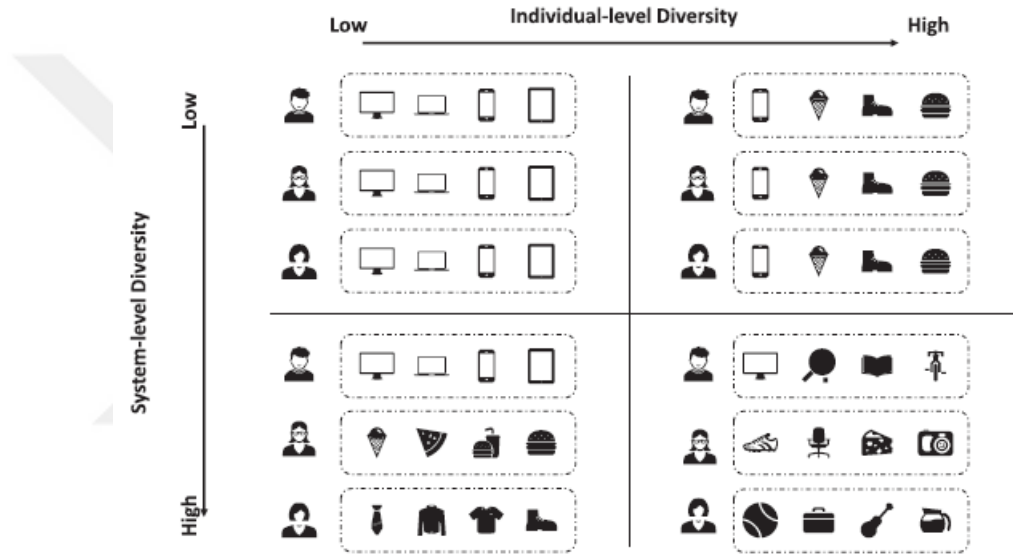


FIGURE 2.1 – Illustration of individual-level diversity and system-level diversity (Gao et al., 2023)

Data Sparsity

User-item interaction data serves as the foundation for recommender systems to understand user preferences. The typical user behavior involves selecting only a limited number of available items which creates a data sparsity problem. The system struggles to accurately model user preferences because there is insufficient information to determine relevant recommendations due to this sparsity (Jain et al., 2020). Users with limited interaction histories experience poor recommendation quality because there is insufficient data. The insufficient amount of data leads to increased chances of useless recommendations which results in user dissatisfaction and reduces the system’s overall performance. Multiple methods exist to reduce data sparsity problems. The problem requires multiple solutions which include identifying hidden similarity patterns between users and items through synthetic connection establishment (Hu et al., 2017) and

using social links (Xiao and Chen, 2024) in addition to using user or items’ auxiliary information.

Cold Start

The cold start problem occurs when recommendation systems need to handle new users or items which have minimal or no interaction data available. Traditional collaborative filtering methods fail when making recommendations because they depend on existing interaction data. New users who do not have prior engagement will receive useless recommendations and new items will not get discovered (Panteli and Boutsinas, 2023a). This challenge manifests itself in several key scenarios. The system faces an issue when trying to suggest items to users who have no interaction history in the system. The system needs to suggest new items to users who have never interacted with the system before. The most demanding situation occurs when the system needs to suggest completely new items to completely new users since there is no available data for either group. The cold start scenarios reduce user engagement and block new item discovery which leads to negative impacts on the recommendation system’s overall performance. Researchers have developed multiple approaches to overcome these challenges by using side information such as user demographics and item attributes (Cao et al., 2023) and meta-learning techniques (Luo et al., 2024) and transfer learning from related domains (Hao et al., 2021).

Scalability

Modern recommender systems process enormous datasets containing millions of users and items which produce extensive interaction data (Xia et al., 2024). The increasing amount of data creates a substantial bottleneck in the process of creating personalized recommendations. A recommender system demonstrates scalability through its capacity to manage expanding data volumes and user traffic without compromising its operational efficiency (Fayyaz et al., 2020). The ability of a scalable system allows it to process extensive datasets swiftly so it can provide immediate recommendations to users. Several strategies are employed to improve scalability. Dimensionality reduction serves as a standard method to simplify complex data structures while maintaining critical information content (Sarwar et al., 2000). User segmentation techniques segment users into groups which both minimizes data sparsity and decreases the amount of data that needs processing (Zhu and Liu, 2023). Singular Value Decomposition (SVD) serves as a powerful method to decompose user-item interaction matrices into lower-

dimensional spaces which facilitates efficient recommendation calculations (Osmanlı, 2010). Incremental and online learning methods are also common methods to handle data scalability (Papagelis et al., 2005). The model can update its parameters through incremental learning by incorporating new data points without requiring a complete retraining process which saves substantial computational power. Online learning builds upon incremental learning by streaming data for processing while the model updates its parameters with each new interaction.

2.2 Graph Neural Networks

Graph Neural Networks (GNNs) have emerged as a transformative paradigm for learning on graph-structured data, which inherently represents relationships and interactions in various domains (Scarselli et al., 2008). Unlike traditional machine learning methods that operate on tabular or sequential data, GNNs leverage the topological structure of graphs, enabling them to effectively model both the attributes of nodes (entities) and the relationships (edges) between them (Zhou et al., 2020).

A graph $G = (V, E)$ consists of a set of nodes V and edges E , where nodes often represent entities, and edges denote interactions or relationships between these entities. Depending on the application, graph types may vary. The flexibility to model such diverse structures has made GNNs widely applicable across various fields, including social networks, knowledge graphs, biological networks, and recommendation systems (Khemani et al., 2024).

The core of GNNs lies in message passing. In this process, each node aggregates information from its neighbors and updates its representation iteratively (Wu et al., 2020). Two fundamental operations enable this : aggregation and update. Aggregation collects information from a node’s neighbors, while update combines this aggregated information with the node’s existing representation to generate a new, enriched representation. Formally, the generic message-passing mechanism in GNNs can be expressed as :

$$h_v^{(l+1)} = \text{UPDATE} \left(h_v^{(l)}, \text{AGGREGATE} \left(\{h_u^{(l)} : u \in \mathcal{N}(v)\} \right) \right), \quad (2.1)$$

here $h_v^{(l)}$ represents the feature vector of node v at layer l , $\mathcal{N}(v)$ denotes the neighbors of node v , AGGREGATE is a function to combine neighbor features (e.g., sum, mean,

or attention) and UPDATE is a function to update the node’s feature vector.

There are different implementations of GNNs, the one of the most popular adaption of GNNs is Graph Convolutional Networks (GCN) (Kipf and Welling, 2016). The GCNs utilize a convolutional mechanism to iteratively combine and transform node features, while Graph Attention Networks (GATs) enhance this process by assigning attention weights to neighbors based on their relevance (Veličković et al., 2017). The generic GNN framework provides flexibility in defining AGGREGATE and UPDATE functions, while the GCN uses a specific normalized adjacency matrix and a linear transformation with learnable weights. This normalization ensures proper scaling of node features during aggregation, preventing numerical instability. The GCN message-passing equation is :

$$H^{(l+1)} = \sigma \left(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right), \quad (2.2)$$

here $H^{(l)}$ is the feature matrix at the l -th layer, with $H^{(0)} = X$ (the initial node features), $\hat{A} = A + I$ is the adjacency matrix of the graph with added self-loops, $\hat{D}$ is the degree matrix of $\hat{A}$, $W^{(l)}$ is the learnable weight matrix at the l -th layer, σ is an activation function, such as ReLU.

The ability to capture local and global graph structures through multihop neighborhood aggregation has positioned GNNs as a powerful tool for tasks such as node classification, link prediction, and graph classification. Furthermore, GNNs are highly flexible, allowing the incorporation of node and edge attributes, heterogeneous graph structures, and external knowledge, thus making them adaptable to a wide range of applications.

2.2.1 Graph Construction Types for GNNs

Graphs are fundamental data structures that represent relationships between entities (Zadgaonkar and Agrawal, 2020). These relationships are important for GNN-based models, as their relationships allow information to propagate and be learned across networked nodes. There are different approaches for building graphs depending on the problem domain. The relationships can differ based on factors such as temporal nature, directionality, etc (Thomas et al., 2022). Thus, the type of graph used also is influenced significantly.

In *directed graphs*, edges have a specific direction, indicating one-way relationships - like in citation networks where one paper cites another (Howell et al., 2024). Undirected graphs contain edges with no direction, representing mutual relationships like friendship networks (Pfaltz, 2018). Similarly, edges can carry different weights to represent the strength or importance of connections (Canale, 2022). In *weighted graphs*, numerical values might indicate distances between the nodes such as transportation networks or interaction frequencies in social networks. In contrast, unweighted graphs focus solely on the presence or absence of relationships by treating all connections equally. Graphs can also be characterized by their temporal nature. Static graphs represent relationships at a fixed point in time (Hood et al., 2020), while *dynamic graphs* evolve to capture changing relationships over time (Michail, 2015). In *hypergraphs*, an edge can connect more than two vertices. This allows hypergraphs to represent higher-order relationships that are difficult to encode within a traditional graph structure (Cameron et al., 2023). As a result of this structure, hypergraphs offer a suitable framework for representing ideas like group interactions, multi-way affiliation, and shared resources. This flexibility in representing relationships makes graphs highly adaptable to diverse applications. This section will explore a subset of graph types considered crucial within the context of this thesis. We categorize these graph types into two primary groups : Homogeneous Graphs and Heterogeneous Graphs which are illustrated in Figure 2.2.

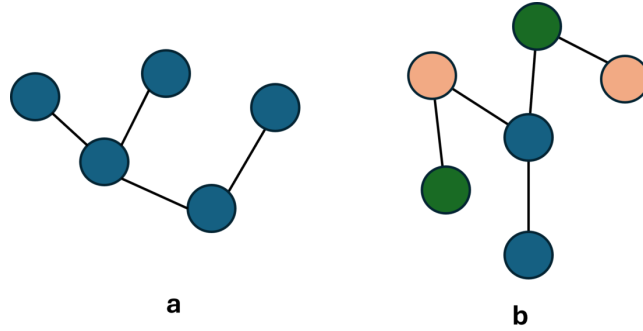


FIGURE 2.2 – An Illustration Of Main Graph Types. Different Colored Nodes Correspond To The Different Node Type. Figure a Represents The Homogenous Graphs And Figure b Represents The Heterogenous Graphs

Homogeneous Graphs

Homogeneous Graph consists of only one kind of node type and edge type (Xiong et al., 2023). These graphs are commonly used to model situations where the entities and relations are the same, e.g. social networks where all nodes correspond to people, and links refer to friendships. Homogeneous graphs are especially appealing when simplicity

and uniformity is desired, because they avoid the added complexity trying to work across multiple node or edge types. Such uniformity also endows all nodes with a common feature space, and simplified information propagation in GNN models.

Heterogeneous Graphs

Heterogeneous Graphs contain multiple types of nodes and edges, allowing them to capture the complexity of systems with diverse entities and relationships (Bing et al., 2023). These graphs are particularly useful when different types of data need to be integrated into a unified framework. This diversity provides a richer representation of the underlying system, enabling more nuanced analysis and predictions. The ability to model multiple types of interactions, such as user-purchases-item or item-associated-with-tag, makes heterogeneous graphs especially suitable for applications like multi-modal recommendation systems.

A common subclass of heterogeneous graphs is bipartite networks, where nodes are divided into two distinct sets. Edges exist only between nodes in different sets. For example, in a user-item graph, one set represents users and the other items, with edges denoting interactions such as purchases or ratings. Bipartite graphs are particularly suited for recommendation systems, as they inherently model relationships between distinct entity types. Extending this concept, tripartite graphs involve three distinct node sets, such as users, items, and tags, with edges capturing interactions across these sets. These structures allow for even richer modeling of complex systems, making heterogeneous graphs a powerful choice for tasks requiring detailed and varied relational data.

Bipartite and Tripartite Graphs are specialized heterogeneous graphs where nodes are partitioned into distinct sets. Figure 2.3 presents example networks for bipartite (Fig. a) and Tripartite (Fig. b) graphs. In bipartite graphs, nodes are divided into two sets with edges connecting only nodes from different sets (Wang et al., 2024). A classic example is a user-item graph in recommendation systems, where edges represent interactions between users and items. Tripartite graphs extend this to three distinct node sets - for instance, connecting users, items, and tags through their relationships.

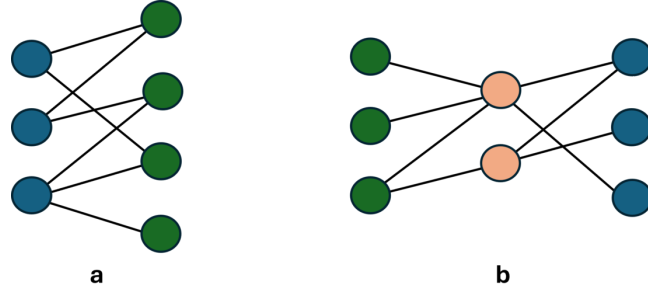


FIGURE 2.3 – An Illustration of Bipartite (Figure a) and Tripartite (Figure b) Graphs. Each Color Represents Different Node Type.

2.2.2 Recommendation Systems with GNNs

This section explores GNN applications across different recommendation contexts : social recommendation, sequential recommendation, bundle recommendation, multi-behavior recommendation and user-item collaborative filtering.

Social Recommendation systems leverage both social relationships and user-item interactions to enhance recommendation accuracy (Oubalahcen and El Ouadghiri, 2023). As illustrated in Figure 2.4 (a), these systems model the influence of social connections, such as friendships or shared interests, alongside users' individual preferences. By integrating social network data, they can capture trust and influence dynamics, enabling more personalized and context-aware suggestions. In addition to recommending items to users, these systems can also recommend users to each other. This feature is particularly valuable in social or professional platforms, where fostering relationships, collaborations, or networks is a primary goal. *Sequential Recommendation* systems are designed to predict a user's next interaction based on their previous sequential behaviors (Betello et al., 2024) as illustrated in Figure 2.4 (b). These systems analyze ordered data, such as clickstreams or purchase histories, to capture temporal patterns. Unlike the previous approaches, sequential systems emphasize the importance of interaction order. Constructing directed networks is a common practice in these type of systems to capture the order importance by linking successively interacted items (Qin et al., 2023). *Bundle Recommendation* focuses on suggesting collections of items that complement each other, rather than recommending individual items (Sun et al., 2024) illustrated in Figure 2.4 (c). This approach is particularly useful in scenarios where items are commonly purchased or consumed together. By recommending bundles, the system aims to enhance user satisfaction by addressing multiple needs in a single re-

commendation. *Multi-behavior Recommendation* systems aim to improve recommendations by leveraging multiple types of user-item interactions, such as clicks, purchases, reviews, and likes (Kim et al., 2025) as illustrated in Figure 2.4 (d). Instead of relying on a single behavior type, these systems integrate diverse signals to capture a more comprehensive understanding of user preferences.

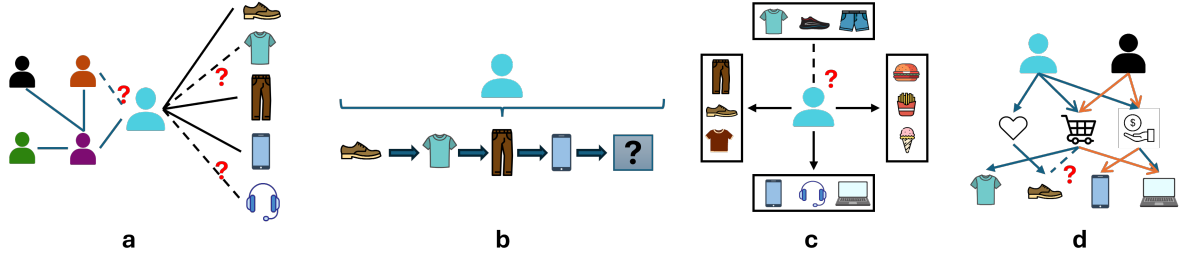


FIGURE 2.4 – An Illustration of some common recommendation tasks. Figure a represents the social recommendation which includes both user-user and user-item edges. Figure b represents the sequential recommendation where the order of the interactions also considered. Figure c represents the bundle recommendation where a set of items are recommended at once and Figure d represents the multi-behavior recommendation where there are multiple user-item edge types

User-item Collaborative Filtering with GNNs is one of the most widely studied scenarios in recommendation systems (Wu et al., 2022). In this type of recommendations, the goal is to predict user preferences for items based on observed user-item interactions, as illustrated in Figure 2.5. In GNN-based approaches, constructing bipartite graphs is the most common approach where one side represents the set of users and the other side represents the items (Giamphy et al., 2023). In this thesis, we concentrated on the user-item collaborative filtering by constructing the bipartite graphs that's why we explain this part in more detail than previous approaches.

NGCF (Wang et al., 2019) introduced embedding propagation layers that aggregate information from neighbors to capture collaborative signals in the embeddings. Multi-GCCF (Sun et al., 2019) explicitly modeled multiple graphs. Besides the user-item

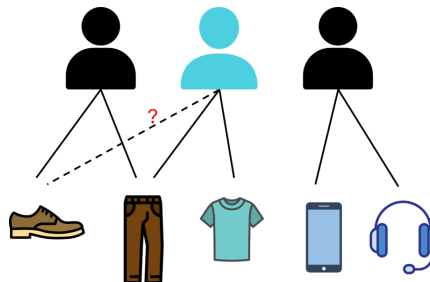


FIGURE 2.5 – An Illustration of User-Item Recommendation

bipartite graph, it constructed separate user-user and item-item graphs to capture proximity information. This helped address data sparsity and improved recommendation quality by incorporating these additional relationship signals. DGCF (Wang et al., 2020) creates multiple versions of the user-item interaction graph, where each version assigns different weights to the same interactions based on discovered patterns. These weighted graphs are used simultaneously during training to learn how different interaction patterns contribute to user preferences. The model uses this multi-graph approach to learn better user-item representations. This approach improves both recommendation accuracy and the ability to explain recommendations by identifying which interaction patterns were most influential. LightGCN (He et al., 2020) simplifies NGCF by focusing solely on neighborhood aggregation, removing nonlinear transformations and feature projections for greater efficiency and interpretability. MixGCF (Huang et al., 2021), while not a standalone model, introduced an effective training method for GNN-based recommenders. MixGCF synthesizes hard negative samples by combining embeddings from different neighborhood hops of raw negatives to learn more discriminative features. SGL (Self-supervised Graph Learning) (Wu et al., 2021) proposed using graph data augmentation and contrastive learning to improve GNN-based recommendation. It created different graph views through node/edge dropout and maximized agreement between views of the same node. This helped capture intrinsic patterns in the graph structure and improve robustness against noise. NCL (Neighborhood Contrastive Learning) (Lin et al., 2022) enhanced contrastive learning by explicitly incorporating potential neighbors into contrastive pairs. It introduced structural neighbors from graph topology and semantic neighbors from representation space. SimGCL (Yu et al., 2022) simplified previous contrastive approaches by replacing graph augmentations with noise-based embedding perturbations, achieving similar or better results with reduced complexity. XSimGCL (Yu et al., 2023) further simplified SimGCL by unifying recommendation and contrastive tasks into a single forward pass with cross-layer contrasting, maintaining performance while significantly reducing computational costs.

We employed several of these algorithms in our experiments and provided a detailed explanation of them in Chapter 4.

2.3 Graph Topology and Recommendation Systems

Graph-based PCFM have been explored from various perspectives to enhance their effectiveness. While LightGCN advocates for simplifying graph-based models to improve efficiency, approaches like MixGCF and SGL highlight the necessity of augmentation techniques, and NCL demonstrates the benefits of contrastive learning. These studies provide different perspectives on how to improve recommendation performance. Research has also focused on designing optimal graph topologies by experimenting with different architectures and aggregation functions (Wei et al., 2022; Wang, Zhao and Shi, 2022). Other studies have examined the impact of message-passing functions on recommendation diversity (Anelli et al., 2022) or addressed fundamental GNN challenges such as over-smoothing (Rusch et al., 2023).

As Anelli et al. (Anelli et al., 2023) pointed out, model rankings can vary when tested on datasets different from those used in the original studies. However, their analysis of the impact of graph topology on these variations is limited, as they only consider basic structural properties such as the total number of users, items, and edges, network density, and the average degree of users and items. These features are insufficient to fully capture the characteristics of the graph structure. Moreover, no statistical analysis has been conducted to assess the significance of the observed performance variations. Instead, the findings are primarily explained based on average node degrees, which may not provide a comprehensive understanding of how dataset properties influence model rankings. The underlying reasons for performance shifts across datasets remain unexplored and require further investigation. As listed in Table 2.1, previous studies have also examined the relationship between graph topology and model performance, not only in graph-based models but also in non-graph-based models. In addition, some studies have investigated how graph topology affects GNN models in tasks such as node classification and link prediction. However, many of these studies rely on a limited set of structural features, evaluation metrics, models, or datasets, leaving key aspects of the topology-performance relationship underexplored.

Adomavicius et al. (Adomavicius and Zhang, 2012) analyzed the impact of data characteristics on the performance of recommendation methods. They used only two KNN-based models and one matrix factorization model. They did not experience any modern deep learning approaches. They identified six key features to describe the graph : density, user-item ratio, interaction distribution skewness, total space size ($U * I$), Gini- i

coefficient, and Gini- u coefficient. They retained these features and performed regression analysis with the RMSE as the dependent variable. The results showed that some data characteristics significantly influence recommendation performance. Total space size and density had a positive effect on all three models, while the user/item ratio negatively impacted KNN-based models but positively affected the matrix factorization model. However, it has some limitations due to the small quantity of data features and performance metrics, which makes it less generalizable.

Deldjoo et al. (Deldjoo et al., 2021) conducted an experimental study analyzing the performance of traditional PCFM across three datasets. Compared to Adomavicius et al., they used a larger set of models, but still relied only on KNN-based and matrix factorization models. Their analysis included 19 topological metrics in a regression analysis. While the number of features increased, many were similar to those in Adomavicius’ study. The feature set was expanded by calculating mean and standard deviation values for several metrics. Additionally, the long-tail item ratio was introduced as a new feature, where the top 20% of items by degree distribution were classified as long-tail. They also employed regression analysis in their study with one recommendation accuracy metric (NDCG) and one fairness metric (MAD (Deldjoo et al., 2019)) as dependent variables. Their findings revealed a strong relationship between dataset characteristics and accuracy performance, with data characteristics explaining up to 80% of accuracy variations. However, they were unable to establish similar strong connections for fairness. The study has several limitations : it only evaluates traditional PCFM, lacks a diverse set of datasets, and relies on single metrics for both accuracy (NDCG) and fairness (MAD), which may not capture the full spectrum of model performance and fairness aspects. While the influence of individual features varied across datasets, user/item ratio and interactions per user were found to be the most significant factors affecting model performance.

Malitesta et al. (Malitesta et al., 2023) distinguished their work from previous work by focusing specifically on the impact of dataset topology on graph-based collaborative filtering (Graph CF) models. They evaluated four Graph CF models using two real-world datasets. Unlike earlier studies, they included assortativity and clustering coefficient values from user and item projected graphs, expanding the feature set to 11 topological properties. Their regression analysis showed that average user and item degree, Gini- i , and density significantly influenced model performance. However, the study has notable limitations. It only considers accuracy as the evaluation metric, uses

a small number of datasets, and does not analyze how these topological features affect other types of PCFM.

Dietz et al. (Dietz et al., 2025) analyzed the impact of topological features on user-location recommendation systems using regression analysis. Unlike most of the previous studies, they included not only accuracy but also fairness and diversity metrics, such as novelty and item exposure. Using a single dataset, they extracted 32 characteristic features. Their topological features contain scenario-specific metrics like duration of activity, user mobility, or demographic and contextual information. Consistent with earlier findings, density and Gini coefficients were identified as key factors influencing model performance. They did not evaluate recent deep learning approaches. They used 2 KNN based approach, 5 different variations of matrix factorization and 1 hybrid model.

The impact of basic topological features on specialized GNN-based models for different tasks has also been studied. Tang et al. (Tang et al., 2020) conducted a preliminary analysis on whether node classification accuracy is related to node degree. Their study was limited in scope, using only one model and three datasets, but their findings suggested a correlation between higher node degree and better classification performance. Wang et al. (Wang and Derr, 2022) focused on the link prediction task. They grouped users based on their degree and measured the average prediction accuracy within each group. Their experiments included four GCN-based models and one matrix factorization model. The results showed a positive correlation between link prediction accuracy and user degree. Fang et al. (Fang et al., 2023) analyzed node classification performance using GCN-based models. Instead of real-world datasets, they used three artificially generated networks and tested them with four GCN models. They examined five topological features (average degree, power law exponent, homophily, average clustering coefficient, and average path length). Their findings consistently highlighted average degree as the most influential factor in node classification performance.

TABLE 2.1 – Comparison of studies examining the impact of topological properties on model performance for various tasks

Reference	#Datasets	Acc. Metrics	Fair. Metrics	Div. Metrics	Methods	#Features	Objective Task
(Adomavicius and Zhang, 2012)	4	RMSE	-	-	UserKNN, ItemKNN, MF	6	Recommendation
(Tang et al., 2020)	4	Accuracy	-	-	GCN, DSGNN, SL-DSGNN	1	Node Classification
(Deldjoo et al., 2021)	4	NDCG, MAP	MAD	-	UserKNN, ItemKNN, BPR, SVD, MF, PMF, NMF	19	Recommendation
(Wang and Derr, 2022)	1	NDCG, Recall, F1	-	-	MF, NDGCF, LightGCN, CAGCN	1	Link Prediction
(Malitesta et al., 2023)	3	NDCG, Recall	-	-	LightGCN, SVD-GCN, UltraGCN, DGCF	11	Recommendation
(Anelli et al., 2023)	5	NDCG, Recall	-	-	NGCF, DGCF, LightGCN, SGL, UltraGCN, GFCF	6	Recommendation
(Fang et al., 2023)	3	F1	-	-	GCN, H2GCN, FAGCN	5	Node Classification
(Dietz et al., 2025)	1	NDCG	Item Exposure	Novelty	UserKNN, ItemKNN, MF, ItemMF, GeoBPRMF, RankGeoFM, PopGeoNN, HKV	32	Recommendation
Ours	10	NDCG, Recall, Precision, MRR, Hit Rate	Gini, Item Exposure, Average Popularity	Novelty, Item Coverage, Personalization, Intra-list Distance	UserKNN, ItemKNN, Deep-MF, SimpleX, RecVAE, MultiVAE, DiffRec, LightGCN, MixGCF, NCL DirectAU, SGL, XSimGCL	25	Recommendation

Apart from the previously mentioned diverse literature, in this thesis, we also concentrate on the coherency between qualitative and quantitative performance measurements such as the relation between recommendation ranking accuracy and diversity. Previous studies have shown that traditional recommendation algorithms tend to favor popular items, leading to low diversity and fairness in recommendations (Kowald et al., 2023; Vrijenhoek et al., 2024).

Zhao et al. (Zhao et al., 2025) investigated the relationship between fairness and diversity metrics in their survey. Their experiments include item exposure fairness and user fairness as fairness metrics. Besides, intra-list distance and aggregate diversity as diversity metrics. They utilized different matrix factorization (MF) variations as recommendation models. Their findings indicate that increasing item fairness led to a rise in aggregate diversity, as unpopular items were recommended more frequently. But this study may be considered as limited since it employs a small number of evaluation metrics and only one data set. In addition, the robustness of their results may be challenged and difficult to generalize, as they employ old-fashioned recommendation methods.

Isufi et al. (Isufi et al., 2021) analyzed the accuracy-diversity trade-off in GCN-based models. By modifying convolutional methods, they created 16 different GCN variations and evaluated them on three real-world datasets using two accuracy and two diversity metrics. Their results indicated that models using linear filters were more robust to hyperparameter selection and better at balancing accuracy and diversity. Additionally, they found that bipartite graph-based models achieved higher accuracy but lower diversity compared to user-user or item-item graph-based models. Despite some variations, their findings generally suggest that higher accuracy in GCN-based models often comes at the expense of diversity.

Besides, previous studies also have shown that the diversity and fairness of recommendations made by traditional approaches can be lacking due to the bias of recommendation algorithms towards user or item popularity (Kowald et al., 2023; Vrijenhoek et al., 2024). More clearly, algorithms tend to suggest more popular items widely. Although we come across the works dedicated to revealing the possible bias of traditional recommendation systems (Abdollahpouri et al., 2021; Klimashevskaja et al., 2024) there is not much comparative studies that focuses mostly on deep-learning based models.

3 REAL WORLD BIPARTITE GRAPHS

In this chapter, we first introduced the graph-related topological features utilized in this study, providing an understanding of their relevance and role in our analysis. Following this, we introduce the different real-world datasets used in our experiments. We then discuss the topological characteristics of the resulting graphs.

3.1 Graph Topological Features

This study employs a bipartite graph representation, where the graph is defined as $G = (U, I, E)$. Here, U represents the set of user nodes, I represents the set of item nodes, and the overall node set is given by $V = U \cup I$. The edge set, denoted as E , consists of connections between users and items, satisfying $E \subseteq U \times I$, and the term $N(v)$ denotes the nodes that are directly connected to node v . Notably, no edges exist between nodes within the same set, meaning there are no direct connections between two users or two items. The topological features are explained with this bipartite graph structure in mind. While these features can also be applied to homogeneous graphs, their interpretation may change depending on the graph structure. In a recommendation system, where we focus on a user-item graph, we also explain how some of these features appear from both the user and item perspectives.

Node degree measures the number of connections a node has, offering insights into the activity level of users or items. High-degree nodes often represent popular items or highly active users, which can bias recommendations toward these nodes. In our experiments, we separately calculated the minimum, maximum, average, and standard deviation of user and item node degrees. We also calculated the *entropy* of the user degree distribution and the item degree distribution separately, which reflect the diversity in connectivity among users and items, respectively. High entropy suggests a more balanced network where many nodes have similar degrees.

