Anime Recommendation System

CMP 49412 Project

Ahmed Mehaisi B00094989

Mohammed Imtiaz B00097219

Dheeman Gangopadhyay B00092265

Dr. Alex Abraham Aklson

April 25, 2025

Abstract

Anime fans today face an overwhelming choice of 20,000-plus titles spread across streaming platforms. Our project builds an end-to-end recommendation engine that allows users to enjoy their anime experience and uncover hidden gems. We combine three main techniques—(i) non-personalized popularity models (weighted and Bayesian scoring), (ii) content-based recommendations based on synopsis and genres (TF-IDF and Jaccard), and (iii) User- and Item-based collaborative filtering into a hybrid pipeline that adapts to every stage of the user life-cycle. Cold-start users first receive high-quality, universally enjoyed anime as recommendations from the non-personalized section, and as soon as they have a handful of ratings, the content-based section takes over, and once sufficient ratings are present, the collaborative filtering (CF) section dominates. Our findings show that item-based collaborative filtering achieved the best accuracy (RMSE = 1.21 on a 10 scale) while User-based was just below (RMSE = 1.36 on a 10 scale). Additionally, content details allowed us to improve diversity without sacrificing relevance. The resulting system provides timely, personalised, and sound recommendations even for brand-new users or freshly released shows.

Introduction

Anime refers to hand-drawn and computer-generated animation originating from Japan. Outside Japan and in English, anime refers specifically to animation produced in Japan (Ashcraft, 2022). Streaming services and community sites have changed anime consumption, viewers can instantly access thousands of series involving comedy, romance, shōnen, action, and more. Trying to find another anime to watch can be simply overwhelming. Additionally, platforms such as Netflix combine recommendations across all kinds of across various categories, combining western movies, TV shows, and documentaries, resulting in anime recommendations being diluted. While choice is abundant, deciding what to watch next has become a genuine issue, with the search for another anime often replacing the joy of discovery. Recommendation systems solve this information overload by filtering the catalogue down to a shortlist that matches each viewer's taste, helps new users get started, and exposes veterans to unfamiliar but rewarding anime. Overall, a well-designed recommendation system can significantly improve user experience by filtering out irrelevant options and highlighting content that matches individual interests. The primary goal of this project is to develop a recommendation system tailored for anime, specifically, we aim to answer the research question:

How can we combine content features (such as synopsis and genres) with user ratings to provide relevant and diverse anime recommendations for both new and experienced viewers?

Data Description & Exploratory Data Analysis (EDA)

The data was compiled by webscraping a major anime discussion forum called My Anime List (Hernan, 2020). The ratings date back to 2020. My Anime List is the largest non-Japanese platform for anime watchers, with a global user base that actively rates, reviews, and discusses thousands of anime titles. This vast and diverse dataset helps provide valuable insights into user preferences and viewing patterns.

Two primary datasets were used:

anime_meta.csv-

- anime_id: A unique ID for each item
- anime_name: Name of the anime
- Genres: The categories the item belongs to.
- Synopsis: A brief description/ Summary of the item
- Score: Individual user score for the item (Out of 10)

This dataset was used for Content-Based Recommendations.

ratings.csv-

user_id: A unique ID for each user.

anime_id: A unique ID for each item

rating: User-given rating out of 10.

anime_name: Name of the anime

This dataset was used for Collaborative Filtering methods.

Data Preprocessing: Several steps were taken to clean and prepare the data:

1. **Missing values** – For the anime_df, rows with ‘Unknown’ or NaN in name, genre, or score were dropped. While for rating_df, rows with ‘Unknown’ or Nan in user_id, anime_id, or rating were checked, but none were found.

This was done because missing or unknown values could cause errors during model training.

2. **Score recalculation** – In anime_df, items repeated with individual user scores. The user scores were combined, resulting in each item appearing once with one Score - the average user rating.

   This was done to give a more meaningful measure of each anime's popularity and quality.

```python
# Compute the true average Score per anime
avg_scores = (
    anime_df
      .groupby('anime_id')['Score']
      .mean()
      .reset_index()
)
#Pull one representative row per anime_id (keep all other columns)
#drop duplicates based on anime_id, keeping the first
anime_meta = anime_df.drop_duplicates(subset='anime_id').copy()

# 3) Drop the old per-user Score from that
anime_meta = anime_meta.drop(columns=['Score'])

# 4) Merge the average scores back
anime_df = anime_meta.merge(avg_scores, on='anime_id', how='left')
```

The shape of anime_df: (3000, 6)

Number of unique animes in anime_df: 3000

The shape of rating_df: (1435890, 4)

