Movie Recommendation Systems: A comparative study of Collaborative filtering, Content-based filtering and a Hybrid model. Project Option: 1. Comparative Analysis

RITIK VERMA, University At Buffalo, USA
GURWINDER KAUR, University At Buffalo, USA
ABDUL RAHMAN HUSSAIN SIDDIQUE, University At Buffalo, USA

Recommendation systems are essential tools for navigating vast digital content libraries, but standard approaches like Collaborative Filtering (CF) and Content-Based Filtering (CBF) face challenges such as data sparsity, the cold-start problem, and balancing accuracy with recommendation diversity. This paper presents a comparative study evaluating the effectiveness of CF (using Alternating Least Squares), CBF (using TF-IDF and Bisecting K-Means), and a Hybrid model combining both approaches for movie recommendation. Using the MovieLens 100k dataset, we analyze these models based on standard metrics (MSE, RMSE, MAE, Precision@K, Recall@K, F1-score) and advanced diagnostics including Out-of-Distribution (OOD) robustness, heteroscedasticity analysis, double descent curve analysis, and t-SNE visualizations. Our results confirm that while CF achieves superior rating prediction accuracy, it suffers significantly in cold-start scenarios and offers limited recommendation coverage. CBF provides broader coverage but can lead to over-specialization. The Hybrid model demonstrates the most balanced performance, achieving the highest ranking metrics (Precision@K, Recall@K, F1-score) and effectively mitigating the cold-start problem, making it the most practical approach for enhancing user experience in movie recommendation tasks despite a slight trade-off in raw predictive accuracy compared to pure CF.

1 Introduction

The objective of this project is to compare and evaluate the effectiveness of different models of the recommendation system to enhance the user experience. Recommendation systems are used by many top platforms such as Amazon, Netflix, and Walmart. These systems not only drive significant business value, but also offer fascinating insights into human behavior and preferences. Its uses vary from movie recommendation systems for streaming services such as Netflix and Prime to item recommendation for E-Commerce sites such as Amazon and Walmart. Recommendation systems are interesting as they Reflect Human behavior. Because so many people use these systems and machine learning technology is changing quickly, it is important to continue improving these systems. This way it helps them to better satisfy what the user wants and keep up with the fast trending changes.

1.1 What has been done before in this space?

Recommendation systems have evolved significantly from basic collaborative and content-based models to more sophisticated hybrid systems that influence deep learning technologies. The field largely revolves around core paradigms such as Collaborative Filtering (CF), Content-Based Filtering (CBF), and various Hybrid approaches. Each of these attempts to model user preferences in distinct ways. While these approaches have advanced the field, significant challenges remain, particularly in effectively handling data sparsity and the cold-start problem for new users or items. Furthermore, balancing the goals of recommendation accuracy with coverage and diversity is an ongoing challenge for many systems. Addressing these issues is crucial for enhancing user satisfaction and engagement on platforms reliant on effective content discovery. Therefore, comparative studies evaluating the trade-offs between these established methods, particularly within specific domains like movie recommendation, remain valuable for understanding their relative strengths and optimal application contexts.

1.2 Usecase Context

The large amount of digital content available today can overwhelm users. Recommendation systems are therefore very important for helping users navigate choices on different platforms. Modern platforms are complex and require advanced methods that go beyond simple filtering techniques. For example, large streaming services like Netflix and YouTube use advanced techniques like deep neural networks. These help model complex user-item interactions and personalize different tasks, such as ranking content, finding new items, and designing the user interface Dong et al.[1], Steck et al.[21].

Specifically within the movie domain, platforms manage massive catalogs containing tens of thousands of titles. This scale inherently leads to significant data sparsity, as individual users rate or watch only a tiny fraction of the available content. Effectively modeling user preferences from such sparse data is a core challenge. Therefore, techniques that can leverage deeper user behavior patterns or richer item features are essential not just for accuracy, but also for driving user engagement and retention by suggesting content that truly matches nuanced tastes and promotes discovery.

Further advancements specifically for movie recommendations explore hybrid session-based models, which use sequence modeling and rich item features from language models like BERT to better understand what a user might want based on recent activity Potter et al.[27]. Different types of information can also help; for instance, using visual details from movie posters and stills in matrix factorization models can improve recommendations by capturing style aspects that other data might miss Zhao et al.[28]. More recently, researchers are exploring how Large Language Models (LLMs) can be used in recommendations. This includes using LLMs for movie recommendations by potentially improving how user preferences are understood or explained Zhao et al.[29]. These examples highlight the need for adaptable and robust recommendation strategies

Authors' Contact Information: Ritik Verma, ritikver@buffalo.edu, University At Buffalo, Buffalo, New York, USA; Gurwinder Kaur, gkaur36@buffalo.edu, University At Buffalo, Buffalo, New York, USA; Abdul Rahman Hussain Siddique, Ah244@buffalo.edu, University At Buffalo, Buffalo, New York, USA.

designed for the specific problems within media recommendation environments.