Graph sparsity quantifies the proportion of missing connections in the graph relative to the total possible connections, providing a measure of how incomplete the interaction data is. Sparse graphs pose challenges for learning effective embeddings and capturing

meaningful relationships. Graph sparsity calculation is given in Eq. 3.1.

$$\text{Sparsity} = 1 - \frac{|E|}{\binom{|V|}{2}} \quad (3.1)$$

where $|E|$ is the number of edges in the graph and $\binom{|V|}{2} = \frac{|V|(|V|-1)}{2}$ is the total possible number of edges for a graph with $|V|$ nodes.

Gini Coefficient measures inequality in the degree distribution. Captures how skewed the degree distribution is. It shows whether a few nodes have very high degrees, while most have low degrees. Ranges from 0 to 1. A higher *Gini* value suggests a more imbalanced network, where a few nodes dominate the connectivity. We calculated the $Gini_u$ and $Gini_i$ separately for users and items, respectively. Given a network with N nodes, the Gini coefficient is computed using the sorted degree distribution :

$$G = \frac{\sum_{i=1}^N (2i - N - 1)k_i}{N^2 \bar{k}} \quad (3.2)$$

where k_i represents the degree of the i -th node in the sorted degree sequence, and $\bar{k}$ is the mean degree.

Eigenvector Centrality is a measure of a node's influence within a network, considering not only its direct connections but also the importance of its neighbors. Unlike degree centrality, which counts direct links, eigenvector centrality assigns higher scores to nodes connected to other influential nodes. Its recursive nature makes it particularly effective in capturing hierarchical structures and ranking nodes based on global network significance.

$$x_v = \frac{1}{\lambda} \sum_{j \in N(v)} A_{vj} x_j \quad (3.3)$$

where x_v is the centrality score of node v , A_{vj} is the adjacency matrix entry indicating the connection between nodes v and j , λ is the largest eigenvalue of the adjacency matrix A

Eigenvector Centralization (EC) shows how much the overall influence in a network is focused on a few nodes instead of being spread evenly across all nodes. While

eigenvector centrality measures the influence of individual nodes based on their connections to other influential nodes, eigenvector centralization is a network-level metric that summarizes the overall inequality or concentration of this influence across all nodes. The calculation of eigenvector centralization is provided in Eq. 3.4.

$$EC = \frac{\sum_{i=1}^{|V|} (C_{\max} - C_v)}{|V| - 1} \quad (3.4)$$

where C_v is the eigenvector centrality score of node v . $C_{\max}$ is the maximum eigenvector centrality score in the graph and $|V|$ is the total number of nodes in the graph.

Clustering coefficient measures the tendency of nodes to form tightly-knit groups. It evaluates how likely a node's neighbors are to be connected to each other. This can reveal the presence of local clusters or communities in the graph. The clustering coefficient calculation is given in Eq. 3.5.

$$C_v = \frac{2e_v}{k_v(k_v - 1)} \quad (3.5)$$

where e_v is the number of edges between the neighbors of node v and k_v is the degree of node v .

Average path length measures the average shortest path distance between all pairs of nodes in the graph. It indicates how well-connected the graph is and can reveal how easily information or influence propagates through the graph. Average path length calculation is given in Eq. 3.6.

$$APL = \frac{1}{|V|(|V| - 1)} \sum_{v \neq j} d(v, j) \quad (3.6)$$

where $|V|$ is the total number of nodes in the graph, $d(v, j)$ is the shortest path distance between nodes v and j .

L3 similarity is a topological metric that measures the likelihood of two nodes interacting based on paths of length three between them (Kovács et al., 2019). Unlike local similarity measures that consider only direct neighbors, L3 accounts for indirect interactions by evaluating paths of exactly three hops in the network.

Mathematically, the L3 score between nodes u and v is given by :

$$S_{uv} = \sum_{i,j} \frac{A_{ui}A_{ij}A_{jv}}{\sqrt{k_u k_i k_j k_v}} \quad (3.7)$$

where A is the adjacency matrix of the network, $A_{ui}A_{ij}A_{jv}$ represents the existence of a three-hop path between nodes u and v through intermediate nodes i and j , k_u, k_i, k_j, k_v denote the degrees of the corresponding nodes, ensuring that high-degree nodes do not disproportionately affect the score.

To characterize the overall network structure, we compute the average L3 similarity, which is the mean L3 score across all node pairs :

$$\bar{S} = \frac{1}{|V|} \sum_{v \in V} S_v \quad (3.8)$$

Long tail analysis examines the distribution of item popularity in the network by distinguishing between *head* (popular items) and *tail* (less popular items). A network with a strong long-tail effect suggests that a small subset of items accounts for a large proportion of interactions, which can lead to recommendation biases.

To quantify this effect, we compute the *long tail item ratio* using a threshold of 5%. The top 5% most popular items are compared against the remaining 95% :

$$\text{LTR} = \frac{\sum_{i \in \text{head}} k_i}{\sum_{j \in \text{tail}} k_j} \quad (3.9)$$

where k_i represents the degree of item i . A higher LTR indicates a more skewed distribution, meaning popular items dominate interactions, while a lower LTR suggests a more balanced item exposure.

Power law degree exponent refers to the exponent that characterizes the decay rate of the degree distribution in networks. The degree distribution of many real-world networks follows a power-law (Zhang et al., 2022). This means that the probability $P(k)$ of a node having degree k is proportional to a power of k . This relationship is expressed as :

$$P(k) \sim k^{-\alpha} \quad (3.10)$$

where α is the degree exponent, which characterizes the shape of the distribution.

The exponent α is a critical parameter that determines the structure of the network :

- If $2 < \alpha < 3$, the network is *scale-free*, meaning a few nodes have very high degrees (hubs), while most nodes have low degrees.
- If $\alpha > 3$, the degree distribution decays more rapidly, leading to fewer hubs and a network closer to a random structure.
- If $\alpha \leq 2$, the network becomes highly heterogeneous, where extremely high-degree nodes dominate.

Given a set of observed degrees $k_1, k_2, \dots, k_n$ with a minimum degree $k_{\min}$, α is estimated as :

$$\hat{\alpha} = 1 + n \left[\sum_{i=1}^n \ln \frac{k_i}{k_{\min}} \right]^{-1} \quad (3.11)$$

where n is the number of nodes with $k \geq k_{\min}$, $k_{\min}$ is the lower bound beyond which the power-law holds.

Homophily in bipartite networks quantifies the tendency of similar nodes to interact with each other. In our study, we measure homophily separately for items and users using pairwise Jaccard similarity, which captures the overlap in their respective interaction patterns.

Item homophily measures the extent to which users interacting with the same item also interact with similar items. To compute this, we first construct an item-user mapping, where each item is associated with the set of users who have interacted with it. Then, for each user, we retrieve the list of items they have engaged with and compute the pairwise Jaccard similarity between every pair of items :

$$J(I_1, I_2) = \frac{|U_{I_1} \cap U_{I_2}|}{|U_{I_1} \cup U_{I_2}|} \quad (3.12)$$

where U_{I_1} and U_{I_2} denote the sets of users who interacted with items I_1 and I_2 , respectively. The final *item homophily score* is obtained by averaging these similarity values

across all users :

$$H_{\text{item}} = \frac{1}{|U|} \sum_{u \in U} \frac{1}{|P_u|} \sum_{(I_1, I_2) \in P_u} J(I_1, I_2) \quad (3.13)$$

where P_u represents the set of item pairs interacted with by user u . A high item homophily score indicates that users tend to engage with items that attract similar user groups, suggesting *preference clustering* in the network.

User homophily captures the similarity among users who interact with the same item, reflecting whether users engaging with a particular item tend to have overlapping interests in other items. This is computed similarly by constructing a user-item mapping, where each user is linked to the set of items they have interacted with. Then, for each item, we retrieve the list of users who have engaged with it and compute the Jaccard similarity between every pair of users :

$$J(U_1, U_2) = \frac{|I_{U_1} \cap I_{U_2}|}{|I_{U_1} \cup I_{U_2}|} \quad (3.14)$$

where I_{U_1} and I_{U_2} denote the sets of items interacted with by users U_1 and U_2 , respectively. The final user homophily score is computed as :

$$H_{\text{user}} = \frac{1}{|I|} \sum_{i \in I} \frac{1}{|P_i|} \sum_{(U_1, U_2) \in P_i} J(U_1, U_2) \quad (3.15)$$

where P_i is the set of user pairs who interacted with item i . A high user homophily score suggests that items attract users with similar interaction histories, indicating strong *user preference overlap*.

3.2 Real World Datasets as Bipartite Graphs and Their Properties

In this section, we describe the real-world datasets used in our experiments. As outlined in Section 3.1, each dataset is represented as a bipartite graph with undirected and unweighted edges. We incorporate neither metadata nor predefined features for users or items. Instead, the structure relies solely on binary interactions between nodes. Datasets that were not already split into train-test were randomly assigned to the test

group. Users who interacted with less than two items were removed from the dataset. Train and test groups were ensured to have at least one sample for the same user. We listed the experienced datasets below.

Movielens¹ The MovieLens datasets consist of movie ratings provided by users. The datasets are collected from MovieLens website. There are several versions with different sizes of data such as MovieLens-100K, MovieLens-1M and MovieLens-20M. Each dataset contains user-movie interaction data and metadata of both users and items. Metadata include additional information like release year and genre of items, as well as user demographics like age, gender, and occupation.

Yelp2018² Items in Yelp2018 represent local businesses such as restaurants, bars, retail shops, and service providers. Each edge in the graph represents an interaction indicating whether a user has reviewed or rated that business.

Amazon-Kindle³ The Amazon-Kindle dataset consists of user interactions with ebooks on the Amazon Kindle platform.

iFashion⁴ The Alibaba-iFashion dataset consists of user interactions with fashion items on the Alibaba e-commerce platform.

BookCrossing⁵ The BookCrossing dataset consists of user interactions with books collected from the Book-Crossing community. The dataset contains user-book interaction data and side information. The side information includes book metadata like title, author and year of publication, as well as user demographics like age and location.

LastFM⁶ The LastFM dataset consists of user interactions with music tracks collected from the Last.fm music platform. The dataset contains user-track interaction data and side information. The side information includes track metadata like artist name and tags, as well as user listening history and preferences.

ModCloth⁷ The ModCloth dataset consists of user ratings with clothing items collected from the ModCloth e-commerce platform. The dataset contains user-item interaction data and side information. The side information includes item metadata like images, descriptions and size measurements, as well as user purchase history and fit feedback.

Song-Data⁸ The song dataset contains user-song interaction data.

-
1. <https://grouplens.org/datasets/movielens/>
 2. <https://github.com/Coder-Yu/SELFRec/tree/main/dataset/yelp2018>
 3. <https://github.com/Coder-Yu/SELFRec/tree/main/dataset/amazon-kindle>
 4. <https://github.com/Coder-Yu/SELFRec/tree/main/dataset/iFashion>
 5. <https://grouplens.org/datasets/book-crossing/>
 6. <http://millionssongdataset.com/lastfm/>
 7. https://cseweb.ucsd.edu/~jmcauley/datasets.htmlclothing_fit
 8. <https://www.kaggle.com/datasets/rymnikski/dataset-for-collaborative-filters/data>

Setur The Setur dataset was obtained from Setur, a travel agency operating in Turkey. This dataset is not publicly available. It contains the hotel purchase history of the users and the properties of the hotels. We used only user-hotel interactions without considering hotel characteristics.

Table 3.1 presents the total number of users, items, and interactions for all the datasets used in this study. These statistics are derived from the full graphs, prior to any train-test separation. Among the datasets, MovieLens-100k has the fewest users, with 943, while iFashion has the largest user base, with 300,000 users. In terms of item count, ModCloth has the smallest set, with 994 items, whereas BookCrossing contains the most, with 237,186 items. Regarding interactions, Setur has the lowest number, totaling 57,671, while Song-data has the highest, with 2,000,000 interactions.

TABLE 3.1 – Total number of users, items and interactions of experienced datasets.

Dataset	#User	#Item	#Interaction
LastFM	1,883	6,953	82,349
MovieLens-100k	943	1,680	99,998
MovieLens-1m	6,040	3,679	1,000,209
Yelp	31,668	38,048	1,561,406
BookCrossing	4,151	237,186	800,521
Kindle	138,333	98,572	1,909,965
Modcloth	14,476	994	69,561
iFashion	300,000	81,614	1,607,813
Setur	17,910	2,506	57,671
Song-data	200,000	123,178	2,000,000

Table 3.2 presents the degree distribution statistics for each dataset, separately for users and items. The table reports the minimum, maximum, average degree, and standard deviation for both users and items. The datasets we analyzed display diverse interaction patterns across different domains. BookCrossing exhibits extremely active users with a maximum of 13,602 interactions and an average of 192.88 interactions per user. This contrasts sharply with its items, which receive minimal attention at only 2.93 interactions on average, suggesting a long-tail distribution effect. The MovieLens datasets demonstrate high engagement for both users and items, with MovieLens-1m containing highly popular items receiving up to 3,428 interactions. Song-data presents perfect uniformity as every user has exactly 10 interactions, indicating predetermined sampling methodology. LastFM shows consistent user activity patterns with low standard deviation, while item popularity varies significantly. Yelp maintains balanced engagement metrics between users and items, implying a more evenly distributed inter-

action ecosystem. Modcloth reveals an asymmetric pattern where users engage minimally but items receive substantial attention, suggesting concentrated user preferences on popular items. Setur users are particularly inactive with only 2.09 interactions on average, potentially indicating a challenging recommendation environment with limited user signals.

TABLE 3.2 – Degree statistics for experienced datasets. μ : Average Degree, σ : Standard Deviation, min : Minimum Degree, max : Maximum Degree.

Dataset	User				Item			
	min	max	μ	σ	min	max	μ	σ
LastFM	1	50	43.62	8.03	2	610	11.82	31.75
MovieLens-100k	20	736	105.97	100.90	1	620	59.62	80.97
MovieLens-1m	20	2,314	165.56	192.73	1	3,428	269.91	383.94
Yelp	18	2,311	49.30	56.35	10	1,492	41.03	59.64
BookCrossing	40	13,602	192.88	415.76	1	924	2.93	8.28
Kindle	1	1,370	13.71	26.78	1	1,805	19.30	29.53
Modcloth	1	250	4.80	11.76	2	1,259	69.98	142.37
iFashion	1	426	5.36	5.46	1	2,881	19.70	64.02
Setur	1	80	2.09	2.06	1	995	14.94	41.84
Song-data	10	10	10	0.00	1	1,904	15.65	54.57

In addition, Figure 3.1 provides a visual representation of the degree distributions across our datasets using box plots. The plot distinctly separates user and item degree patterns for each dataset, allowing for direct comparison. The top figure shows the user degree distribution and the bottom figure shows the item degree distribution. Each blue dot represents an individual user or item sample, with the box plots clearly indicating minimum, maximum, and mean values.

For both users and items, the degree distributions are not symmetrical. Especially for the item degree distributions, it can be seen that the mean and median lines are separated from each other. For users, most datasets show a skewed pattern where a few users are super-active while most have fewer connections. As example, BookCrossing and Kindle stand out with many highly-connected users, while MovieLens-100k and Yelp have a much more concentrated pattern where most users have similar connection counts. Since all users in Song-data have the same number of interactions, we do not see a special distribution for this dataset. All of these distributions highlight the structural differences in user-item interactions that can impact the performance and generalizability of PCFM.

It is well established that the degree distributions of many real-world networks, whether bipartite or not, often follow a power-law pattern (Newman, 2003). Therefore, we

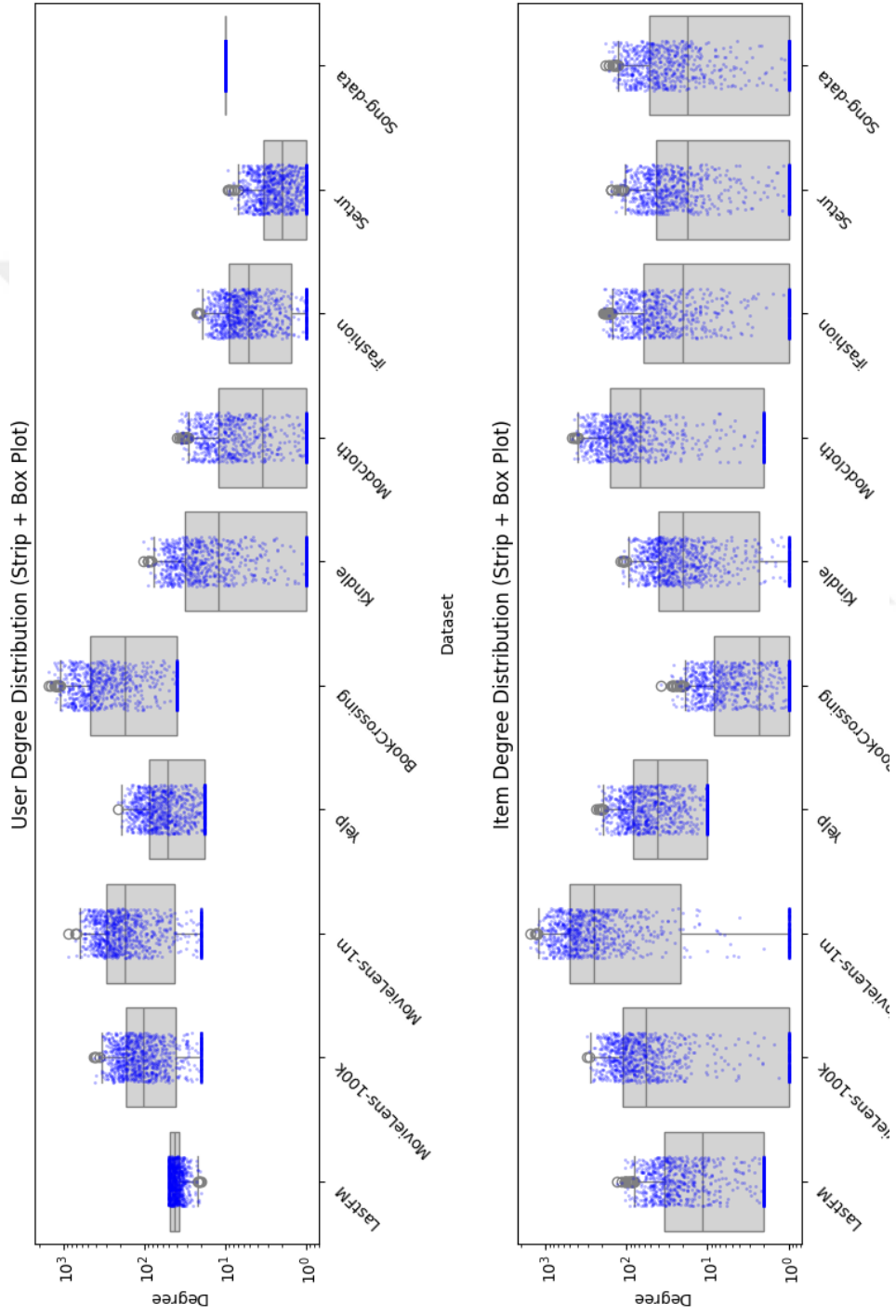


FIGURE 3.1 – Degree Distributions of Entire Datasets

fitted power-law distributions to the degree distributions of our datasets and examined their degree exponents. Table 3.3 presents the estimated power-law degree exponents (α) for our datasets. The power-law exponents across the entire graphs (A- α) range from 1.19 to 1.83, with MovieLens-1m having the lowest value and Yelp the highest. Lower exponents typically indicate more uneven degree distributions with stronger hubs, which aligns with our previous observations of MovieLens-1m having extreme maximum degrees. When examining user-specific exponents (U- α), Yelp shows a user exponent at 2.27, suggesting a more evenly distributed user activity pattern compared to other datasets. Song-data doesn't have a valid user-side α because all users have same number of interactions. For item-specific exponents (I- α), LastFM has the highest value (1.83), indicating its item popularity follows a more balanced distribution, while MovieLens-1m has the lowest (1.19), reflecting its tendency toward extremely popular items.

TABLE 3.3 – α Values for Degree Exponents. A- α : alpha for the entire graph, U- α : alpha for only the user nodes' degree distribution, I- α : alpha for only the item nodes' degree distribution

Dataset	A- α	U- α	I- α
LastFM	1.36	1.22	1.83
MovieLens-100k	1.23	1.76	1.26
MovieLens-1m	1.19	1.61	1.19
Yelp	1.83	2.27	1.97
BookCrossing	1.78	1.97	1.82
Kindle	1.34	1.35	1.31
Modcloth	1.51	1.55	1.38
iFashion	1.45	1.46	1.41
Setur	1.71	1.79	1.41
Song-data	1.36	20.49	1.42

In addition to the exponents presented in the Table 3.3, the corresponding distribution plots are provided in Figures 3.2, 3.3, and 3.4. Figure 3.2 illustrates the overall degree distribution including both users and items while Figure 3.3 focuses solely on user degrees, and Figure 3.4 presents item degree distributions. In these plots, the red lines represent the fitted power-law curves, while the green lines correspond to exponential distribution fits. For the Song-data dataset, user degrees are uniform and therefore no distribution is shown. Except for LastFM, the degree distributions of both users and items in all other datasets fit the power-law distribution better than the exponential distribution. In the case of LastFM, high-degree users are more concentrated than low-degree ones, indicating a deviation from the expected pattern; however, the item degree distribution in this dataset still aligns well with a power-law distribution.

Additionally, Table 3.4 presents other fundamental graph properties, such as network sparsity, eigenvector centralization, and average L3 Similarity, which further describe the topological differences across datasets.

TABLE 3.4 – Graph properties for the experienced datasets. EC : Edge clustering coefficient, U-I Ratio : User-item ratio.

Dataset	Sparsity	User Homophily	Item Homophily	L3 Similarity	EC	U-I Ratio
LastFM	0.9937	0.0800	0.0763	0.52×10^{-3}	0.18	0.27
MovieLens-100k	0.9368	0.1820	0.1796	1.62×10^{-3}	0.08	0.56
MovieLens-1m	0.9550	0.1440	0.1522	0.47×10^{-3}	0.05	1.63
Yelp	0.9987	0.0283	0.0282	5.41×10^{-5}	0.12	0.83
BookCrossing	0.9992	0.0153	0.1802	9.57×10^{-5}	0.39	0.01
Kindle	0.9998	0.0505	0.0549	1.76×10^{-5}	0.09	1.41
Modcloth	0.9951	0.0882	0.0605	1.03×10^{-3}	0.13	0.59
iFashion	0.9999	0.0872	0.0136	1.71×10^{-5}	0.41	3.67
Setur	0.9987	0.2528	0.0265	2.55×10^{-4}	0.55	7.14
Song-data	0.9999	0.0536	0.0174	1.32×10^{-5}	0.13	1.52

MovieLens-100k shows the lowest sparsity (0.9368), indicating it has relatively more interactions compared to its potential maximum, while datasets like iFashion and Song-data approach near-complete sparsity (0.9999). Setur demonstrates the highest user homophily (0.2528), suggesting its users tend to have more similar interaction patterns than other datasets. For item homophily, BookCrossing and MovieLens-100k show the highest values (0.1802 and 0.1796 respectively), indicating items in these datasets are more likely to be consumed by similar user groups. The L3 Similarity value, representing the average similarity between nodes based on their third-order connections, is highest for MovieLens-100k (1.62×10^{-3}), while Yelp, BookCrossing, and e-commerce datasets have considerably lower values, suggesting limited third-order connectivity patterns. The Eigenvector Centrality (EC) values vary significantly across datasets, with Setur showing the highest value (0.55), followed by iFashion (0.41) and BookCrossing (0.39). Higher EC values indicate greater influence of central nodes in the network, suggesting these datasets contain more prominent users or items that dominate the interaction patterns. In contrast, MovieLens-1m has the lowest EC (0.05), indicating a more distributed influence across nodes. Setur has an extremely high U-I ratio (7.14), indicating many more users than items, while BookCrossing has the lowest (0.01), with far more items than users.