Number of unique users in rating_df: 10000

\

The unique genres that exist:

```
['Action', 'Adventure', 'Comedy', 'Drama', 'Sci-Fi', 'Space',
 'Mystery', 'Shounen', 'Police', 'Supernatural', 'Magic', 'Fantasy',
 'Sports', 'Josei', 'Romance', 'Slice of Life', 'Cars', 'Seinen',
 'Horror', 'Psychological', 'Thriller', 'Super Power',
 'Martial Arts', 'School', 'Ecchi', 'Vampire', 'Military',
 'Historical', 'Dementia', 'Mecha', 'Demons', 'Samurai', 'Game',
 'Shoujo', 'Harem', 'Music', 'Shoujo Ai', 'Shounen Ai', 'Kids',
 'Hentai', 'Parody', 'Yuri', 'Yaoi'], dtype=object)
```

The top 10 anime and their score according to anime_df:

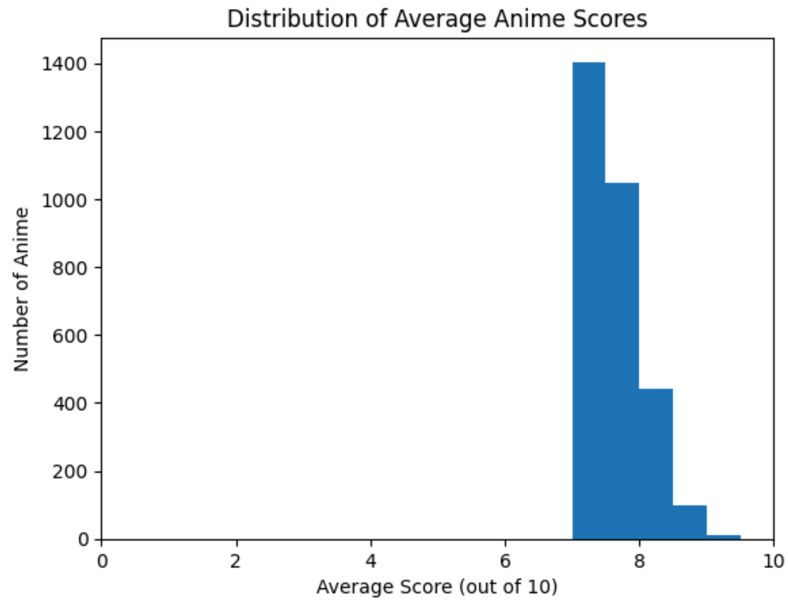
Column: anime_name		Score
1	Fullmetal Alchemist: Brotherhood	9.19
2	Shingeki no Kyojin: The Final Season	9.17
3	Steins;Gate	9.11
4	Hunter x Hunter (2011)	9.10
5	Shingeki no Kyojin Season 3 Part 2	9.10
6	Gintama°	9.10
7	Gintama'	9.08
8	Ginga Eiyuu Densetsu	9.07
9	Gintama': Enchousen	9.04
10	Koe no Katachi	9.00

This matched myAnimeList's top anime rating from the end of 2020.

Rank	Title	Score
1	**Fullmetal Alchemist: Brotherhood** TV (64 eps) Apr 2009 - Jul 2010 1,777,286 members **Manga Store** Volume 1 **$6.99** Preview	9.23
2	**Steins;Gate** TV (24 eps) Apr 2011 - Sep 2011 1,450,929 members	9.13
3	**Gintama°** TV (51 eps) Apr 2015 - Mar 2016 313,855 members	9.12
4	**Hunter x Hunter (2011)** TV (148 eps) Oct 2011 - Sep 2014 1,197,308 members	9.12
5	**Ginga Eiyuu Densetsu** OVA (110 eps) Jan 1988 - Mar 1997 189,727 members	9.11
6	**Gintama'** TV (51 eps) Apr 2011 - Mar 2012 300,658 members	9.09
7	**Shingeki no Kyojin Season 3 Part 2** TV (10 eps) Apr 2019 - Jul 2019 588,666 members **Manga Store** Volume 1 **$10.99** Preview	9.07
8	**Gintama': Enchousen** TV (13 eps) Oct 2012 - Mar 2013 180,871 members	9.05

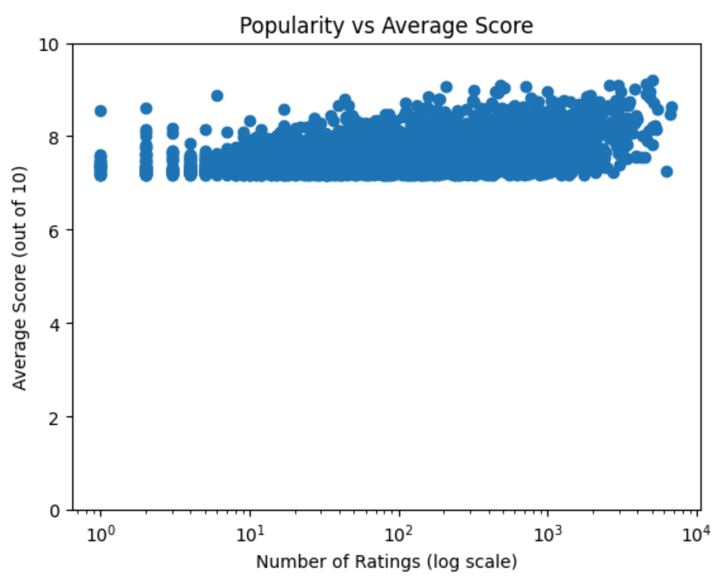
[2]

