# KNN & Redige Regression Applied Movie Recommendation Model

Bohan Shan (bohans3)

Pete Chen(xc74)

Jingchen Liu(jl288)

School of Information Sciences, UIUC

IS 507: Final Paper

Professor Haohan Wang, Issac Joy

Dec 13, 2024

## Abstract

The rapid growth of streaming platforms and the production of thousands of movies annually have made it increasingly difficult for users to find movies that suit their preferences. This study develops a personalized movie recommendation system that combines machine learning techniques and diverse movie metadata to provide accurate and tailored recommendations. Using a Kaggle dataset, the system integrates content-based and collaborative filtering approaches, employing K-Nearest Neighbors (KNN) and Ridge Regression to analyze user preferences and movie attributes. Key features include prioritizing movies from the same collection, matching genres, ranking by popularity when necessary, and ensuring recommendations align with user interests. Rigorous testing demonstrated the system's adaptability and effectiveness in predicting preferences, reducing search time, and helping users discover hidden gems. This project addresses the growing demand for efficient and personalized entertainment solutions, laying the groundwork for future advancements in recommendation algorithms.


*Keywords*: Movie recommendation system, personalized recommendations, streaming platforms, collaborative filtering, K-Nearest Neighbors (KNN), Ridge Regression, user preferences, hybrid model, movie metadata, Kaggle dataset, content-based filtering, machine learning, recommendation accuracy, data analysis, predictive modeling.

## Introduction

In today's fast-paced world, the growing number of streaming platforms and the production of 7,000 to 10,000 movies each year make it harder for people to choose what to watch. With so many options, users often feel overwhelmed and struggle to decide on a movie during their leisure time. This highlights the need for a compelling movie recommendation system that can simplify the process. A well-designed system can analyze movie data, including genres, popularity, and ratings, to guide users toward movies that match their interests. In conclusion, Movie Recommendation Systems have become indispensable tools for users to navigate the vast amount of available Content(Modi, 2023).

Movie recommendation systems help users discover films they will enjoy by focusing on the features they like most (Jayalakshmi et al., 2022). These systems make choosing a movie easier and improve the overall viewing experience by helping people connect with stories that resonate with them. At the same time, they allow streaming platforms to better serve their audiences by tailoring content to individual preferences.

Most of the movie recommendation systems in the industry don't use users' previous view or rating history as they require data (Gupta et al., 2020). Our study focuses on building a recommendation system that makes movie selection easy and personalized. Using a detailed movie dataset from Kaggle, we combined collaborative filtering through K-Nearest Neighbors (KNN) with Ridge Regression to predict user preferences. By finding patterns in user behavior and analyzing movie features, the system aims to recommend films that match each user's tastes.

The hybrid approach taken here has several benefits. In previous studies, the hybrid approach shows a significant increase in the accuracy of the recommendation engine than the traditional approaches (Salmani & Kulkarni, 2021). Collaborative filtering identifies similarities between users and movies, while Ridge Regression provides more precise predictions by analyzing specific movie attributes. Together, these techniques make the system more flexible and accurate, catering to many users. The goal is to create a smooth and enjoyable experience, helping people quickly find movies they will love in an increasingly crowded market.

## Background/Related Work

Personalized recommendation systems have become more and more significant in the entertainment sector in recent years due to the growth of online streaming platforms and the rise in the consumption of digital content. One important application in this area is movie recommendation systems, which are revolutionizing how people find and watch cinema content.

We reviewed online sources and engaged in in-depth discussions on the topic, and determined to build a Movie Recommendation Model. From the article “What Is Movie Recommendation System and how we build it” , we learned some basic information about movie recommendation systems, including system filtering movies, and how the system chooses recommended movies to users(Sharma,2024). Movie Recommendation is an intelligent system that applies machine learning techniques to analyze user behavior and movie features, thereby providing personalized movie recommendations. Its core objective is to predict a user's preference for movies they have not yet seen through algorithms and recommend movies they are most likely to find interesting. One of the important algorithms we planned to use is Machine learning. Machine learning algorithms are often the foundation of movie recommendation systems, which analyse user behaviour and data to produce tailored recommendations(Reddy & Swathi, 2020).

Based on the article, recommendation systems are primarily divided into two categories: content-based filtering and collaborative filtering(Sharma, 2024). Content-based algorithms utilize the similarity between movies to recommend new ones, while collaborative filtering leverages overlapping movie ratings from other users to generate recommendations. In recent years, with the rapid advancement of machine learning technologies, recommendation systems have achieved significant improvements in both accuracy and personalization. From early content-based filtering to collaborative filtering and now deep learning models, recommendation algorithms have evolved continuously. One of the typical examples would be Netflix. The Netflix Prize competition further spurred research in this field, leading to numerous innovative algorithms(Sharma, 2024).