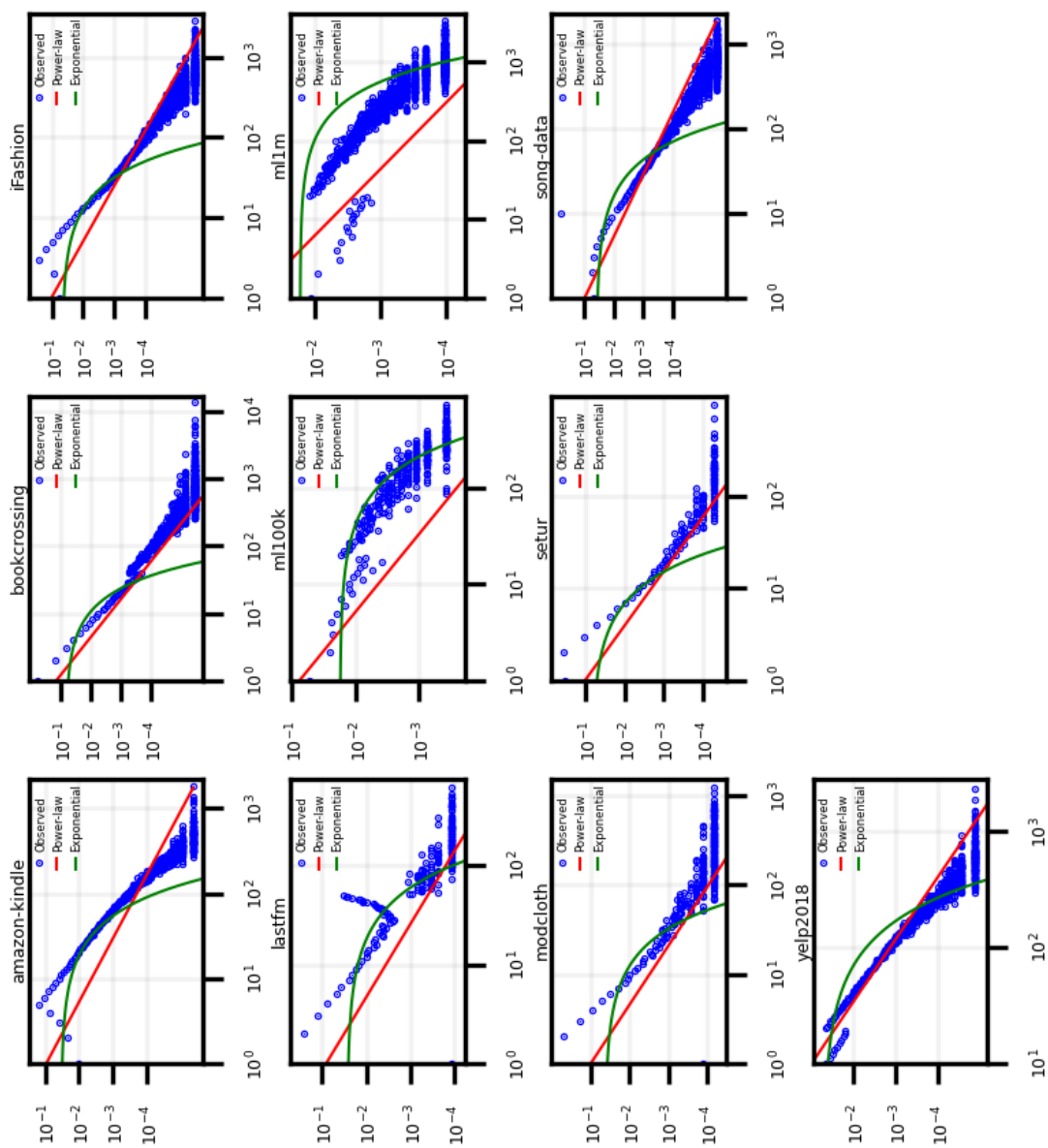


FIGURE 3.2 – Degree Exponents of Entire Datasets

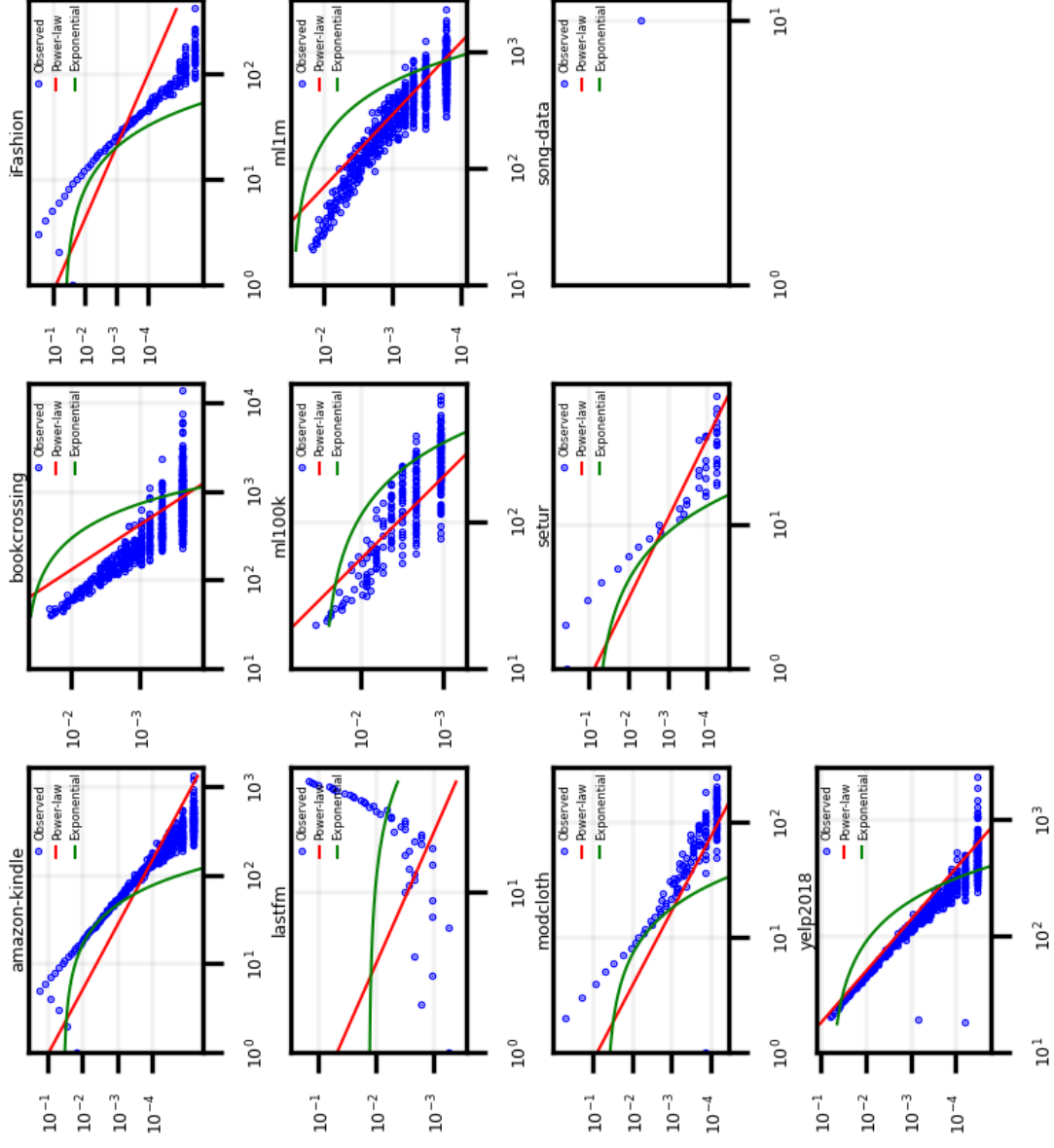


FIGURE 3.3 – Degree Exponents of Only the User Nodes

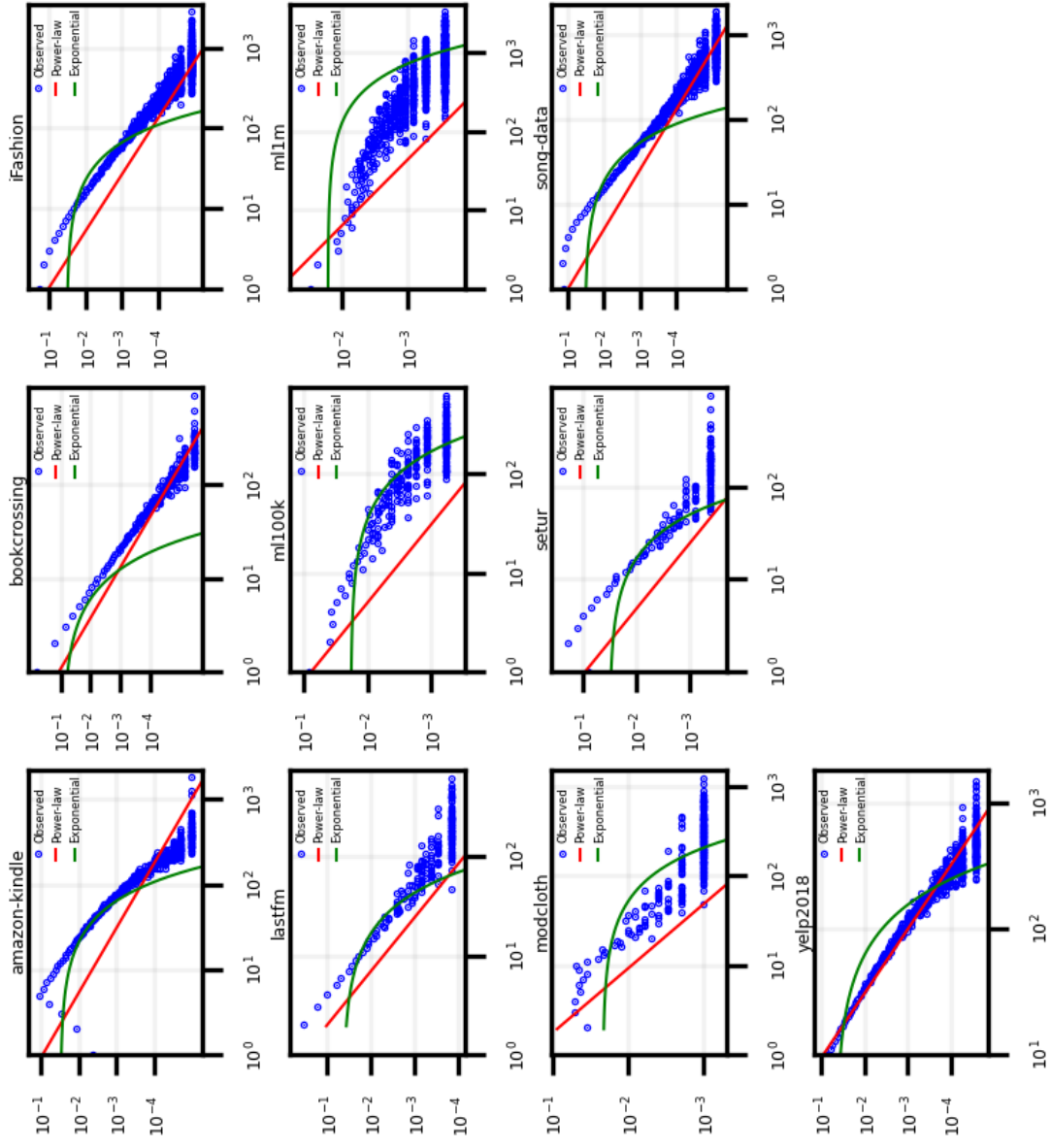


FIGURE 3.4 – Degree Exponents of Only the Item Nodes

4 METHOD

We propose an experimental evaluation of the performances with different perspectives of different recommendation approaches. There are hundreds of recommendation strategies in the literature. Here, we concentrate on the most recent and well-performing ones. We choose the methods having different strategies. Thus, in our experiments, deep learning-based matrix factorization (Xue et al., 2017), the most popular GNN-based model LightGCN (He et al., 2020), graph augmentation models SGL (Wu et al., 2021) and XSimGCL (Yu et al., 2023) and negative sampling based MixGCF (Huang et al., 2021) are used. The performances are evaluated on three axes : *Accuracy*, *Fairness*, and *Diversity*. In the following parts, we both explain the algorithms and the performance evaluation metrics.

4.1 Personalized Recommendation Algorithms

In this thesis, we selected and experimented with 13 collaborative filtering (CF) models from those discussed in Sect. 2.1.2 and Sect. 2.2.2. The chosen models span a diverse range of architectures. We listed all the evaluated models in Table 4.1. In this table, we compared whether these models are used for any accuracy, fairness or diversity evaluation either the model’s referenced paper or other previous recommendation studies listed in Table 2.1. While all models have been evaluated at least once under the accuracy criteria, fairness evaluations of current models have been made less frequently. However, none of our reference studies performed an analysis on diversity.

We also present a taxonomy developed in this thesis to organize and present the models we experienced. We gave our classification in Fig. 4.1 for a easier interpretation. Since our input structures remain the same, using only user-item interactions without any additional features, we focused on how the models generate outputs when developing the taxonomy.

In our taxonomy, we split the PCFM into two main groups : traditional ones and neural network-based models. The traditional methods include clustering-based models such

TABLE 4.1 – Comparison of experienced recommendation models

Model	Category	Sub-Category	Accuracy Eval.	Fairness Eval.	Diversity Eval.
UserKNN (Resnick et al., 1994)	Traditional	-	✓	✓	×
ItemKNN (Deshpande and Karypis, 2004)	Traditional	-	✓	✓	×
Deep-MF (Xue et al., 2017)	Neural Net.	MLP	✓	✓	×
SimpleX (Mao et al., 2021)	Neural Net.	MLP	✓		
MultiVAE (Liang et al., 2018)	Neural Net.	VAE	✓	×	×
RecVAE (Shenbin et al., 2020)	Neural Net.	VAE	✓	×	×
DiffRec (Wang et al., 2023)	Neural Net.	VAE	✓	×	×
LightGCN (He et al., 2020)	Neural Net.	GNN	✓	✓	×
MixGCF (Huang et al., 2021)	Neural Net.	GNN	✓	×	×
SGL (Wu et al., 2021)	Neural Net.	GNN	✓	✓	×
NCL (Lin et al., 2022)	Neural Net.	GNN	✓	×	×
DirectAU (Wang, Yu, Ma, Zhang, Chen, Liu and Ma, 2022)	Neural Net.	GNN	✓	×	×
XSimGCL (Yu et al., 2023)	Neural Net.	GNN	✓	✓	×

as ItemKNN (Deshpande and Karypis, 2004) and UserKNN (Resnick et al., 1994), which are commonly found in the literature. Subclasses of the neural network models are as follows : *(i)* The first category includes models built on top of a straightforward Multilayer Perceptron (MLP). These models attempt to model the relationship between entities based only on linear layers and activations and without making use of more sophisticated mechanisms. *(ii)* The second category is VAE-based models, with the encoder-decoder structure. Such models encode user-item interactions to a latent space (via encoder layers) and decode them for reconstruction or new interaction prediction (via decoder layers). *(iii)* The last class is graph-based models that exploit graph structures for modeling the interactions between users and items. To make it more clear, we further divide the similarity graph-based models into two additional sub-categories. The first difference is whether contrastive learning is adopted during training of the model. The non-contrastive learning category is represented by the model LightGCN which is commonly used in the survey. Some models with constrastive learning define the positive and negative pairs from predefined rules. These belong to

the family of the explicit contrastive learning. Some other models integrate contrastive learning with augmentation methods to enhance the information generated from the data and improve learning behaviour. These belong to the family of graph augmentation.

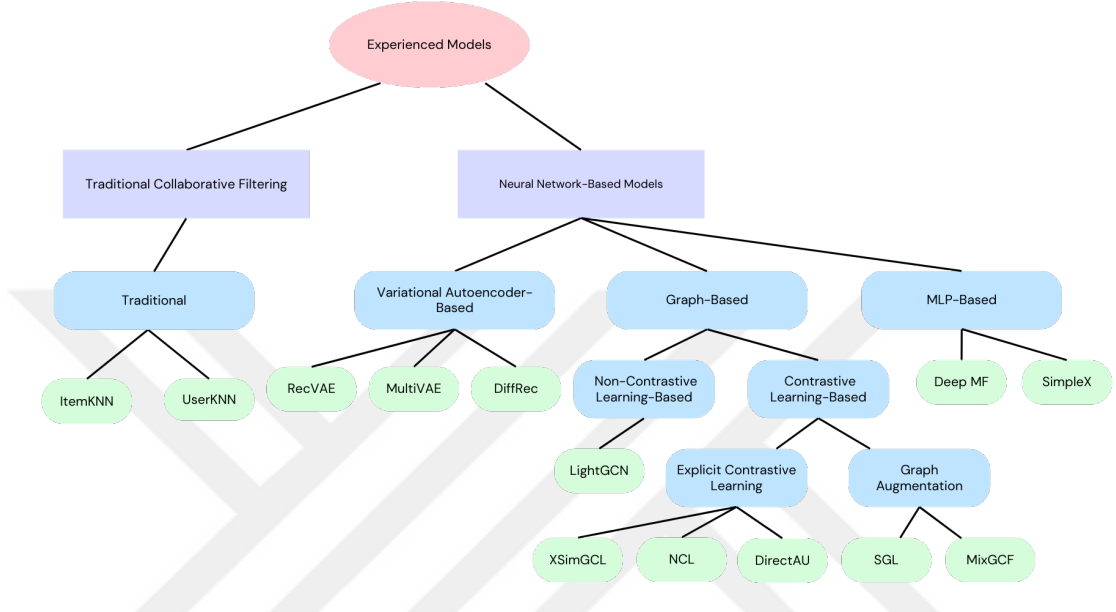


FIGURE 4.1 – A taxonomy diagram of the evaluated models in the thesis. Purple rectangles represent the main model categories, while blue rounded boxes indicate sub-categories based on architectural or learning principles. Green ovals list the individual models evaluated under each category.

UserKNN (Resnick et al., 1994) operates on the principle that similar users tend to like similar items. The model calculates similarity between users based on their interaction histories using metrics such as cosine similarity or Pearson correlation. For a target user u , the model identifies the k most similar users (nearest neighbors) and aggregates their preferences to predict items of interest.

ItemKNN (Linden et al., 2003), conversely, exploits similarities between items rather than users. This approach assumes that users tend to like items similar to those they have previously interacted with. The model computes item-item similarities based on user interaction patterns. For recommendation, the model aggregates a user’s preferences for similar items to predict ratings for unseen items.

Deep Matrix Factorization (DeepMF) (Xue et al., 2017) is one of the most popular personalized recommender approach (Koren et al., 2009). Traditional MF operates by breaking down the user-item interaction matrix using singular value decomposition

(SVD) (Sankari et al., 2024). This operation results in two smaller, lower-dimensional matrices. These matrices reveal latent factors that represent the relations between users and items. We employed a deep learning-based version of MF (Xue et al., 2017) in this study. This approach differs from the traditional ones, which rely on linear algebra. Instead of linear techniques, it uses neural networks to learn complex relationships between users and items. The model encodes users and items as dense vectors in a low-dimensional space. It optimizes these embeddings using the Bayesian Personalized Ranking (BPR) loss and stochastic gradient descent.

SimpleX (Mao et al., 2021) combines matrix factorization with user behavior modeling. Each user u and item i has embedding vectors e_u and e_i . SimpleX also looks at items a user has interacted with in the past. For a user u with an interacted item set H_u , the model calculates an aggregated preference vector :

$$p_u = \sum_{k \in H_u} I_k \cdot \alpha_k e_k$$

where I_k equals 1 for actual items and 0 for padding, and α_k are weights based on different aggregation methods like average pooling or attention mechanisms.

The final user representation combines the user embedding and the preference vector :

$$h_u = g \cdot e_u + (1 - g) \cdot V p_u$$

where g is a parameter that controls the mixing ratio and V is a learnable transformation matrix.

SimpleX uses a cosine contrastive loss (CCL) that measures cosine similarity between users and items :

$$L_{CCL}(u, i) = (1 - \hat{y}_{ui}) + \frac{w}{|N|} \sum_{j \in N} \max(0, \hat{y}_{uj} - m)$$

where $\hat{y}_{ui}$ is the cosine similarity between user u and item i , N is a set of negative samples, m is a margin value, and w controls the weight of negative samples. This loss function helps the model focus on difficult negative samples.

Before the **LightGCN** (He et al., 2020), NGCF (Wang et al., 2019) was leveraging GCNs to refine user and item embeddings by aggregating information from their neighbors. But, NGCF and similar models consist of feature transformation and nonlinear activation. These operations may not contribute significantly to CF tasks and can increase the training complexity. In response to these drawbacks, LightGCN simplifies GCN-based recommendation by removing feature transformation and nonlinear activation. It keeps only the neighborhood aggregation process. This simplification not only improves the ease of training, but also accuracy for recommendations.

LightGCN operates on a user-item interaction graph. Each user u and item i is initialized with an embedding vector $e_u^{(0)}$ and $e_i^{(0)}$, respectively. LightGCN then refines these embeddings through multiple layers of neighborhood aggregation.

Unlike traditional GCNs, LightGCN propagates embeddings without feature transformation or nonlinear activation. The propagation rule for user and item embeddings at layer $(k + 1)$ is :

$$e_u^{(k+1)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}} e_i^{(k)} \quad (4.1)$$

$$e_i^{(k+1)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}} e_u^{(k)} \quad (4.2)$$

where the normalization factor $\frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}$ ensures stability in propagation.

After K propagation layers, LightGCN combines embeddings across layers using a weighted sum :

$$e_u = \sum_{k=0}^K \alpha_k e_u^{(k)}, \quad e_i = \sum_{k=0}^K \alpha_k e_i^{(k)} \quad (4.3)$$

where α_k is the weight assigned to each layer's embedding. Typically, equal weights ($\alpha_k = \frac{1}{K+1}$) are used.

The final recommendation score for a user-item pair (u, i) is computed as :

$$\hat{y}_{ui} = e_u^T e_i \quad (4.4)$$

which represents the similarity between user and item embeddings.

For training, LightGCN employs the *Bayesian Personalized Ranking (BPR)* loss (Rendle et al., 2012) :

$$\mathcal{L}_{\text{BPR}} = - \sum_{(u,i,j) \in \mathcal{D}} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \|E^{(0)}\|^2 \quad (4.5)$$

where $\sigma(\cdot)$ is the sigmoid function, $\mathcal{D}$ is the training set containing triplets (u, i, j) , where user u prefers item i over item j , λ is a regularization coefficient.

A key difficulty in CF is to efficiently sample negative instances from implicit feedback as negative interactions are usually unobserved (Zhao et al., 2023). Traditional negative sampling methods calculates the importance weight of negative items by using uniform random sample or complex hard-negative mining (Yang et al., 2024). **MixGCF** (Huang et al., 2021) brings in a new negative sampling mechanism. Rather than sampling the real negatives directly, MixGCF generates synthetic negatives by mixing embeddings of multiple layers from the GNN according to both positive side and negative side. This generates artificially negative samples, which provides an augmentation on the learning task. This is not a CF model itself; rather, it is a technique for enhancing existing models.

MixGCF also works on a bipartite graph. Each user u and item i is initialized with an embedding vector $e_u^{(0)}$ and $e_i^{(0)}$, respectively.

MixGCF blends positive and negative embeddings to create harder negatives :

$$e_{v-}^{(l)} = \alpha e_{v+}^{(l)} + (1 - \alpha) e_{v-}^{(l)}, \quad (4.6)$$

where $\alpha \sim \text{Uniform}(0, 1)$. This forces the model to distinguish subtle differences between positives and hard negatives.

Instead of using a single-layer embedding, MixGCF aggregates multi-hop negatives to generate a more challenging negative representation :

$$e_{v-} = \sum_{l=0}^L \gamma_l e_{v_l-}, \quad (4.7)$$

where v_l^- is a sampled negative at layer l , and γ_l is a weight parameter.

Using these synthetic negatives, the *MixGCF-enhanced BPR loss* becomes :

$$\mathcal{L}_{MixGCF} = \sum_{(u,v^+) \in E} \sum_{v^- \sim \mathcal{H}(u,v^+)} \ln \sigma(e_u^\top e_{v^+} - e_u^\top e_{v^-}), \quad (4.8)$$

where $\mathcal{H}(u, v^+)$ represents hard negatives synthesized through positive and hop mixing.

Graph-based collaborative filtering models like LightGCN which are solely on bipartite graphs struggle with degree bias and noisy interactions (Fan et al., 2023). To close the gap, **SGL** (Wu et al., 2021) proposes *self-supervised learning (SSL)* as a novel graph-based recommendation approach. The key idea is to generate many augmented views of the user-item graph and to contrastively learn between the views. This enhances the recommendation robustness and fairness. More specifically, SGL constructs two augmented graph views and seeks to make them as similar as possible. It also defines the user-item bipartite graph as G , similar to previous models.

SGL applies three augmentation techniques to generate alternative graph structures :

- **Node Dropout (ND)** : Randomly removes nodes and their connections.
- **Edge Dropout (ED)** : Randomly drops edges while keeping node embeddings.
- **Random Walk (RW)** : Generates dynamic subgraphs through random walks, capturing high-order connectivity.

Formally, let G_1 and G_2 be two augmented versions of the original graph G . We obtain embeddings $e_u^{(1)}$ and $e_u^{(2)}$ from these views.

SGL enforces consistency between the two augmented views of a node u using the InfoNCE loss :

$$\mathcal{L}_{SSL} = \sum_{u \in U} -\log \frac{\exp(s(e_u^{(1)}, e_u^{(2)})/\tau)}{\sum_{v \in U} \exp(s(e_u^{(1)}, e_v^{(2)})/\tau)}, \quad (4.9)$$

where $s(e_u^{(1)}, e_u^{(2)})$ is the cosine similarity between the two representations of user u . τ is a temperature hyperparameter. Negative pairs are sampled from the entire user set U .

This loss forces different views of the same node to be similar while different nodes

become different.

To obtain the final embeddings, SGL combines the BPR loss and contrastive loss :

$$\mathcal{L} = \mathcal{L}_{BPR} + \lambda \mathcal{L}_{SSL} + \eta \|\Theta\|^2, \quad (4.10)$$

where λ controls the contribution of the self-supervised loss. η is the regularization coefficient. Θ represents model parameters.

As a whole, the current contrastive learning algorithms primarily adopt random contrastive pairs crafting or only take simple graph augmentations into account, which neglects the potential information of the inner graph relationships. **Neighborhood-enriched Contrastive Learning (NCL)** (Lin et al., 2022) strengthens the graph-based collaborative filtering algorithm by directly incorporating both structural and semantic neighborhoods into contrastive learning. The approach revolves around two critical objectives : structure-aware contrastive learning and prototype-based contrastive learning.

The structure-aware contrastive objective captures the large-range interactions between nodes and their neighbors from various layers of a GNN. The embeddings on each layer inherently aggregate information from higher-order neighbors that NCL treats as the embeddings on different layers being positive contrastive pairs. It guarantees the nodes to be consistent in different neighborhood expanses.

The prototype-based contrastive objective incorporates semantic neighbors by grouping users (or items) based on their representation similarities rather than direct connections in the graph. To do this, it clusters nodes into prototype groups based on the expectation-maximization (EM) algorithm, guaranteed that semantically similar nodes get closer to each other in a graph embedding space. Unlike traditional contrastive learning which only learns structural relationships, the proposed method allows NCL to discover latent similarities in low-resource settings.

For training the model, NCL minimizes the BPR loss with the intra-modality structure and prototype contrastive losses. This multi-task learning method ensures the accuracy of the recommendation and also enhances the robustness to sparsity.

DirectAU (Wang, Yu, Ma, Zhang, Chen, Liu and Ma, 2022) introduces a new lear-

ning objective that leverages two aspects : *alignment* and *uniformity*. The alignment constraint allows the representations for user-item pairs with to share the same distribution in the representation space. This allows capturing strong preference signals, and keeping close users and items in the embedding space. The uniformity property, however, pushes the embedding points evenly spread on the hypersphere, avoiding the collapse of representation and learning of meaningful distinctions among entities.

To achieve this, DirectAU introduced the alignment and uniformity combined loss. The alignment loss minimizes the squared Euclidean distance between embeddings of positive user-item pairs :

$$l_{\text{align}} = \mathbb{E}_{(u,i) \sim p_{\text{pos}}} \|\tilde{f}(u) - \tilde{f}(i)\|^2 \quad (4.11)$$

where $f(u)$ and $f(i)$ are the embeddings of user u and item i , and P_{pos} represents the set of user-item pairs that have been observed in the dataset, representing positive interactions.

The uniformity loss ensures that embeddings are evenly distributed by minimizing the expected Gaussian potential between all pairs of embeddings :

$$l_{\text{uniform}} = \log \mathbb{E}_{x,y \sim p_{\text{data}}} e^{-2\|\tilde{f}(x) - \tilde{f}(y)\|^2} \quad (4.12)$$

where P_{data} is the distribution of all users and items.

The final objective function balances these two losses :

$$L_{\text{DirectAU}} = L_{\text{align}} + \lambda L_{\text{uniform}} \quad (4.13)$$

where λ is a hyperparameter that controls the contribution of the uniformity loss. Unlike traditional methods that rely on negative sampling, DirectAU ensures representation quality by directly optimizing these two properties. This approach improves recommendation performance without requiring complex encoder architectures. In this thesis, we used DirectAU with LightGCN.

eXtremely Simple Graph Contrastive Learning (XSimGCL) (Yu et al., 2023) is a contrastive learning recommendation model to ease the augmentation procedure. In

contrast to previous contrastive learning models that utilize graph augmentation techniques like edge or node dropout, XSimGCL applies these operations with noise-based embedding augmentation. This method injects controlled noise into the embedding space in order to achieve better representation uniformity, and still preserves important structural information. Instead of comparing different graph augmentations, XSimGCL contrasts embeddings from different layers of the model, allowing more efficient training and avoiding the need for extra graph structures.

To generate contrastive views, XSimGCL perturbs embeddings by adding uniform random noise :

$$\tilde{e}_i = e_i + \epsilon \cdot \xi_i, \quad \xi_i \sim U(-1, 1) \quad (4.14)$$

where e_i is the original embedding, ϵ is a hyperparameter controlling the noise magnitude, and ξ_i is a random vector sampled from a uniform distribution.

XSimGCL optimizes a joint loss function that combines the BPR loss with a cross-layer contrastive loss :

$$L_{\text{XSimGCL}} = L_{\text{BPR}} + \lambda \sum_i \log \frac{e^{\text{sim}(\tilde{e}_i, e_i)/\tau}}{\sum_j e^{\text{sim}(\tilde{e}_i, e_j)/\tau}} \quad (4.15)$$

where $\text{sim}(\cdot, \cdot)$ represents cosine similarity, and τ is a temperature scaling parameter.

DiffRec (Wang et al., 2023) applies diffusion models to denoise user interactions. Instead of using traditional generative models like GANs or VAEs, DiffRec gradually corrupts user interaction histories by adding Gaussian noise in a forward process and then recovers the original interactions through an iterative denoising process.

The forward process corrupts the interaction history $\mathbf{x}_0$ step by step :

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}) \quad (4.16)$$

where β_t controls the amount of noise added at each step. The reverse process attempts to recover the original interactions :

$$p_{\theta}(\mathbf{x}t - 1|\mathbf{x}t) = \mathcal{N}(\mathbf{x}t - 1; \mu_{\theta}(\mathbf{x}t, t), \Sigma_{\theta}(\mathbf{x}t, t)) \quad (4.17)$$

where μ_{θ} and Σ_{θ} are learned by a neural network.

The training objective minimizes the denoising reconstruction loss :

$$L_{\text{DiffRec}} = \sum_{t=1}^T \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} [|\mu_{\theta}(\mathbf{x}_t, t) - \mathbf{x}_0|^2] \quad (4.18)$$

By leveraging diffusion processes, DiffRec enhances the robustness of recommendations and effectively models interaction uncertainties.

MultiVAE (Liang et al., 2018) extends variational autoencoders to collaborative filtering by modeling user-item interactions through a latent variable framework. It assumes that each user’s preferences are encoded in a latent vector z sampled from a Gaussian prior :

$$z \sim \mathcal{N}(0, I) \quad (4.19)$$

The user’s observed interactions are then generated based on a multinomial likelihood :

$$p(x|z) = \text{Mult}(x|n, \pi(z)) \quad (4.20)$$

where $\pi(z)$ represents the probability distribution over items obtained through a neural network transformation of z .

The model is trained using the evidence lower bound (ELBO), which consists of a reconstruction term and a KL-divergence regularization :

$$L_{\text{MultiVAE}} = \mathbb{E}q\phi(z|x)[\log p_{\theta}(x|z)] - \beta \text{KL}(q_{\phi}(z|x)|p(z)) \quad (4.21)$$

where β controls the strength of the KL-divergence regularization.

RecVAE (Shenbin et al., 2020) is a variational autoencoder-based model designed for top-N recommendation with implicit feedback. It builds upon Multi-VAE by intro-

ducing a new composite prior, a dynamic regularization strategy, and an alternating training approach to improve recommendation performance.

The model encodes user preferences in a latent variable z , where the likelihood of observed interactions is modeled as :

$$p(x|z) = \text{Mult}(x|n, \pi(z)) \quad (4.22)$$

where $\pi(z)$ is obtained through a neural network transformation followed by a softmax function. Mult refers to the Multinomial distribution.

To improve training stability, RecVAE introduces a composite prior combining a standard Gaussian with the posterior from the previous epoch :

$$p(z) = \alpha \mathcal{N}(0, I) + (1 - \alpha) q_{\phi_{\text{old}}}(z|x) \quad (4.23)$$

where α balances between the static Gaussian prior and the learned prior.

The loss function consists of a reconstruction term and a weighted KL-divergence :

$$L_{\text{RecVAE}} = \mathbb{E} q_{\phi}(z|x) [\log p_{\theta}(x|z)] - \beta_u \text{KL}(q_{\phi}(z|x) | p(z)) \quad (4.24)$$

where β_u adjusts regularization strength based on user interaction count.

RecVAE also employs an alternating training strategy, updating the encoder and decoder separately for improved convergence. These modifications enhance training stability and recommendation performance.

4.2 Recommendation System Performance Evaluation

In this subsection, we present 12 different evaluation metrics which are used for measuring the performance of recommendation systems from different perspectives. In general, we call all these metrics as *Recommendation System Performance Evaluation (RSPE) Metrics*. Table 4.2 lists all the used RSPE metrics in this thesis.

TABLE 4.2 – System Evaluation Metrics

Metric	Measures	Objective
NDCG	Ranking quality of recommendations	Accuracy
Recall	Proportion of items recommended successfully	Accuracy
Precision	Proportion of recommended items that are relevant	Accuracy
Hit Rate	Proportion of users with at least one relevant recommendation	Accuracy
MRR	How early the first relevant item appears	Accuracy
Intra List Dist.	Distance within a recommendation list	Diversity
Item Coverage	Proportion of items appearing in at least one recommendation list	Diversity
Personalization	Distance between users' recommendation lists	Diversity
Novelty	How unpopular recommended items are	Diversity
Avg. Popularity	How popular recommended items are	Fairness
Gini	How recommendations are distributed across items	Fairness
Item Exposure	How evenly recommendations are distributed across popularity groups	Fairness

4.2.1 Quantitative Performance Evaluation : Accuracy Metrics

We used 5 different accuracy metrics to measure the performance of the items recommended to users. These metrics measure different aspects of how efficiently users interact with the items recommended to them.