1.3 Overall plan

In this project we conduct a comparative study of two different models used in recommendation systems: Collaborative filtering and Content-based filtering. Our focus in this project will be on movie recommendation platforms. We are going to analyze each model independently to identify its strengths and challenges. We will explore the potential of a hybrid model that combines the best features of both Collaborative filtering and Content-based filtering.

1.4 Expectation

Our expectation in the group project is to develop algorithms that capture individual preferences and behaviors. By combining collaborative filtering with content-based filtering will provide a more robust recommendation. We are going to explore areas where recommendation systems will handle large datasets while keeping in mind the speed and accuracy. Our project will also be able to have methods to deal with data sparsity.

1.5 Why is this interesting?

The contributions that we discussed in the previous section are really important because today there is a demand of having more sophisticated, accurate and fair recommendation systems. Having personalized recommendations are beneficial for user satisfaction and also drive business success. By providing users with content they are more likely to engage with, platforms can increase user retention and revenue. They can handle problems related to lack of personalization, sparse data and cold start.

With so much digital content available, users often feel overwhelmed. Recommendation systems help make choices easier by filtering and suggesting relevant content to users. This is especially important in areas like movies, where the huge number of options can be overwhelming for viewers. These systems help users navigate through vast content libraries to find what truly interests them, improving their overall experience.

2 Related Work

Movie recommendation systems have been through various types of development over the years and can be broadly categorized into three main categories: content-based filtering, collaborative filtering, and hybrid models. This section gives an overview of the contributions and limitations in each area presented by the research over the years

2.1 Collaborative Filtering Approaches

Collaborative filtering relies on inter-user behavioral correlations. CF works by collecting interests from many users to make automatic predictions (filtering) about the interests of other users. The underlying premise of CF is that a person's preference does not change much over time[24]. It has been widely studied, with the main focus on researching the distinction between user-based and item-based collaborative filtering.[7]

Literature such as Wu et al. [8], Sarwar et al. [5] have extensively evaluated these approaches on datasets such as MovieLens, which we also use in our research. These works highlight the effectiveness of collaborative filtering by using the collective user preferences to generate recommendations, but their reliance on dense interaction matrices introduces vulnerabilities i.e. if the data is sparse, the quality of recommendations suffers. Additionally, in collaborative filtering if a user has unique choices he wont get good recommendations due to the inconsistent alignment with any user group.

Rendle et al.[20] present a framework for understanding collaborative filtering (CF) methodologies, categorizing them between **memory-based** approaches that leverage similarity metrics among users or items, and **model-based** techniques that construct predictive frameworks from available data.

Rendle et al.[20] also explores several Bayesian variants of matrix factorization, including Bayesian MF, which characterizes latent factors as probabilistic distributions to better manage uncertainty and reduce overfitting challenges in sparse data environments. Their analysis extends to temporally-aware models like Bayesian timeSVD, which accounts for evolving user preferences through temporal variables, and Bayesian SVD++, which enriches the recommendation process by incorporating implicit feedback mechanisms. These sophisticated approaches demonstrate significant improvements in recommendation quality by capturing complex patterns in user behavior and leveraging diverse interaction signals.

Hongyi[19] mentions the concept of temporal collaborative filtering, which considers the time aspect of user preferences and also privacy-aware collaborative filtering, where users have control over their data, User-controlled data filtering techniques, such as removal of out-dated records based on users' time preferences. This approach is useful where the user interest changes over time and gives importance to the most recent interactions rather than focusing on past interactions.

Chen et al.[12] further distinguish between these two categories. Memory-based algorithms, such as user-based methods [25] and item-based methods [5], establish neighborhood relationships for each user and predict missing ratings using a weighted sum of known ratings. However, these methods struggle with data sparsity, as accurate user or item similarities require sufficient rating data. In contrast, model-based algorithms develop a predictive model of user preferences to generate recommendations. Matrix factorization is a prominent example of model-based CF [6], alongside other techniques like Restricted Boltzmann Machines (RBM) [23] and neural autoregressive approaches such as CF-NADE [24]. CF-NADE, inspired by RBM-CF, leverages a neural autoregressive architecture with parameter sharing to enhance performance and supports extensions to deep learning models.

In the paper by Liu et al.[3] they focus on the deep collaborative filtering with probabilistic graphical modeling. It uses a deep neural network to model nonlinear interactions. It also maintains probabilistic structure which is important for making decisions and interpretation. The overall model of this paper shows that this approach improves recommendation when used as content. Because it explores a way to improve recommendation systems using extra information that it derived directly from the users.

The early development of collaborative filtering systems and their human-centered origins, which aimed to address information overload by filtering based on social relationships, as seen in early systems like GroupLens[15] and Tapestry[18]. These systems collected explicit or implicit ratings data from multiple users to deliver personalized recommendations, laying the foundation for modern CF techniques.

Several studies such as Lavanya et al. [9], Dong et al. [1] identify the main challenges with collaborative filtering i.e. the cold-start problems where new users or new items with very less or no interaction data are difficult to accurately be recommended, the scalability issues and including data sparsity, particularly for long tail movies with limited engagement. These limitations are particularly relevant to our research as we evaluate collaborative filtering's performance against content-based alternatives.

