BT4222 Mining Web Data for Business Insights Final Report

Movie Recommendation System

Recommendation System Group 5

Name	Matric Number
Bernice Liew Yee Ling	A0257618U
Chong Ke Lin, Mark	A0252467Y
Koh Ken Tze	A0251805J
(Andre) Lim Tze Chuan	A0217714H
Foo Jia Yi	A0257340J
Tan Hui Rong	A0216246J

Table of Contents

Abstract

In the era of digital expansion, managing data overload is paramount for improving user engagement and satisfaction. This project tackles the business challenge of delivering personalized movie recommendations to users to prevent information overload and enhance the viewing experience. The core machine learning problem addressed is the prediction of individual movie preferences based on historical viewing data, ratings, and demographic details of users.

The data used in this project comprises:

- The TMDB Dataset featuring 1 million movie entries across various genres and years.
- The MovieLens 1M Dataset, which includes user demographic data and movie ratings for films released before 2000.

These datasets were integrated to predict user preferences for a broad array of movies, including recent releases not covered by existing user data in the MovieLens dataset. Our methodology involved developing three distinct models:

1. A collaborative filtering model transitioning into a content-based filtering approach.
2. Two hybrid models employing both collaborative and content-based methods through advanced deep-learning techniques.

Key achievements and noteworthy highlights of the project:

- Resolution of the cold start and data sparsity issues through the use of embedding layers in deep neural networks, which provide dense, informative representations of users and movies, facilitating accurate predictions even with limited user interaction data.
- Implementation of diverse recommendation strategies in our models, ranging from random to engineered diversity, ensuring tailored recommendations that align with user preferences.
- Continuous improvement in model's performance with each iteration by experimenting with several architectural designs.
- Effective management of large-scale data challenges associated with a dataset of 1 million movies, adapting our models to handle hardware limitations.
- Development and engineering of a novel gender preference rating feature, to analyze and predict movie preferences across different gender demographics.

Proposal Review

Potential Contributions	Satisfactory Level
Integration of diverse movie options in recommendations	Satisfactory
Prediction of user preferences without historical data (Cold Start)	Satisfactory
Overcoming data sparsity and preventing model overfitting	Satisfactory
Incorporation of temporal dynamics in recommendations	Not Achieved
Development & refinement of a scoring system for recommendation relevance	Under Expectations
Effective merging of TMDB and MovieLens datasets for enhanced feature integration	Satisfactory
Hyperparameter tuning to enhance model performance	Satisfactory
Continuous improvement in model performance across multiple iterations	Exceed Expectations
Exploration of various models to identify optimal recommendation strategies	Satisfactory
Assessment of different features' impact on model performance	Satisfactory

Justifications:

- We achieved diversity by implementing both engineered and random inclusion strategies.
- Cold Start and Data Sparsity were addressed through the use of embedding layers.
- We attempted temporal binning of movies by decades; however, this approach did not yield a significant improvement in the model's performance. This led us to maintain a simpler, more effective model strategy.
- We developed a scoring system in Model 1 to assess the relevance of our movie recommendations. While it achieved high precision, the recall was low. This imbalance likely stemmed from Model 1's reliance on TMDB movies that exceeded a certain vote count threshold, suggesting a need for further balance and refinement in future enhancements.
- We observed clear and measurable performance enhancements across our model iterations, as evidenced by the consistent decrease in the MSE from Model 1 to Model 3. This progression showcases the effective evolution and optimization of our recommendation system.

Data Description

Our primary dataset, `Merged_df.csv`, is an amalgamation of the MovieLens dataset (GroupLens, 2003), which contains over 1 million ratings from approximately 6,040 users on 3,900 movies, and the TMDB Movies Dataset (Asaniczka, 2024), which features metadata for over 1,000,000 movies. The resulting merged dataset, `Merged_df`, consists of 995,655 rows and 38 columns, encompassing extensive user data and movie metadata.