Normalized Discounted Cumulative Gain (NDCG) is a ranking-based metric used to evaluate the quality of the top-k recommendations in system. It takes into account both the relevance and the position of recommended items within the top-k list. NDCG@k uses a graded relevance scale, giving higher weights to highly relevant items that appear earlier in the recommendation list. The metric is calculated by comparing the Discounted Cumulative Gain (DCG) of the actual top-k recommendations to the DCG of an ideal ranking of the top-k items. NDCG@k values range from 0 to 1, with 1 indicating a perfect ranking where the most relevant items are at the top of the list. The NDCG@k is calculated as follows :

$$NDCG@k = \frac{DCG@k}{IDCG@k} \quad (4.25)$$

where $DCG@k$ is the Discounted Cumulative Gain for the top-k recommendations :

$$DCG@k = \sum_{i=1}^k \frac{2^{rel_i} - 1}{\log_2(i + 1)} \quad (4.26)$$

and $IDCG@k$ is the Ideal DCG for the top-k recommendations (the DCG of the perfect ranking of the top-k items). Here, rel_i is the graded relevance of the item at position

i , and the denominator $\log_2(i + 1)$ is the discount factor that reduces the impact of lower-ranked items.

Recall is a metric that measures the proportion of relevant items found in the top- k recommendations out of all relevant items for a user. It focuses on the model's ability to find and suggest relevant items within a limited set of recommendations. A high Recall indicates that the system is effective at surfacing relevant items within the top- k recommendations. However, Recall does not consider the ordering of items within the top- k list or the relevance of items beyond the k -th position. The equation for Recall@ k is :

$$Recall@k = \frac{|\text{Relevant} \cap \text{Recommended}@k|}{|\text{Relevant}|} \quad (4.27)$$

where $|\text{Relevant} \cap \text{Recommended}@k|$ is the number of relevant items in the top- k recommendations, and $|\text{Relevant}|$ is the total number of relevant items for the user.

Precision measures the proportion of relevant items in the top- k recommendations. It focuses on the accuracy of the recommendations within a fixed number of items, which is particularly important in scenarios where users are presented with a limited number of recommendations. A high Precision indicates that a large portion of the top- k recommendations are relevant to the user. The equation for Precision@ k is :

$$Precision@k = \frac{|\text{Relevant} \cap \text{Recommended}@k|}{k} \quad (4.28)$$

where $|\text{Relevant} \cap \text{Recommended}@k|$ is the number of relevant items in the top- k recommendations, and k is the number of items recommended.

Hit Rate (HR) measures the proportion of users for whom at least one relevant item appears in the top- K recommendations. It evaluates how effectively a recommendation system recommends at least one relevant item for each user. A higher HR@ K indicates that more users receive at least one correct recommendation. Unlike precision or recall, which consider multiple relevant items, HR@ K focuses on whether any relevant item is retrieved.

$$HR@k = \frac{\sum_{u \in U} \delta(R_u \cap T_u \neq \emptyset)}{|U|} \quad (4.29)$$

where U is the set of users present in both the recommendation and test sets. R_u is the set of top- k recommended items for user u . T_u is the set of relevant items for user

u in the test set. $\delta(R_u \cap T_u \neq \emptyset)$ is an indicator function that returns 1 if at least one relevant item is recommended, otherwise 0.

Mean Reciprocal Rank (MRR) measures how early the first relevant item appears in the recommendation list, averaged across all users. A higher MRR@K indicates that relevant items are ranked closer to the top, improving ranking quality.

$$MRR@k = \frac{1}{|U|} \sum_{u \in U} \sum_{r=1}^k \frac{\delta(R_{u,r} \in T_u)}{r} \quad (4.30)$$

where U is the set of users present in both the recommendation and test sets. $R_{u,r}$ is the item recommended at rank r for user u . T_u is the set of relevant items for user u in the test set. $\delta(R_{u,r} \in T_u)$ is an indicator function that returns 1 if the recommended item at rank r is relevant, otherwise 0.

4.2.2 Qualitative Performance Evaluation Metrics

Qualitative performance evaluation metrics offer a complementary perspective by assessing aspects such as diversity and fairness. We used 4 diversity metrics and 3 fairness metrics as qualitative evaluation. This section delves into the qualitative metrics employed in this study.

4.2.2.1 Fairness Metrics

We used four metrics to assess model fairness in handling items of varying popularity. These metrics gauge bias towards popular items and evaluate how recommendations are balanced across different popularity levels.

Average Popularity (AP) measures how much a recommender system favors popular items in its recommendations. It's normalized between 0 and 1, where values closer to 1 indicate that recommendations are heavily biased towards the most popular items, while values closer to 0 suggest the system tends to recommend less popular items. A value around 0.5 would indicate a balanced distribution between popular and unpopular items.

$$P_{\text{avg}} = \frac{1}{\sum_{u \in U} |I_u^K|} \sum_{u \in U} \sum_{i \in I_u^K} \frac{p(i)}{p_{\text{max}}} \quad (4.31)$$

GINI coefficient measures the degree of inequality between items in the system and expresses this inequality in a numerical value. The value ranges from 0 to 1, with 0 meaning that all items are equally recommended, and 1 representing complete inequality. Gini plays a critical role in determining the degree bias of the model. It can help to identify an unfair model.

$$\text{Gini} = 1 - 2 \cdot \text{AUL} \quad (4.32)$$

Where AUL is the Area Under the Lorenz curve, calculated as :

$$\text{AUL} = \frac{1}{n} \sum_{i=1}^n L(i) \quad (4.33)$$

The Lorenz curve $L(i)$ is defined as :

$$L(i) = \frac{\sum_{j=1}^i r_j}{\sum_{j=1}^n r_j} \quad (4.34)$$

Here n is the number of unique recommended items, r_j is the number of recommendations for the j -th item when items are sorted in ascending order of recommendation count.

Item Exposure Fairness (IEF) evaluates how evenly the recommendations are distributed across different item popularity groups. It first divides items into groups based on their popularity, then measures how balanced the recommendations are across these groups using entropy. Higher values indicate that the recommender system is providing more balanced exposure across all popularity segments, rather than concentrating recommendations within specific popularity groups.

$$IEF = \frac{H(E)}{H_{\text{max}}} = \frac{-\sum_{n=1}^N e_n \log(e_n)}{\log(N)} \quad (4.35)$$

where $H(E)$ is the entropy of exposure distribution, N is the number of groups and given as 5 in experiments. H_{max} represents the maximum entropy. e_g is the exposure rate of group g , defined as :

$$e_g = \frac{\sum_{i \in g} \text{rec}_i}{\sum_{i \in I} \text{rec}_i} \quad (4.36)$$

where rec_i represents the number of times item i is recommended

4.2.2.2 Diversity Metrics

We used four metrics to evaluate the diversity of recommendations. These metrics measure how varied the recommendations are for each user and assess the system's coverage of the entire item catalog. They also highlight differences in recommendations between users. Each metric offers specific insights into recommendation diversity.

Intra-List Distance (ILD) quantifies the diversity within a single user's recommendation list by measuring the average pairwise distance between recommended items. Higher ILD values indicate more diverse recommendations for each user, suggesting that the recommender system is providing a varied selection of items rather than a homogeneous set. By averaging the ILD across all users, we obtain a global measure of the recommender system's ability to provide diverse recommendations at the individual user level. The maximum value of ILD is theoretically unbounded and depends on the maximum possible pairwise distances between items, minimum value is 0.

$$ILD = \frac{1}{|U|} \sum_{u \in U} \frac{1}{\binom{k}{2}} \sum_{i, j \in R_u, i \neq j} dist(i, j) \quad (4.37)$$

Here U is the set of users, R_u is the recommendation list for user u , and $dist(i, j)$ is the degree distance between items i and j . Euclidean distance was used as the distance metric.

Item Coverage (IC) measures the proportion of all available items that appear in the recommendations across all users, indicating the system's ability to expose the entire item catalog. It is calculated as the ratio of unique items recommended to any user to the total number of items in the system. Higher values indicate that the recommender system is utilizing a broader range of items rather than focusing on a limited subset of popular items

$$Coverage_{@k} = \frac{|\bigcup_{u \in U} R_k(u)|}{|I|} \quad (4.38)$$

Here U is the set of users, I is the set of items, $R_k(u)$ is the top k recommended items for user u and $\bigcup_{u \in U}$ is the number of unique items recommended to all users.

Personalization metric quantifies the degree to which recommendations differ across

users, measuring how tailored the suggestions are to individual preferences. It is calculated by comparing the overlap of recommended items between users, with lower overlap indicating higher personalization. Higher values of this metric suggest that the recommender system is providing more unique, user-specific recommendations rather than generic, widely-popular items.

$$Personalization = 1 - \frac{|\bigcup_{u \in U} R_u|}{|U| \cdot k} \quad (4.39)$$

Where k is the number of recommendations per user.

Novelty metric quantifies the degree to which a recommender system suggests items that are relatively unknown or unpopular to users, measuring the system's ability to introduce users to new and potentially interesting content. It is typically calculated as the inverse of the average popularity of recommended items, where popularity can be defined by factors such as interaction frequency. Higher novelty values indicate that the system is recommending less well-known items, potentially broadening users' horizons and mitigating the "popularity bias" where already popular items dominate recommendations.

$$Novelty = 1 - \frac{1}{|U| \cdot k} \sum_{u \in U} \sum_{i \in R_u} \frac{pop(i)}{\max_{j \in I} pop(j)} \quad (4.40)$$

where $pop(i)$ is the popularity of item i .

5 EXPERIMENTAL DESIGN AND RESULTS

This chapter first presents our research questions and the designed experiments we follow to address the research questions. We give our experimental setup including the software libraries and computational resources used during the experiments at the end of this section. Second, we describe our train datasets used during training phase which play a crucial role in understanding performance variations. Final section systematically presents the findings for each research question, offering insights into model behaviors, metric relationships, and the influence of graph features on evaluation outcomes.

5.1 Experiment Design

We examine the relationships between three major components of recommendation systems : performance metrics, graph topological features, and personalized recommendation algorithms. Each of these components includes many subcategories, which increases the complexity of the experimental setup. Due to the large number of variables and possible interactions, it is not practical to rely only on simple or direct relationships. Therefore, we designed a structured and systematic experimental plan. This plan is guided by four research questions, each addressing a specific aspect of our main goal. In the following sections, we describe the design of our experiments for each question and explain how they help us find answers.

RQ1. How do the recommendation models’ performance differ according to evaluation criteria ?

The performance of recommendation systems can be evaluated across various dimensions. A model that excels in accuracy may perform poorly in diversity or fairness. Understanding these variations is crucial for selecting models aligned with the specific goals of a recommendation system.

To address this question, we compare how recommendation models perform across different evaluation metrics. We evaluated all models using accuracy, diversity, and fairness metrics (see Table 4.2). Our aim is to assess whether models exhibit consistent

performance across these dimensions or if their effectiveness varies significantly depending on the metric.

We used min-max normalization to enable fair comparisons between metrics that had different scales and ranges. We normalized all model scores to the $[0, 1]$ range for each dataset and metric combination, where 1 corresponds to the best-performing model and 0 to the worst. This normalization brings performance differences across metrics and datasets onto a common scale, which enables meaningful visual and statistical comparisons.

Formally, the normalized score for model i on metric j and dataset k is calculated as :

$$\hat{x}_{i,j,k} = \frac{x_{i,j,k} - \min(x_{j,k})}{\max(x_{j,k}) - \min(x_{j,k})} \quad (5.1)$$

where $x_{i,j,k}$ is the original score, and $\min(x_{j,k})$ and $\max(x_{j,k})$ denote the minimum and maximum scores across all models for metric j on dataset k , respectively.

The resulting normalized values allow us to analyze the distribution, stability, and trade-offs in model performance across various metrics. A model that consistently achieves high normalized scores across evaluation criteria can be considered robust and well-rounded. In contrast, substantial variability may indicate specialization or sensitivity to certain objectives.

RQ2. Are the evaluation metrics coherent with each other ?

Prior research indicates that enhancing accuracy in recommendation systems may compromise other objectives, such as fairness or diversity, suggesting inherent trade-offs among evaluation metrics. However, it remains uncertain whether these trade-offs are consistent across different models or if some models are better equipped to balance multiple evaluation criteria. Clarifying these relationships is crucial for developing recommendation systems that can optimize multiple objectives simultaneously without disproportionately favoring one.

To address this question, we analyzed how different evaluation metrics relate to each other across recommendation models. The Algorithm 1 below gives the procedure of the general approach followed.

Algorithm 1 Metric Correlation Analysis for RQ2

Require: Models, Datasets, Metrics, Scores

Ensure: CorrelationMatrices (one for each model)

```

1: Initialize CorrelationMatrices as an empty list
2: for each model index  $m$  in Models do
3:   Initialize ScoreMatrix as an empty matrix
4:   for each dataset index  $d$  in Datasets do
5:     Initialize row as an empty list
6:     for each metric index  $k$  in Metrics do
7:       value  $\leftarrow$  Scores[m][d][k]
8:       append(value, row)
9:     end for
10:    append(row, ScoreMatrix)
11:  end for
12:   $C \leftarrow$  calculate_pearson_correlation(ScoreMatrix)
13:  append( $C$ , CorrelationMatrices)
14: end for
15: return CorrelationMatrices

```

We collected performance values from ten datasets and calculated correlation matrices for each model. These matrices were computed by measuring Pearson correlation coefficients between pairs of evaluation metrics across all datasets. Pearson correlation between two metric x and y is calculated as follows :

$$[hbt]r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (5.2)$$

Where x_i and y_i represent the values of two different evaluation metrics for the i -th dataset, and $\bar{x}$ and $\bar{y}$ are their respective means across all datasets.

These matrices show us how metrics influence each other and reveal which models demonstrate stronger or weaker relationships between specific evaluation criteria. Our analysis helps identify models that can minimize trade-offs and achieve balanced performance across multiple evaluation dimensions.

The output will consist of correlation matrices that quantify the relationships between different evaluation metrics across various datasets for each model. If we observe strong

positive correlations between certain metrics, it could suggest that they measure similar underlying properties of recommendation performance. In contrast, negative or weak correlations may point to trade-offs or independence between objectives. This analysis will help us understand whether evaluation metrics tend to move together or in opposite directions, and whether some metrics provide redundant or complementary perspectives on model performance.

RQ3. Are performance metrics sensitive to the used recommendation model?

We investigate whether different evaluation metrics produce consistent rankings across various recommendation models. Besides, we examine if their values are influenced by the specific model applied. To address this, we compute the correlation of each metric across multiple models as demonstrated in Algorithm 2.

Algorithm 2 Metric Sensitivity Analysis for RQ3

Require: Models, Datasets, Metrics, Scores

Ensure: CorrelationDistributions (pairwise metric correlation values across models)

```

1: Initialize CorrelationDistributions as a dictionary {Keys : metric pairs, Values :
   list of correlation values across models}
2: for each model index  $m$  in Models do
3:   Initialize ScoreMatrix as an empty matrix
4:   for each dataset index  $d$  in Datasets do
5:     Initialize row as an empty list
6:     for each metric index  $k$  in Metrics do
7:       value  $\leftarrow$  Scores[m][d][k]
8:       append(value, row)
9:     end for
10:    append(row, ScoreMatrix)
11:  end for
12:   $C \leftarrow$  calculate_pearson_correlation(ScoreMatrix)
13:  for each metric pair  $(i, j)$  in Metrics do
14:    corr_value  $\leftarrow$  C[i][j]
15:    append(corr_value, CorrelationDistributions[(i, j)])
16:  end for
17: end for
18: return CorrelationDistributions

```

By quantifying the average correlation for each metric, we aim to understand which evaluation metric depends on the choice of recommendation model rather than the inherent quality of recommendations. Pearson correlation coefficients were used for the correlation calculation.

Answering this question provides insights into the robustness and generalizability of commonly used performance metrics. If certain metrics exhibit high sensitivity to the model, their reliability in comparing different recommendation approaches may be questioned. This analysis helps determine which metrics offer more stable and model-independent evaluations.

The output will include correlation values showing how consistently each evaluation metric ranks datasets across different recommendation models. These correlation patterns will reveal which metrics produce similar assessments regardless of algorithm choice versus which are highly model-dependent. For metrics with high average correlation values, we would interpret this as indicating stability and model-independence—suggesting these metrics rank datasets similarly regardless of which recommendation algorithm is used. Such metrics could be considered more reliable for general evaluation purposes.

RQ4. Can the recommendation model performance explained by topological characteristics ?

We examined whether model performance can be explained using only the topological features. Additionally, we investigated whether certain topological features have a greater impact on recommendation model performance compared to others.

Previous studies have primarily analyzed the relationship between topological features and model performance using linear assumptions. Their experiments demonstrate that sparsity might have an important impact on recommendation models' performance. However, their studies are limited because when using graph structure, the non-linear relationships can also be highly possible. So that linear assumption based approaches can be misleading or can not explain whole perspective of the relationship.

Given the scale of our experimental design—encompassing 12 evaluation metrics, 13 models, and 10 datasets—basic correlation analysis is not sufficient to uncover the potential effects of topological features on model performance. Capturing these complex interactions requires a more expressive modeling approach. Furthermore, linear regression falls short in identifying non-linear dependencies between graph structure

and evaluation outcomes. Therefore, we employed tree-based models, which are better suited for identifying such non-linear relationships (Rožman et al., 2024). Given that decision trees are prone to overfitting (Vaishnavi, 2017), we opted for a random forest model. Random forests address this issue by averaging the predictions of multiple decision trees, reducing noise and mitigating overfitting (Halabaku and Bytyçi, 2024). Each tree-based model was trained by using the topological features of the datasets as explanatory variables, with the corresponding recommendation performance metric serving as the target variable. We repeated this training process for each recommendation model-performance metric pair. We analyzed the feature importance derived from each trained model. Using these values, we computed an importance score for each topological feature. This helped identify which characteristics contribute most to model performance.

In random forests, feature importance is calculated using Mean Decrease in Impurity (MDI) (Louppe et al., 2013). Impurity measures how mixed the classes or values are at a node, with higher impurity indicating less predictive power. MDI works by measuring how effective each feature is at reducing this impurity when used for splitting nodes across all trees in the forest. When a node splits on a particular feature, the algorithm calculates the weighted reduction in impurity. These impurity reductions are aggregated for each feature across all trees and normalized to create a relative importance score. Features that consistently produce larger impurity reductions are considered more important, as they contribute more significantly to the model’s predictive performance. All the individual scores are non-negative and their sum is 1. Although there is no hard rule, practically features with higher than 0.1 importance scores can be considered as important. We utilized these importance scores to identify which dataset characteristics have the strongest influence on recommendation performance metrics.

The output will include feature importance values derived from training random forest models that predict evaluation scores using topological features of the dataset. If certain topological properties consistently show high importance across models and metrics, they may serve as strong indicators of recommendation performance. Alternatively, if importance scores vary significantly between metrics or models, it may suggest that model or metric is influenced by different structural factors.

Besides, we used ANOVA to analyze if there is a significant difference between two groups. ANOVA tests the hypothesis that the mean values of the given groups are

equal where $\mu_1 = \mu_2 = \mu_3$. ANOVA has some assumptions while analyzing the given data : (i) Measured values from a group should not affect the values from another group, (ii) Independent variable should be nominal and dependent variable should be metric variable, (iii) The given groups should follow normal distribution. There are two crucial metrics obtained from ANOVA tests : F-value and P-value. F-value is the ratio of two different estimates of variance and calculated as :

$$F = \frac{\text{Between-group variability}}{\text{Within-group variability}} = \frac{MS_{\text{Between}}}{MS_{\text{Within}}} \quad (5.3)$$

where MS_{Between} is the mean sum of squares between groups and MS_{Within} is the mean sum of squares within groups. A larger F-value means that there is a higher chance that the differences between groups are statistically significant and not just due to randomness. The P-value reflects this probability, specifically the proportion of finding the observed F-value or something more extreme, given that the null hypothesis itself is true. The P-value is calculated using the cumulative distribution function (CDF) of the F-distribution :

$$p\text{-value} = 1 - F_{\text{CDF}}(F; df_1, df_2) \quad (5.4)$$

where F is the calculated F-value. df_1 is the degrees of freedom for the numerator (between-groups) calculated as the number of groups minus one, and df_2 is the degrees of freedom for the denominator (within-groups) calculated as the total sample size minus the number of groups. Generally, a P-value below a predetermined significance level (typically 0.05) leads to rejecting the null hypothesis, suggesting that at least one group mean is significantly different from the others.

In our experimental setup, we employed ANOVA to test for significant differences between recommendation models across evaluation metrics, helping establish whether performance variations between architectures are meaningful or simply due to chance.

All deep learning-based models were trained with a fixed layer size of 3. The SelfRec framework¹ was used for training LightGCN, DirectAU, Deep-MF, MixGCF, NCL, SGL, and XSimGCL, while MultiVAE, RecVAE, DiffRec, and SimpleX were implemented using the RecBole framework². UserKNN and ItemKNN were implemented

1. <https://github.com/Coder-Yu/SELFRec>
2. <https://recbole.io/>

using the scikit-learn library³.

In line with prior studies (Wang et al., 2019; He et al., 2020; Wu et al., 2021; Fei et al., 2023), we set the latent dimension to 64 and the learning rate to 0.001, as these values are widely used as standard baseline configurations in recommendation models. Training was stopped if no improvement was observed for 30 epochs. All experiments were conducted on a system with 32GB RAM, an AMD Ryzen 9 8945HS processor, and an NVIDIA GeForce RTX 4070 GPU.

5.2 Train Datasets

We concentrated on user-item interactions. When splitting the data into training and testing subsets, we employed a random sampling method while ensuring that each user was represented by at least one rating in both the training and testing sets. We distributed the interactions into train and test sets following an 80-20 ratio. We presented the user, item and interaction counts in Table 5.1.

TABLE 5.1 – Total users, items and interactions of training datasets

Dataset	#User	#Item	#Interaction
LastFM	1,883	6,953	58,019
MovieLens-100k	943	1,680	90,570
MovieLens-1m	6,040	3,679	800,167
Yelp	31,668	38,048	1,237,259
BookCrossing	4,151	237,186	638,754
Kindle	138,333	9,8572	1,527,913
Modcloth	14,476	994	14,642
iFashion	300,000	81,614	1,255,447
Setur	17,910	2,506	57,671
Song-data	200,000	123,178	1,600,000

Our data sets have user, item and interaction numbers in different scales. iFashion has the largest number of user's as 300,000 and the MovieLens-100k with 943 users. Besides, Yelp has the largest number of items with 38,048 items while Modcloth is the smallest with 994. Similarly, Modcloth has the smallest number of interactions with 14,642 while it is 1,600,000 for Song-data. And minimum, maximum, average and standard deviation of degree statistics are given in Table 5.2. These values show a

3. <https://scikit-learn.org/>

similar distribution with the non-splitted datasets described in Section 3.2.

TABLE 5.2 – Degree statistics for the train sets. μ : Average Degree, σ : Standard Deviation, min : Minimum Degree, max : Maximum Degree.

Dataset	User				Item			
	min	max	μ	σ	min	max	μ	σ
LastFM	1	45	30.81	7.04	1	442	8.35	23.80
MovieLens-100k	10	727	95.98	100.91	1	550	54.00	71.91
MovieLens-1m	16	1,851	132.45	154.16	1	2,723	217.51	307.70
Yelp	16	1,848	39.07	45.11	1	1,258	32.72	49.27
BookCrossing	32	10,881	153.91	332.61	1	731	2.69	7.06
Kindle	1	1,100	11.02	21.52	1	1,443	15.47	23.69
Modcloth	1	5	1.01	0.12	1	329	14.73	34.14
iFashion	1	342	4.18	4.44	1	2,239	15.38	49.60
Setur	1	5	1.02	0.15	1	565	7.28	22.33
Song-data	8	8	8	0.00	1	1,562	12.99	45.03

Table 5.3 presents the estimated power-law degree exponents for the datasets. Figure 5.1 visualizes the estimated degree exponent for the entire dataset, Figure 5.2 visualizes the degree exponent for using only user nodes and Figure 5.3 visualizes degree exponents using only the item nodes.

TABLE 5.3 – α Values for Degree Exponents of train sets. A- α : alpha for all nodes, U- α : alpha for only the user nodes' degree distribution, I- α : alpha for only the item nodes' degree distribution

Dataset	A- α	U- α	I- α
LastFM	1.44	1.24	1.56
MovieLens-100k	1.24	1.56	1.27
MovieLens-1m	1.19	1.61	1.20
Yelp	1.25	2.49	1.27
BookCrossing	1.80	1.97	1.85
Kindle	1.36	1.38	1.34
Modcloth	2.27	2.42	1.50
iFashion	1.51	1.53	1.45
Setur	2.21	2.41	1.60
Song-data	1.39	16.49	1.45

Except the item node degree of LastFM, all distributions fits better to power-law distribution then exponential distribution. In LastFM, high degree nodes appear more than low degree nodes. Additionally, Table 5.4 presents the sparsity, user and item homophily, L3 Similarity, EC and user-item ratio values of the training datasets. Similar to the values of the original datasets, the most dense dataset is Movielens-100k and the

most sparse are iFashion and Song-data. The dataset with the highest user homophily value is Modcloth, while the lowest is Song-data. Song-data also has the lowest item homophily along with iFashion, while Kindle has the highest item homophily. Modcloth and Setur have the highest EC and U-I ratios due to their densely connected structures. For similar reasons, Setur has the highest L3 Similarity rate and BookCrossing has the lowest L3 Similarity rate.

TABLE 5.4 – Graph properties for the training sets. Sparsity : Graph sparsity, Item Homophily : Tendency of items to be connected to similar items, L3 Similarity : Average 3-hop connectivity, EC : Edge clustering coefficient, U-I Ratio : User-item ratio.