2.2 Content-Based Filtering Approaches

Content-based recommendation systems analyze item features such as genre, cast, directors, narrative themes, etc. to infer the user preferences. The core idea is to recommend items that are similar to those a given user has liked in the past. This involves representing both users and items in the same feature space, allowing for the computation of similarity scores between them. Recommendations are then made based on these similarity scores, suggesting items with high similarity to a user's previously preferred items[17]. We have seen several publications that have explored the effectiveness of these approaches in the movie recommendation context, such as the comparative analysis of Kurniawan et al. [10], says that content-based systems demonstrate higher accuracy with limited user data, making them particularly valuable for addressing the cold-start problem where sparse user data makes collaborative methods less effective. This is because CBF relies on item features rather than user-item interactions[22]

In the paper by Garden et al.[2], they use semantic features for example tags and annotations. These features can be important in describing the item characteristics and user preferences. In other words, they improve personalization. Their model uses these semantic features to bridge the gaps in sparse interaction matrices. This approach is useful when the data in not available.

Literature such as Lavanya et al. [9] constantly identify a major limitation in content based approaches, which is their inability to capture unexpected recommendations since they rely heavily on previous rated items leading to a filter bubble which will confine the user recommendations rather than diversifying them. Also, the Content-based systems may face challenges when item content is not readily available or difficult for a computer to analyze[22].

In the paper by Melville et al.[4], the content-based classifiers are trained using item features. After that the predictions are used to fill the missing entries of the matrix. The next step is to take this matrix and use for collaborative filtering. In simple words the predictions are added to the rating matrix which makes it more complete. This is useful when a new user has no interaction history. This approach makes recommendations more accurate especially when data is sparse.

2.3 Hybrid Approaches

Due to the established limitations of both content-based and collaborative filtering when used independently, significant research attention has turned towards hybrid recommendation approaches that aim to synergize their respective strengths [11]. Integrating CF and CBF techniques can yield substantial benefits, and many successful contemporary recommendation methods employ such hybrid architectures [15]. Burke [11] provides a comprehensive survey detailing various hybridization strategies, including weighted, mixed, switching, feature combination, cascade, feature augmentation, and meta-level techniques, while noting that many potential combinations remain underexplored and that pure methods can suffer from recommending items already known to the user (the "portfolio effect"). It provides a support for our research into developing and also evaluating a hybrid model.

Early pioneering hybrid systems laid important groundwork. For instance, Fab utilized content analysis of web pages to construct user profiles for collaborative recommendation [16]. Other formative approaches included the use of "filterbots", they are automated agents generating content-based ratings to enrich the user-item matrix for CF, and methods employing content-based predictors, such as naive Bayes, to impute missing values and create pseudo-user profiles, thereby enhancing the data available for CF algorithms [4]. Extending this idea of using content to bolster CF, Basu et al. [26] proposed framing recommendation as a classification task that leverages both social (collaborative) and content-based information.

More sophisticated integration techniques have followed. Lu et al. [17] developed Content-based Collaborative Filtering (CCF) specifically for news recommendation, addressing challenges related to rich content features and sparse data from long-tail users. CCF models user preferences via baseline parameters combined with latent factors derived from both the target item and its content-based neighbors, weighted by similarity metrics computed using techniques like Probabilistic Latent Semantic Analysis (PLSA) and empirical word distributions. They also explored using external search queries as auxiliary data to better link news items. In a related vein, Basilico et al. [15] proposed a unifying hybrid approach that learns a specific similarity function applicable to user-item pairs, demonstrating the power of merging diverse data sources within a single model.

The advent of deep learning has opened new avenues for hybrid systems. Dong et al. [1] introduced a hybrid model coupling matrix factorization with deep neural networks capable of capturing complex, nonlinear user-item interactions, thereby creating more informative representations and improving rating prediction accuracy compared to traditional matrix factorization alone. Their subsequent work introduced the additional stacked denoising autoencoder (aSDEA), extending the stacked denoising autoencoder framework to integrate auxiliary side information (e.g., user demographics, item metadata) explicitly to address cold-start and data sparsity issues [1]. This model tightly couples deep representation learning for side information with CF applied to the ratings matrix, showing improved performance over prior methods. However, the effectiveness of such models can depend on the quality and availability of side information, and like many deep learning approaches, they may face challenges related to interpretability and the need for large training datasets.

Knowledge graphs (KGs) represent another promising direction for enhancing hybrid recommender systems. Wang et al. [14] proposed the Knowledge-aware Path Recurrent Network (KPRN), which explicitly leverages KG structures by generating path representations between users and items and capturing sequential dependencies along these paths using a weighted pooling operation to discern path importance. The connectivity within KGs provides rich semantic information about entities and relations, complementing traditional user-item interaction data and facilitating a deeper understanding of user interests. Further emphasizing the role of KGs, Dhelim et al. [13] demonstrated improvements by integrating KGs to infer topic-item associations via semantic reasoning. Their Meta-Interest system utilizes graph-based meta-path discovery to model both implicit and explicit user interests. Wang et al. [14] also highlight the importance of explicit reasoning over KGs, developing end-to-end models to learn path semantics directly.