Our project aims to develop a Python-based movie recommendation system that leverages open-source datasets and machine learning algorithms to provide personalized movie recommendations for users. We will employ collaborative filtering techniques to analyze user behavior and preference patterns to generate recommendations. Through this system, we hope to enhance users' movie-watching experience while also offering valuable user insights to streaming platforms. The project will be implemented in Python, using a combination of collaborative filtering and content-based recommendation algorithms. By analyzing user rating data and movie metadata from the Movie dataset, we will build a system capable of accurately predicting user preferences and providing personalized recommendations.

In the following sections, we will provide a detailed overview of the system's design concepts, the dataset used, the core algorithms, and the implementation process. With this project, we aim to gain a deeper understanding of how recommendation systems function and explore how machine learning techniques can be applied to solve real-world problems.

## Proposed Method / Experiment Setup

1. Data Collection: The first step of our project was to identify the right dataset to meet our requirements. Choosing an appropriate dataset is crucial for ensuring accurate and effective computations in later stages.

   We sourced our dataset from Kaggle, which contains 45,468 movies in a single CSV file. This dataset includes comprehensive information such as:

   a. Movie collections
   b. Genres
   c. Titles
   d. Movie IDs and IMDb IDs
   e. Popularity scores
   f. Vote counts
   g. Average voting scores

   With this rich dataset, all subsequent computations were performed using the Python programming language.

2. Data Preparing:
   a. Users input a movie name they are interested in, and expect the model will return top 5 most related high score movies.
   b. Model reads the dataset and user ratings from the .csv file from Kaggle.
   c. Extract key information like “collection_id”, “genres”, “popularity”, “vote_average” to prepare for related movie filtering.
   d. Drop invalid or unrelated data from the dataset to make sure the data will be clean for training.
3. Data Training:
   a. Constructs a matrix where rows represent users, columns represent movies, and cells contain user ratings (defaulting to 0 if missing).
   b. Training the Ridge Regression Model - every movie’s features includes “CollectionID”, “Genre”, “Vote Average”, and “Popularity”. Then the model will automatically pair these features with user ratings to train an Ridge Regression Model. And the trained model will predict the user's preference for other movies.
   c. KNN: Uses KNN to predict ratings based on movie similarity and user preferences.
4. Data Filtering:
   a. The data filtering process includes several steps, results need to meet several requirements:
      i. Same Collection: Recommended movies need to have the same collection with the input movie.
      ii. Exact Genre: Recommended movies need to have exact same genres with the input movie.
      iii. Partial Genre Overlapping: If you can not meet the first two requirements, find movies with overlapping genres with the input movie.

iv. Popular Movies: If you can not meet all three requirements above, rank by popularity.

b. Rank and Sorting: The model will assign scores to each movies it found:
   i. **3** for collection-based matches.
   ii. **2** for genre-based matches.
      - For individual movies that have multiple genres, the function prioritizes selecting movies with the exact matched multiple genres.
   iii. **1** for popularity-based matches.
   iv. After all the weight scores are assigned, the model sorts the list based on the highest review scores descending.

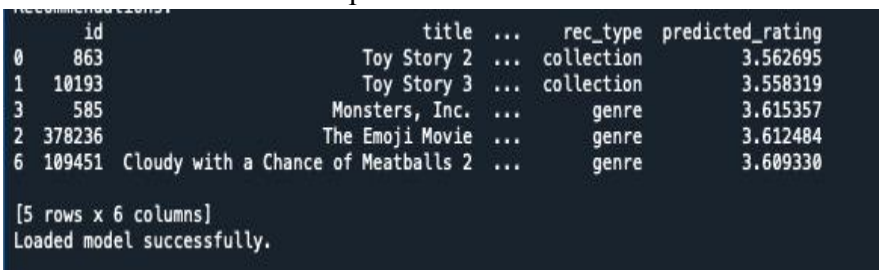
c. Evaluate the accuracy of prediction:
   i. Splits user ratings into training and control groups.
   ii. Tests the Ridge and KNN models on the control group.
   iii. Evaluates accuracy using RMSE and MAE for both models.
   iv. Identifies the better model (Ridge or KNN) for the current user.

5. Data Testing:

The last step is data training. In this step, we input different movie names to our model, testing the system to make sure the final results (Top 5 recommended movies related to the movie name we input) are meeting our expectations.