Dataset	Sparsity	User Homophily	Item Homophily	L3 Similarity	EC	U-I Ratio
LastFM	0.9955	0.06	0.07	5.39×10^{-4}	0.17	0.27
MovieLens-100k	0.9427	0.17	0.17	1.68×10^{-3}	0.08	0.56
MovieLens-1m	0.9640	0.11	0.11	4.83×10^{-4}	0.05	1.64
Yelp	0.9989	0.02	0.03	5.39×10^{-5}	0.12	0.83
BookCrossing	0.9993	0.01	0.19	9.85×10^{-5}	0.46	0.01
Kindle	0.9998	0.05	0.51	1.78×10^{-5}	0.09	1.40
Modcloth	0.9989	0.97	0.15	2.26×10^{-4}	0.70	14.56
iFashion	0.9999	0.11	0.01	1.68×10^{-5}	0.52	3.67
Setur	0.9995	0.97	0.24	1.29×10^{-4}	0.70	7.14
Song-data	0.9999	0.06	0.02	1.32×10^{-5}	0.15	1.62

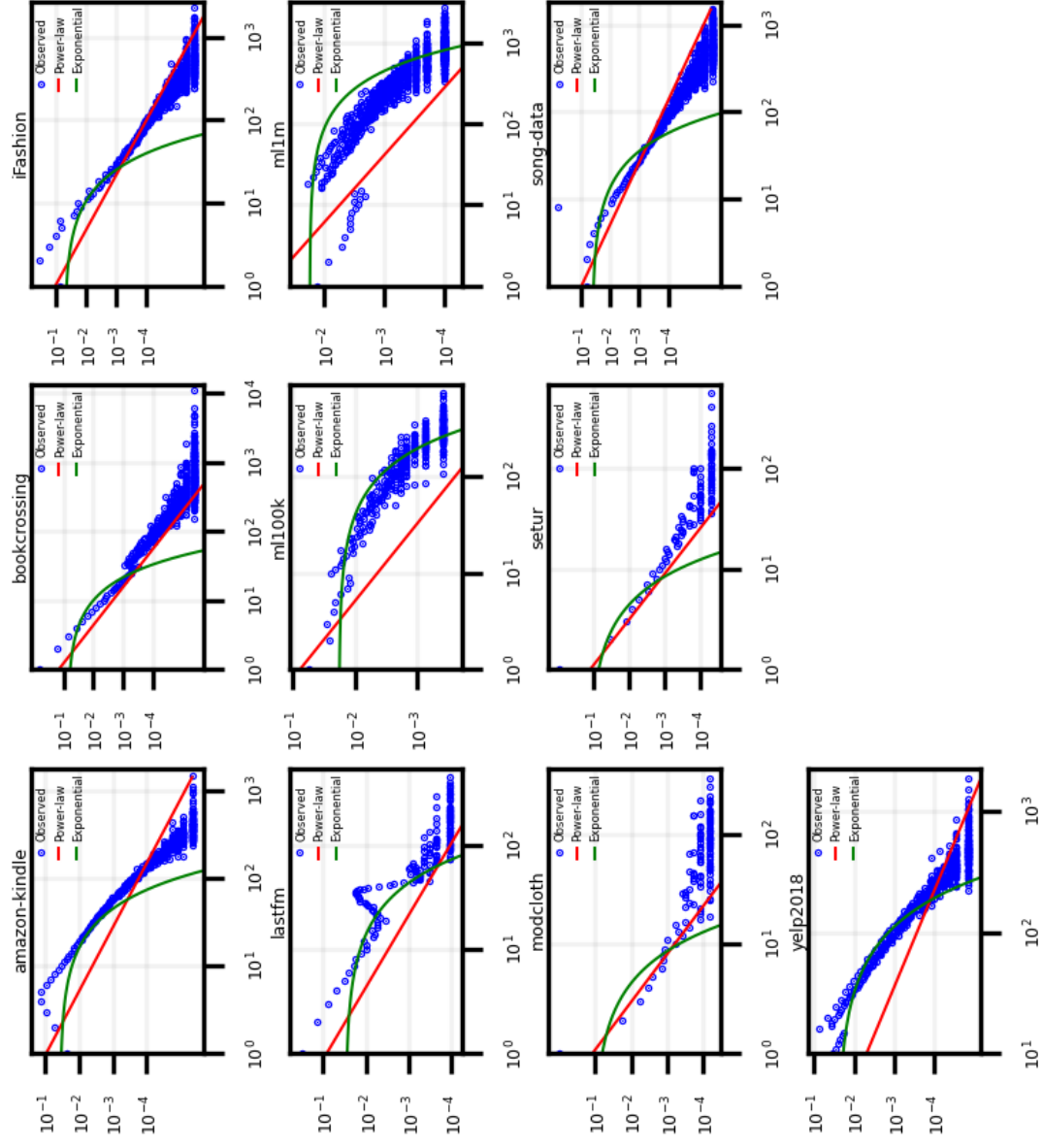


FIGURE 5.1 – Degree Exponents of All Nodes in Train Sets

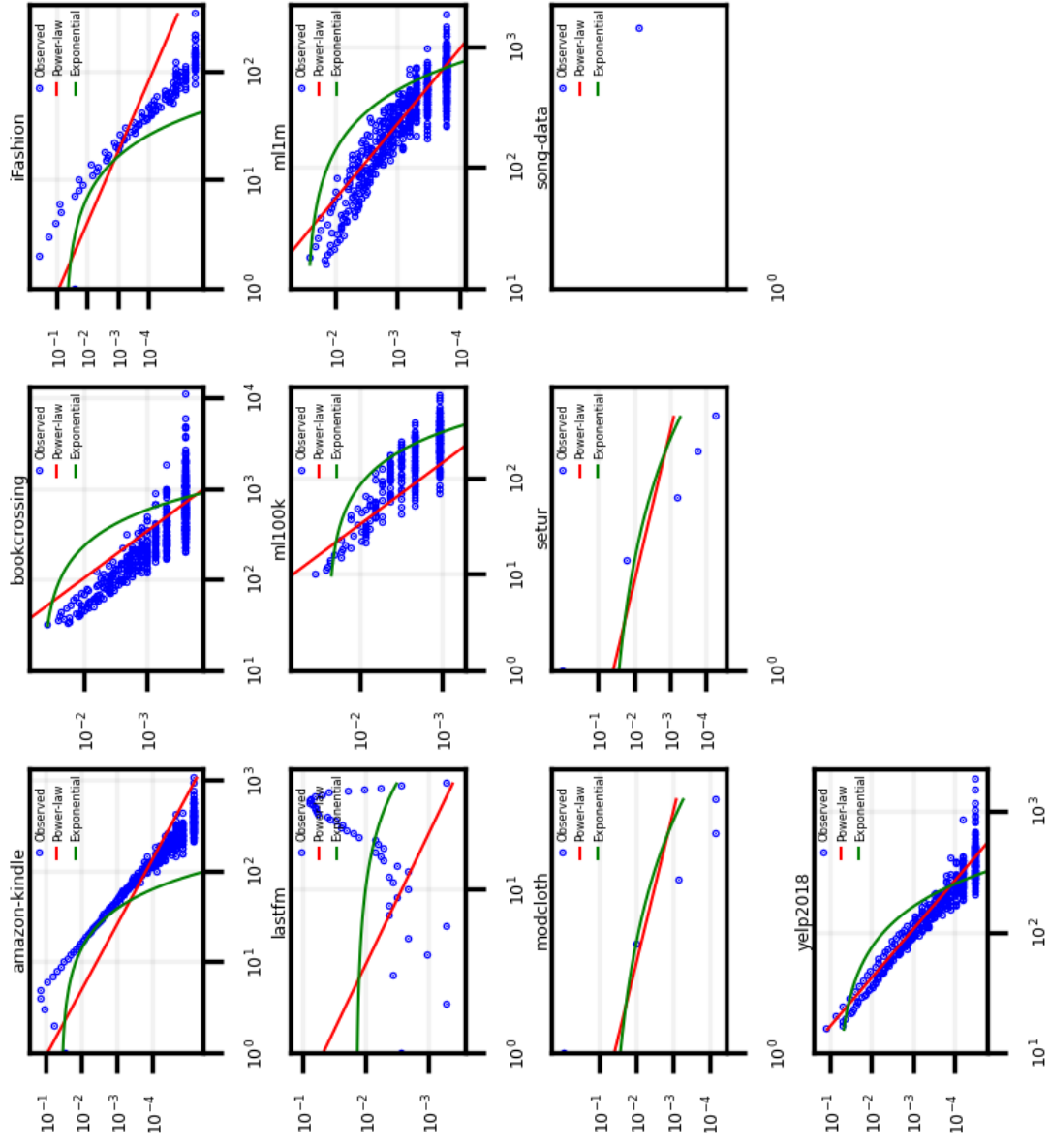


FIGURE 5.2 – Degree Exponents of Only the User Nodes in Train Sets

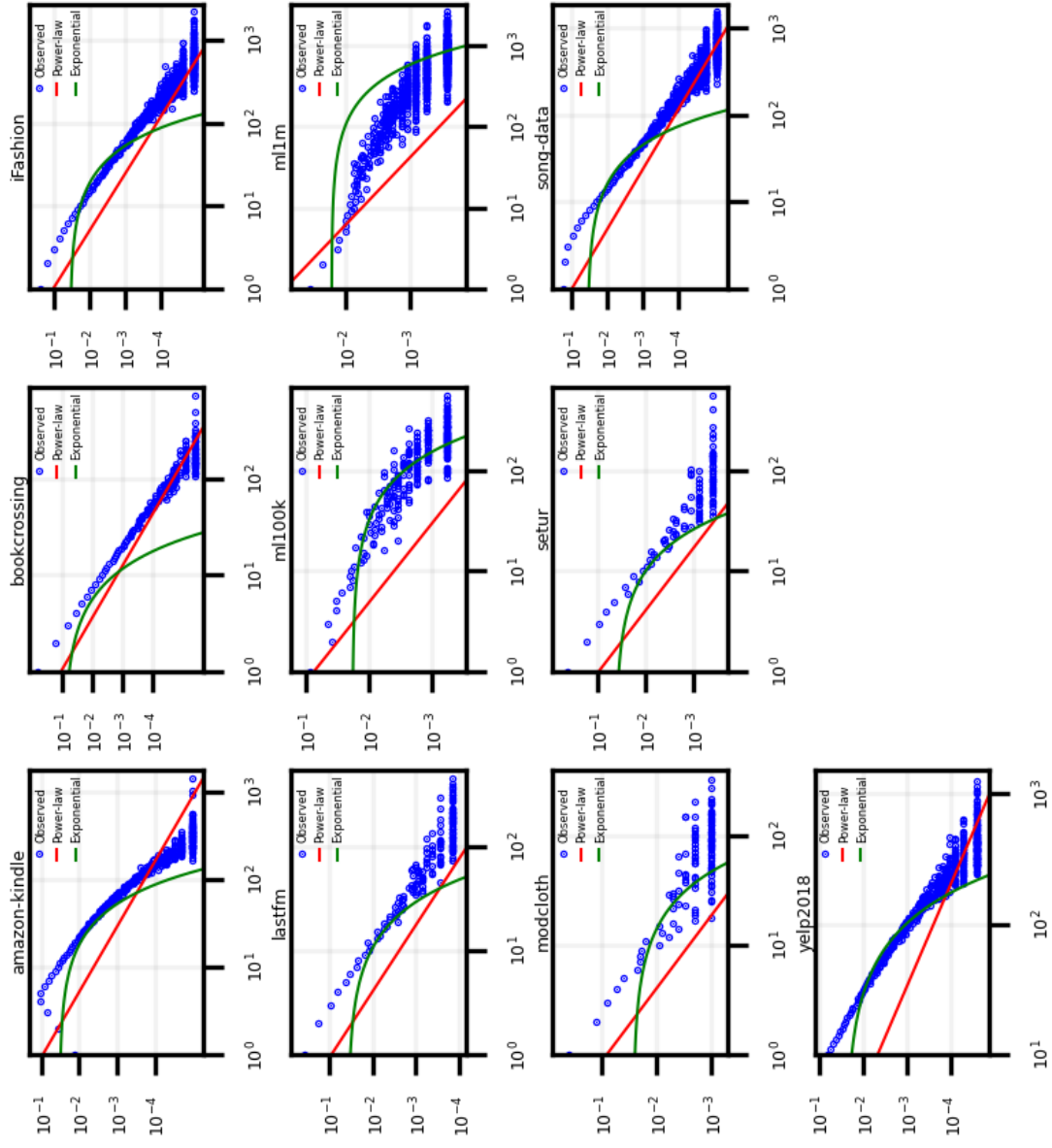


FIGURE 5.3 – Degree Exponents of Only the Item Nodes in Train Sets

5.3 Results

In the experiments, we concentrate on answering four research questions. First, we compare the performance of the recommendation system models among themselves from accuracy-diversity and fairness perspectives. Second, we examine whether the evaluation metrics are correlated with each other on a model-by-model basis. Third, we investigate whether different evaluation metrics produce consistent rankings across various PCFM. And finally, we investigate any relationship between the topological features and the evaluation results.

In this section, we present the results for each of the four research questions.

5.3.1 Models' Performance According to Evaluation Dimensions (RQ1)

We present all the measured metrics across all models in APPENDIX A. To objectively compare the performance of models across all metrics, we transform their scores to a common range using min-max normalization. This scaling procedure adjusts all metric values to fall within the $[0, 1]$ interval, where 1 represents the best-performing model and 0 the worst for a given dataset-metric combination. As a result, we can fairly assess and visualize model performance across different evaluation dimensions. Figure 5.4 presents candlestick charts showing the distribution of these normalized scores for each model across multiple datasets and metrics. The candles are color-coded based on model architecture. The black diamond within each candle represents the mean score, while the horizontal line inside the box denotes the median. Within each individual plot, the models are grouped by their architecture. Narrow boxes indicate consistent performance, whereas wider boxes suggest higher variability across datasets.

Graph-based models achieve higher accuracy scores compared to other models. Graph-based approaches naturally incorporate collaborative filtering principles by exploiting higher-order connectivity patterns that traditional models might miss. The structural information encoded in these models provides a richer representation space that translates to more accurate recommendations, as evidenced by the strong and stable performance of SGL across accuracy metrics. Although they exhibit high accuracy, these models show considerable variance in performance across different datasets, suggesting that their effectiveness may be context-dependent or sensitive to specific data cha-

racteristics. Overall, SGL achieves the highest accuracy with minimal variation across datasets. Even with its straightforward approach, LightGCN delivers strong and reliable accuracy. We do not observe a common trend for other metrics, indicating that different recommendation tasks and evaluation criteria may require specialized model architectures tailored to specific objectives rather than a one-size-fits-all approach. When compared to more sophisticated models, traditional models appear to provide recommendations that are close to random suggestions, highlighting the limitations of simple similarity-based approaches that fail to capture the complex, non-linear relationships present in user-item interaction data. In addition to graph-based and traditional models, we also evaluated VAE- and MLP-based architectures. The three VAE-based models exhibited similar accuracy levels across datasets, with no significant differences in their overall performance. Notably, MultiVAE, despite being the earliest among them, produced the most consistent results, indicating its robustness across varying data conditions. Among MLP-based approaches, Deep-MF stood out by consistently achieving competitive accuracy scores. In many scenarios, it ranked just behind the graph-based models, suggesting that MLPs can still offer a viable alternative under certain conditions. SimpleX, although built on a much simpler architecture, delivered surprisingly strong results. However, its high variance across evaluation metrics indicates that while lightweight models like SimpleX can perform well in some cases, they may struggle with consistency or generalization in others. For the diversity and fairness metrics, although traditional models show outlier results in some metrics, there is no significant difference between model architectures.

To answer the RQ1, we compared how different model architectures perform across various evaluation criteria. The following list summarizes the key findings based on accuracy, diversity, and fairness metrics :

- Graph-based models consistently achieved the highest accuracy scores across datasets.
- SGL and NCL appear to be the models that achieve balanced results across all metrics, suggesting that the self-supervised learning approach effectively captures both global and local structural information while preserving robustness.
- MLP-based models, frequently ranked just below graph-based models in accuracy. Deep-MF shows more stable performance over SimpleX.
- VAE-based models produced similar accuracy results among themselves, with MultiVAE offering slightly more stability.

- Traditional models exhibited the weakest accuracy scores, often close to random.
- According the diversity and fairness criterias, there is not any consensus. The most significant performance observed for traditional algorithms on Novelty. There is no significant difference for the other metrics.

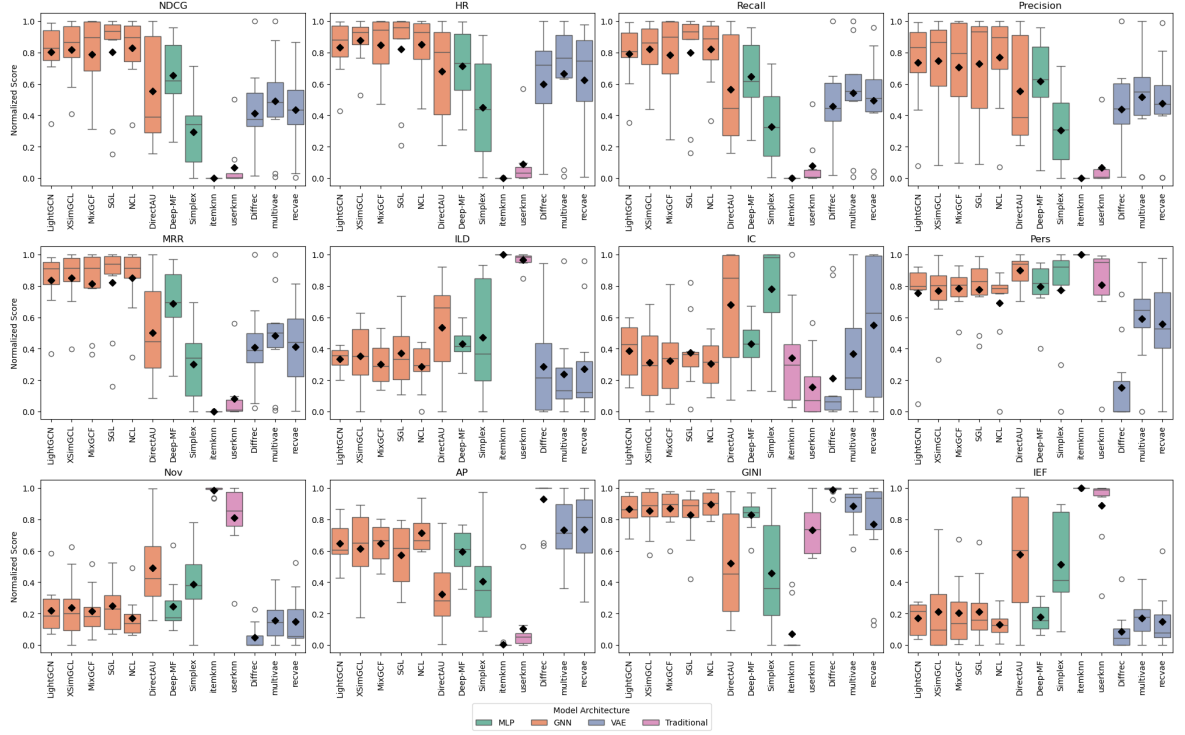


FIGURE 5.4 – Distribution of scaled model performances across evaluation metrics.

Each subplot corresponds to a different metric, showing the distribution of performance scores after min-max normalization within each dataset-metric group. Boxes represent normalized score ranges with the median marked as a horizontal line, and the mean is highlighted with a black diamond. Colors indicate the architectural category of each model.

5.3.2 Inter-Metric Relationships and Consistency (RQ2)

We presented the Pearson correlations of evaluation metrics across PCFM in Figure 5.5. A common pattern observed across all models is that accuracy metrics exhibit strong internal correlations. For example, in the NCL, the correlation between MRR and NDCG was calculated as 0.98. Among accuracy metrics, Recall showed the lowest correlation with the other four accuracy metrics. In the DirectAU, the Recall-MRR correlation was observed to be 0.6. The high correlations among accuracy metrics indicate a significant overlap in the aspects of recommendation quality these metrics capture. This redundancy implies that reporting every accuracy metric may be unnecessary, as they often reflect overlapping evaluation criteria. The lower correlation between Recall

and other accuracy metrics is notable and suggests that Recall captures a slightly different aspect of ranking performance making it a complementary metric to others like NDCG, MRR and Precision.

In contrast to accuracy metrics, fairness metrics exhibit significant variations in their internal correlations. While Gini and IEF consistently demonstrate a strong negative correlation across all models, the magnitude of this relationship varies substantially by model architecture. For instance, in the RecVAE model, the Gini-IEF correlation reaches a near-perfect negative value of -0.98, suggesting these metrics capture almost exactly opposing aspects of recommendation fairness. However, this relationship weakens considerably in the traditional models, where the correlation drops to -0.25, highlighting that fairness outcomes vary based on model architecture. Moreover, average popularity emerges as relatively independent from both Gini and IEF across all evaluated models, with correlation coefficients typically below ± 0.4 . This independence suggests that average popularity captures a distinct dimension of fairness not reflected by distribution-based metrics like Gini and IEF.

Diversity metrics generally exhibit positive correlations among themselves with most coefficients exceeding 0.5, indicating substantial alignment in how they measure recommendation variety. However, item coverage stands as a notable exception to this trend, displaying either negligible (near-zero) correlations with other diversity metrics across all evaluated models. Item coverage stands apart from other diversity metrics by focusing on catalog-level distribution instead of list-based composition, and it operates independently of the similarity calculations that underpin metrics like intra-list distance and personalization.

Regarding the relationship between accuracy and diversity, IC is generally positively correlated with accuracy-based metrics for many models. For example, the correlation between NDCG and IC for the XSimGCL was found to be 0.82. However, deviations from this trend exist; for instance, RecVAE exhibited a negative correlation of -0.41 between NDCG and item coverage. Furthermore, ILD demonstrated an inverse relationship with accuracy metrics. Specifically, the correlation between precision and ILD reached -0.76 for the SGL model, whereas XSimGCL exhibited a weaker relationship with a correlation of -0.08. When examining accuracy-novelty interactions, results indicate that autoencoder-based models tend to exhibit negative relationships, whereas other models do not. Similarly, personalization metrics generally show no significant

correlation with accuracy metrics, except for the SGL and SimpleX, which demonstrated a clear negative correlation (around -0.75) with accuracy metrics.

The relationship between accuracy and fairness metrics did not reveal a consistent trend across models. For example, in the DirectAU, the correlation between the Gini coefficient and MRR was -0.70, whereas the same correlation was only 0.09 for the SGL. AP exhibited a strong correlation with accuracy metrics in most models. For instance, in the DiffRec model, the correlation between AP and precision was found to be 0.79. However, exceptions were also observed; in the DirectAU, the correlation between AP and ILD was close to zero (-0.03). Finally, the relationship between diversity and fairness metrics did not exhibit a strong positive or negative correlation across models. The only notable trend was a negative correlation between the Gini coefficient and IC. For instance, the SGL exhibited a correlation of -0.92 between these two metrics, indicating a strong inverse relationship.

The measurements show that metric relationships vary substantially by model, rather than following a universal pattern. Our findings to answer RQ2 are summarized below :

- Accuracy metrics exhibit strong internal correlations, indicating overlap in what they measure.
- Fairness metrics capture distinct aspects of model behavior and show more complex, model-dependent relationships.
- Diversity metrics generally correlate positively with each other, except for item coverage, which behaves independently from other diversity measures.
- The relationship between accuracy, fairness, and diversity metrics shifts depending on the overall model architecture, suggesting that each type of model prioritizes different trade-offs.
- Improving one metric does not necessarily come at the expense of another.
- Individual recommendation models exhibit unique correlation patterns, reflecting how specific design choices influence metric interactions.

5.3.3 Model Dependency of Evaluation Metrics (RQ3)

We presented the Pearson correlation coefficients between accuracy metrics as a heatmap in Figure 5.6. Overall, accuracy metrics exhibit strong correlations. For each accuracy

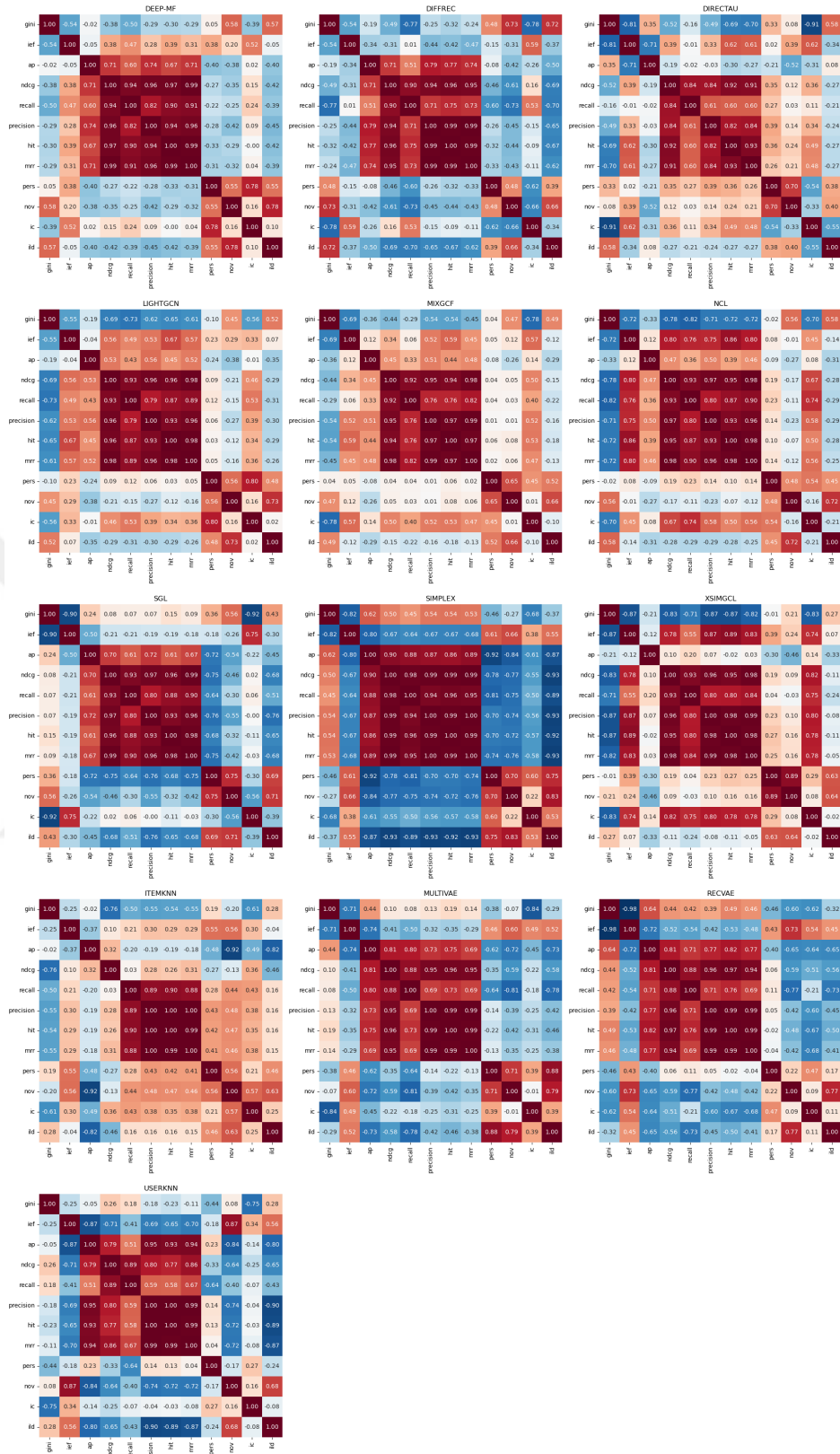


FIGURE 5.5 – Pearson Correlation matrices displaying metric relationships within individual recommendation models. Each heatmap represents a different recommendation model, showing the correlations between different evaluation metrics. Red cells indicate positive correlations while blue cells represent negative correlations.

metric, models generally show a high correlation above 0.8. This suggests that accuracy trends remain consistent regardless of the recommendation model used. Simplex, MultiVAE, RecVAE, UserKNN and ItemKNN deviate from this pattern. They exhibit lower correlations across all performance metrics compared to other models, indicating that these models respond differently to evaluation criteria.

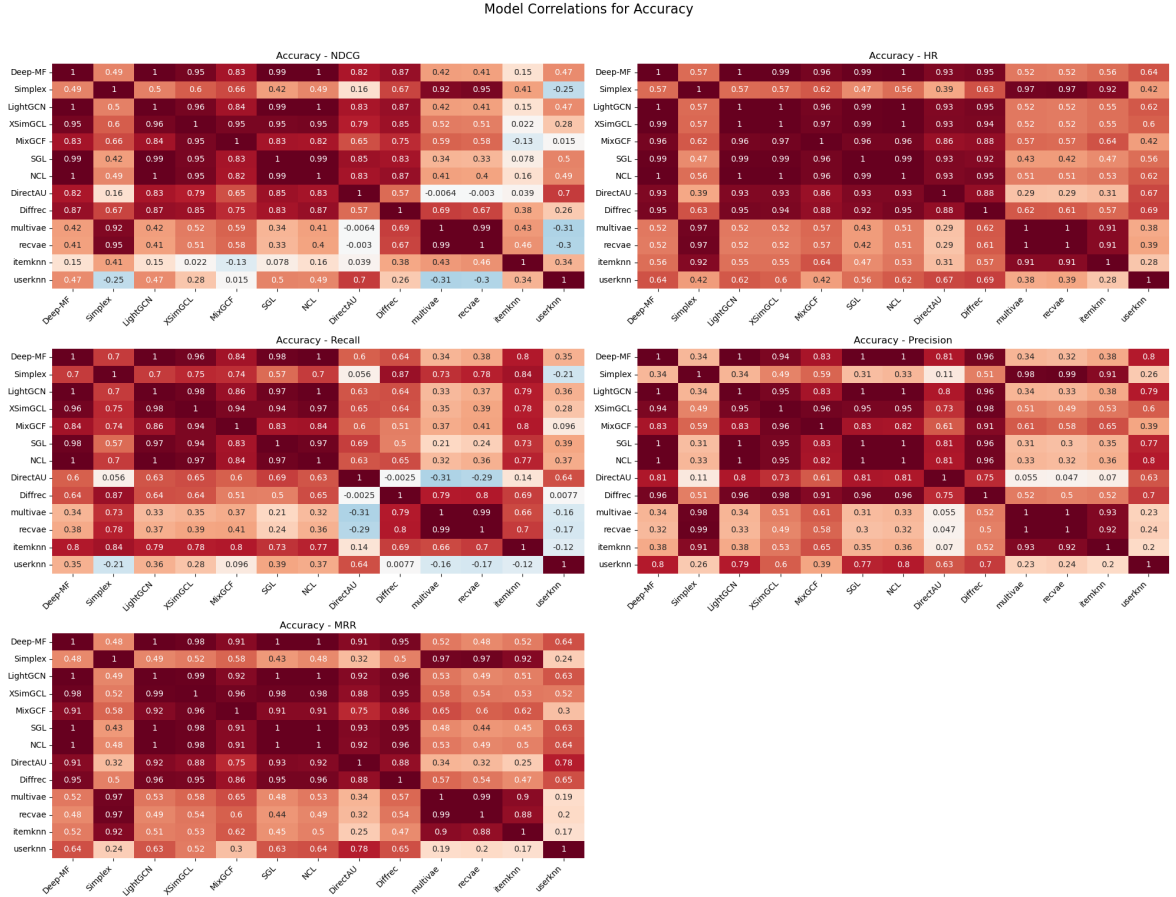


FIGURE 5.6 – Correlation matrices illustrating the Pearson correlations between performance metrics across recommendation models for accuracy measures (NDCG, HR, Recall, Precision, and MRR). Darker red represents stronger positive correlations, while darker blue indicates stronger negative correlations.

Figure 5.7 shows the correlation matrices for diversity metrics. For diversity metrics, no consistent trend is observed across all metrics. ILD and Novelty behave similarly to accuracy metrics, showing strong positive correlations of around 0.8 across different models. This indicates that these metrics exhibit low sensitivity to model choice. However, NCL and ItemKNN stands out as the most distinct model for these two metrics, with weak correlations generally below 0.5. In contrast, item coverage and personalization metrics do not follow a clear positive or negative correlation pattern. These metrics appear to be highly sensitive to model selection, making them less stable for cross-model comparisons. Traditional models show negative relationship between other

models in terms of the personalization evaluation.

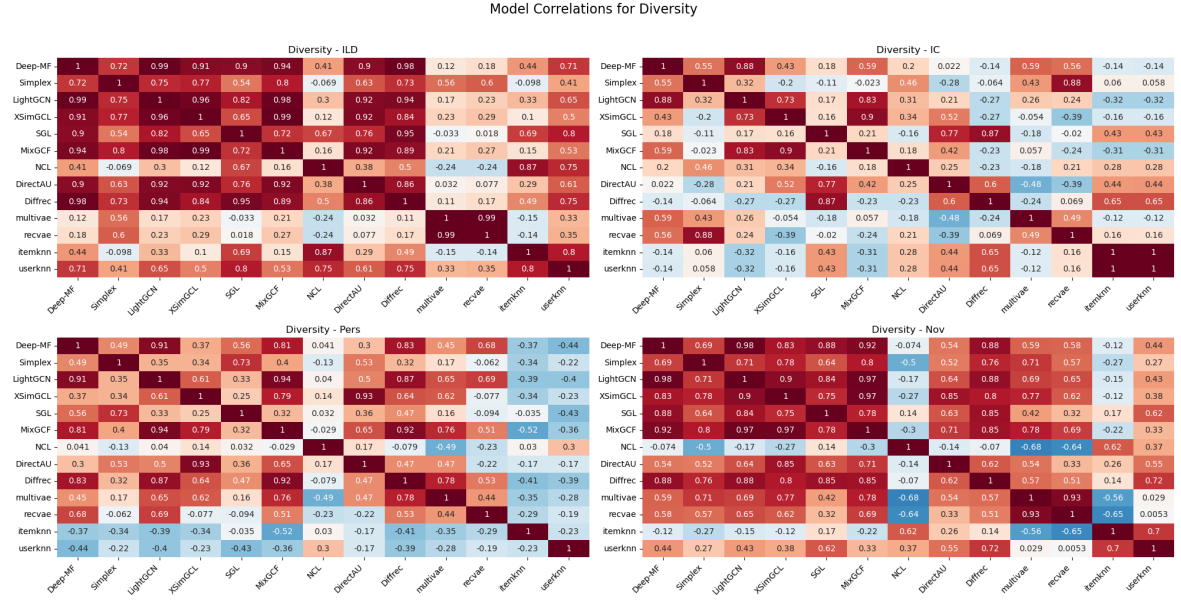


FIGURE 5.7 – Model correlations for diversity metrics across recommendation systems.