1) Distribution of Average Anime Scores (Histogram)



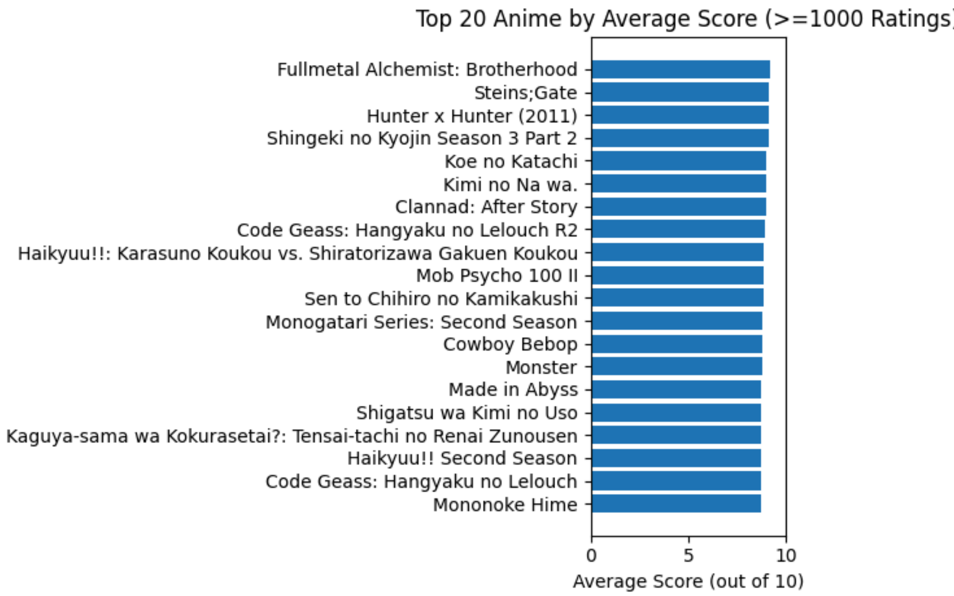
Most anime cluster between **7.0 and 9** out of 10. Very few titles score below 6 or above 9. It confirms the community is generous: ratings are left-skewed. When we later compute RMSE, errors can look large on paper but are small in the context of such a compressed range.

2) Popularity vs Average Score (Scatter Plot)



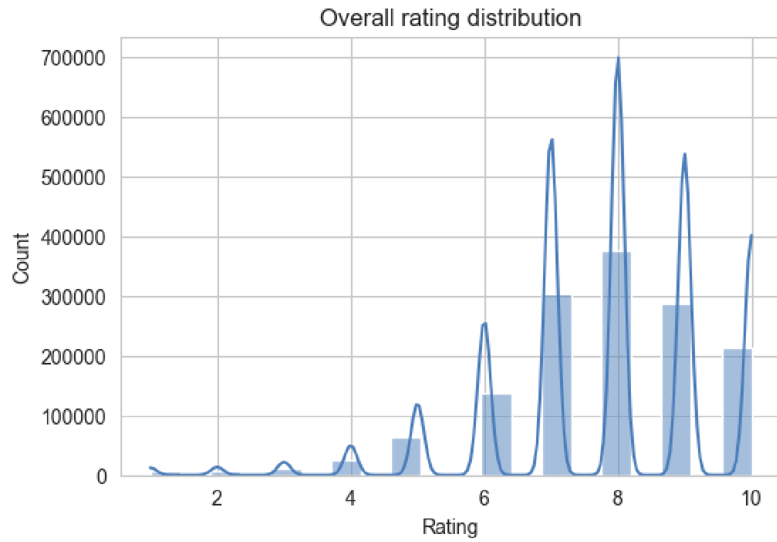
A weak upward trend: shows with thousands of ratings tend to have slightly higher average scores, but the spread is wide. Many obscure anime also sit in the 7–8 zone. Popularity does not mean quality, so a good recommendation system should balance both. This will be revisited when we discuss the fundamental principles of recommendation systems.

3) Top-20 Anime by Score



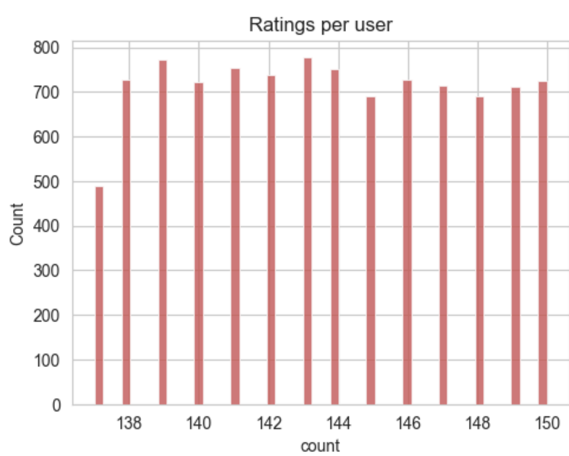
Fan-favourites such as “Fullmetal Alchemist: Brotherhood” and “Steins; Gate” dominate the leaderboard, all scoring around 9/10. This list will help formulate some of our Non-Personalized Recommendation algorithms.

