Team - 192: Jenny Goldsher, Kelvin Ervais, Vishal Vala

FlickFlow: A People-driven Movie Recommendation Engine

Introduction

In today's media climate, real-world applications of recommendation systems are limited to organizations with personal or monetary incentives. Companies implement recommendation models heavily biased towards pushing their original work exclusive to their streaming services, movies with high contract costs, or films with the potential to sell specific merchandise. Our intended users will be those looking for a movie recommendation based on their personal preferences, not what streaming companies identify to be the most profitable. While previous research has investigated static sentiment-based movie preferences, FlickFlow is an implementation that dynamically generates movie recommendations based on user-inputs and a deliberate model implementation.

Problem Definition

Dynamically generate movie recommendations that users will enjoy watching based on their personal film preferences. The method must respond to a user's evolving preference over time as well as deploy in a way that users will readily adopt due to familiarity and ease of use (dating app swiping style).

Literature Survey

Numerous studies have examined the advantages and practical uses of recommendation systems, as evidenced by the works of Osman & Azman (2018), Kesharwani & Bharti (2017), Kumar & Roy (2020), Chauhan et al. (2021), Marappan & Bhaskaran (2022), and Baid et al. (2017). Early in our investigation, we identified the necessity for adequate data quantity as a key factor. This was highlighted by an investigation of sentiment-based recommendation systems across multiple technology platforms conducted by Osman and Azman (2018). In their study, they noted that for a recommendation system to be valuable, large amounts of user ratings data were necessary in the modeling process.

Wakil et al. (2015) explored combining collaborative and content-based filtering and discussed integrating emotion into recommendation systems. This aligns with our approach, including the potential development of an affect classification system. Sappadla et al's (2017) 'zero state' and 'base state' concept based on global and user average similarity inspired us, while Nakhli et al's work guided our feature design, emphasizing the incorporation of revenue metrics for viewership analysis in our recommendation system.

As it specifically relates to movie recommendation systems, Kesharwani & Bharti (2017) implemented a recommendation system based on Twitter sentiment analysis. They collected hundreds of sentiment-related statements per movie from Twitter, intentionally choosing Twitter due to its large number of users to ensure extensive data collection. Using their algorithm, they converted the raw text data into movie ratings on a likert scale of 1-5, creating a dataset for predictive analysis. We used a similar methodology to quantify the similarity between user preferences.

In terms of model applications to sentiment analysis, Kumar and Roy (2020) provided an example of movie-sentiment related analysis using various modeling techniques. Their experiment suggested that Linear SVM demonstrated a slightly higher predictive accuracy compared to other predictive models. Kumar and Roy (2020) emphasized the importance of pre-processing and feature engineering stages in achieving such accuracy, indicating their relevance for similar studies. In our project, we have deliberately implemented data processing and carefully reviewed relationships between variables to enable effective modeling.

Previous research has shown many relevant examples of collaborative filtering systems. Among these projects, consistent themes have been data processing, intentional model selection, and extensive data acquisition. Our methodology and implementation reflect our gained knowledge from reviewing these projects.

Methodology

FlickFlow innovates the movie recommendation experience through a breadth of data, dynamic UI and a unique form of collaborative filtering (Counts and Means Based CF - *CMCF*) without any profit motivated bias. No other movie recommendation platform today analyzes user's preferences and recommends what they might like based solely on improving the user's experience instead of improving an organization's profit margin.

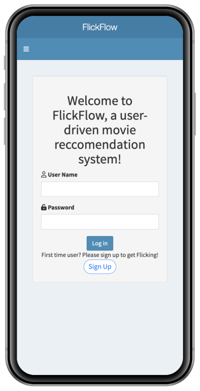
Data Creation and EDA

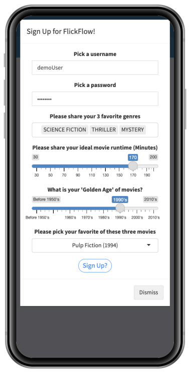
For FlickFlow, our main movie data comes from TMDB, specifically a Kaggle dataset that is updated daily. Due to computational restraints, we have limited our movie dataset to just under 100,000 movies, mainly based on a combination of popularity, number of TMDB votes, and overall missing data. However, this allowed us to one-hot encode an extra 100+ columns based on genre, production company, and decade. Additionally, for the purposes of being able to use them in our recommendation system, we converted the TMDB vote average, TMDB popularity score, revenue, budget, and runtime columns to percentile values.