Some User Features from the 'Merged_df' dataset

Feature	Description	Value Type	Statistics/ Categories
UserID	Distinct IDs of each user	Int	E.g. '1', '2' Range: [1, 6040] Number of Missing Values: 0
Rating	Rating of a movie given by a particular user	Int	E.g. '1', '2', '3', '4', '5' Range: [1, 5] Number of Missing Values: 0
Gender	Gender of user, 'M' or 'F'	String	E.g. 'M', 'F' Number of Missing Values: 0
Age	Age group of user, * 1: "Under 18" * 18: "18-24" * 25: "25-34" * 35: "35-44" * 45: "45-49" * 50: "50-55" * 56: "56+"	Int	E.g. '1', '18', '50' Number of Missing Values: 0
Occupation	Occupation of user in numerical form, from 0-20 * 0: "other" or not specified * 1: "academic/educator" * 2: "artist" * 3: "clerical/admin" * 4: "college/grad student" * 5: "customer service" * 6: "doctor/health care" * 7: "executive/managerial" * 8: "farmer" * 9: "homemaker" * 10: "K-12 student" * 11: "lawyer" * 12: "programmer" * 13: "retired" * 14: "sales/marketing"	Int	E.g. '0', '1', '2', '20' Range: [0, 20] Number of Missing Values: 0

	* 15: "scientist" * 16: "self-employed" * 17: "technician/engineer" * 18: "tradesman/craftsman" * 19: "unemployed" * 20: "writer"		

Some Movie Features from the 'Merged_df' dataset

Feature	Description	Value Type	Statistics/ Categories
MovieID	Distinct IDs of each movie	Int	E.g. '1', '500' , '3230' Range: [1, 1000000] Number of Missing Values: 0
Title	Title of a movie followed by its release year	String	E.g. 'The Crying Game (1992)' Number of Missing Values: 0
Genres	All applicable genres of a movie	String	E.g. 'Drama|Romance|War' Number of Missing Values: 0
vote_average	Average vote or rating given by viewers	float	E.g. '8.364', Mean: 6.95 ; Median: 7.00 ; SD: 0.908 Number of Missing Values: 0
vote_count	Total number of votes received for the movie	Int	E.g. '34495', Mean: 3313.43 ; Median: 1416 ; SD: 4651.41 Number of Missing Values: 0
adult	Indicates if the movie is suitable only for adult audiences	bool	E.g. 'True', 'False' Number of Missing Values: 0
overview	Brief description or summary of the movie	String	E.g. 'Irish Republican Army member Fergus forms an u…', Number of Missing Values: 495
runtime	Duration of the movie in minutes	Int	E.g. '110', Number of Missing Values: 0
popularity	Popularity score of the movie	float	E.g. '83.952', Mean: 24.731 ; Median: 20.140 ; SD: 19.008, Number of Missing Values: 0

Models and Performance

Hybrid Model 1

This model adopts a two-step approach, beginning with collaborative filtering and proceeding to content-based recommendations. Initially, we calculate a user similarity matrix using cosine similarity based on ratings of older movies, which predicts how users might rate similar movies. This matrix is then used to predict movie ratings for each user. The Root Mean Square Error (RMSE) of 1.02 was evaluated to check the accuracy of predicted ratings against actual ratings, confirming the model's effectiveness. Based on the highest predicted ratings, each user receives a recommendation for one older movie. In content-based filtering, we recommend five new movies from the TMDB dataset using both textual and numerical features, and a newly engineered feature, 'gender_pref', which predicts a movie's appeal based on gender. Text data from TMDB is processed using Word Soup and TF-IDF vectorization, chosen over BERT for its superior ability to capture keyword relevance and semantic features aligned with user preferences with lower computational costs. We selected the XGBoost classifier for its high accuracy in predicting 'gender_pref'. To manage computational demands, we divided the TMDB dataset into `tmdb_above_200` for robust cosine similarity calculations and `tmdb_below_200` to incorporate serendipity through less-reviewed movies. We compute a batch cosine similarity matrix, prioritizing 'vote_count', 'vote_average', 'popularity', 'year', to ensure recommendations focus on newly released, highly-rated movies. Ultimately, each user's top collaborative recommendation is succeeded by four tailored new movie suggestions based on content similarities and one random, less-reviewed movie to introduce a serendipitous surprise.