The heatmaps illustrate Pearson correlation coefficients between different recommendation models for four diversity metrics : Intra-List Distance (ILD), Item Coverage (IC), Personalization (Pers), and Novelty (Nov). Red indicates positive correlations while blue represents negative correlations, with color intensity corresponding to correlation strength.

For fairness metrics, as shown in Figure 5.8, item exposure fairness (IEF) exhibits inconsistent correlation patterns across models. While graph-based models and DeepMF generally demonstrate strong correlations with each other, SGL stands as a notable exception to this pattern. In contrast to the variable patterns seen with IEF, both average popularity (AP) and Gini coefficient metrics show more consistent correlation structures. Traditional models in particular display strong internal correlations for these metrics, with correlation values reaching up to 0.89. Graph-based models consistently exhibit positive correlations for both AP and Gini coefficient measurements, with some pairs like LightGCN and NCL achieving nearly perfect correlation (0.99) for AP. Among graph-based models, DirectAU is distinctive in showing only weak positive correlations with other models in terms of AP metric.

Figure 5.9 shows the mean correlation coefficients for each evaluation metric across models, including their minimum and maximum ranges. Metrics are color-coded by category (accuracy, diversity, and fairness). Metrics with lower correlations may indicate their sensitivity to algorithmic differences between models, while metrics with higher correlations suggest more consistency in how they rank models across different

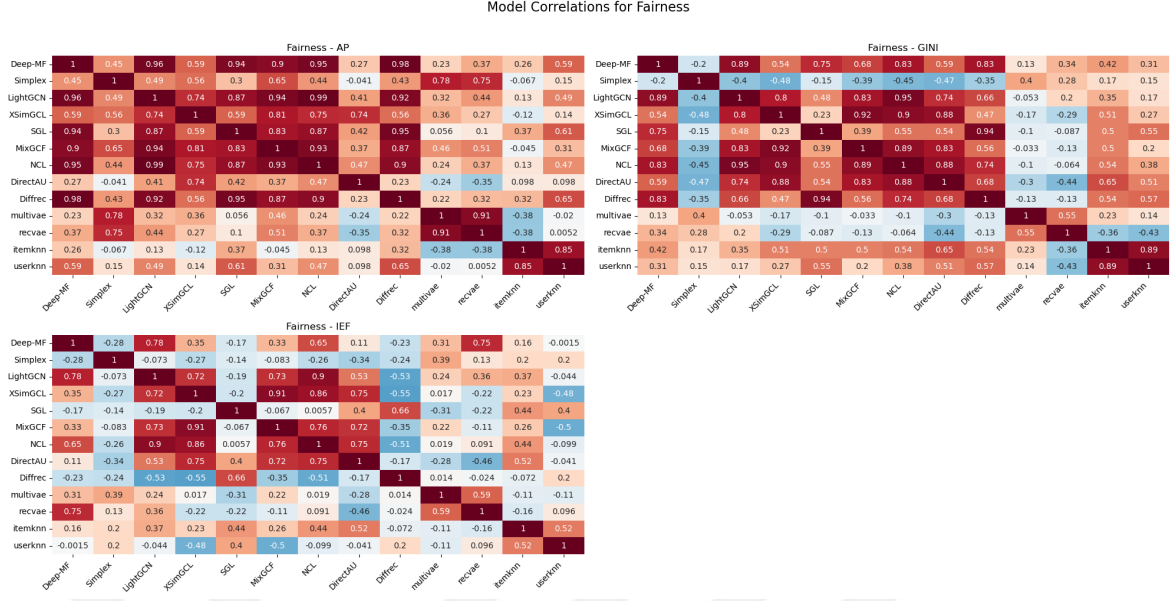


FIGURE 5.8 – Model correlations for fairness metrics across recommendation systems.

The heatmaps display Pearson correlation coefficients between different recommendation models for three fairness metrics : Average Popularity (AP), Gini Index (GINI), and Item Exposure Fairness (IEF). Strong positive correlations appear in dark red, while strong negative correlations are shown in dark blue, illustrating how consistently different models rank datasets on these fairness dimensions.

datasets. Among accuracy metrics, NDCG and Recall demonstrate lower cross-model correlations (around 0.6) compared to HR and Precision (approximately 0.75), though all accuracy metrics maintain correlations above 0.5. For diversity metrics, only ILD exhibits a correlation exceeding 0.5, while IC demonstrates the lowest correlation, potentially making it more responsive to model variations. Similarly, all fairness metrics show mean correlations below 0.5, with IEF having the lowest correlation (around 0.25).

To answer the RQ3, we investigated whether evaluation metrics produce consistent results across different models. The following points summarize our key findings based on model sensitivity and cross-model correlation patterns :

- Accuracy metrics are relatively stable across different models, showing consistently high correlations in most cases.
- Especially traditional and VAE-based models deviate from this trend and produce lower agreement among accuracy metrics.
- Diversity metrics show mixed sensitivity : while ILD and Novelty are generally stable, metrics like personalization and item coverage vary more with model type.
- Fairness metrics are the most model-sensitive, especially item exposure fairness,

which shows highly inconsistent patterns across models.

- Metric sensitivity depends not only on the metric itself but also on the nature of the model.
- Most fairness metrics and some diversity metrics have lower average cross-model correlation than accuracy metrics, confirming their sensitivity to model design.
- Researchers should consider metric robustness when comparing models, as some metrics may exaggerate or obscure performance differences based on the architecture.

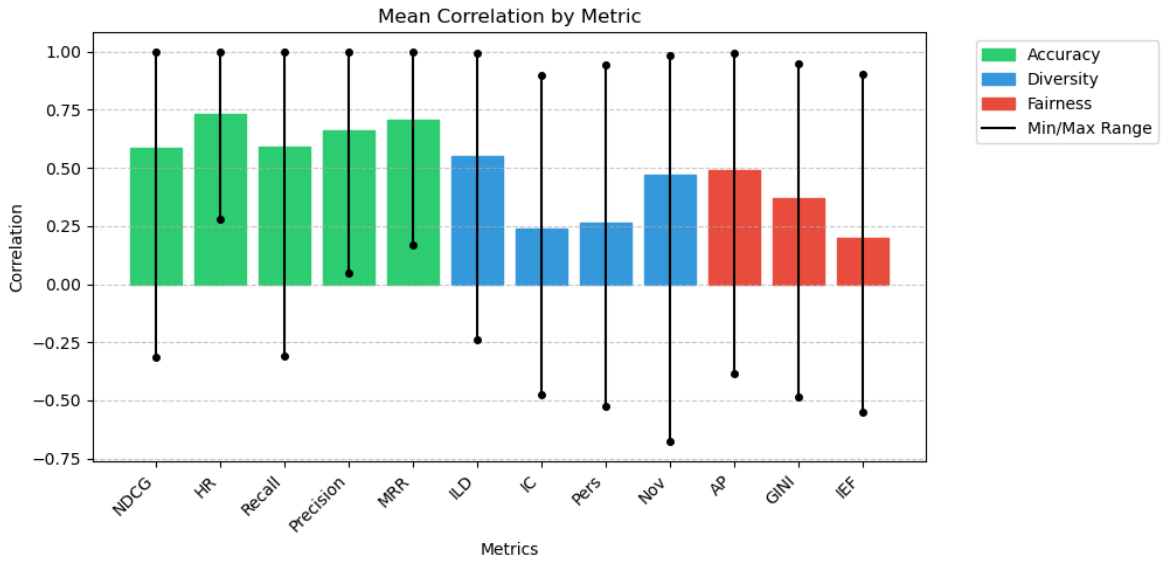


FIGURE 5.9 – Mean correlation analysis of evaluation metrics across recommendation models. Bar height indicates the average Pearson correlation coefficient for each metric when comparing between different models, with error bars showing minimum and maximum values. Metrics are color-coded by category (green for accuracy, blue for diversity, red for fairness). Higher correlation values suggest metrics that are more consistent across different model architectures, while lower values indicate metrics that better differentiate between models.

5.3.4 Impact of Graph Topology on Model Performance (RQ4)

Due to the large number of possible combinations, it is not feasible to visualize all interactions. Nevertheless, the scatter plots in Figure 5.10 illustrate selected examples of metric-topological feature relationships. Each point represents a different dataset and each color corresponds to a different recommendation model. The figure demonstrates that none of the examined performance metrics exhibit a strictly linear relationship with any topological feature. As seen in the figure, small changes in sparsity do not cause radical differences in NDCG for any recommendation model. However, once sparsity

exceeds a certain threshold, NDCG declines significantly. A similar pattern is observed in the relationship between NDCG and L3. While changes in L3 initially influence NDCG, beyond a certain point, further variations in L3 have little impact. We also observe that Gini and ILD metrics exhibit similar non-linear relationships with these two topological features.

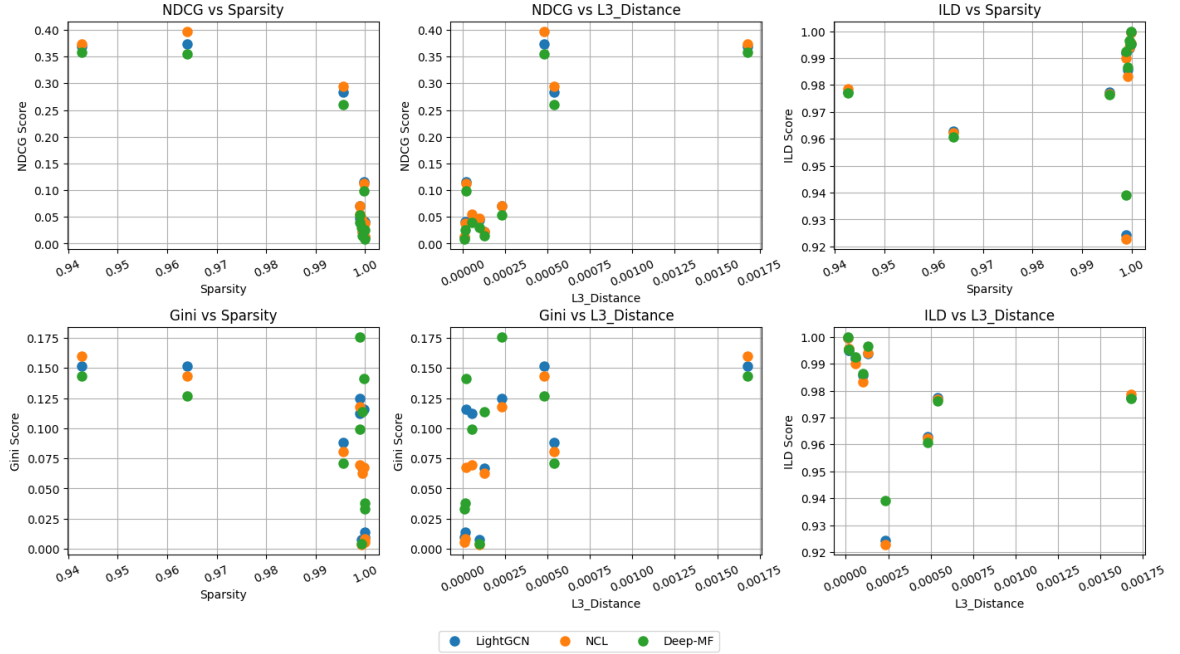


FIGURE 5.10 – Scatter plot illustrating the relationship between a performance metric and a topological feature. Each color represents a different recommendation model.

Almost all of the random forest models achieved more than 0.75 R^2 scores which points to the models' capability to explain recommendation performance across different metrics and PCFM (see APPENDIX B). Our analysis sought to determine whether recommendation model performance can be explained by topological features and to identify which features have the greatest impact. Figures 5.11, 5.12, and 5.13 present feature importance heatmaps obtained using MDI for accuracy, diversity, and fairness metrics, respectively.

The heatmaps for accuracy metrics shown in Figure 5.11 demonstrate that certain topological features consistently exhibited higher importance. Sparsity emerged as the most influential feature for accuracy metrics. This suggests that the density of user-item interactions strongly affects a model's ability to recommend relevant items. L3 Similarity also showed high importance for accuracy metrics, indicating that higher-order connectivity patterns impact ranking quality. In contrast to these general patterns, for the DirectAU model, user degree alpha values demonstrated greater importance than

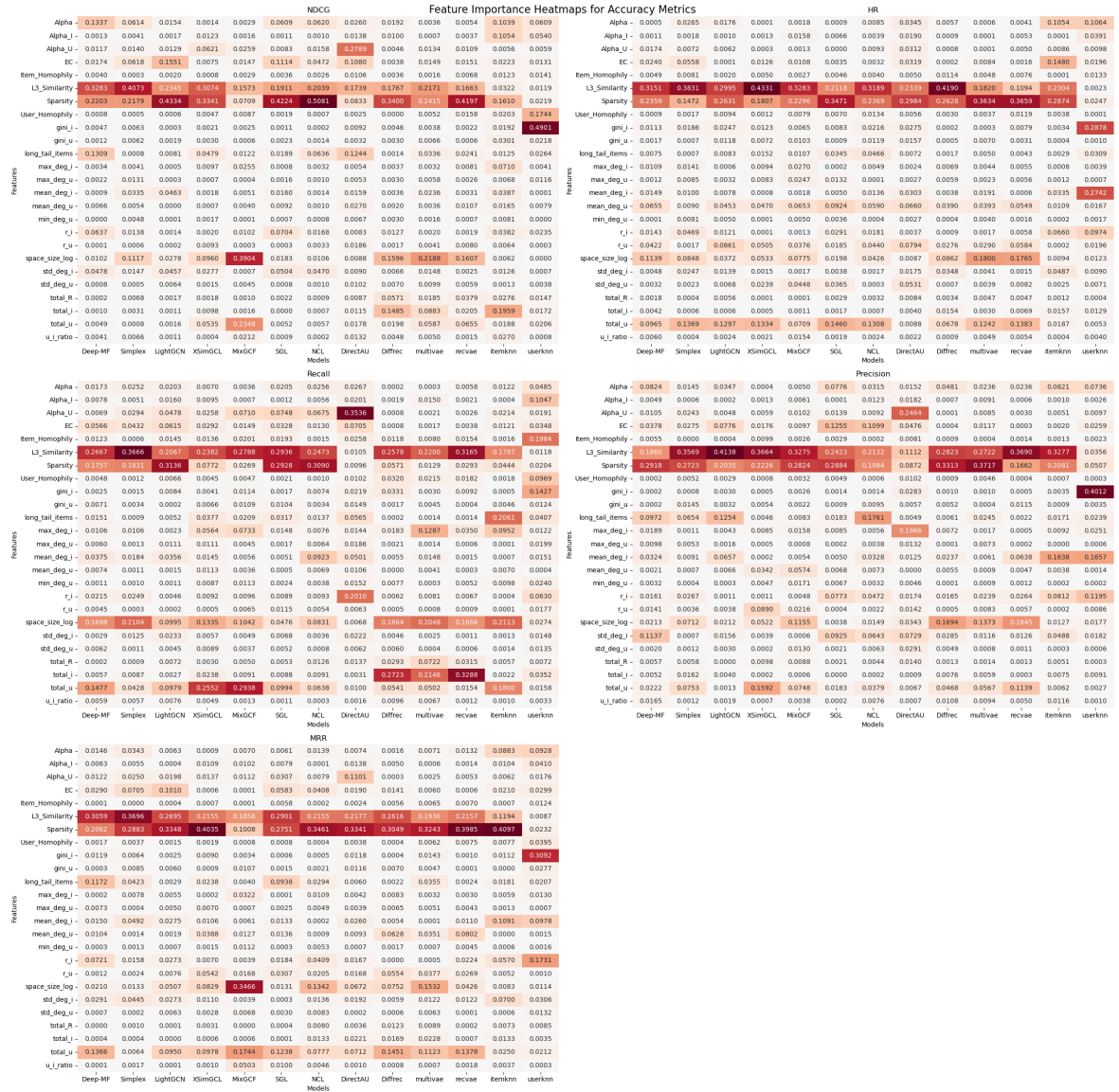


FIGURE 5.11 – MDI Feature importance heatmaps for accuracy metrics across different recommendation models. The heatmap shows the relative importance of various topological features (rows) for predicting NDCG, HR, Recall, Precision, and MRR performance for different recommendation algorithms (columns). Color intensity indicates feature importance, with darker red representing higher importance values.

Sparsity or L3 Similarity specifically for NDCG, Recall, and Precision metrics. This distinct pattern suggests that DirectAU’s performance is more heavily influenced by the distribution of user activity levels than by overall graph density or higher-order connectivity.

For diversity metrics illustrated in Figure 5.12, we observed that there is no common pattern across models. However, total user, item and interaction counts and the degree distributions proved more effective than sparsity and L3 similarity measurements in predicting diversity outcomes. For intra-list distance specifically, gini-item coefficient emerged as particularly important. When items follow a more unequal popularity distribution, recommendation algorithms may struggle to include diverse items in the same list, as they tend to favor either very popular or very niche items rather than a balanced mix. For item coverage, all degree-related features demonstrated similar levels of importance across different models, indicating that the distribution of connections in the user-item graph uniformly affects how many unique items get recommended. When examining personalization, degree alpha values showed greater importance than other features, suggesting that the power-law characteristics of user and item degree distributions particularly influence how tailored recommendations are to individual users. For novelty, total item count clearly emerged as the most important predictor.

The relationship between topological features and fairness metrics as demonstrated in Figure 5.13 revealed that there is no common pattern among the fairness metrics. For AP, standard deviation of item degrees and sparsity values appear more important. For the GINI, each model is affected by different topological features. Overall, eigenvector centrality, item homophily, L3 similarity, space size, and total item count seem important for different models separately. These variations are also observed for item exposure fairness, where each recommendation model focuses on different topological features. However, user-based features generally demonstrate importance across models, such as total user count, standard deviation of user degrees, or total user interactions, in addition to item homophily. This diversity in influential features suggests that fairness in recommendation systems is highly dependent on the specific algorithm used, with no single topological characteristic serving as a universal predictor of fair recommendations.

Our findings on feature importance values summarized as follows :

- Sparsity emerged as the most influential topological feature for accuracy metrics

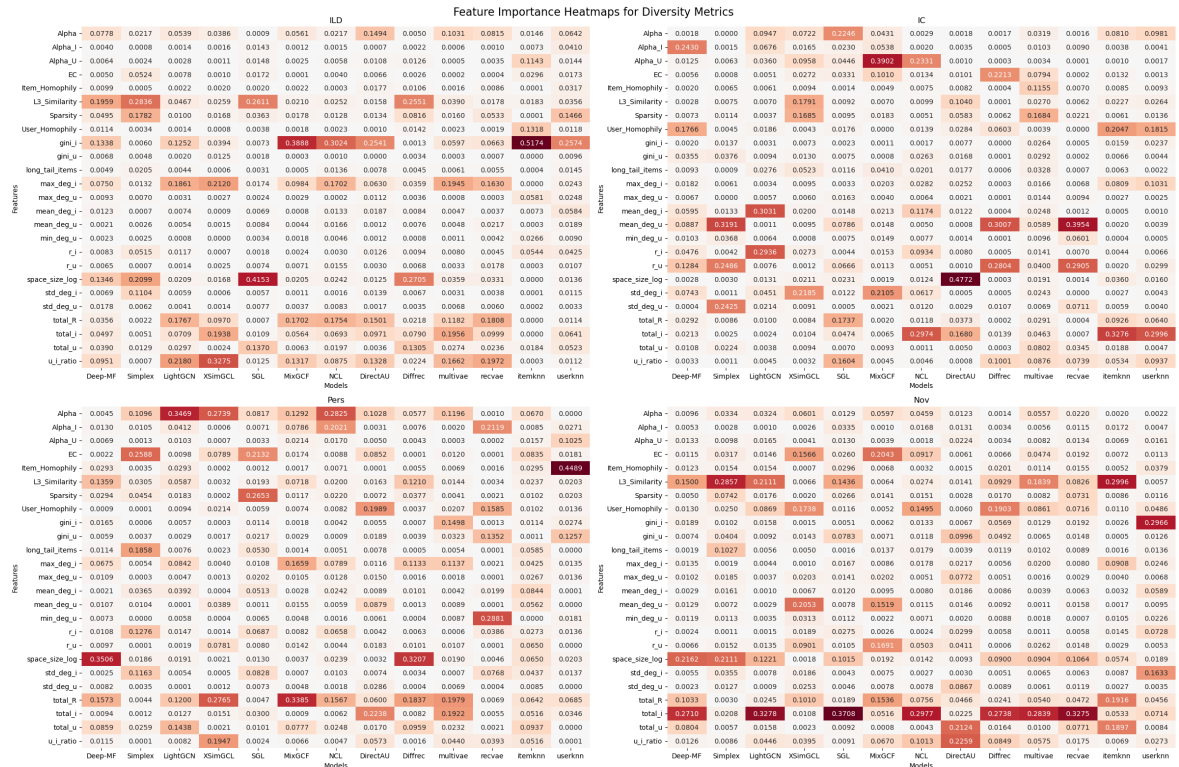


FIGURE 5.12 – MDI Feature importance heatmaps for diversity metrics across different recommendation models. The heatmap illustrates the importance of topological features (rows) for predicting ILD, IC, Personalization, and Novelty metrics for different recommendation algorithms (columns). Darker red cells indicate higher feature importance for the corresponding model-metric pair.

such as NDCG and Recall, indicating that denser user-item graphs generally lead to better ranking quality.

- L3 similarity also contributed notably to accuracy, showing that higher-order structural patterns improve recommendation precision in many models.
- There is no common important feature for both diversity and fairness metrics. Each metric is effected by different topological features.
- In predicting intra-list diversity (ILD), item degree inequality (Gini- i) stood out as an important factor, suggesting that skewed popularity distributions reduce list-level diversity.
- Personalization was more sensitive to the power-law exponents, implying that the shape of the degree distribution influences how tailored the recommendations are.
- The total number of items in a dataset had the strongest impact on novelty. Additionally, L3 Similarity has high impact on Novelty.
- While the importance of topological features for diversity metrics varies across models, features such as item/user homophily, sparsity, and L3 similarity gene-

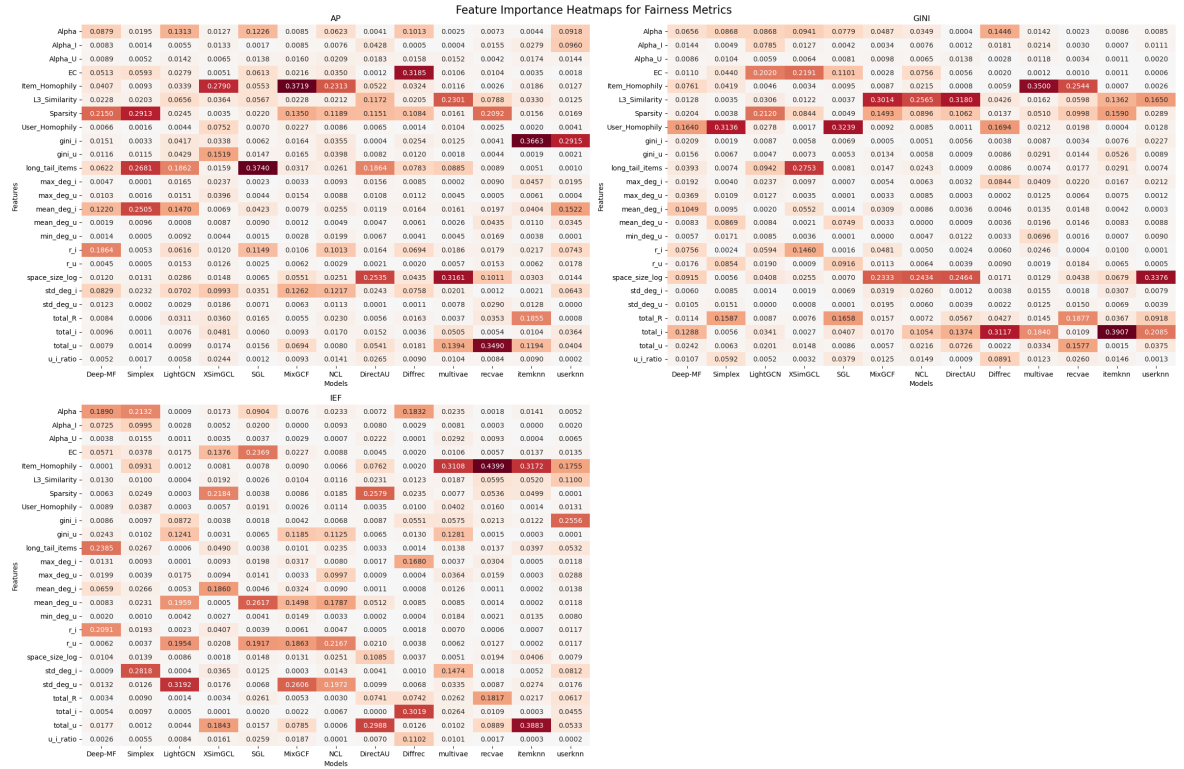


FIGURE 5.13 – MDI feature importance heatmaps for fairness metrics across different recommendation models. The heatmap displays the importance of topological features (rows) for predicting AP, GINI, and IEF metrics for different recommendation algorithms (columns). Higher importance values are represented by darker red coloration.

rally have the highest influence, indicating that both local similarity patterns and broader structural properties contribute to how diverse the recommendations are.

- Across all metric groups, we found that no single topological feature dominates universally. Their impact varies depending on the metric and model architecture. This implies that modifying these features through preprocessing could enhance the quality of recommendations in terms of desired metric.

Besides the feature importance values, we observed positive relationship between L3 Similarity and all accuracy metrics while ILD effected positively from high space size and total interactions. Additionally, HR comes higher if standard deviation of item degrees are higher. For the negative relationships, we observe lower accuracy results with the higher sparsity and space size values. Moreover, ILD generally comes lower when the datasets have high user-item ratio.

5.3.5 Complementary Statistical Analysis

In addition to the main analyses conducted to address our research questions, we performed a complementary statistical evaluation to validate whether PCFM exhibit significant performance variation across evaluation metrics. This analysis builds on the candlestick boxplots presented in Figure 5.14, where each subplot corresponds to a different model and displays the distribution of that model’s performance scores across various evaluation metrics. The line inside each box represents the median score, while the black diamond indicates the mean score. In the figure, noticeable differences in distribution and central tendency across metrics for each model visually suggest potential statistical variation.

To formally test this observation, we applied a one-way ANOVA test separately for each model, where the evaluation metric served as the independent variable and the model’s performance scores as the dependent variable. The analysis aimed to determine whether the mean performance significantly varied across different metrics. For each test, the degrees of freedom were given by $df_{\text{between}} = k - 1$ and $df_{\text{within}} = N - k$, where k is the number of metrics ($k = 12$), and N is the total number of performance observations for that model ($N = 120$). Thus, each ANOVA test was conducted with $df_{\text{between}} = 11$ and $df_{\text{within}} = 108$. As expected, the results yielded statistically significant differences across metrics for all models, with all p -values below 0.001.

To better understand which pairs of metrics were responsible for these differences, we performed a post-hoc Tukey test. The results were broadly consistent with the findings discussed in RQ2 and RQ3, although from a different methodological perspective. As shown in the RQ2 analysis, accuracy metrics exhibited high internal correlations—suggesting redundancy. The Tukey analysis corroborates this by showing that nearly all pairwise comparisons among accuracy metrics (e.g., NDCG, HR, Recall, Precision, MRR) were statistically non-significant across models. This reinforces the idea that accuracy metrics measure highly overlapping aspects of model performance.

The post-hoc analysis also echoes the insights of RQ3, which examined the consistency of metric-based model rankings. Metrics that ranked models similarly across experiments—such as MRR and NDCG—also showed no statistically meaningful difference in their score distributions. On the other hand, metrics that yielded inconsistent rankings across models (e.g., IC, IEF) also exhibited statistically significant gaps when compared

to accuracy metrics. This alignment supports the view that both score magnitude and ranking behavior of evaluation metrics vary by type and role in evaluation. Furthermore, the analysis showed that differences between accuracy metrics and diversity or fairness metrics were consistently significant ($p < 0.001$), highlighting the multidimensional nature of recommender system evaluation. Even within diversity and fairness categories, pairwise comparisons often revealed significant differences, indicating that these metrics are not interchangeable.

In summary, this analysis confirms and strengthens findings by providing statistical evidence that metric selection has a measurable impact on evaluation outcomes. It further justifies the use of a diverse set of metrics to capture different aspects of recommendation quality and warns against over-relying on any single evaluation criterion.