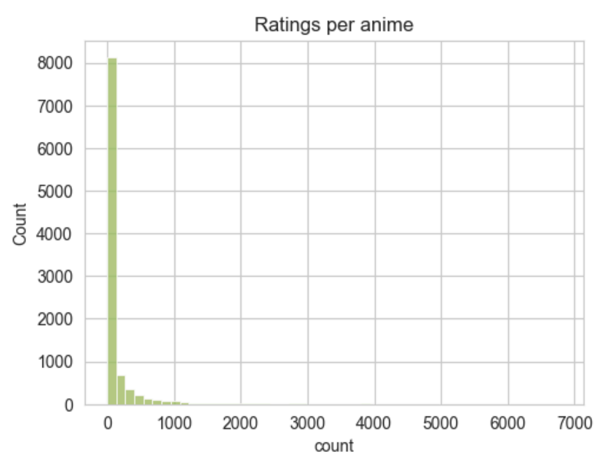
4) Overall Rating Distribution (Histplot)



The peaks reveal that users mostly prefer round scores/ integers. Very low ratings (<4) are almost absent. This means that most users are optimistic about the anime they watched.

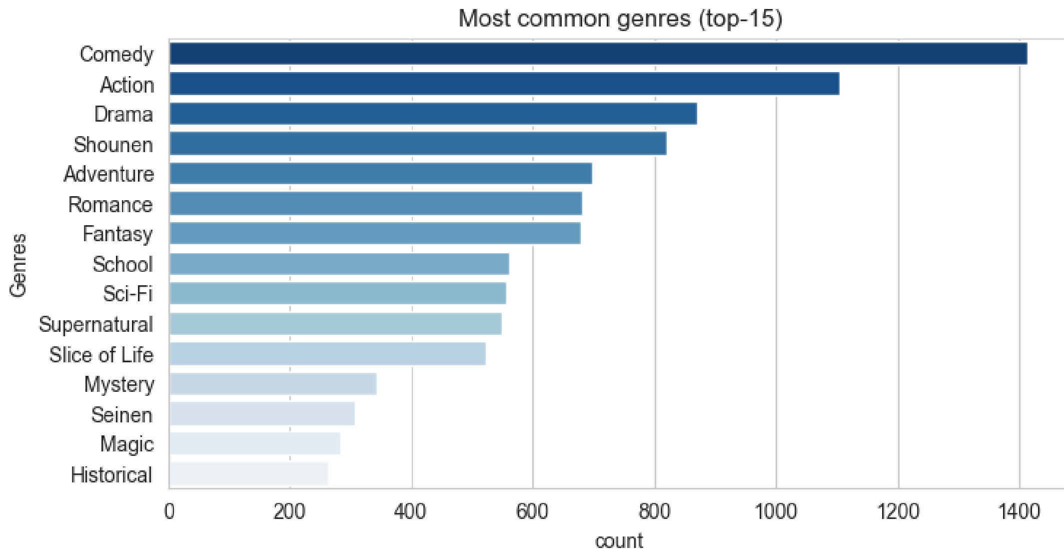
5) Ratings-per-User & Ratings-per-Anime





Most users rate fewer than 150 shows; a tiny minority rate hundreds. Likewise, most anime receive <100 ratings, while a handful have thousands. This shows that we have a sparse user-item matrix.

6) Most Common Genres (top-15 bar chart)



Comedy and Action lead by a wide margin, followed by Drama, Shounen, and Adventure. This imbalance could end up pushing the most common genres over and over, drowning out niche categories like Mystery or Historical.

Methodology

Several recommendation approaches were used, each with distinct characteristics:

Non-Personalized Recommendations

Popularity-Driven Average Rating:

This method involved taking the average of all items whose number of ratings is above a certain threshold. We first drop any anime that has fewer than T ratings (e.g., T = 100). Among the remaining items, we rank them by their plain arithmetic mean. The threshold reduces small-sample bias and ensures quality, not just engagement.

Weighted Scoring:

This ranking balances *how many* people rated an anime with *how well* they rated it. We rescale each show's number of ratings and its average rating to a 0–1 range (so the two quantities are comparable). We combine these values with weights. We chose W1 and W2 as 0.5 as we want to give equal importance to popularity and average rating. Solves the problem of small sample bias by ensuring that both rating and popularity contribute to ranking. Weighted scoring balances popularity and quality, ensuring that items with high ratings but low votes don't dominate.

$$\text{Weighted Score} = (\mathrm{W}_1 \times \text{Norm. Popularity}) + (\mathrm{W}_2 \times \text{Norm. Average Rating})$$ [5]

Bayesian Scoring:

Here we treat every average rating with statistical caution: titles with only a handful of votes are "pulled" toward the catalogue-wide mean until they collect more evidence. The adjusted score is

$$\text{Bayesian Scoring} = \frac{C \times m + \sum r_i}{C + n}$$ [5]

When n is small, the global mean m dominates, and as n grows, the anime's average takes over. This automatically accounts for small-sample bias while letting genuinely popular, well-rated shows rise to the top as more viewers weigh in.