	Model 1 using unscaled features and excluding TF-IDF embeddings	**Model 1 using scaled features and TF-IDF embeddings**
Precision	1.0	1.0
Recall	0.36336021975009747	0.4290272228173163
F1 score	0.5330362650843679	0.6004465358910273

Table 1: Performance results of Hybrid Model 1

Hybrid Models 2 and 3

Models 2 and 3 incorporate both collaborative and content-based filtering techniques enhanced by deep learning to offer personalized recommendations. Both models share a fundamental architecture tailored for a recommendation system.

Common Architecture:

Input layer	Take in User ID, Movie ID and user attributes, catering to personalized recommendations.

Embedding layer	Convert User IDs and Movie IDs into dense vectors, encapsulating latent user and movie factors critical for capturing interaction patterns.
Flatten layer	Flatten embedded vectors to 1D for processing in subsequent network layers.
Dot product	Computes the interaction between user and movie vectors through their dot product, a key element in collaborative filtering, predicting user ratings based on these interactions.
Concat	Merges outputs from collaborative and content-based components for a unified approach.
Output	A final dense layer outputs the final predicted rating, summarizing learned information into a single prediction rating value for a particular movie.

Hybrid Model 2 focuses on integrating collaborative and content-based filtering techniques. The model utilizes dense layers to process user attribute vectors with ReLU activation, aiming to transform raw user data into a predictive format effectively. However, it exhibits signs of overfitting, potentially attributed to high dimensionality and the intentional omission of movie features during training. *Hybrid Model 3* advances upon the foundations set by Model 2 by incorporating more sophisticated neural network techniques to further leverage both user and movie metadata. This model includes additional dense layers with dropout regularization to prevent overfitting and discern complex patterns (Figure 1). It also utilizes a sigmoid activation function scaled to the typical 1-5 rating scale to refine the accuracy of the output predictions. These enhancements significantly improve performance by effectively addressing data sparsity and the cold start problem, leading to more robust and accurate recommendations.

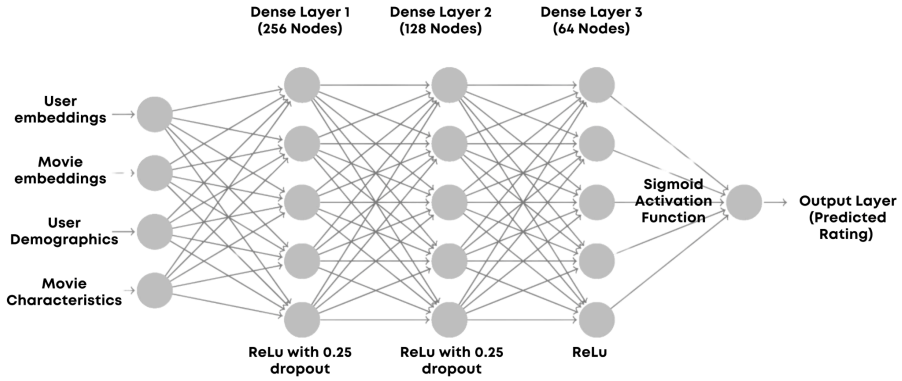

Figure 1: Hybrid Model 3's Neural Network Architecture

MSE of Predicted Ratings	Model 2	Model 3
(Training Data)	(After 5 Epochs) 0.6098	(After 5 Epochs) 0.3684
(Test Data)	(After 5 Epochs) 0.7734	(After 5 Epochs) 0.9198

Table 2: Performance results of Hybrid Models 2 and 3

Contribution and Justification

Enhancing Dataset Integration

The enhancement of datasets in our movie recommendation system significantly contributed to the robustness and accuracy of our recommendations. By strategically merging the MovieLens dataset, which provided rich user information, with the TMDB dataset, known for its comprehensive movie metadata, we created a holistic view of both user preferences and movie characteristics. This integration involved meticulous data cleaning and merging based on movie titles and release years. We employed fuzzy matching techniques to ensure high accuracy in data alignment, focusing particularly on the normalization of title formats and removal of extraneous punctuation to facilitate accurate data merging. These steps helped handle discrepancies in movie titles, ensuring that titles from both datasets were accurately linked, even with minor variations. This careful attention to data integration significantly enhanced the dataset's comprehensiveness, reducing errors and enriching the final dataset to serve as a foundational element of our project, demonstrating high effort and effectiveness.