Despite the proliferation of hybrid models and techniques, comparative studies remain relatively scarce, especially those evaluating multiple distinct approaches (CF, CBF, and various hybrids) under consistent experimental conditions. Furthermore, while the development of hybrid systems continues rapidly, optimal strategies for integrating different components to effectively balance accuracy, scalability, diversity, and serendipity require further exploration.

3 Dataset

We are going to use the famous 100k Movielens dataset for our recommendation system project. The data set is sourced from the University of Minnesota, which is known for its reliability and availability to the public. This dataset also contains data about the user's demography which would be required for content-based recommendations. This dataset includes a wide range of movies and a diverse group of users which is helpful in this project to test the algorithms under various conditions. This data set consists of 1,00,000 ratings from 943 users on 1682 movies, each user has rated around 20 movies and simple demographic info for the users. The MovieLens dataset provides a broad spectrum of movies and diverse group of users, which means that it is ideal for testing the algorithms under various conditions. This diversity is going to help us in evaluating the robustness of our recommendation models.

3.1 Challenges

This dataset presents certain challenges. Some of these challenges are really common in real world recommendation systems. They can have a significant impact on the outcome of the model performance.

The first challenge is that users are not required to rate all movies which can lead to users giving ratings to movies that they particularly liked or disliked. This makes the data more sparse which makes it difficult for collaborative filtering model to give accurate results.

The second challenge is the date of the dataset which is 1998. This is not perfect to reflect on current viewers and trends. The expectations and the content has drastically changes. This dataset might not align with the present-day trends.

The third challenge: it can give a cold start problem. A cold start is when recommendation systems faces difficult because there is not information about the new users or the items. This is because the collaborative filtering heavily relies on past ratings. Instead the content-based filtering might solve this issue because it does rely on the historical data. it uses item features to compare new items with what the user already seen.

The fourth challenge that we can face is the bias in the ratings of the users. A users can change preferences over time because they maybe found new interest. But those new change in their preferences might not be noted. This is going to have an effect on the outcome for example give irrelevant suggestions and it might not be beneficial for the user.

4 Approach

We are using a Databricks Community Edition cluster for our model. We upload all our datasets into DBFS and created respective RDD's. After that we implemented our 3 said models in the following way.

4.1 Content-Based Recommendation System

We are trying to get similar movies using Feature Engineering. This will transform the movie dataset into a machine readable format easier for computation.

4.1.1 Load Movie Data: The first step is to load the movie dataset. Read movie metadata(genres, descriptions) from (u.item).

4.1.2 Feature Extraction: Convert these features into numerical vectors using TF-IDF (Term frequency-inverse document frequency).These vectors are then normalized.

4.1.3 Clustering or Similarity Computation: Use k-means clustering specifically BisectingKmeans() to group similar movies. BisectingKmeans is a type of hierarchical version of Kmeans clustering.

4.1.4 Generate Recommendations: For a given movie rated by a user, find similar movies in the same cluster. These movies shares same genre and have the similar taste.

4.2 Collaborative-Based Recommendation System

We are using Matrix Factorization using ALS (Alternating Least Squares). ASL is beneficial when solving problems related with large-scale recommendation.

4.2.1 Load Ratings Data: Read the user-item interactions dataset (u.data).

4.2.2 Preprocess Data: Convert raw data into a Spark DataFrame with columns userId, movieId, and rating.

4.2.3 Train ALS Model: Use ALS() from Spark ML to train a collaborative filtering model. Set parameters
ALS.train(ratings,rank,iterations,lambda) as ALS.train(ratingsRDD, 10, 10, 0.01).

4.2.4 Generate Recommendations: Predict ratings for test data(User 100) and recommend top movies for the given user.

4.3 Hybrid Recommendation System

We combined both the scores of CF and CBF each 50This provides a more balanced recommendation. The normalization of the scores gives a fairness in the contribution of each model.

4.3.1 Try Collaborative Filtering First: Get recommendations using ALS same as CF model

4.3.2 Fallback to Content-Based if No Ratings Exist: If a user has no rating i.e we are faced with cold start problem ,recommend movies based on similar movie clusters(as in CNB)

4.3.3 Combine Both: Else , merge both CF and CNB.

4.4 Light Graph Convolutional Networks(GCN)

This is the first latest model that we implemented based on the works of He et al.[30]. In contrast to traditional GCN's, lightGCN removes feature transformation and activation,while focusing solely on neighborhood aggregation in hopes to improve both accuracy and scalability in recommendations.

4.4.1 Load Ratings Data: Read the user-item interactions dataset(u.data)

4.4.2 Preprocess and Encode: Each user and movie is encoded to create mappings which are required to create a graph.The edges are then constructed from user-item interactions resulting in a bipartite graph.