Personalized Recommendations -

- Content-Based Filtering:
    - Description: Content-Based Filtering is a recommendation technique that suggests similar items by analyzing the content itself such as genres, synopsis, and themes without relying on user-related features like ratings or interactions.
    - Strengths:
        - It can be personalized since it directly matches items to a user's past preferences and interests.
        - Can recommend items to new users as long as some of their item preferences are known, which can be viewed as a solution to the cold start problem.
        - Easy to explain why an item was recommended (e.g., "because you liked action movies, here are similar action movies").
    - Weaknesses:
        - The system tends to recommend items similar to what the user has already seen, making it harder to introduce diverse or surprising recommendations.
        - New items that have little content information or metadata may not be recommended effectively.
        - Over time, users may only be shown very similar types of content, limiting discovery.
        - The model can not be evaluated using any metric other than A/B testing or manually deciding if the recommendations make sense.

Collaborative Filtering (CF):

User-Based:

$$\hat{r}_{u^{(1)}i} = \frac{\sum\limits_{j \in N_3(u^{(1)})} r_{u^{(j)}i}}{|N_3(u^{(1)})|}$$

` [5]

Item Based:

$$\hat{r}_{u,i} = \frac{\sum_{j \in N_k(i)} \mathrm{sim}_{\mathrm{adj}}(i,j)\, r_{u,j}}{\sum_{j \in N_k(i)} \mathrm{sim}_{\mathrm{adj}}(i,j)}$$

- Description: Collaborative Filtering relies on ratings, and user-based CF takes the position that users with similar past behaviour will have similar future behaviour. Item-based CF finds items similar to those the user liked based on co-occurrence patterns in user ratings.
- Strengths:
  - It works purely from user behavior (e.g., ratings, clicks), so detailed content information is not required
  - The more user interaction data available, the better and more accurate the recommendations become.
- Weaknesses:
  - As the number of users and items grows, the computational cost to find similar users or items can become very high.
  - Cannot make recommendations for new users or new items due to lack of rating data.
  - In large systems, users typically interact with only a small subset of items, leading to sparse interaction matrices, which can reduce recommendation quality.

- Hybrid Methods:
  - Description: Integrating multiple recommendation techniques to improve accuracy and overcome the limitations of individual methods.
  - Strengths:
    - Adapts to different scenarios (e.g., new users vs. established ones).

- Weaknesses:
    - Requires careful tuning to balance contributions from different methods.
    - Needs continuous monitoring to balance method contributions.

To serve users at every stage of their journey, we deploy three purpose-built hybrid recommenders and switch between them as soon as more ratings given by the user are available. For brand-new visitors, who have provided no interactions, we rely on a Hybrid Non-Personalized model that merges a weighted-score ranking with a Bayesian Scoring. This combination promotes animes that are both widely rated and reliably good, while still giving new animes a fair chance. We assign equal weights to Weighted Scoring and Bayesian Scoring because they correct complementary biases—Weighted boosts popular titles while Bayesian protects against small-sample hype— Equal weights benefit both without favouring one bias-reduction over the other.

Once a user has expressed only a handful of interests (e.g., clicked a few shows or stated favourite genres), we move to a Hybrid Content-based Model, combining TF-IDF similarity of plot synopses with Jaccard similarity on genres. As it needs no rating history, this model can handle the initial/new user phase well and maintain recommendation diversity. Within Hybrid Content, we give TF-IDF the larger share because synopsis text captures richer narrative detail than broad genre tags, making cosine similarity a stronger signal of true story-level likeness; Jaccard genres merely refine that list for thematic consistency.

After a user rates more anime, we upgrade to a Hybrid Collaborative Filtering model. This model allows for a mix of user-based and item-based CF, where the item component receives the larger weight (0.7) (it delivers lower RMSE, is less prone to user-bias, and as items are more stable long-term), while the user component adds finer personal nuances. This strategy lets the system evolve from broad, trustworthy suggestions to highly personalised lists, smoothly dealing with sparsity and cold-start issues along the way.