Creative Feature Engineering

We also developed a sophisticated movie scoring algorithm (Figure 2) that refines our movie predictions and enables us to measure the precision and relevance of our recommendations. This algorithm incorporates a range of weighted attributes: revenue, runtime, budget, gender preference, and adult content, which are normalized between -1 and 1 to moderate their influence. Furthermore, vote average, vote count, and popularity are scaled between -2 and 2 to emphasize movies with higher user ratings, while release year is scaled between -3 and 3 to favor recent movies. We also employed TF-IDF normalization on text vectors to capture the semantic significance of each movie. The sum of these components yields a comprehensive score that reflects the nuanced relevance of movies to individual users. By setting a relevance threshold, our system filters and flags only movies that meet this criterion, effectively addressing the challenge posed by the absence of labeled test data in the TMDB dataset. This methodology is particularly innovative as it allows us to gauge the relevancy of our predictions effectively, overcoming a common issue in recommendation systems where scoring cannot be performed without a labeled test set (Bondarenko, 2019). This approach not only enhances the interpretability of our model but also addresses a significant challenge that has historically impacted the field of machine learning in recommendation systems. In summary, our creative feature engineering reflects a high level of effort, while showing promising effectiveness. Moving forward, further tuning of weighted attributes based on insights into their importance will be crucial for enhancing the robustness and effectiveness of our scoring system.

$$\text{movie\_score} = S_1(X_{N1}) + S_2(X_{N2}) + S_3(X_{N3}) + TFIDF$$

$S_1(X_{N1})$: 'revenue', 'runtime', 'budget', 'gender_pref', 'adult' scaled to [-1, 1], balancing influence

$S_2(X_{N2})$: 'vote_average', 'vote_count', 'popularity' scaled to [-2, 2], prioritising highly-rated movies

$S_3(X_{N3})$: 'year' scaled to [-3, 3], prioritising newer movies

$TFIDF$: Text vectors, normalized, capturing semantic importance

Figure 2: Our movie scoring system to determine relevance

Design/Adaptation of new ML methods/architecture

Our recommendation system incorporates a sophisticated hybrid neural network that utilizes both collaborative and content-based filtering techniques. This architecture is designed to effectively leverage deep learning to enhance both the accuracy and personalization of recommendations.

Our model builds upon seminal research and recent advancements in neural recommendation systems:

- He et al. (2017), "Neural Collaborative Filtering (NCF)" — This foundational paper introduces neural architectures into the collaborative filtering framework, and demonstrates how NCF can significantly outperform traditional methods like matrix factorization and item-based collaborative filtering, especially on sparse data. Our model extends these concepts by incorporating user demographics and movie characteristics into the embeddings, thus enhancing the model's ability to understand nuanced user preferences.
- Covington et al. (2016), "Deep Neural Networks for YouTube Recommendations" — Demonstrating the effectiveness of deep neural networks in large-scale recommendation systems, this study inspired our adaptation of similar techniques to the movies domain, supplemented with enriched contextual data like genres and popularity, for a more tailored approach.

The integration of these sophisticated neural techniques demanded a high degree of effort, particularly in adapting and tuning the models to work effectively with our specific dataset and objectives. The effectiveness of our approach is rated as high due to its substantial impact on prediction accuracy as the MSE of the predicted ratings for model 3 is the lowest compared to the models 1 and 2. Post prediction, our system employs a novel method to introduce diversity into the recommendations. This step is distinct from the neural network architecture and is influenced by:

- Kunaver and Požrl (2017), "Diversity in Recommender Systems – A Survey": Acknowledging the importance of diversity, this survey highlights various strategies to achieve it. Inspired by these concepts, our model calculates a diversity score for the next top 10 movies based on genre

dissimilarity and recommends the movie that attained the highest diversity score. This provides a set of quality recommendations, while exposing users to a broader range of content.