4.4.3 Create Edge Index and Embeddings: To create index and embeddings we utilise Pytorch built-in format to represent connections in the graph we created.After initial embeddings are learned , they are refined through multiple layers.

4.4.4 Train LightGCN:. The model is finally trained using BPR or Bayesian Personalized Ranking(BPR) loss to optimize pairwise ranking.The final result of the user-item embeddings are obtained by summation across all GCN layers.

4.4.5 Generate Predictions: The predictions are computed by taking the dot product of user-item embedding.The recommendations are generated for given user ranking items based on the predicted scores.

5 Result

After implementing the four models i.e Content-Based, Collaborative Filtering, Hybrid model and LightGCN. We conducted multiple studies to compare the performance of the four models including ablation studies and **t-SNE plots** for Content-Based and Hybrid models and **OOD robustness, heteroscedasticity, and double descent** for Collaborative filtering and LightGCN.

Tests like **RMSE** and **MSE** mainly focus on absolute rating errors,which works well for collaborative filtering and LightGCN which is more like a regression problem but may not be the best metric for content-based or hybrid models,where recommendations are based on ranking.**Precision@K**,**Recall@K** and **f1-scores** are better suited for evaluating ranking based recommendations.

Where as testing parameters such **t-SNE plots** work great for visualising high-dimensional feature vectors in 2D to see how well movies cluster by genre or recommendations as is the case with Content-based recommendations. While ALS(Collaborative Filtering) does not create interpretable content vectors rather builds latent matrices for users and items hence its not suitable with it. Also, with LightGCN the embeddings are leaerned in a global colaborative space while t-SNE works locally. **OOD robustness** simulates cold start scenarios and shows how well the model generalizes to unseen distributions. Content-based recommender recommends based on user item-similarity hence there is no "new-user" or no cold-start problem. **Heteroscedasticity** checks whether prediction error increases with rating magnitude so works well with CF but not so much useful with CNB.Double Descent is a phenomenon wheres the models error first decreases, then increases and then again decreases.CF learns from data by minimizing prediciton error, and it includes a tunable model complexity parameter called rank, which controls the number of latent features.As we increases rank, the model changes from underfitting to overfitting and potentially changes back to better generalization which results in a double descent curve. In contrast, CNB relies on fixed feature extraction(in our case **TF-IDF**) without any learnable model parameters. Since there is no learning parameter or capacity to tune double descent curve is not applicable to it. The outcomes we got for our models were :-

Note:- All the tests were done for User ID:100

5.1 Collaborative Filtering

MSE: 0.397 **RMSE:** 0.630 **MAE:** 0.509
Precision@10: 0.78 **Recall@10:** 0.076 **F1-score:** 0.053

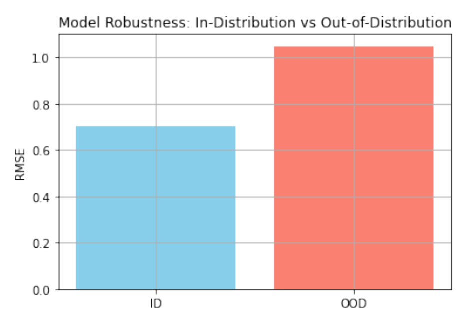

Fig. 1. Model Robustness: In-Distribution vs Out-of-Distribution (CF)

The above plot shows that the ALS model performs significantly better on users that were present during training.The RMSE is much higher for OOD users, which highlights that CF is vulnerable to cold-start problems.

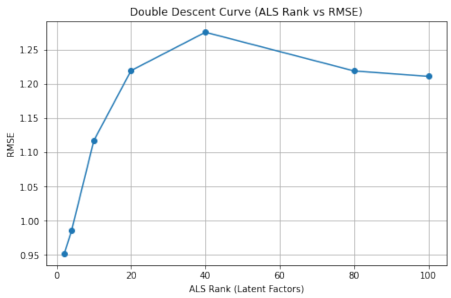

Fig. 2. Double Descent Curve (ALS Rank vs RMSE)

The double descent curve illustrates how error (RMSE) first decreases(underfitting),then increases (overfitting)and then decreasesagain as model complexity increases(rank increases).

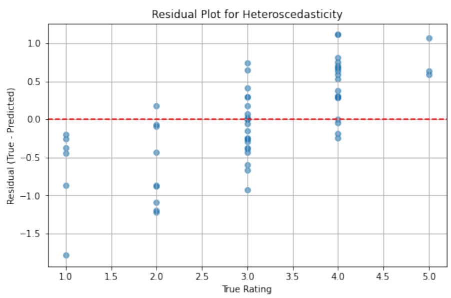

Fig. 3. Residual Plot for Heteroscedasticity (CF)

The residual plot signifies that prediciton errors are not uniformly distributed.More spread can be seen at higher true ratings,indicating heteroscedasticity.This shows the ALS model is less confident when predicting high ratings

5.2 Content-based Filtering

MSE: N/A **RMSE:** N/A **MAE:** N/A
Precision@10: 0.68 **Recall@10:** 0.081 **F1-score:** 0.061