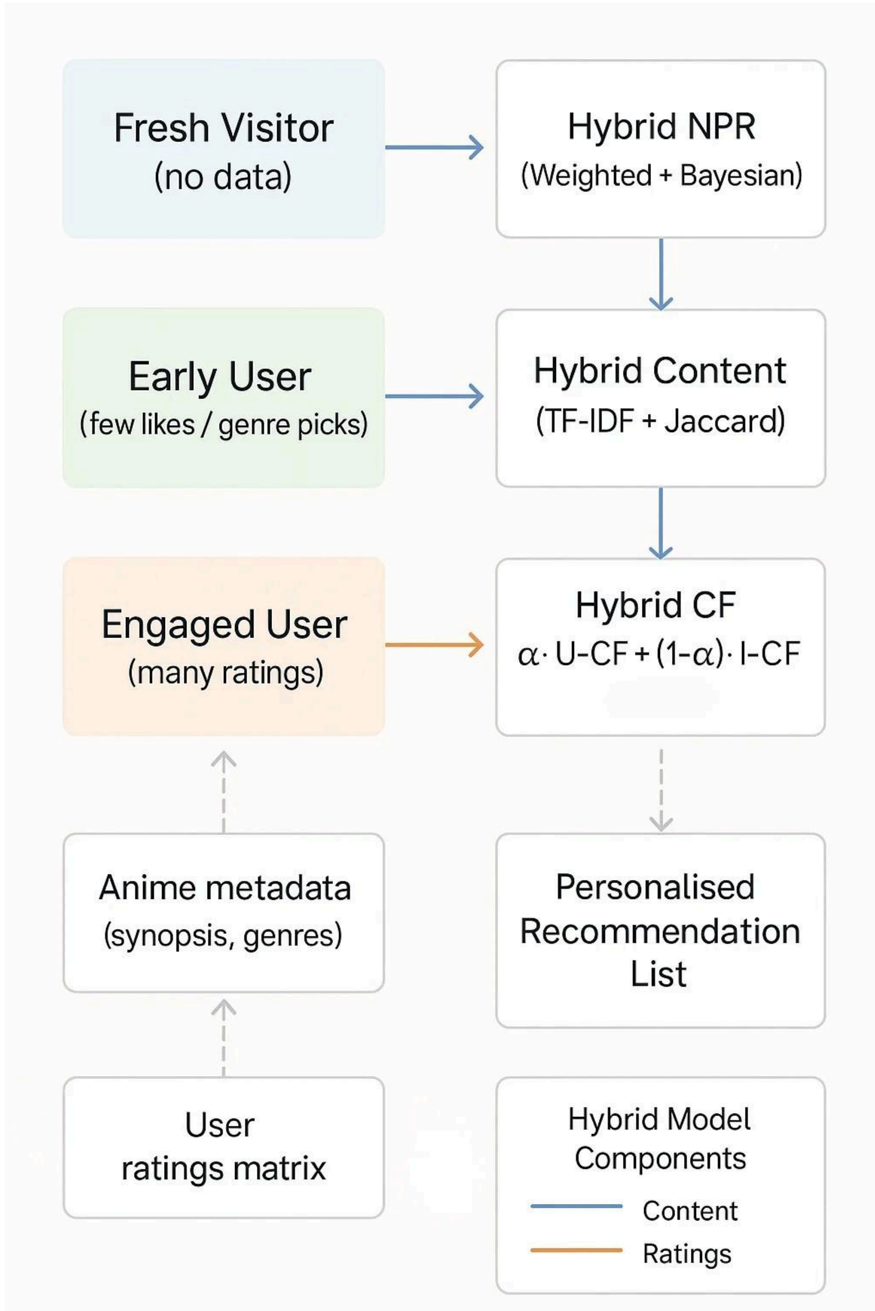


Diagram depicting the User Pipeline

Results

Popularity Driven-Average Rating:

```
                       anime_name
             Ginga Eiyuu Densetsu
Fullmetal Alchemist: Brotherhood
                         Gintama°
                      Steins;Gate
                         Gintama'
           Hunter x Hunter (2011)
                          Gintama
              Gintama': Enchousen
       Owarimonogatari 2nd Season
       3-gatsu no Lion 2nd Season
```

Weighted Scoring:

```
                       anime_name
                       Death Note
               Shingeki no Kyojin
Fullmetal Alchemist: Brotherhood
                    One Punch Man
                Sword Art Online
 Code Geass: Hangyaku no Lelouch
                        Toradora!
                     Angel Beats!
                      Steins;Gate
                    Kimi no Na wa.
```

Bayesian Scoring:

```
                       anime_name
  Fullmetal Alchemist: Brotherhood
                       Steins;Gate
             Hunter x Hunter (2011)
Shingeki no Kyojin Season 3 Part 2
                           Gintama
              Clannad: After Story
                          Gintama'
                          Gintama°
Code Geass: Hangyaku no Lelouch R2
                   Koe no Katachi
```

Non-Personalized Hybrid (Bayesian and Weighted):

To generate a reliable list of top anime, we developed a hybrid scoring system that combines both Bayesian average rating and a weighted rating score based on the number of user ratings. This method balances the quality of ratings with the quantity of ratings to avoid biases toward either niche or overly popular titles.

```
We recommend you these 10 Anime:
Fullmetal Alchemist: Brotherhood
Steins;Gate
Death Note
Code Geass: Hangyaku no Lelouch R2
Kimi no Na wa.
Hunter x Hunter (2011)
Shingeki no Kyojin
Code Geass: Hangyaku no Lelouch
Sen to Chihiro no Kamikakushi
Koe no Katachi
```

Content-Based:

As aforementioned, evaluating a content-based approach in the form of quantifying some metric is not possible. Therefore, recommendations were manually (subjectively) evaluated and the model interestingly performed very well. The model was asked to generate its top four recommendations for the famous anime called One Punch Man. Below are the resulting recommendations from the genre filtering side of the model:

```
One Punch Man 2nd Season
One Punch Man 2nd Season Specials
One Punch Man Specials
One Punch Man: Road to Hero
```

Furthermore, we used One Punch Man a second time as a tester for the hybrid content-based model (text and genre-based), with a higher weight given to the text based approach due to our datasets rich synopsis feature.

```
{'top': One Punch Man 2nd Season
 One Punch Man Specials
 One Punch Man 2nd Season Specials
 One Punch Man: Road to Hero
```

Even though it was subjectively evaluated, it would be very easy to convince someone that the content based aspect of the model is working swimmingly.

User-Based:

```
Anime title
-----------------------------------------------------
Kimi to Boku. 2
Texhnolyze
Megalo Box
Kimi to Boku.
JoJo no Kimyou na Bouken Part 4: Diamond wa Kudake
Yama no Susume: Second Season
Yakusoku no Neverland
Kaguya-sama wa Kokurasetai: Tensai-tachi no Renai
Kizumonogatari III: Reiketsu-hen
Owarimonogatari 2nd Season
```

An Example Run for User 36

We evaluated our user-based model using the Root Mean Squared Error (RMSE) metric, which measures the average magnitude of prediction errors between the actual and predicted user ratings. Our model achieved a test RMSE of **1.3639**, meaning that the predicted ratings, on average, were about 1.36 points away from the true ratings. We treated this value as a baseline for future improvements and optimizations to the model.

Item-Based:

```
Kore ga UFO da! Soratobu Enban
Futatsu no Kurumi
Glass no Kamen: Sen no Kamen wo Motsu Shoujo
Penguin's Memory: Shiawase Monogatari
Moonfesta
True Tears Epilogue
Genshiken Nidaime OVA
Houkago Initiation
Zetsubou Funsai Shoujo ∞ Amida
Amada Anime Series: Super Mario Brothers
```

An Example Run for User 36

We also evaluated our item-based collaborative filtering model using the Root Mean Squared Error (RMSE) metric. The model achieved a test RMSE of **1.2099**, indicating that, on average, the predicted ratings were about 1.21 points away from the true ratings.

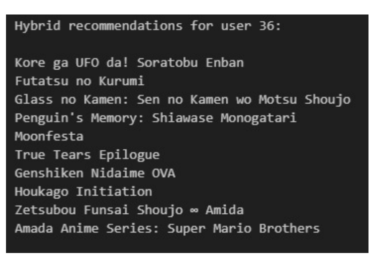
An RMSE of 1.2099 suggests stronger predictive performance compared to the user-based model, showing that item-based filtering was better able to capture and generalize user preferences in our dataset. This result highlights the advantage of using item similarities in cases where users' rating patterns are sparse or inconsistent.

Important Note:

It is incredibly important to note that the ratings used are based on a scale of 1 to 10 and not the common 1 to 5; this distinction makes a massive difference when considering the RMSE score; higher rating ranges naturally lead to higher RMSE scores compared to smaller scales. Considering this, both obtained RMSE scores are reasonable and reflect a relatively great performance given the dataset and project constraints.

Hybrid Collaborative Filtering:

To further personalize recommendations, we implemented a hybrid collaborative filtering approach tailored to individual users. The image above shows the output for User 36, where the system generated a top ten list by combining both user-based and item-based collaborative filtering signals. This hybrid method leverages explicit user ratings to identify similar users and items, then combines the results to produce a better recommendation list. The recommendations include lesser-known titles such as *Kore ga UFO da! Soratobu Enban* and *Glass no Kamen: Sen no Kamen wo Motsu Shoujo*, demonstrating the system's ability to find relevant but non-mainstream content. This supports our goal of increasing serendipity and personalization, as the recommendations are tailored to the user's taste profile rather than plain popularity. Overall, this output highlights the model's capacity to adapt to individual user preferences, especially for users with rich rating histories.



```
Hybrid recommendations for user 36:

Kore ga UFO da! Soratobu Enban
Futatsu no Kurumi
Glass no Kamen: Sen no Kamen wo Motsu Shoujo
Penguin's Memory: Shiawase Monogatari
Moonfesta
True Tears Epilogue
Genshiken Nidaime OVA
Houkago Initiation
Zetsubou Funsai Shoujo ∞ Amida
Amada Anime Series: Super Mario Brothers
```

Discussion

Effectiveness:

The model is effective in every stage of user interaction. Whether it is a new user, a user who showed explicit interest after answering a prompt question, or even a user with thousands of ratings, including whether that user is optimistic or pessimistic. All these user types are accounted for, the cold start problem can be solved with non-personalized and content-based recommendations. Furthermore, if the model is presented to the user with a rich rating depth, CF will be used, and behavioural biases will be taken care of.