On the user side, our data was stored in a MongoDB with a similar structure to our one-hot-encoded movie data. Additional columns were added to this database to track user interactions, including number of swipes, directions, and which movies were swiped in which direction. This user data is used to craft our similarity matrix and power our based model, which we will discuss more below. Finally, for data that is pulled on demand (Movie posters and cast/director info), we are using the TMDB API.

Computation- Zero State User Model (Content-Based Filtering)

For new users, they will be prompted with a three question preference quiz to narrow down genre, ideal runtime, and movie generation. From this, we take an algorithmic approach (content-based filtering) to present the user with three options. Once the user selects one of the three movies, we run the algorithm again to get the initial fifty recommendations that will be presented to the user. Every time a user swipes, they are presented with a random choice from these initial fifty recommendations. Once a user has run through twenty five of these fifty recommendations, we will then switch to our Base State model.


FlickFlow
Welcome to FlickFlow, a user-driven movie reccomendation system!
User Name
Password
Log in
First time user? Please sign up to get Flicking!
Sign Up



Pick a username: demoUser

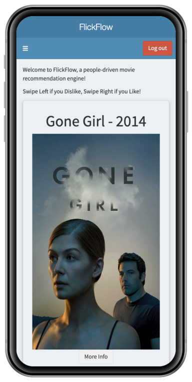
Pick a password: ••••••••

Please share your 3 favorite genres: SCIENCE FICTION, THRILLER, MYSTERY

Please share your ideal movie runtime (Minutes): 170 (30–200)

What is your 'Golden Age' of movies?: 1990's (Before 1950's – 2010's)

Please pick your favorite of these three movies: Pulp Fiction (1994)


FlickFlow
Log out
Welcome to FlickFlow, a people-driven movie recommendation engine!
Swipe Left if you Dislike, Swipe Right if you Like!
Gone Girl - 2014
GONE GIRL
More Info

Computation- Base State User Model (CMCF)

Once a user has reached twenty five interactions, rather than receiving recommendations from our content-based filtering model, the outputs will be made by our custom-made collaborative filtering system. While many approaches to collaborative filtering exist, no libraries exist for our needs. Specifically, we needed a similarity matrix based on counts and means rather than binary or ratings. We also needed a hybrid format that incorporated both user data and the movie data. In order to accomplish this, the first step we took was creating a similarity function using cosine similarity.

$$\boxed{\text{similarity}(\text{user}_x, \text{user}_y) = \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \times \sqrt{\sum_{i=1}^{n} y_i^2}}}$$

From here, we then make our similarity matrix of users. Given the nature of our project and the need for user data, for our MVP we have used simulated user data to build the similarity matrix. Considerations were taken to accurately simulate user data, including distributing the preferences based on our actual movie data. For our final iteration, we have simulated 500 users. Using this similarity matrix, we then find *k* users with the highest similarity ranking. From these users, we aggregate their preferences together, and use these aggregated preferences to find suitable movies for recommendation.