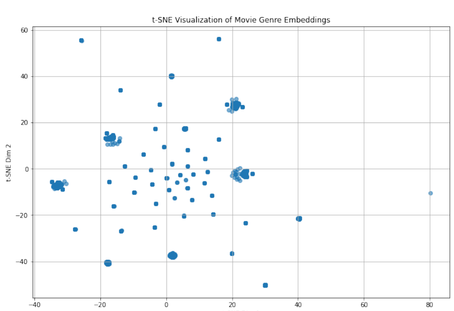

Fig. 4. t-SNE Visualization of Genre Embeddings (CNB)

This t-SNE plot shows that content-based TF-IDF algorithm combines movies group into meaningful clusters.This proves that movies with similar genres have close proximity in the embedding space,which supports effective similarity-based recommendations.

5.3 Hybrid Model

MSE: 2.747 **RMSE:** 1.657 **MAE:** 1.516
Precision@10: 0.85 **Recall@10:** 0.093 **F1-score:** 0.071

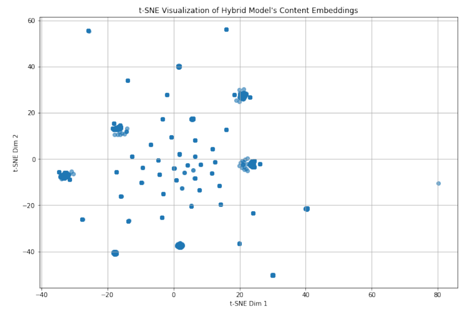

Fig. 5. t-SNE Visualization of Genre Embeddings (Hybrid)

The hybrid model combines the strengths of both Collaborative Filtering(ALS)and content-based clusters. This visualization confirms that the Content side still maintains the clusters allowing the model to fall back to genre-based when ALS fails(i.e in case of cold-start problem)

5.4 LightGCN

MSE: 22.09 **RMSE:** 4.7406 **MAE:** 1.7631
Precision@10: 0.76 **Recall@10:** 0.106 **F1-score:** 0.073

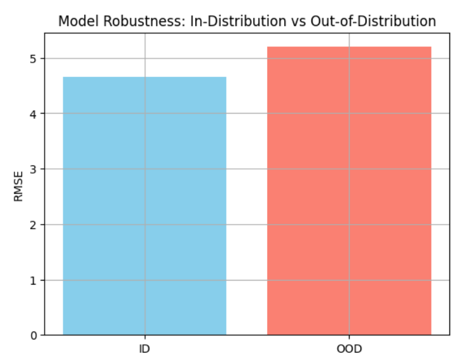

Fig. 6. Model Robustness: In-Distribution vs Out-of-Distribution (lightGCN)

The above plot shows that our models works a little better on OOD as compared to ID ,which indicates that on unseen data it performs siginificantly worse.

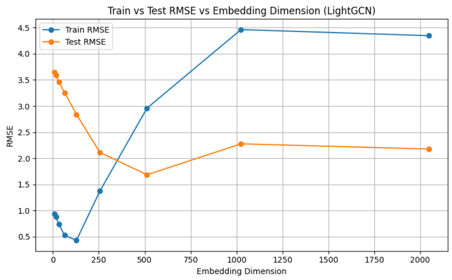

Fig. 7. Double Descent Curve:Train vs Test RMSE vs Embedding Dimension

The above plot shows that LightGCN exhibits double descent with respect to embedding dimension. Smaller embeddings underfit,while mid-size overfit, and large embeddings improve with high dimension learning.

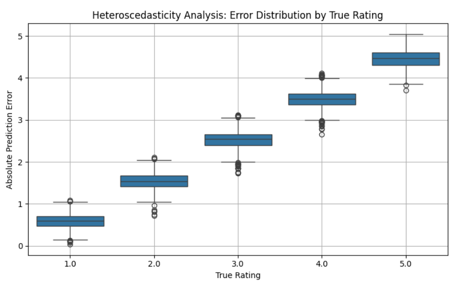

Fig. 8. Residual Plot for Heteroscedasticity (lightGCN)

The residual plot signifies that prediciton errors are not uniformly distributed.More spread can be seen at higher true ratings,indicating heteroscedasticity.This shows the ALS model is less confident when predicting high ratings

6 Conclusion

Based on our studies(both ablation and advanced evaluations like t-SNE,OOD,heteroscedasticity and double descent curve) we have arrived at the following observations regarding the three recommendation approaches :

6.1 Collaborative Filtering

Collaborative Filtering (CF) achieves the best performance in rating prediction showed by highest metrics such as MSE,RMSE and MAE.This confirms its strength in modeling user-item interactions which shows numerical accuracy.However, its lower recall and F1-scores shows that though accurate in nature the recommendationsa are limited in coverage.Furthermore,additional diagnostics revealed two key issues which are prevalent in Collaborative filtering techniques.