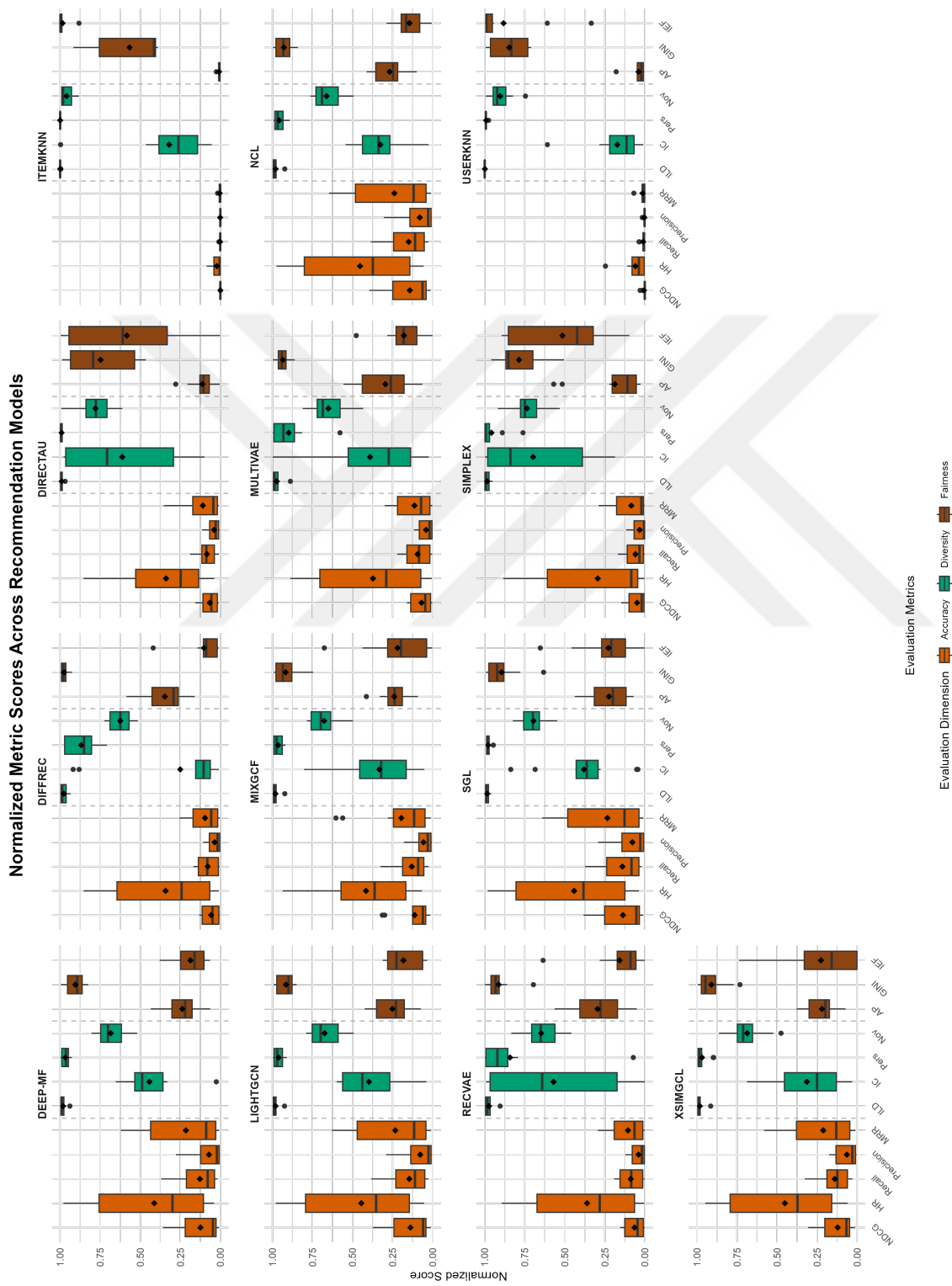


FIGURE 5.14 – Boxplots of normalized metric scores for each recommendation model. Each subplot represents a model, and the boxplots within each show the distribution of scores across datasets for 12 different evaluation metrics. Metrics are grouped by color to reflect evaluation dimensions : accuracy (red), diversity (green), and fairness (brown).

6 CONCLUSION

In modern digital platforms, such as e-commerce websites or online streaming services, recommender systems play a crucial role for both companies and customers (Rao et al., 2024). Among the variety of available items, a good recommender system should suggest proper items that increase user interaction (Abdalla, 2020). A successful recommender system should produce a recommendation set that contains items relevant to the user’s historical interests to enhance user interaction. Besides, the system should recommend a range of options that might engage the user’s curiosity or expose them to new possibilities (Fu and Niu, 2023). While prioritizing the most similar items to the user’s preferences is generally termed *accuracy*, the extent to which the recommended items differ from one another is explored under the concept of *diversity*. In addition to these two terms, which mainly focus on user satisfaction, digital platforms must also consider the visibility of new items. As newly added items have fewer interactions than the older and popular ones, many recommendation systems may tend to ignore them (Panteli and Boutsinas, 2023b; Klimashevskaja et al., 2024). This bias against newer or less popular items is called *item fairness*. It’s better for each platform to provide fair recommendations to make them attractive to new vendors. There are various recommendation system performance evaluation (RSPE) metrics that analyze the recommender system from the perspectives of accuracy, diversity, and fairness separately.

Although accuracy, diversity, and fairness are all essential in the design of recommender systems, there is still no comprehensive study that examines the relationships and trade-offs among these evaluation dimensions. At the same time, real-world datasets vary significantly in their topological features, yet the effect of these variations on model performance remains underexplored too. The literature lacks a clear understanding of the effect of the topological features on evaluation metrics. Recent studies, such as Zhao et al. (Zhao et al., 2025), have investigated fairness and diversity relationships, while Isufi et al. (Isufi et al., 2021) explored accuracy and diversity, but both remained limited in dataset and model variety. Malitesta et al. (Malitesta et al., 2023) examined topological effects on graph-based recommenders, yet focused mainly on accuracy and lacked deeper metric diversity. Dietz et al. (Dietz et al., 2025) introduced fairness and diversity metrics in user-location recommendations, but only over a single dataset,

limiting generalizability. In this thesis, we aimed to fill these gaps. We conducted one of the most detailed experimental evaluations to date, systematically examining the possible relationships between algorithms, topological features, and evaluation metrics from different dimensions.

We present several key contributions to address the mentioned thesis goals. The first major contribution of this thesis is the development of a comprehensive taxonomy to group the personalized collaborative filtering models (PCFM). We segment the PCFM based on their architectural principles and learning strategies. Second, we provide a comprehensive ranking of the PCFM across the RSPE metrics and datasets. The third contribution of this thesis is the systematic analysis of the coherence between RSPE metrics. As the fourth contribution, we investigate the relationship between recommendation performance and topological features through a comprehensive experimental approach.

To achieve the listed contributions, we design a tailored experimental approach. We determined 13 popular PCFM with diverse strategies. Our taxonomy is constructed based on the core architectural principles of these models. We first categorized the PCFM into two primary categories : Neural network-based and non-neural network-based (ItemKNN and UserKNN) models. Whereas non-neural network-based models have no further subcategories, we organized neural network-based models into three subcategories : Variational Autoencoder-based (RecVAE, MultiVAE and DiffRec), Graph-based, and Multi-layer Perception-based (Deep Matrix Factorization and SimpleX) models. Further, Graph-based models are divided into two categories : non-constractive learning (LightGCN) and constractive learning-based models. In the last sub-layer, models that use constractive learning are split into two : graph augmentation-based (SGL and MixGCF) and explicit constractive learning-based (XSimGCL, NCL, DirectAU) ones.

We used 12 well known RSPE metrics to measure the recommendation quality from three different aspects : accuracy (NDCG, recall, precision, mean reciprocal rank and hit rate), diversity (intra list distance, item coverage, personalization, and novelty), and fairness (average popularity, GINI, and item exposure). We evaluated the performance of the recommendation algorithms across 10 different real-world datasets. In this thesis we consider only the collaborative information which means user-item interactions without incorporating any user or item attributes.

Firstly, we rank of the PCFM per RSPE metrics. We observe that, although there is not

a single model that dominates all evaluation metrics, graph-based models perform best in terms of accuracy and able to balance results on different evaluation perspectives better than other model architectures. Especially, SGL and NCL achieve the most balanced results which shows the effectiveness of contrastive learning approaches. Besides, traditional models can not capture the complex relationships in user-item interaction data, often resulting in performance close to random suggestions compared to more sophisticated models. Additionally, MLP and VAE-based models demonstrate moderate to high accuracy. Although Deep-MF is one of the oldest and simplest approaches, it outperforms many of the more recent models. Moreover, MultiVAE, another earlier model, delivers more balanced results compared to other VAE-based models. We present a comprehensive picture of performance differences resulting from architectural differences according to topological variations by assessing these models on a range of real-world datasets in terms of accuracy, diversity, and fairness. Our findings fill a gap in the literature by providing a strong understanding of models' rankings. In addition to highlighting performance patterns, this analysis identifies recurrent strengths and shortcomings among model families.

Second, we examined the correlations between the metrics. Our analysis shows that while some metrics exhibit strong correlations, others reveal potential trade-offs. Accuracy metrics exhibit strong internal correlations, pointing to a major overlap in what they measure. Among accuracy metrics, Recall shows the lowest correlation with other accuracy metrics which indicates that it captures a distinct aspect of recommendation performance. This makes Recall a complementary measure. Diversity metrics generally correlate positively with each other, except for item coverage, which behaves independently. Fairness metrics demonstrate more complex, model-dependent relationships. Instead observing a single pattern, correlations among fairness metrics vary according to chosen model. Furthermore, this research underscores that the relationships between accuracy, diversity, and fairness metrics vary based on the model architecture, instead of universal trends. The common assumption that improving one metric necessarily reduces performance in another is not universally true. These results suggest that characterizing a recommender system using a single diversity or fairness metric is insufficient. Most metrics capture distinct aspects of recommendation behavior and often produce differing model rankings, indicating that they are not interchangeable and should be interpreted in context.

Third, we examined the correlation of model performance per metric. We observed that

the accuracy metrics behave similar trends regardless of the recommendation model used except the non-deep learning based models and VAE-based models. That models show lower correlations which indicates they respond differently. In terms of the diversity and fairness metrics, there is no consistent trends. Except for metrics other than ILD, Novelty and AP, the models did not show a regular positive or negative correlation. We observed these three metrics generally shows strong positive correlations within each PCFM which indicates their behavior are more consistent across different models.

Finally, we take advantage of explainable AI to observe the effect of topological features on model performance. In this task, we first trained separate random forest models for each recommendation model-RSPE metric pair. We used topological features as dependent variables and tried to predict metric results. Then, we analysed MDA based feature importance of each topological feature. As a result, on one hand, we observed common feature importance patterns for each recommendation model on accuracy metrics. In accuracy metrics, sparsity emerged as the most influential topological feature, with denser user-item interaction generally leading to better ranking quality. L3 similarity also played a significant role, demonstrating that higher-order structural patterns improve recommendation precision. On the other hand, we observed that there is no consensus on the feature importance explainability on diversity and fairness. Nevertheless, item degree inequality (Gini- i), total item count, user-item ratio and Alpha values seem more effective than other features for diversity metrics. Besides, item homophily, sparsity, L3 similarity, total item and user counts are the key features for fairness metrics.

As a result, the evaluation and selection practices of recommendation systems today need to be seriously questioned. Many newly developed algorithms are published after being tested with limited datasets and limited metrics. We have shown that model performances can vary depending on the datasets used. In the study, inconsistencies between metrics, dependence of models on topological features and significant variation in performance across datasets are systematically demonstrated. Our findings strongly indicate that it is not reliable to declare the superiority of models based on only a few benchmark results. At the same time, the study emphasizes that widely used accuracy-oriented evaluation approaches ignore critical dimensions such as diversity and fairness, and this deficiency may hinder model development processes. In addition, although various evaluation metrics are commonly categorized as fairness and diversity metrics in

the literature, we did not observe consistent patterns or generalizable behaviors across all models. Previous studies, often constrained by limited experimental scales, have reported that different metrics appear coherent or that clear trade-offs exist between them. However, our comprehensive experiments show that such coherence or trade-offs do not consistently hold across models or datasets. Similarly, we have shown that the effect of topological features change accross different models and also different metrics. The thesis open another discussion whether the common evaluation method used in the literature is reliable.

This work can be expanded in future research by including information from sources other than binary user-item interactions. The observed model performance rankings, might be validated by incorporating external node or edge features, such as item metadata, user demographics, or contextual data. Additionally, building artificial graphs with controlled structural characteristics is a possible direction for future work. By generating networks that target specific graph features, it would be possible to design controlled experiments that further refine the findings of this thesis. Such experiments would help isolate the conditions under which different models succeed or fail, offering a deeper understanding of their robustness and limitations. In addition, using a large number of artificial datasets would increase the number of samples for each model. This could make it possible to train a machine learning model that learns the relationship between graph topologies and model performances more accurately. A trained model could then be used to predict the most suitable recommendation algorithm for a given dataset based solely on its topological features. Finally, future research could concentrate on creating meta-evaluation frameworks that identify, analyze, and resolve these disparities in light of the frequent inconsistencies among evaluation metrics. This would improve the dependability and interpretability of model comparisons.

REFERENCES

- Abdalla, A. (2020). Using recommendation systems to recommend relevant items, *Fourth Int. Conf. Basic Sci. their Appl.*, no.
- Abdollahpouri, H., Mansoury, M., Burke, R., Mobasher, B. and Malthouse, E. (2021). User-centered evaluation of popularity bias in recommender systems, *Proceedings of the 29th ACM conference on user modeling, adaptation and personalization*, pp. 119–129.
- Adomavicius, G. and Zhang, J. (2012). Impact of data characteristics on recommender systems performance, *ACM Trans. Manage. Inf. Syst.* **3**(1).
URL: <https://doi.org/10.1145/2151163.2151166>
- Albalawi, A. and Alharbi, S. T. (2023). E-commerce recommender systems : A comprehensive survey, *2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)*, IEEE, pp. 503–509.
- Althbiti, A., Alshamrani, R., Alghamdi, T., Lee, S. and Ma, X. (2021). Addressing data sparsity in collaborative filtering based recommender systems using clustering and artificial neural network, *2021 IEEE 11th annual computing and communication workshop and conference (CCWC)*, IEEE, pp. 0218–0227.
- Anand, P. B. and Nath, R. (2020). Content-based recommender systems, *Recommender system with machine learning and artificial intelligence : Practical tools and applications in medical, agricultural and other industries* pp. 165–195.
- Anelli, V. W., Deldjoo, Y., Di Noia, T., Di Sciascio, E., Ferrara, A., Malitesta, D. and Pomo, C. (2022). How neighborhood exploration influences novelty and diversity in graph collaborative filtering., *MORS@ RecSys* **3268**.
- Anelli, V. W., Malitesta, D., Pomo, C., Bellogin, A., Di Sciascio, E. and Di Noia, T. (2023). Challenging the myth of graph collaborative filtering : a reasoned and reproducibility-driven analysis, *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, Association for Computing Machinery, New York, NY, USA, p. 350–361.
URL: <https://doi.org/10.1145/3604915.3609489>

- Antikacioglu, A., Bajpai, T. and Ravi, R. (2019). A new system-wide diversity measure for recommendations with efficient algorithms, *SIAM Journal on Mathematics of Data Science* **1**(4) : 759–779.
- Arora, K., Bali, V. and Singh, S. (2017). Recommendation systems : A review report., *International Journal of Advanced Research in Computer Science* **8**(7).
- Beattie, L., Taber, D. and Cramer, H. (2022). Challenges in translating research to practice for evaluating fairness and bias in recommendation systems, *Proceedings of the 16th ACM Conference on Recommender Systems*, pp. 528–530.
- Betello, F., Purificato, A., Siciliano, F., Trappolini, G., Bacciu, A., Tonellotto, N. and Silvestri, F. (2024). A reproducible analysis of sequential recommender systems.
URL: <https://arxiv.org/abs/2408.03873>
- Bhat, S. S., P, P., K V, S., Raghunandan, A. and Mohan, B. R. (2022). Comparative performance evaluation of web-based book recommender systems, *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 985–991.
- Bing, R., Yuan, G., Zhu, M., Meng, F., Ma, H. and Qiao, S. (2023). Heterogeneous graph neural networks analysis : a survey of techniques, evaluations and applications, *Artificial Intelligence Review* **56**(8) : 8003–8042.
- Bux, T. U., Varshney, B. et al. (2022). Role of collaborative filtering in recommendation systems, *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, IEEE, pp. 1–4.
- Cameron, P. J., Ajith, M. V. et al. (2023). Hypergraphs defined on algebraic structures, *arXiv preprint arXiv :2303.00546* .
- Canale, E. A. (2022). From weighted to unweighted graphs in synchronizing graph theory, *arXiv preprint arXiv :2209.06362* .
- Çano, E. and Morisio, M. (2017). Hybrid recommender systems : A systematic literature review, *Intelligent data analysis* **21**(6) : 1487–1524.
- Cao, Y., Yang, L., Wang, C., Liu, Z., Peng, H., You, C. and Yu, P. S. (2023). Multi-task item-attribute graph pre-training for strict cold-start item recommendation, *Proceedings of the 17th ACM Conference on Recommender Systems*, pp. 322–333.
- Carnovalini, F., Rodà, A. and Wiggins, G. A. (2025). Popularity bias in recommender systems : The search for fairness in the long tail, *Information* **16**(2) : 151.

- Castells, P. and Moffat, A. (2022). Offline recommender system evaluation : Challenges and new directions, *AI magazine* **43**(2) : 225–238.
- Chen, Z., Silvestri, F., Wang, J., Zhang, Y. and Tolomei, G. (2023). The dark side of explanations : Poisoning recommender systems with counterfactual examples, *Proceedings of the 46th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 2426–2430.
- Chia, E. J. and Najafabadi, M. K. (2022). Solving cold start problem for recommendation system using content-based filtering, *2022 International Conference on Computer Technologies (ICCTech)*, IEEE, pp. 38–42.
- Deldjoo, Y., Anelli, V. W., Zamani, H., Bellogín, A. and Di Noia, T. (2019). Recommender systems fairness evaluation via generalized cross entropy, *arXiv preprint arXiv :1908.06708* .
- Deldjoo, Y., Bellogin, A. and Di Noia, T. (2021). Explaining recommender systems fairness and accuracy through the lens of data characteristics, *Information Processing Management* **58**(5) : 102662.
URL: <https://www.sciencedirect.com/science/article/pii/S0306457321001503>
- Deshpande, M. and Karypis, G. (2004). Item-based top-n recommendation algorithms, *ACM Transactions on Information Systems (TOIS)* **22**(1) : 143–177.
- Dietz, L. W., Sánchez, P. and Bellogín, A. (2025). Understanding the influence of data characteristics on the performance of point-of-interest recommendation algorithms, *Information Technology & Tourism* pp. 1–50.
- Duricic, T., Kowald, D., Lacic, E. and Lex, E. (2023). Beyond-accuracy : a review on diversity, serendipity, and fairness in recommender systems based on graph neural networks, *Frontiers in big data* **6** : 1251072.
- Dwivedi, D. N. and Mahanty, G. (2025). Leveraging personalized ai recommendations to enhance user experience in streaming services (ott platform) : An empirical evaluation of recommendations, *Transforming Cinema with Artificial Intelligence*, IGI Global Scientific Publishing, pp. 265–290.
- Fan, Z., Xu, K., Dong, Z., Peng, H., Zhang, J. and Yu, P. S. (2023). Graph collaborative signals denoising and augmentation for recommendation, *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pp. 2037–2041.

- Fang, J., Liu, D. and Tse, C. K. (2023). Impact of structure of network based data on performance of graph neural networks, *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5.
- Fareed, A., Hassan, S., Belhaouari, S. B. and Halim, Z. (2023). A collaborative filtering recommendation framework utilizing social networks, *Machine Learning with Applications* **14** : 100495.
- Fayyaz, Z., Ebrahimian, M., Nawara, D., Ibrahim, A. and Kashef, R. (2020). Recommendation systems : Algorithms, challenges, metrics, and business opportunities, *applied sciences* **10**(21) : 7748.
- Fei, Z., Wang, J., Liu, K., Attahi, E. and Huang, B. (2023). Deep feature fusion-based stacked denoising autoencoder for tag recommendation systems, *IET Cyber-Systems and Robotics* **5**(3) : e12095.
- Fu, Z. and Niu, X. (2023). Modeling users' curiosity in recommender systems, *ACM Transactions on Knowledge Discovery from Data* **18**(1) : 1–23.
- Gao, C., Zheng, Y., Li, N., Li, Y., Qin, Y., Piao, J., Quan, Y., Chang, J., Jin, D., He, X. et al. (2023). A survey of graph neural networks for recommender systems : Challenges, methods, and directions, *ACM Transactions on Recommender Systems* **1**(1) : 1–51.
- Gasmi, I., Seridi-Bouchelaghem, H., Hocine, L. and Abdelkarim, B. (2015). Collaborative filtering recommendation based on dynamic changes of user interest, *Intelligent Decision Technologies* **9**(3) : 271–281.
- Giamphy, E., Guillaume, J.-L., Doucet, A. and Sanchis, K. (2023). A survey on bipartite graphs embedding, *Social Network Analysis and Mining* **13**(1) : 54.
- Godinot, A. and Tarissan, F. (2023). Measuring the effect of collaborative filtering on the diversity of users' attention, *Applied Network Science* **8**(1) : 9.
- Goethals, S., Calders, T. and Martens, D. (2024). Beyond accuracy-fairness : Stop evaluating bias mitigation methods solely on between-group metrics.
- Greenwood, S., Chiniah, S. and Garg, N. (2024). User-item fairness tradeoffs in recommendations, *Advances in Neural Information Processing Systems* **37** : 114236–114288.

- Guan, N. (2024). Machine learning algorithms in the development of recommender systems, *Applied and Computational Engineering* **51**(1) : 119–128.
- Gunathilaka, T. M. A. U., Manage, P. D., Zhang, J., Li, Y. and Kelly, W. (2025). Addressing sparse data challenges in recommendation systems : A systematic review of rating estimation using sparse rating data and profile enrichment techniques, *Intelligent Systems with Applications* p. 200474.
- Halabaku, E. and Bytyçi, E. (2024). Overfitting in machine learning : A comparative analysis of decision trees and random forests, *Intelligent Automation and Soft Computing* **39**(6) : 987–1006.
- Hao, B., Zhang, J., Yin, H., Li, C. and Chen, H. (2021). Pre-training graph neural networks for cold-start users and items representation, *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 265–273.
- He, X., Deng, K., Wang, X., Li, Y., Zhang, Y. and Wang, M. (2020). Lightgcn : Simplifying and powering graph convolution network for recommendation, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 639–648.
- Hood, J. C., Graber, C. and Brase, G. L. (2020). Comparing the efficacy of static and dynamic graph types in communicating complex statistical relationships, *Frontiers in Psychology* **10** : 2986.
- Howell, J., Kavirathne, S. D. and Pike, D. A. (2024). Brushing directed graphs, *arXiv preprint arXiv :2410.04559* .
- Hu, Y., Shi, W., Li, H. and Hu, X. (2017). Mitigating data sparsity using similarity reinforcement-enhanced collaborative filtering, *ACM Transactions on Internet Technology (TOIT)* **17**(3) : 1–20.
- Huang, T., Dong, Y., Ding, M., Yang, Z., Feng, W., Wang, X. and Tang, J. (2021). Mixgcf : An improved training method for graph neural network-based recommender systems, *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, Association for Computing Machinery, p. 665–674.
- Isufi, E., Pocchiari, M. and Hanjalic, A. (2021). Accuracy-diversity trade-off in recommender systems via graph convolutions, *Information Processing & Management* **58**(2) : 102459.

- Jain, A. F., Vishwakarma, S. K. and Jain, P. (2020). An efficient collaborative recommender system for removing sparsity problem, *ICT Analysis and Applications : Proceedings of ICT4SD 2019, Volume 2*, Springer, pp. 131–141.
- Ji, Y., Ding, Y., Liu, C., Lu, Y., Xin, X. and Lu, H. (2024). Topology-aware popularity debiasing via simplicial complexes, *arXiv preprint arXiv :2411.13892* .
- Jia, Z., Yang, Y., Gao, W. and Chen, X. (2015). User-based collaborative filtering for tourist attraction recommendations, *2015 IEEE international conference on computational intelligence & communication technology*, IEEE, pp. 22–25.
- Kecki, V. and Said, A. (2024). Understanding fairness in recommender systems : A healthcare perspective, *Proceedings of the 18th ACM Conference on Recommender Systems*, pp. 1125–1130.
- Khemani, B., Patil, S., Kotecha, K. and Tanwar, S. (2024). A review of graph neural networks : concepts, architectures, techniques, challenges, datasets, applications, and future directions, *Journal of Big Data* **11**(1) : 18.
- Kim, K., Kim, S., Lee, G., Jung, J. and Shin, K. (2025). Multi-behavior recommender systems : a survey.
URL: <https://arxiv.org/abs/2503.06963>
- Kingma, D. P., Welling, M. et al. (2019). An introduction to variational autoencoders, *Foundations and Trends® in Machine Learning* **12**(4) : 307–392.
- Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks, *arXiv preprint arXiv :1609.02907* .
- Klimashevskaya, A., Jannach, D., Elahi, M. and Trattner, C. (2024). A survey on popularity bias in recommender systems, *User Modeling and User-Adapted Interaction* .
- Koren, Y., Bell, R. and Volinsky, C. (2009). Matrix factorization techniques for recommender systems, *Computer* **42**(8) : 30–37.
- Kovács, I. A., Luck, K., Spirohn, K., Wang, Y., Pollis, C., Schlabach, S., Bian, W., Kim, D.-K., Kishore, N., Hao, T. et al. (2019). Network-based prediction of protein interactions, *Nature communications* **10**(1) : 1240.
- Kowald, D., Mayr, G., Schedl, M. and Lex, E. (2023). A study on accuracy, miscalibration, and popularity bias in recommendations, *arXiv :2303.00400* .

- Li, J., Ren, Y., Sanderson, M. and Deng, K. (2024). Explaining recommendation fairness from a user/item perspective, *ACM Transactions on Information Systems* **43**(1) : 1–30.
- Li, M., Liu, Y., Jullien, S., Ariannezhad, M., Yates, A., Aliannejadi, M. and de Rijke, M. (2024). Are we really achieving better beyond-accuracy performance in next basket recommendation ?, *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2024, ACM, p. 924–934.
URL: <http://dx.doi.org/10.1145/3626772.3657835>
- Liang, D., Krishnan, R. G., Hoffman, M. D. and Jebara, T. (2018). Variational autoencoders for collaborative filtering, *Proceedings of the 2018 world wide web conference*, pp. 689–698.
- Lin, Z., Tian, C., Hou, Y. and Zhao, W. X. (2022). Improving graph collaborative filtering with neighborhood-enriched contrastive learning, *Proceedings of the ACM web conference 2022*, pp. 2320–2329.
- Linden, G., Smith, B. and York, J. (2003). Amazon. com recommendations : Item-to-item collaborative filtering, *IEEE Internet computing* **7**(1) : 76–80.
- Louppe, G., Wehenkel, L., Sutura, A. and Geurts, P. (2013). Understanding variable importances in forests of randomized trees, *Advances in neural information processing systems* **26**.
- Luo, Y., Jiang, Y., Jiang, Y., Chen, G., Wang, J., Bian, K., Li, P. and Zhang, Q. (2024). Online item cold-start recommendation with popularity-aware meta-learning.
URL: <http://arxiv.org/pdf/2411.11225>
- Malitesta, D., Pomo, C., Anelli, V. W., Mancino, A. C. M., Di Sciascio, E. and Di Noia, T. (2023). A topology-aware analysis of graph collaborative filtering, *arXiv preprint arXiv :2308.10778* .
- Mao, K., Zhu, J., Wang, J., Dai, Q., Dong, Z., Xiao, X. and He, X. (2021). Simplex : A simple and strong baseline for collaborative filtering, *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 1243–1252.
- Metsai, A. I., Karamitsios, K., Kotrotsios, K., Chatzimisios, P., Stalidis, G. and Goulianas, K. (2022). Evolution of neural collaborative filtering for recommender systems, *2022 14th International Conference on Knowledge and Smart Technology (KST)*, IEEE, pp. 86–90.

- Michail, O. (2015). An introduction to temporal graphs : An algorithmic perspective, *Algorithms, Probability, Networks, and Games : Scientific Papers and Essays Dedicated to Paul G. Spirakis on the Occasion of His 60th Birthday*, Springer, pp. 308–343.
- Morales Murillo, V. G., Pinto Avendaño, D. E., Rojas Lopez, F. and Gonzales Caleros, J. M. (2022). A systematic literature review on the hybrid approaches for recommender systems, *Computación y Sistemas* **26**(1) : 357–372.
- Nath, R. K. and Ahmad, T. (2022). Content based recommender system : Methodologies, performance, evaluation and application, *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, IEEE, pp. 423–428.
- Newman, M. E. (2003). The structure and function of complex networks, *SIAM review* **45**(2) : 167–256.
- Nguyen, H. H. L., Le, T. N., Ha, T.-T. and Nguyen, L. V. (2024). Collaborative filtering-based personalized recommendations : Challenges, limitations, and applications, *International Conference on Intelligence of Things*, Springer, pp. 12–23.
- Osmanlı, O. N. (2010). *A singular value decomposition approach for recommendation systems*, Master’s thesis, Middle East Technical University.
- Oubalahcen, H. and El Ouadghiri, M. D. (2023). Recommender systems for social networks : A short review, *Proceedings of the 6th International Conference on Networking, Intelligent Systems & Security*, pp. 1–7.
- Panteli, A. and Boutsinas, B. (2023a). Addressing the cold-start problem in recommender systems based on frequent patterns, *Algorithms* **16**(4) : 182.
- Panteli, A. and Boutsinas, B. (2023b). Addressing the cold-start problem in recommender systems based on frequent patterns, *Algorithms* **16**(4) : 182.
- Papadakis, H., Papagrigoriou, A., Panagiotakis, C., Kosmas, E. and Fragopoulou, P. (2022). Collaborative filtering recommender systems taxonomy, *Knowledge and Information Systems* **64**(1) : 35–74.
- Papagelis, M., Rousidis, I., Plexousakis, D. and Theoharopoulos, E. (2005). Incremental collaborative filtering for highly-scalable recommendation algorithms, *Foundations of Intelligent Systems : 15th International Symposium, ISMIS 2005, Saratoga Springs, NY, USA, May 25-28, 2005. Proceedings 15*, Springer, pp. 553–561.