The Fundamental Principles of Recommendation Systems:

As the model was being built, we had the fundamental principles of recommendation systems almost split-screened, as if they were actively guiding our creation process.

1) Personalization

This is achieved through user-based collaborative filtering and hybrid models that tailor recommendation lists to each individual user's past behaviors and expressed interests.

2) Exploitation vs Exploration

We start with a popularity-style list (exploration) for brand-new users, then gradually lean on content-based and collaborative filtering (exploitation) once we learn what they enjoy. This allows us to ensure users are recommended according to their tastes and interests, as well as being exposed to new anime/genres, diversifying their engagement.

3) Relevance

By prioritizing explicit feedback in our CF implementation, we are creating a pipeline that allows relevant recommendations to our users. Additionally, by basing scores on explicit 1-to-10 ratings (rather than implicit feedback), our CF predictors aim for items that the target user is statistically likely to rate ≥ 8. In other words, fewer "Why on earth did it show me this?" moments.

4) Fairness and Bias Mitigation

We store only numeric IDs for both users and anime. No gender, nationality, language, or other sensitive fields enter the model, so it cannot learn to prefer one demographic over another. Popularity metrics are normalised, and techniques like Bayesian help smoothing stop cult classics from unfairly outranking quality anime.

5) Cold start problem

This is done by integrating a non-personalized hybrid model of bayesian and weighted scoring, in addition to a hybrid content model, ensuring that new users and newly added items still receive meaningful recommendations even without prior interaction data.

Challenges:

Due to constraints in processing power and memory, we had to carefully manage how we built and stored recommendation models. All methods were computationally expensive, forcing us to optimize by limiting the dataset size or precomputing similarities where possible.

Evaluating the performance of content-based filtering was challenging because traditional metrics like RMSE and precision-recall are easier to apply to user-based collaborative systems with explicit ratings. For content-based methods, we attempted to design evaluation strategies based on indirect measures but ultimately settled on manual subjective evaluation.

Since we adopted a hybrid approach, another challenge was deciding how to balance and combine the outputs of text-based, genre-based, user-based, and item-based models. Tuning the contribution of each method required experimentation and careful observation. We mitigated this by separating the two approaches and applying them on a need-now basis. Furthermore, this was done by designing the model to use a hybrid content based approach of text based filtering and genre based filtering if the user has zero animes rated but explicitly expressed his interest in specific animes or genres.

Limitations and Future Work

The main limitation of our implementations is the scalability, as the dataset grows, similarity computations become more resource-intensive and will heavily negatively affect system performance. Furthermore, precomputing and caching similarities or moving towards matrix factorization methods like ALS could reduce the need for full pairwise similarity calculations, making the system more efficient for large-scale deployment. In addition, the system currently relies on explicit user ratings for collaborative filtering, overlooking valuable implicit signals like click-through rates, watch duration, or browsing history. Integrating these behavioral cues could enhance personalization, particularly for passive users who rarely rate items. Additionally, our static hybrid weighting fails to adapt to individual engagement patterns. Moreover, a dynamic strategy where the model increases collaborative filtering weight, after detecting binge-watching behavior, for example, could improve responsiveness. The implementation also lacks real-time feedback mechanisms; implementing "thumbs down" buttons or skipped-item tracking would create a closed-loop system to continually recalibrate recommendations. Beyond these enhancements, we envision adopting session-aware algorithms to capture temporal preferences (e.g., holiday movie trends) and exploring knowledge graphs to connect items through multidimensional relationships (directors, themes, cultural relevance).

Conclusion

In conclusion, this project successfully demonstrated the design and implementation of a hybrid recommendation system that combines content-based filtering, collaborative filtering, and non-personalized methods. By leveraging both explicit user feedback and item metadata, we were able to personalize recommendations, handle cold start scenarios, and provide relevant suggestions even for new users. Our experiments showed that item-based collaborative filtering yielded the best predictive accuracy (RMSE: 1.2099), while the hybrid scoring model effectively surfaced top-rated anime by balancing rating quality with popularity. Though scalability remains a key limitation, the system provides a solid foundation for further optimization through techniques like matrix factorization or approximate nearest neighbor search. Overall, the project highlights the importance of the fundamental principles in building an effective recommendation system.

References

[1]

Ashcraft, B. (2022, May 27). What “Anime” means. Kotaku. https://kotaku.com/what-anime-means-1689582070

[2]

Top anime - MyAnimeList.net. (2020). MyAnimeList.net. https://web.archive.org/web/20200603002033/https://myanimelist.net/topanime.php

[3]

Hernan. (2020). *GitHub - Hernan4444/MyAnimeList-Database*. GitHub. https://github.com/Hernan4444/MyAnimeList-Database

[4]

Anime-Planet Recommendation Database 2020. (2021, June 28). Kaggle. https://www.kaggle.com/datasets/hernan4444/animeplanet-recommendation-database-2020

[5]

Aklson, A. A. (2025). Non-personalized recommendations. CMP 49412 – Intelligent Recommendation Systems, American University of Sharjah.