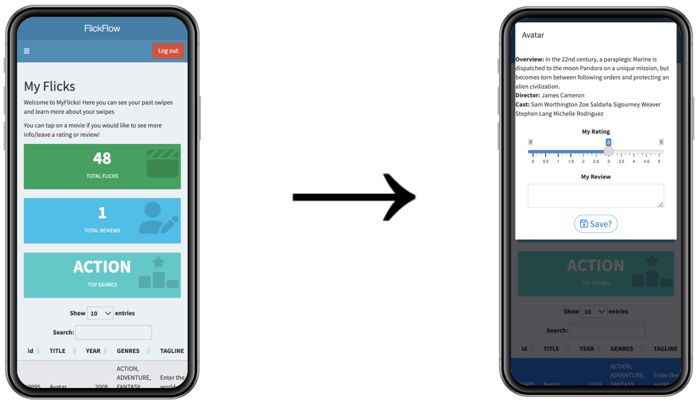
$$\boxed{\text{Aggregated Preferences}_x = \frac{1}{k} \sum_{i=1}^{k} \text{user}_{s_i}}$$

After this, we compute a similarity score for the aggregated preferences and the suitable movies and the output is the highest similarity score. Using these scores, we display the movie to the user.

$$\boxed{\text{score}_i = \text{similarity}(\text{Aggregated Preferences}_x, \text{movie}_i)}$$

User Interface

Once a user has signed up, they are met with a movie card with the title, poster, and a "More Info" button. Similar to dating apps, a user swipes right for a movie they are interested in seeing/have seen and liked; and swipe left for movies they have no interest in/did not enjoy. These movies are stored in our MongoDB and can be viewed by the user in the "My Movies" tab. Here, a user can access what movies they have swiped on, get more information on the movies, and leave a review and rating.



Welcome to MyFlicks! Here you can see your past swipes and learn more about your swipes

You can tap on a movie if you would like to see more info/leave a rating or review!

id	TITLE	YEAR	GENRES	TAGLINE
19995	Avatar	2009	ACTION, ADVENTURE, FANTASY	Enter the world

Avatar

Overview: In the 22nd century, a paraplegic Marine is dispatched to the moon Pandora on a unique mission, but becomes torn between following orders and protecting an alien civilization.
Director: James Cameron
Cast: Sam Worthington Zoe Saldaña Sigourney Weaver Stephen Lang Michelle Rodriguez

Experiments/Evaluation

When designing our user backend, we kept in mind a number of metrics that we wanted to track. In regard to the overall performance of our approach (combining our zero and base state models), we felt it important to set some goals for the number of users and associated metrics. The specific goals can be seen in the table below. Our actual metrics are displayed as of 4/20/24:

Metric	**Number of Users**	**Total Flicks**	**Flicks Right**	**Flicks Right % of Total Flicks**
Goal	**30**	**300**	**200**	**66**
Actual	**39**	**579**	**362**	**63**

While we did not quite reach our goal of ‘yes’ swipes (flicks right), we are quite happy with these results. Additionally, when diving into the distribution of zero state and base state users, we were very pleased to see that our base model outperformed the basic content filtering of the zero state model.

Model	**Number of Users**	**Total Flicks**	**Flicks Right**	**Flicks Right % of Total Flicks**
Zero	**30**	**234**	**128**	**55**
Base	**9**	**345**	**234**	**68**

From this, we see that the near quarter of our users who reached the base model threshold did see an increased number of flicks right during their experience.

Outside of metrics tracked, another evaluation we wanted on our model was any potential biases in the recommendations, specifically amongst genre, production company, or decade. The table below shows the top five most swiped right for each of these:

Genre	**Company**	**Decade**
Adventure (395 Flicks Right)	Marvel (152 Flicks Right)	2010’s (346 Flicks Right)
Action (365 Flicks Right)	Warner Bros (116 Flicks Right)	2000’s (162 Flicks Right)
Sci-Fi (255 Flicks Right)	Paramount (98 Flicks Right)	1990’s (69 Flicks Right)
Drama (175 Flicks Right)	20th Cent. Fox (44 Flicks Right)	1980’s (31 Flicks Right)
Fantasy (148 Flicks Right)	New Line (32 Flicks Right)	1970’s (14 Flicks Right)

From this, the biggest flag comes from company and decades, specifically a bias towards Marvel and a pretty significant degree of recency bias. While we did not ask for user age, one explanation could be that our users geared Gen-Z and Millennial, accounting for the recency bias. In regards to Marvel, this could also be explained by a potential bias in the user base, as it is not representative of the nearly 100,000 movies available in our dataset.