- Pfaltz, J. L. (2018). A category of “undirected graphs” a tribute to hartmut ehrig, *Graph Transformation, Specifications, and Nets : In Memory of Hartmut Ehrig*, Springer, pp. 223–230.
- Prent, J. and Mansoury, M. (2024). Correcting for popularity bias in recommender systems via item loss equalization, *arXiv preprint arXiv :2410.04830* .
- Qin, Y., Ju, W., Wu, H., Luo, X. and Zhang, M. (2023). Learning graph ode for continuous-time sequential recommendation, *arXiv (Cornell University)* .
URL: <https://scispace.com/papers/learning-graph-ode-for-continuous-time-sequential-26jqak2e>
- Ranjan, A., Gidwani, K., Patil, D. and Uike, K. (2023). Book recommendation system using collaborative and content based filtering with web scraping, *International Journal of Advanced Research in Science, Communication and Technology* pp. 58–63.
- Rao, T., Gupta, A., Sharma, Y. and Dhondiyal, S. A. (2024). Revolutionizing the user experience using content-based recommendation system in digital platforms, *Available at SSRN 5077707* .
- Raza, S., Rahman, M., Kamawal, S., Toroghi, A., Raval, A., Navah, F. and Kazemeini, A. (2024). A comprehensive review of recommender systems : Transitioning from theory to practice, *arXiv preprint arXiv :2407.13699* .
- Rendle, S., Freudenthaler, C., Gantner, Z. and Schmidt-Thieme, L. (2012). Bpr : Bayesian personalized ranking from implicit feedback, *arXiv preprint arXiv :1205.2618* .
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. (1994). GroupLens : An open architecture for collaborative filtering of netnews, *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pp. 175–186.
- Rožman, M., Kišić, A. and Oreški, D. (2024). Comparative analysis of nonlinear models developed using machine learning algorithms, *WSEAS transactions on information science and applications* **21**.
- Rusch, T. K., Bronstein, M. M. and Mishra, S. (2023). A survey on oversmoothing in graph neural networks, *arXiv preprint arXiv :2303.10993* .
- Sankari, A., Masih, S. and Ingle, M. (2024). Exploring matrix decomposition methods for recommender systems, *Journal of Scientific Research* **16**(3) : 705–712.

- Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms, *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295.
- Sarwar, B., Karypis, G., Konstan, J., Riedl, J. et al. (2000). Application of dimensionality reduction in recommender system-a case study, *ACM WebKDD workshop*, Vol. 1625, Citeseer, pp. 285–295.
- Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M. and Monfardini, G. (2008). The graph neural network model, *IEEE transactions on neural networks* **20**(1) : 61–80.
- Seth, R. and Sharaff, A. (2022). A comparative overview of hybrid recommender systems : Review, challenges, and prospects, *Data mining and machine learning applications* pp. 57–98.
- Shani, G. and Gunawardana, A. (2011). Evaluating recommendation systems, *Recommender systems handbook* pp. 257–297.
- Sharma, A., Raj, R. and Yamini, B. (2024). Novel algorithmic recommendation engine for diverse content discovery, *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, IEEE, pp. 1–4.
- Sharma, J., Sharma, K., Garg, K. and Sharma, A. K. (2021). Product recommendation system a comprehensive review, *IOP conference series : materials science and engineering*, Vol. 1022, IOP Publishing, p. 012021.
- Shenbin, I., Alekseev, A., Tutubalina, E., Malykh, V. and Nikolenko, S. I. (2020). Recvae : A new variational autoencoder for top-n recommendations with implicit feedback, *Proceedings of the 13th international conference on web search and data mining*, pp. 528–536.
- Shi, L., Li, S., Ding, X. and Bu, Z. (2023). Selection bias mitigation in recommender system using uninteresting items based on temporal visibility, *Expert Systems with Applications* **213** : 118932.
- Singhal, A., Sinha, P. K. and Pant, R. (2017). Use of deep learning in modern recommendation system : A summary of recent works, **180**(7) : 17–22.
URL: <https://doi.org/10.5120/ijca2017916055>

- Sun, J., Zhang, Y., Ma, C., Coates, M., Guo, H., Tang, R. and He, X. (2019). Multi-graph convolution collaborative filtering, *2019 IEEE International Conference on Data Mining (ICDM)*, IEEE, pp. 1306–1311.
- Sun, M., Li, L., Li, M., Tao, X., Zhang, D., Wang, P. and Huang, J. X. (2024). A survey on bundle recommendation : Methods, applications, and challenges.
URL: <https://arxiv.org/abs/2411.00341>
- Tang, X., Yao, H., Sun, Y., Wang, Y., Tang, J., Aggarwal, C., Mitra, P. and Wang, S. (2020). Investigating and mitigating degree-related biases in graph convolutional networks, *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, ACM, pp. 1435–1444.
- Thomas, J. M., Moallem-Oureh, A., Beddar-Wiesing, S. and Holzhüter, C. (2022). Graph neural networks designed for different graph types : A survey, *arXiv preprint arXiv :2204.03080* .
- Thorat, S. A., Ashwini, G. and Seema, M. (2023). Survey on collaborative and content-based recommendation systems, *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, IEEE, pp. 1541–1548.
- Vaishnavi, S. (2017). Ensemble methods and random forests.
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P. and Bengio, Y. (2017). Graph attention networks, *arXiv preprint arXiv :1710.10903* .
- Vhatkar, A. (2024). A collaborative filtering-based recommender systems approach for multifarious applications, *Journal of Electrical Systems* **20** : 478–485.
- Vrijenhoek, S., Daniil, S., Sandel, J. and Hollink, L. (2024). Diversity of what ? on the different conceptualizations of diversity in recommender systems.
- Wang, C., Yu, Y., Ma, W., Zhang, M., Chen, C., Liu, Y. and Ma, S. (2022). Towards representation alignment and uniformity in collaborative filtering, *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 1816–1825.
- Wang, H., Wang, K., Zhang, W. and Zhang, Y. (2024). Bipartite graph analytics : Current techniques and future trends, *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, IEEE, pp. 01–07.

- Wang, W., Xu, Y., Feng, F., Lin, X., He, X. and Chua, T.-S. (2023). Diffusion recommender model.
URL: <https://arxiv.org/abs/2304.04971>
- Wang, X., He, X., Wang, M., Feng, F. and Chua, T.-S. (2019). Neural graph collaborative filtering, *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 165–174.
- Wang, X., Jin, H., Zhang, A., He, X., Xu, T. and Chua, T.-S. (2020). Disentangled graph collaborative filtering, *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pp. 1001–1010.
- Wang, Y. and Derr, T. (2022). Degree-related bias in link prediction, *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, IEEE, pp. 757–758.
- Wang, Z., Zhao, H. and Shi, C. (2022). Profiling the design space for graph neural networks based collaborative filtering, *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pp. 1109–1119.
- Wei, L., Zhao, H. and He, Z. (2022). Designing the topology of graph neural networks : A novel feature fusion perspective, *Proceedings of the ACM Web Conference 2022*, pp. 1381–1391.
- Wu, J., Wang, X., Feng, F., He, X., Chen, L., Lian, J. and Xie, X. (2021). Self-supervised graph learning for recommendation, *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, pp. 726–735.
- Wu, S., Sun, F., Zhang, W., Xie, X. and Cui, B. (2022). Graph neural networks in recommender systems : A survey, *ACM Computing Surveys* **55**(5) : 1–37.
URL: <https://dl.acm.org/doi/10.1145/3535101>
- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C. and Yu, P. S. (2020). A comprehensive survey on graph neural networks, *IEEE transactions on neural networks and learning systems* **32**(1) : 4–24.
- Xia, Z., Sun, A., Xu, J., Peng, Y., Ma, R. and Cheng, M. (2024). Contemporary recommendation systems on big data and their applications : A survey, *IEEE Access* .

- Xiao, B. and Chen, D. (2024). Explicitly exploiting implicit user and item relations in graph convolutional network (gcn) for recommendation, *Electronics* **13**(14) : 2811.
- Xiong, X., Zhou, B. and Wang, Y. G. (2023). Graph representation learning for interactive biomolecule systems, *arXiv preprint arXiv :2304.02656* .
- Xue, H.-J., Dai, X.-Y., Zhang, J., Huang, S. and Chen, J. (2017). Deep matrix factorization models for recommender systems, *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI'17*, AAAI Press, p. 3203–3209.
- Yang, S., Zhang, L., Xia, F. and Lin, L. (2023). Graph exploration matters : improving both individual-level and system-level diversity in wechat feed recommendation, *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 4901–4908.
- Yang, Z., Ding, M., Huang, T., Cen, Y., Song, J., Xu, B., Dong, Y. and Tang, J. (2024). Does negative sampling matter? a review with insights into its theory and applications, *IEEE Transactions on Pattern Analysis and Machine Intelligence* .
- Yu, J., Xia, X., Chen, T., Cui, L., Hung, N. Q. V. and Yin, H. (2023). XSimGCL : Towards extremely simple graph contrastive learning for recommendation, *IEEE Transactions on Knowledge and Data Engineering* .
- Yu, J., Yin, H., Xia, X., Chen, T., Cui, L. and Nguyen, Q. V. H. (2022). Are graph augmentations necessary? simple graph contrastive learning for recommendation, *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pp. 1294–1303.
- Yurchak, I. and Hryhlevych, M. (2023). Recommender system based on collaborative filtering, *COMPUTER-INTEGRATED TECHNOLOGIES : EDUCATION, SCIENCE, PRODUCTION* (53) : 78–85.
- Zadgaonkar, A. V. and Agrawal, A. J. (2020). Graph : An efficient data structure to represent and interpret semantic information, *ICCCE 2020 : Proceedings of the 3rd International Conference on Communications and Cyber Physical Engineering*, Springer, pp. 809–817.
- Zhang, T., Yang, L., Xiao, Z., Jiang, W. and Ning, W. (2024). On practical diversified recommendation with controllable category diversity framework, *Companion Proceedings of the ACM Web Conference 2024*, pp. 255–263.

- Zhang, X., He, Z., Zhang, L., Rayman-Bacchus, L., Shen, S. and Xiao, Y. (2022). The analysis of the power law feature in complex networks, *Entropy* **24**(11) : 1561.
- Zhao, Y., Chen, R., Lai, R., Han, Q., Song, H. and Chen, L. (2023). Augmented negative sampling for collaborative filtering, *Proceedings of the 17th ACM Conference on Recommender Systems*, pp. 256–266.
- Zhao, Y., Wang, Y., Liu, Y., Cheng, X., Aggarwal, C. C. and Derr, T. (2025). Fairness and diversity in recommender systems : a survey, *ACM Transactions on Intelligent Systems and Technology* **16**(1) : 1–28.
- Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C. and Sun, M. (2020). Graph neural networks : A review of methods and applications, *AI open* **1** : 57–81.
- Zhu, T. and Liu, Y. (2023). Learning personalized preference : A segmentation strategy under consumer sparse data, *Expert Systems with Applications* **215** : 119333.

APPENDIX A RSPE METRIC RESULTS



Table APPENDIX A.1 - Accuracy Performance for Each Model

Model Metric data _{n,ame}	Deep-MF		DiffRec		DirectAU		LightGCN		MixGCF						
	HR	MRR	NDCG	Prec.	Rec.	HR	MRR	NDCG	Prec.	Rec.	HR	MRR	NDCG	Prec.	Rec.
BookCrossing	0.30	0.09	0.03	0.02	0.02	0.21	0.06	0.02	0.01	0.01	0.13	0.04	0.01	0.01	0.39
Kindle	0.30	0.11	0.10	0.02	0.16	0.01	0.01	0.01	0.01	0.01	0.35	0.13	0.11	0.03	0.19
LastFM	0.90	0.54	0.26	0.16	0.24	0.79	0.25	0.13	0.10	0.15	0.70	0.27	0.11	0.07	0.12
ML-100k	0.98	0.62	0.36	0.18	0.37	0.76	0.21	0.13	0.08	0.17	0.57	0.19	0.09	0.05	0.11
ML-1m	0.95	0.60	0.36	0.28	0.23	0.85	0.26	0.13	0.11	0.12	0.85	0.35	0.16	0.11	0.13
Modcloth	0.21	0.06	0.05	0.02	0.10	0.27	0.05	0.06	0.02	0.14	0.14	0.04	0.03	0.01	0.06
Setur	0.05	0.01	0.01	0.00	0.03	0.17	0.06	0.06	0.01	0.13	0.04	0.01	0.01	0.00	0.02
Song-data	0.04	0.01	0.01	0.00	0.02	0.03	0.00	0.01	0.00	0.02	0.06	0.01	0.01	0.00	0.03
Yelp-2018	0.32	0.08	0.04	0.02	0.05	0.32	0.06	0.03	0.02	0.05	0.41	0.01	0.06	0.03	0.07
iFashion	0.07	0.02	0.03	0.00	0.06	0.01	0.00	0.00	0.00	0.01	0.14	0.04	0.06	0.07	0.11
Model Metric data _{n,ame}	NCL		SGL		Simplex		XSimGCL		itemknn						
	HR	MRR	NDCG	Prec.	Rec.	HR	MRR	NDCG	Prec.	Rec.	HR	MRR	NDCG	Prec.	Rec.
BookCrossing	0.41	0.13	0.05	0.04	0.03	0.42	0.15	0.05	0.04	0.03	0.06	0.01	0.01	0.01	0.39
Kindle	0.34	0.12	0.11	0.03	0.19	0.36	0.14	0.13	0.03	0.20	0.01	0.01	0.00	0.01	0.36
LastFM	0.93	0.60	0.29	0.18	0.27	0.93	0.59	0.29	0.18	0.28	0.80	0.22	0.12	0.09	0.15
ML-100k	0.97	0.64	0.37	0.19	0.39	0.98	0.61	0.36	0.19	0.37	0.72	0.22	0.14	0.08	0.17
ML-1m	0.97	0.65	0.40	0.30	0.26	0.96	0.64	0.38	0.29	0.25	0.88	0.29	0.15	0.12	0.13
Modcloth	0.27	0.08	0.07	0.02	0.14	0.13	0.04	0.03	0.01	0.06	0.11	0.02	0.02	0.01	0.04
Setur	0.08	0.02	0.02	0.01	0.05	0.04	0.01	0.01	0.01	0.02	0.04	0.02	0.02	0.01	0.03
Song-data	0.06	0.01	0.01	0.00	0.03	0.06	0.01	0.01	0.00	0.03	0.05	0.01	0.01	0.00	0.03
Yelp-2018	0.40	0.11	0.06	0.03	0.07	0.40	0.11	0.05	0.03	0.07	0.27	0.06	0.03	0.02	0.04
iFashion	0.10	0.03	0.04	0.01	0.08	0.12	0.04	0.05	0.01	0.10	0.01	0.01	0.01	0.01	0.12
Model Metric data _{n,ame}	multivae		recvae		userknn		userknn		userknn						
	HR	MRR	NDCG	Prec.	Rec.	HR	MRR	NDCG	Prec.	Rec.	HR	MRR	NDCG	Prec.	Rec.
BookCrossing	0.27	0.07	0.02	0.02	0.02	0.26	0.09	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.01
Kindle	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.04	0.01	0.01	0.00	0.04
LastFM	0.85	0.30	0.15	0.11	0.17	0.82	0.27	0.14	0.10	0.16	0.06	0.01	0.01	0.01	0.01
ML-100k	0.81	0.27	0.16	0.10	0.20	0.79	0.23	0.15	0.10	0.19	0.09	0.02	0.01	0.01	0.01
ML-1m	0.89	0.29	0.15	0.12	0.13	0.89	0.29	0.15	0.12	0.14	0.11	0.02	0.00	0.01	0.01
Modcloth	0.37	0.08	0.10	0.03	0.22	0.33	0.06	0.08	0.03	0.19	0.04	0.01	0.00	0.00	0.01
Setur	0.16	0.05	0.06	0.01	0.12	0.17	0.05	0.06	0.01	0.12	0.01	0.01	0.01	0.00	0.01
Song-data	0.05	0.01	0.01	0.01	0.02	0.03	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01
Yelp-2018	0.31	0.07	0.04	0.02	0.05	0.31	0.07	0.04	0.02	0.05	0.25	0.07	0.03	0.02	0.03
iFashion	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01

Table APPENDIX A.2 - Diversity Performance for Each Model

Model Metric data_name	Deep-MF			DiffRec			DirectAU			LightGCN			MixGCF			NCL			SGL									
	IC	ILD	Nov	Pers	IC	Nov	Pers	IC	ILD	Nov	Pers	IC	ILD	Nov	Pers	IC	ILD	Nov	Pers	IC	ILD	Nov	Pers					
BookCrossing	0.025	0.986	0.749	0.927	0.009	0.984	0.724	0.764	0.139	0.998	0.993	0.999	0.039	0.986	0.757	0.910	0.052	0.986	0.785	0.934	0.025	0.983	0.734	0.892	0.051	0.988	0.824	0.987
Kindle	0.653	0.9953	0.804	0.998	0.087	0.992	0.721	0.975	0.292	0.992	0.777	0.994	0.579	0.998	0.786	0.998	0.459	0.992	0.771	0.999	0.439	0.997	0.759	0.987	0.364	0.992	0.765	0.998
LastFM	0.331	0.976	0.688	0.963	0.103	0.961	0.625	0.858	0.864	0.982	0.918	0.997	0.379	0.977	0.704	0.969	0.396	0.974	0.703	0.967	0.350	0.974	0.685	0.967	0.338	0.974	0.706	0.966
ML-100k	0.488	0.978	0.656	0.938	0.150	0.967	0.607	0.820	0.971	0.984	0.860	0.988	0.498	0.977	0.658	0.948	0.468	0.977	0.656	0.948	0.490	0.976	0.667	0.949	0.425	0.975	0.646	0.952
ML-1m	0.546	0.960	0.558	0.960	0.151	0.946	0.525	0.842	0.984	0.976	0.705	0.987	0.595	0.962	0.570	0.967	0.797	0.976	0.627	0.978	0.542	0.957	0.558	0.967	0.415	0.957	0.548	0.963
Modcloth	0.409	0.939	0.523	0.939	0.885	0.926	0.520	0.798	0.977	0.965	0.603	0.977	0.309	0.919	0.488	0.918	0.239	0.919	0.504	0.923	0.328	0.917	0.491	0.907	0.840	0.967	0.608	0.968
Setur	0.342	0.999	0.602	0.957	0.921	0.997	0.558	0.713	0.959	0.999	0.669	0.977	0.230	0.989	0.563	0.928	0.137	0.986	0.538	0.918	0.280	0.987	0.546	0.929	0.678	0.997	0.668	0.978
Song-data	0.488	0.998	0.720	0.984	0.036	0.997	0.668	0.972	0.097	0.997	0.706	0.987	0.247	0.996	0.698	0.968	0.082	0.997	0.702	0.977	0.107	0.997	0.688	0.967	0.038	0.996	0.686	0.978
Yelp-2018	0.498	0.987	0.740	0.988	0.108	0.987	0.697	0.975	0.547	0.9864	0.782	0.999	0.497	0.987	0.743	0.987	0.239	0.989	0.718	0.988	0.259	0.987	0.713	0.986	0.276	0.9876	0.708	0.987
iFashion	0.653	0.998	0.796	0.998	0.051	0.997	0.621	0.968	0.294	0.987	0.784	0.988	0.579	0.998	0.737	0.998	0.458	0.992	0.768	0.998	0.436	0.997	0.759	0.990	0.360	0.987	0.768	0.998

Model Metric data_name	Simplex			XSimGCL			itemknn			multivae			recvae			userknn								
	IC	ILD	Nov	Pers	IC	ILD	Nov	Pers	IC	ILD	Nov	Pers	IC	ILD	Nov	Pers	IC	ILD	Nov	Pers	IC	ILD	Nov	Pers
BookCrossing	0.189	0.990	0.918	0.908	0.059	0.988	0.857	0.998	0.139	0.999	0.987	0.999	0.020	0.982	0.738	0.814	0.003	0.9829	0.634	0.073	0.076	0.998	0.986	0.987
Kindle	0.968	0.992	0.870	0.998	0.249	0.986	0.747	0.987	0.375	0.999	0.918	0.998	0.557	0.9917	0.807	0.998	0.957	0.987	0.826	0.997	0.007	0.999	0.938	0.982
LastFM	0.746	0.972	0.719	0.967	0.447	0.977	0.726	0.978	0.050	0.999	0.988	0.998	0.239	0.966	0.678	0.938	0.167	0.9621	0.657	0.905	0.026	0.997	0.917	0.988
ML-100k	0.237	0.9580	0.595	0.756	0.658	0.970	0.695	0.967	0.228	0.999	0.987	0.998	0.284	0.963	0.608	0.849	0.168	0.967	0.620	0.840	0.144	0.998	0.949	0.988
ML-1m	0.318	0.947	0.527	0.887	0.686	0.977	0.636	0.987	0.094	0.999	0.977	0.999	0.418	0.947	0.538	0.929	0.318	0.947	0.538	0.929	0.057	0.998	0.849	0.987
Modcloth	0.998	0.978	0.657	0.979	0.178	0.907	0.470	0.898	0.377	0.999	0.987	0.998	0.098	0.8912	0.428	0.578	0.967	0.896	0.460	0.790	0.228	0.992	0.918	0.987
Setur	0.998	0.994	0.779	0.987	0.108	0.987	0.520	0.899	0.149	0.999	0.987	0.998	0.999	0.987	0.568	0.908	0.987	0.985	0.539	0.888	0.087	0.999	0.907	0.989
Song-data	0.927	0.998	0.778	0.999	0.034	0.998	0.678	0.978	0.297	0.998	0.947	0.998	0.048	0.997	0.709	0.987	0.958	0.998	0.779	0.998	0.179	0.999	0.968	0.992
Yelp-2018	0.597	0.987	0.768	0.998	0.248	0.9876	0.718	0.987	0.998	0.999	0.878	1.000	0.267	0.987	0.719	0.988	0.158	0.9862	0.707	0.988	0.607	0.986	0.748	0.988
iFashion	0.988	0.999	0.729	0.999	0.457	0.9874	0.768	0.999	0.458	0.999	0.907	0.999	0.958	0.9985	0.687	0.987	0.987	0.998	0.698	0.987	0.278	0.9994	0.817	0.988

Table APPENDIX A.3 - Fairness Performance for Each Model

Model Metric	Deep-MF AP	DiffRec		DirectAU		LightGCN		MixGCF		NCL		SGL									
		GINI	IEF	AP	GINI	IEF	AP	GINI	IEF	AP	GINI	IEF	AP								
data_name																					
BookCrossing	0.21	1.00	0.10	0.27	1.00	0.11	0.01	0.90	0.92	0.22	0.99	0.24	0.19	0.99	0.44	<u>0.25</u>	1.00	0.16	0.11	0.99	0.29
Kindle	0.06	0.86	0.38	0.16	0.99	0.10	0.09	0.95	0.28	0.07	0.88	0.31	0.09	0.92	0.24	<u>0.10</u>	0.93	0.20	0.09	0.93	0.21
LastFM	0.26	0.93	0.17	0.40	0.98	0.01	0.02	0.52	0.97	0.24	0.91	0.21	0.25	0.92	0.26	0.25	0.92	0.18	0.23	0.92	0.21
ML-100k	0.43	0.86	0.25	0.58	0.95	0.01	0.11	0.51	<u>1.00</u>	0.42	0.85	0.27	0.41	0.86	0.29	0.41	0.84	<u>0.29</u>	<u>0.44</u>	0.87	0.2
ML-1m	0.40	0.87	0.24	0.59	0.97	0.01	0.12	0.47	0.95	0.37	0.85	0.29	0.28	0.75	0.68	0.37	0.86	<u>0.25</u>	<u>0.43</u>	0.91	0.12
Modcloth	0.23	0.82	0.16	0.31	0.93	<u>0.42</u>	0.13	0.57	0.71	0.29	0.88	0.06	0.28	0.87	0.12	<u>0.30</u>	0.88	0.08	0.14	0.63	0.65
Setur	0.09	0.89	0.06	<u>0.19</u>	0.96	0.09	0.06	0.74	0.48	0.14	0.93	0.06	0.15	0.95	0.01	0.15	0.94	0.09	0.07	0.78	0.46
Song-data	0.32	0.97	<u>0.10</u>	0.44	1.00	0.09	0.28	0.99	0.01	0.38	0.99	0.06	0.33	0.99	0.00	<u>0.40</u>	0.99	0.02	0.34	0.99	0.00
Yelp-2018	0.17	0.90	0.31	0.27	<u>0.98</u>	0.04	0.10	0.85	0.51	0.17	0.89	<u>0.28</u>	0.22	0.95	0.16	0.21	0.93	0.18	0.19	0.94	0.13
iFashion	0.20	0.96	0.11	0.26	0.99	0.14	<u>0.21</u>	0.99	0.01	<u>0.21</u>	0.99	0.03	0.20	0.99	0.01	<u>0.25</u>	0.99	0.01	<u>0.21</u>	0.99	0.01
Model Metric	Simplex AP	XSimGCL		itemknn		multivae		recvae		userknn											
		GINI	IEF	AP	GINI	IEF	AP	GINI	IEF	AP	GINI	IEF	AP								
BookCrossing	0.04	0.86	0.87	0.07	0.98	0.32	0.00	0.92	<u>1.00</u>	0.25	1.00	0.23	0.41	1.00	0.01	0.00	0.98	0.99			
Kindle	0.03	0.65	<u>0.79</u>	0.11	0.96	0.14	0.01	0.77	0.99	0.06	0.86	0.48	0.05	0.70	0.64	0.01	0.99	0.99			
LastFM	0.22	0.84	<u>0.42</u>	0.19	0.88	0.33	0.00	0.42	1.00	<u>0.30</u>	0.95	0.17	<u>0.34</u>	0.96	0.09	0.03	0.78	0.94			
ML-100k	<u>0.57</u>	0.96	0.11	0.33	0.77	<u>0.55</u>	0.01	0.40	1.00	0.56	<u>0.94</u>	0.09	0.56	<u>0.95</u>	0.04	0.05	0.71	0.99			
ML-1m	<u>0.52</u>	0.95	0.10	0.20	0.73	<u>0.74</u>	0.01	0.41	1.00	0.48	<u>0.93</u>	0.13	0.48	<u>0.93</u>	0.09	0.05	0.73	<u>1.00</u>			
Modcloth	0.11	0.51	0.89	<u>0.32</u>	<u>0.90</u>	0.00	0.00	0.39	1.00	0.50	0.97	0.00	<u>0.40</u>	0.93	0.05	0.02	0.74	<u>0.99</u>			
Setur	0.03	0.51	0.89	0.17	<u>0.96</u>	0.01	0.01	0.41	1.00	0.17	0.86	0.28	<u>0.18</u>	0.93	0.11	0.01	0.73	<u>0.98</u>			
Song-data	0.15	0.87	0.33	<u>0.38</u>	0.99	0.00	0.01	0.83	<u>1.00</u>	0.27	0.99	0.01	0.17	0.86	0.28	0.01	0.97	1.00			
Yelp-2018	0.11	0.84	0.42	<u>0.19</u>	<u>0.94</u>	0.18	0.03	0.40	0.98	<u>0.19</u>	<u>0.94</u>	<u>0.22</u>	0.21	0.97	0.09	0.18	0.89	0.33			
iFashion	0.10	0.86	0.32	<u>0.23</u>	<u>0.99</u>	0.01	0.01	0.72	0.88	0.16	0.91	<u>0.19</u>	0.14	0.90	<u>0.19</u>	0.04	0.95	0.61			

APPENDIX B R² SCORES OF RANDOM FOREST MODELS

Table APPENDIX B.1 - R² Regression Results Between Graph Topological Features and Accuracy Metrics

Model	HR	MRR	NDCG	Precision	Recall
Deep-MF	0.928	0.929	0.904	0.861	0.676
Simplex	0.897	0.854	0.907	0.798	0.867
DirectAU	0.651	0.671	0.557	0.511	0.587
LightGCN	0.926	0.849	0.830	0.881	0.702
MixGCF	0.860	0.876	0.879	0.947	0.528
NCL	0.885	0.873	0.889	0.878	0.693
SGL	0.898	0.935	0.692	0.875	0.740
XSimGCL	0.804	0.943	0.836	0.878	0.598
Diffrec	0.911	0.888	0.890	0.894	0.908
multivae	0.949	0.910	0.886	0.933	0.816
recvae	0.860	0.907	0.925	0.847	0.879
itemknn	0.911	0.902	0.354	0.906	0.812
userknn	0.663	0.678	0.712	0.729	0.430

Table APPENDIX B.2 - R² Regression Results Between Graph Topological Features and Diversity Metrics

Model	IC	ILD	Nov	Pers
Deep-MF	0.584	0.704	0.562	0.579
Simplex	0.808	0.861	0.631	0.847
DirectAU	0.816	0.611	0.799	0.751
LightGCN	0.692	0.791	0.700	0.648
MixGCF	0.593	0.888	0.677	0.819
NCL	0.544	0.871	0.661	0.688
SGL	0.533	0.733	0.437	0.815
XSimGCL	0.732	0.954	0.657	0.931
Diffrec	0.981	0.735	0.767	0.795
multivae	0.008	0.922	0.685	0.777
recvae	0.947	0.805	0.669	0.959
itemknn	0.845	0.703	0.780	0.406
userknn	0.882	0.478	0.419	0.727

Table APPENDIX B.3 - R² Regression Results Between Graph Topological Features and Fairness Metrics

Model	AP	GINI	IEF
Deep-MF	0.472	0.315	0.776
Simplex	0.882	0.747	0.702
DirectAU	0.658	0.875	0.861
LightGCN	0.177	0.665	0.940
MixGCF	0.407	0.657	0.708
NCL	0.372	0.546	0.746
SGL	0.562	0.793	0.759
XSimGCL	0.337	0.841	0.809
Diffrec	0.457	0.791	0.894
multivae	0.642	0.520	0.499
recvae	0.832	0.793	0.745
itemknn	0.736	0.755	0.826
userknn	0.628	0.830	0.461