- **Cold-Start Sensitivity (OOD Robustness):** CF is unable to perform well for unseen users,which highlights that its dependent of prior user ratings/interaction.
- **Heteroscedasticity:** It increases with increasing rating magnitude,which suggests prediction variance is higher for strong user preferences.
- **Double Descent:** RMSE varies(indecrease then decreases then again increases) with latent factor size or rank,which confirms the model exhibits moder overparametrization effects.

6.2 Content-Based Filtering

Content-Based Filtering (CNB) does not produce numeric predictions,so number based metrics are not really usefull.Instead,it focuses on recommending similar items based on features(genre-based).It performed a bit worse than CF in Precision@10 but had better Recall and F1-score,which suggests broader coverage of relevant items which were not considered earlier.The t-SNE visualization of its TF-IDF model confirmed well structured clusters of movies,which validates its semantic grouping capabilities.However,CNB suffers from over-specialization and lacks adaptability and often recommends similar genres only.

6.3 Hybrid Approach

The Hybrid model which combines the plus points of both CF and CNB,achieved the highest overall performance in ranking metrics i.e Precision@10,Recall and F1-scores which indicates it generates both precise(plus point of CF) and diverse(plus point of CNB) recommendations.While its RMSE is higher than CF due to the introduced noise as a result of combining two models,it overcame CF's weaknesses :-

- **Cold Start:** CNB support improved performance for unseen users/items.
- **Coverage and Diversity:** Recommendations are more varied.(due to CNB)
- **Interpretability:** t-SNE confirms the above claims.

6.4 LightGCN

Instead of using traditional GCN,we used LightGCN,which is a graph-based collaborative filtering model,which simplifies traditional GCN by removing transformation layers.It achieves competitive RMSE :

- **OOD Robustness:** The performance on unseen users is not as good as when compared with ID users,which is a characteristic similar to traditional CF.
- **Heteroscedasticity:** The error variance could be seen increasing with higher ratings,which indicates rating-dependent reliability.
- **Double Descent Curve:** As common with collaborative filtering models RMSE follows a non-monotonic trend where it first underfits , then overfits and then recovers.

Note:-LightGCN's performance is promising for collaborative feature setups,but it requires more tuning and possible hybridization to address cold-start and diversity issue.

6.5 Final Recommendation

If accuracy or metrics is the primary goal then classis collaborative filtering with ALS is the most effective method.However, in real-word more than accuracy we require relevance, and in this aspect the

Hybrid approach shines among all 4 models.It offers the best balance of performance(accuracy) and practicality(relevance).Also,LightGCN shows very good promise for future graph-based recommendation models,it offers scalability,especially with fine tuning and further hybrid strategies.

7 References

[1] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang, "A Hybrid Collaborative Filtering Model with Deep Structure for Recommender Systems," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017, doi: 10.1609/aaai.v31i1.10747.

[2] M. Garden and G. Dudek, "Mixed Collaborative and Content-Based Filtering with User-Contributed Semantic Features." in *Proceedings of The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, (2006)

[3] J. Liu and D. Wang and Y. Ding,'PHD: A Probabilistic Model of Hybrid Deep Collaborative Filtering for Recommender Systems', in *Proceedings of the Ninth Asian Conference on Machine Learning*, 2017, vol.77, pp. 224–239,

[4] P. Melville, R. J. Mooney, R. Nagarajan, "Content-Boosted Collaborative Filtering for Improved Recommendations," *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, 2002, pp. 187-192 doi: 10.1609/aaai.v31i1.10747.

[5] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based Collaborative Filtering Recommendation Algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, 2001, pp. 285–295, doi: 10.1145/371920.372071.

[6] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009, doi: 10.1109/MC.2009.263.

[7] Su, Xiaoyuan & Khoshgoftaar, Tagdoi, "A Survey of Collaborative Filtering Techniques, " *Adv. Artificial Intelligence*, (AAAI) 2009, doi: 10.1155/2009/421425.

[8] C.-S. M. Wu, D. Garg, and U. Bhandary, "Movie Recommendation System Using Collaborative Filtering," in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, 2018, pp. 11–15, doi: 10.1109/ICSESS.2018.8663822.

[9] R. Lavanya, U. Singh, and V. Tyagi, "A Comprehensive Survey on Movie Recommendation Systems," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, Coimbatore, India, 2021, pp. 532–536, doi: 10.1109/ICAIS50930.2021.9395759.

[10] H. H. Kurniawan, W. S. Lukman, R. Fredyan, and M. A. Ibrahim, "Movie Recommendation System: A Comparison of Content-Based and Collaborative Filtering," *Procedia Computer Science*, vol. 245, pp. 860–868, 2024, doi: 10.1016/j.procs.2024.10.313.

[11] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User-Adapted Interaction*, vol. 12, 2002, doi: 10.1023/A:1021240730564.

[12] C. Chen, D. Li, Q. Lv, J. Yan, L. Shang, and S. Chu, "GLOMA: Embedding Global Information in Local Matrix Approximation Models for Collaborative Filtering", *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, Feb. 2017, doi: 10.1609/aaai.v31i1.10752.