Our implementation is novel in the combination of deep learning techniques with a post-model diversity enhancement strategy. This approach allows us to address common issues such as the over-concentration of popular items and the underrepresentation of niche content, fostering a balanced and engaging user experience. The efforts and outcomes associated with this model mark a substantial advancement in the personalization and sophistication of recommendation systems.

Creativity in addressing the gap between ML outcomes and business decision making

Movie recommendation systems face a significant challenge in balancing user preferences with revenue optimization (Ghanem, 2022). Businesses often prioritize user satisfaction by recommending movies based solely on user ratings and preferences. However, achieving long-term revenue goals requires aligning recommendations with broader business objectives, such as increasing sales. This demands innovative approaches in ML models to enhance customer satisfaction while strategically driving revenue through targeted recommendations. While leveraging predicted ratings from ML models is crucial, our model surpasses this by integrating diversity into recommendations. This layer of analysis addresses a critical gap in traditional recommendation systems—the limited diversity often observed in movie recommendations. Unlike conventional models, our system considers diversity metrics to recommend not only popular movies but also 'unusual' ones that users may enjoy. This strategy introduces refreshing content experiences, ultimately enhancing customer satisfaction, retention rates, and aligning short-term user preferences with long-term business goals and revenue optimization. Additionally, we experimented with incorporating temporal dynamics into our recommendation system by binning movies into decades to accurately reflect evolving preferences of users over time. Despite these efforts, the model's performance did not improve significantly, prompting us to adopt a simpler model. Nevertheless, this aspect remains open for further exploration and refinement to better capture dynamic shifts in user interests and optimize revenue generation effectively. In summary, our approach demonstrates a medium level of effort and effectiveness within ML practices, with ongoing potential to refine diversity metrics, address temporal dynamics, and incorporate business-specific KPIs. Ongoing enhancements in this area hold promise for advancing the alignment between ML outputs and broader business decision-making, addressing critical gaps in traditional recommendation systems.

Self-Evaluation Table

Contributions		Low	Medium	High
Use valuable and high-quality new datasets, including integrating existing datasets, scrawling or retrieving data, etc.	Effort			✓
	Effectiveness			✓
Creativity in feature engineering in a way that increases the interpretability of the models.	Effort			✓
	Effectiveness		✓	
Design or adaptation of new ML methods/architecture or the integration of existing methods with a balance of resource and cost	Effort			✓
	Effectiveness			✓
Creativity in identifying and resolving the gap or confusion between ML outcomes and business decision making	Effort		✓	
	Effectiveness		✓	

References

Bondarenko, K. (2019, February 23). Precision and recall in recommender systems. And some metrics stuff. Kirill Bondarenko. Retrieved April 16, 2024, from https://bond-kirill-alexandrovich.medium.com/precision-and-recall-in-recommender-systems-and-some-metrics-stuff-ca2ad385c5f8

Covington, P. (2016, September 7). Deep neural networks for YouTube recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems. Retrieved April 18, 2024, from https://doi.org/10.1145/2959100.2959190

Ghanem, N. (2022, September 12). Balancing consumer and business value of recommender systems: A simulation-based analysis. Electronic Commerce Research and Applications. Retrieved April 15, 2024, from https://www.sciencedirect.com/science/article/pii/S1567422322000783

He, X. (2017, April 3). Neural collaborative filtering. In Proceedings of the 26th International Conference on the World Wide Web. Retrieved April 18, 2024, from https://doi.org/10.1145/3038912.3052569

Kunaver, M. (2017, May 1). Knowledge-Based Systems. ScienceDirect. Retrieved April 19, 2024, from https://doi.org/10.1016/j.knosys.2017.02.009

Feldges, C. (2022, April 2). Text Classification with TF-IDF, LSTM, BERT: a comparison of performance. Retrieved April 19, 2024, from https://medium.com/@claude.feldges/text-classification-with-tf-idf-lstm-bert-a-quantitative-comparison-b8409b556cb3

Dataset links: TMDB Movies Dataset, MovieLens 1M Dataset