Conclusion/Discussion

Based on the results of our implementation and previous research, we gained multiple insights on content recommendation systems. Specifically, as highlighted both by the present project and Osman and Azman (2018), large quantities of user data play a key role in model accuracy and feature engineering. While we were able to exceed our number of target users, a continued investigation with a more substantial quantity of user data may have yielded more effective similarity scores. Further, because of the large number of Flicks involved with the dataset relative to number of users, we may have needed more users to obtain a more exhaustive set of data to construct our similarity scores from. Lastly, we constrained the movie data to only English speaking films. Expanding the field of movies across all languages is a potential future extension of FlickFlow. This future opportunity could improve both the accuracy of the hybrid collaborative filtering algorithm as well as the user experience by exposing them to movies they would otherwise never have come across.

We believe our UI implementation was effective in engaging users and providing an easy-to-use framework to elicit user preferences.The simplicity of the UI allowed us to quickly collect data and invest additional time in developing our model implementation and data-backend. This streamlined approach significantly improved our development efficiency. With additional time and resources, we would have employed more tactful user-recruiting techniques. While our UI implementation and data collection methods were effective within our constraints, further scaling with increased user data would enable the creation of an improved algorithm with greater predictive accuracy.

Overall we consider FlickFlow a success. It is the first movie recommendation platform completely decoupled from company profits and focused solely on the user's benefit. It successfully deployed a custom designed hybrid collaborative filtering algorithm to a user base seamlessly via R Shiny and MongoDB. FlickFlow dynamically generated movie recommendations via the similarity matrix of users and similarity score of aggregated move preferences. We successfully reached our goal for the number of users testing FlickFlow in our experimental trials. Of these test users, we surpassed our goal of total flicks and positive user response. Although we didn't reach our goal of positive response (flicks right) of total participation, we nearly achieved it within a margin of 3%. Across all experimental trials, the user feedback on the UI and mobile friendly "dating app like swiping" was universally positive. All team members contributed an equal amount of effort to the success of the project.

Sources

Baid, P., Gupta, A., & Chaplot, N. (2017) Sentiment Analysis of Movie Reviews using Machine Learning Techniques . *International Journal of Computer Applications*, *179*(7), 0975 – 8887.

Chauhan, A., Nagar, D., & P. Chaudhary (2021) Movie recommender system using sentiment analysis. *International Conference on Innovative Practices in Technology and Management (ICIPTM)*, Noida, India, 190-193, doi: 10.1109/ICIPTM52218.2021.9388340.

.Kesharwani, A. & Bharti, R. (2017). Movie rating prediction based on twitter sentiment analysis. *Journal of Advanced Computing and Communication Technologies, 5*(1), 1-5.

Kumar, S. & Roy, P.P. (2020). Movie recommendation system using sentiment analysis from microblogging data. *IEEE Transactions on Computational Social Systems*, *7*(4), 915-923, doi: 10.1109/TCSS.2020.2993585.

Marappan, R., & Bhaskaran, S. (2022). Movie Recommendation System Modeling Using Machine Learning. *International Journal of Mathematical, Engineering, Biological and Applied Computing*, *1*(1), 12–16.

Osman, N. A. & Azmam N. (2018). Sentiment-Based Model for Recommender Systems. *Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)*, Kota Kinabalu, Malaysia, 1-6, doi: 10.1109/INFRKM.2018.8464694.

Wakil, K., Bakhtyar, R., Ali, K., & Alaadin, K. (2015). Improving web movie recommender system based on emotions. *International Journal of Advanced Computer Science and Applications*, *6*(2), 218-226.

Sappadla, P., Sadhwani, Y., & Arora, P. (2017). *Movie Recommender System Project Report*. NYU. https://www.codeheroku.com/static/workshop/hw/movie_recommendation/MovieRecommenderSystem.pdf

Nakhli, R. E., Moradi, H., & Sadeghi, M. A. (2019, February). Movie recommender system based on percentage of view. In *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)* (pp. 656-660). IEEE.