[13] S. Dhelim, H. Ning, N. Aung, R. Huang and J. Ma, "Personality-Aware Product Recommendation System Based on User Interests Mining and Metapath Discovery," in *IEEE Transactions on Computational Social Systems*, vol. 8, no. 1, pp. 86-98, Feb. 2021, doi: 10.1109/TCSS.2020.3037040.

[14] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, "Explainable Reasoning over Knowledge Graphs for Recommendation", in *Proceedings of the AAAI Conference on Artificial Intelligence*, v. 33, n. 01, p. 5329-5336, 2019. doi: 10.1609/aaai.v33i01.33015329

[15] J. Basilico, T. Hofmann, "Unifying collaborative and contentbased filtering", in *Proceedings of the 21st International Machine Learning Conference*, ACM Press, 2004, NY, USA, pp 9. doi: 10.1145/1015330.1015394

[16] M. Balabanovic, Y. Shoham, "Fab: Content-Based, Collaborative Recommendation" in *Communications of the ACM* 40. 66-72. doi: 10.1145/245108.245124.

[17] Z. Lu, Z. Dou, J. Lian, X. Xie, and Q. Yang, "Content-Based Collaborative Filtering for News Topic Recommendation", in *Proceedings of the AAAI Conference on Artificial Intelligence*, [S. l.], v. 29, n. 1, 2015. doi: 10.1609/aaai.v29i1.9183.

[18] J. Konstan and L. Terveen, "Human-Centered Recommender Systems: Origins, Advances, Challenges, and Opportunities", in *AI* Magazine, [S. l.], v. 42, n. 3, p. 31-42, 2021. doi: 10.1609/aimag.v42i3.18142.

[19] W. Hongyi, "User-aware Recommender Systems: Algorithm and User-Interaction Design." Doctoral dissertation, Cornell University, 2021, doi: 10.7298/ns5e-4e66

[20] S. Rendle, Z. Li, Y. Koren, "On the difficulty of evaluating baselines: A study on recommender systems." in *arXiv preprint* 2019, doi: 10.48550/arXiv.1905.01395

[21] H. Steck, L. Baltrunas, E. Elahi, D. Liang, Y. Raimond, and J. Basilico, "Deep Learning for Recommender Systems: A Netflix Case Study", in *AI Magazine*, [S. l.], v. 42, n. 3, p. 7-18, 2021. doi: 10.1609/aimag.v42i3.18140.

[22] L. Si, J. Rong, "Flexible Mixture Model for Collaborative Filtering" in *International Conference on Machine Learning (2003).* doi: 10.5555/3041838.3041927

[23] R. Salakhutdinov, A. Mnih, G. Hinton, "Restricted Boltzmann machines for collaborative filtering." in *ACM International Conference Proceeding Series.* 227. 791-798. (2007) doi: 10.1145/1273496.1273596

[24] Y. Zheng, B. Tang, W. Ding, H. Zhou, "A Neural Autoregressive Approach to Collaborative Filtering", *Proceedings of The 33rd International Conference on Machine Learning*, in Proceedings of Machine Learning Research 48:764-773 , 2016, doi: 10.48550/arXiv.1605.09477

[25] J. L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. "An algorithmic framework for performing collaborative filtering" In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 1999).* Association for Computing Machinery, New York, NY, USA, 230–237. doi: 10.1145/312624.312682

[26] C. Basu, H. Hirsh, W. W. Cohen, "Recommendation as classification: Using social and content-based information in recommendation." In *Proceedings of the 15th National Conference on Artificial Intelligence*,Association for the Advancement of Artificial Intelligence (AAAI 1998) (pp. 714–720).

[27] M. Potter, H. Liu, Y. Lala, C. Loanzon, and Y. Sun, "GRU4RecBE: A Hybrid Session-Based Movie Recommendation System (Student Abstract)", In *Proceedings of the AAAI Conference on Artificial Intelligence*, [S. l.], v. 36, n. 11, p. 13029-13030, 2022. doi: 10.1609/aaai.v36i11.21651

[28] L. Zhao, Z. Lu, S. J. Plan, Q. Yang. "Matrix factorization+ for movie recommendation." In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (IJCAI'16). AAAI Press, 3945–3951, 2016. doi: 10.5555/3061053.3061171

[29] Z. Zhao, W. Fan, J. Li, Y. Liu, X. Mei, Y. Wang, Z. Wen et al. "Recommender systems in the era of large language models (llms)." In *IEEE Transactions on Knowledge and Data Engineering*, 2024. doi: 10.48550/arXiv.2307.02046

[30] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 639–648. doi: 10.1145/3397271.3401063.

Table 1. Team Member Contributions

Member	**Role/Contributions**
Ritik Verma	- Implementation of Models and Testing - Section 4: Approach - Section 5: Result - Section 6: Conclusion - Procuring Research Papers
Gurwinder Kaur	- Section 1: Introduction - Section 3: Dataset - Section 4: Approach - Procuring Research Papers - Proofreading and editing
Abdul Rahman Hussain Siddique	- Section 1: Introduction - Section 2: Related Work - Section 7: References - Proofreading and editing