## Results

The Results about Ridge Regression prediction based on movie characteristics met our expectations. Below are some examples:



| | id | title | ... | rec_type | predicted_rating |
|---|---|---|---|---|---|
| 0 | 863 | Toy Story 2 | ... | collection | 3.562695 |
| 1 | 10193 | Toy Story 3 | ... | collection | 3.558319 |
| 3 | 585 | Monsters, Inc. | ... | genre | 3.615357 |
| 2 | 378236 | The Emoji Movie | ... | genre | 3.612484 |
| 6 | 109451 | Cloudy with a Chance of Meatballs 2 | ... | genre | 3.609330 |

[5 rows x 6 columns]
Loaded model successfully.

When input is “Toy Story,” the module returns the following top 5 results:

1. **Toy Story 2**
2. **Toy Story 3**
3. **Monsters, Inc.**
4. **The Emoji Movie**
5. **Cloudy with a Chance of Meatballs 2**

These results align with our movie filtering process:

- **Same Collection** → Prioritize movies from the same collection as "Toy Story." (Toy Story 2 and Toy Story 3)
- **Exact Genre Match** → Recommended movies have the exact genre set with “Toy Story”. (Monsters Inc., The Emoji Movie and Cloudy with a Chance of Meatballs 2)
- **Partial Genre Match** → Recommended movies have at least the same genre type with “Toy Story”.

- **Popularity** → Rank remaining options by popularity.

As a result, the module successfully recommended movies that are all cartoons and share genres with "Toy Story."

To ensure the module performs well across different types of movies, we tested it with "The Godfather" (an American gangster film). In this case, we expected results like **The Godfather: Part II**, **The Godfather: Part III**, and **American Gangster.**

```
    id                        title  ...    rec_type  predicted_rating
0  240       The Godfather: Part II  ...  collection          3.594455
1  242      The Godfather: Part III  ...  collection          3.561208
5  627                Trainspotting  ...       genre          3.673067
2  278     The Shawshank Redemption  ...       genre          3.627124
3  311  Once Upon a Time in America  ...       genre          3.584103

[5 rows x 6 columns]
Loaded model successfully.
```

The results met our expectations:

- The Godfather: Part II and The Godfather: Part III were identified as they share the same collection value with "The Godfather."
- Movies like Trainspotting, The Shawshank Redemption, and Once Upon a Time in America were recommended as they share similar genre sets with "The Godfather."

Then we tested K-Nearest Neighbors (KNN) and Ridge Regression to find out the best training model for the specific user according to this user's preferences.

Example: Which model will be the best training model for user 2?

```
In [8]: %runfile '/Users/xuchen/Desktop/untitled folder 2/UserBasedREC_KNN_Ridge.py' --wdir
Please enter target user ID (input 'q' to quit): 2
User 2 control group:
    userId  movieId  rating  timestamp
79       2      527     4.0  835355731
77       2      509     4.0  835355719
82       2      550     3.0  835356109
87       2      587     3.0  835355779
74       2      497     3.0  835355880

Evaluation results for user 2:
Ridge RMSE: 0.6833
Ridge MAE: 0.5987
KNN RMSE: 0.6325
KNN MAE: 0.4000
Best Model: KNN
Number of test movies: 5

Evaluation details (including actual ratings and predicted ratings):
   movieId  actual_rating  best_prediction
0    527.0            4.0              4.0
1    509.0            4.0              3.0
2    550.0            3.0              3.0
3    587.0            3.0              4.0
4    497.0            3.0              3.0
Number of movies used for training user 2: 71
Please enter target user ID (input 'q' to quit):
```

The best training model for user 2 is KNN.

Further, we use Integrated Model to find out recommended movies to user 2 due to user's personal movie preferences, below is the result:

```
Predicted rating by user 2 for movie 222848: 3.216089827237256
Predicted rating by user 2 for movie 439050: 3.6677355232438087
Predicted rating by user 2 for movie 111109: 3.255951910504303
Predicted rating by user 2 for movie 227506: 3.037651160594214
Predicted rating by user 2 for movie 461257: 3.704333383988559
Recommended movies for user 2:
          id                  title  predicted_rating  popularity  \
9752    2357       10 Items or Less               5.0    3.898236
16032  54507  A Very Potter Musical               5.0    1.054258
14067  78320     Blood and Concrete               5.0    0.897851
13842   1673       Comanche Station               5.0    3.657790
1767     845   Strangers on a Train               5.0   15.417730

       vote_average
9752            6.6
16032           7.7
14067           7.3
13842           6.2
1767            7.6
Please enter the target user ID (enter 'q' to exit the program):
```

Using the best training model tailored for User 2, and based on their personal movie preferences, our system identified the movies most likely to interest them (predicted rating: 5.0/5.0). The system also displayed the actual popularity of these movies and their IMDb ratings, along with their vote averages and total points (out of 10).

## Discussion

Summary of Achievements:

1. Our integrated movie recommendation system successfully combines content-based and collaborative filtering approaches to provide personalized movie suggestions.
2. We've successfully implemented a movie recommendation system that contains multiple prediction models including Ridge Regression and K-Nearest Neighbors (KNN), increasing the system's accuracy and versatility.
3. Our movie recommendation system has a comprehensive evaluation framework that can predict according to different user's preferences on movies.

Insights Learned:

1. The system demonstrated strong predictive capabilities, accurately estimating user ratings for unseen movies based on individual user profiles.
2. Since each user has a unique profile and distinct movie preferences, some users benefit more from KNN-based predictions, while others achieve better results with Ridge Regression.

Possible Real-World Impacts:

1. Significantly reduces the time users spend searching for a "good movie" that aligns with their preferences and interests.
2. By predicting ratings for unseen movies, the system helps users discover "hidden gems" they may have overlooked, enhancing overall user satisfaction and engagement.

Limitation :

1. Our integrated model still has much to improve, due to the complexity of the metadata and model structure, there are chances that the KNN & Ridge User prediction regression select the same movies both in the training group and control group. Although the control group sample drawing function ensures that once a movie is selected as a control group, it will be excluded in the training group for each user drawing, but when there is a movie that is reviewed by multiple users, the duplicate existence of a movie in both control group and training group is inevitable, causing a flaw in our KNN & Bridge regression accuracy.
2. Moreover, our model has large future space for efficiency improvement. For our integrate model, it takes minutes to run the data in our dataset file, refactoring and modularizing key components, such as preprocessing, feature extraction, and model training, would enhance readability and maintainability

## Conclusion

The recommended hybrid approach combines content-based and popularity-based recommendation techniques. The underlying premise is to prioritize movies relatively close to the user's interests as depicted by his/her choice. Choosing movies from the same collection and with similar genres means that the system will try to recommend certain movies related to the user's interests. A weighted scoring method allows a trade-off between similarity and popularity, providing the user with some familiar and/or fresh content. This might help address the cold start problem frequently faced by collaborative filtering systems. The movie recommendation system illustrates AI working to enhance the personalized entertainment experience. With diverse data sources, advanced algorithms, and a hybrid approach, this system lays the groundwork toward building an intuitive and accurate content discovery platform.

As the digital entertainment dimension keeps accelerating with time, systems such as the one we devised will play an important role in determining how an audience interacts with media, rendering the vast universe of cinema much more accessible and delightful for everyone. As the system continues to evolve and scale, it is more than just an enhancement of an algorithm; rather, it is an extension of how people discover and engage with cinema. The committed efforts toward improving the functionalities have the potential to redefine the entire experience of video consumption, making it more approachable, intuitive, and equipped to cater to diverse preferences.

This journey is about a broader vision, connecting people to stories that speak to them, with a view of enhancing appreciation for film as one form of universal expression and storytelling.

## References

UpGrad. (n.d.). *Create your own movie recommendation system using Python*. UpGrad. Retrieved December 13, 2024, from https://www.upgrad.com/blog/create-your-own-movie-recommendation-system-using-python/

Jayalakshmi, S., Ganesh, N., Čep, R., & Senthil Murugan, J. (2022). Movie Recommender Systems: Concepts, Methods, Challenges, and Future Directions. Sensors (Basel, Switzerland), 22(13), 4904. https://doi.org/10.3390/s22134904

Reddy, A. B., & Swathi, S. (2020). *Movie Recommendation System Using Collaborative Filtering Techniques*. IRE Journals, 4(7), 77–81. Retrieved from https://www.irejournals.com/formatedpaper/1704718.pdf

Gupta, M., Thakkar, A., Aashish, Gupta, V., & Rathore, D. P. (2020). Movie recommender system using collaborative filtering. *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 415–420. https://doi.org/10.1109/icesc48915.2020.9155879

Modi, P., Kumar, A., & Kapoor, B. (2023). Filmview: A review paper on movie recommendation systems. *IRE Journals, 6*(12). https://www.irejournals.com/formatedpaper/1704718.pdf

Salmani, S., & Kulkarni, S. (2021). Hybrid movie recommendation system using Machine Learning. *2021 International Conference on Communication Information and Computing Technology (ICCICT)*, 1–10. https://doi.org/10.1109/iccict50803.2021.9510058