Movie Recommendation System

Eric Zhang[1], Christine Chen[2], Eric Li[3], Dhruv Chavan[4]

Computer Science Department
Luddy School of Informatics, Computing, and Engineering
Indiana University Bloomington

[1]ez2@iu.edu
[2]cch8@iu.edu
[3]ermili@iu.edu
[4]dvchavan@iu.edu

***Abstract*—** Recommendation systems are widely used to personalize and enhance user experiences, and they are especially useful on media review platforms like Letterboxd and IMDb. This project mimics these platforms and develops a recommendation system that suggests trending movies that align with a user's previously watched movies and preferences. It utilizes a Letterboxd movies dataset from Kaggle, author Eric Li's Letterboxd reviews and ratings, and a trending movies dataset from Kaggle. This data is used alongside data mining techniques including cosine similarity and k-nearest neighbors for the two components of the recommendation system. First, the recommendation system obtains a movie most representative of a user's history and taste based on the movies they have watched and reviewed. Second, the system uses this representative movie to suggest similar movies that are trendings. The system aims to produce accurate movie recommendations for movie watchers of various experiences.

***Keywords*—** Cosine Similarity, Recommendation System, Letterboxd, K-Nearest Neighbor, Sentiment Analysis

I. Introduction

Recommendation systems are essential to digital platforms, enabling personalized content and improving user experience. Movie recommendations help users discover new films that align with their preferences by analyzing past behaviors and movie characteristics. This project aims to design a system using data mining techniques that generates suggestions similar to a given set of movies the user has previously watched.

A. Related Works

Former research includes findings using IMDb attributes [1], while this paper uses a dataset from Letterboxd, another movie rating platform. This paper utilizes Letterboxd due to its recommendation style from the social-media-like interface that utilizes the watch history from not only genre-specific communities, but also a user's following and followers. This paper aims to provide a recommendation system that outputs a list of movies where past research tries to predict box-office outcomes [1,2]. We gain insight from neural networks and machine learning implementation through various works [1,2].

II. Methods

A. Data Gathering

For our purposes, we required two datasets: one dataset with instances detailing the movies watched by an individual, along with their personal ratings and reviews, and another dataset with instances of trending movies not watched by the user. A Letterboxd movies dataset from Kaggle was chosen as the source for all movie information. Each instance contained the following attributes: movie name, date released, movie tagline, movie description, movie duration, and average rating. The dataset considered 950,000 films spanning from 1874-2025. We had author Eric Li compile his Letterboxd movie reviews and ratings, and

add them as features to a subset of aforementioned dataset, solely consisting of the movies he reviewed and rated.

As for the second dataset, movie names were extracted from a trending movies dataset from Kaggle, and the names were used to retrieve instances pertaining to those movies from the same Letterboxd movies dataset from Kaggle used to create the first dataset. Moreover, the movies in the second dataset were kept only if they were not in the first dataset, meaning author Eric Li had not reviewed them on Letterboxd.

B. Data Preprocessing

For both datasets, all non-numerical features such as movie tagline, and movie description, were converted to numerical features or numerical, vector-valued features. Additionally, we handled plot summaries using Word-Embedding to match similar words and text with each other.

C. Representative Selection

Sentiment analysis was conducted on a user's reviews, and the results were used to assign a weight to each movie watched by the user. This analysis utilized natural language processing to discern a review's meaning, and it categorized them as one of 'bad', 'neutral', or 'good'. Next, each review's categorization was mapped to a numerical weight. This way, movie reviews could be utilized meaningfully, and serve as another way of conveying a user's preferences separate from personal ratings.

The set of n movies watched by the user was iterated through n^2 times. In this process, for each of the n movies, a weighted sum of the cosine similarity between the movie and every other movie in the set was computed. In each computation of cosine similarity between movie instances, the weight assigned to the 'other' movie under consideration was used. As a side note, cosine similarity was also computed between vector-valued features to subsequently be used to compute cosine similarity between instances.

The representative movie was selected as the movie with the smallest weighted sum of cosine similarities. This approach aimed to select a movie that aligned with a user's preferences, and if possible, was similar to other movies watched by the user.

D. Trending Movie Suggestion

A k-NN algorithm using Euclidean distance was employed on the trending movies dataset to find the k most similar movies to the representative movie. All k movies served as the trending movie recommendations for the user the representative movie was obtained for.

E. Constraints

On Letterboxd, shows are considered in the rankings and can serve significance to one's preferences as to what they desire to watch. The dataset that was selected for use only considers movies, thus eliminating some meaningful data that could be used to train our model and constricting which instances to consider from one's watch history.

Due to limitations with time, scraping user data from Letterboxd was unfeasible, so the recommendation system could only be tested with a single Letterboxd user's data: author Eric Li.

The recommendation system falls somewhat short in the case that a user has extremely random taste, meaning all the movies they have watched and reviewed are equally suitable candidates to be the representative of the user's history and taste. Furthermore, it is likely that the trending movies most similar to the representative movie would not be suitable recommendations for the user. However, the user's random taste may be suggestive of that fact that they would enjoy any movie recommendation.

III. Author Contributions

A. Eric will provide training data from personal Letterboxd reviews/ratings for Eric

B. Eric will develop a sentiment analysis of the reviews provided by Eric

Figure 1: Eric vs Eric, which is which.

Figure 2: Reaction to movie.

TBD STARTING FROM HERE

IV. Results

TABLE I
DUMMY DATA

Font Size	Appearance (in Time New Roman or Times)		
	Regular	**Bold**	**Italic**
8	table caption (in Small Caps), figure caption, reference item		reference item (partial)
9	author email address (in Courier), cell in a table	abstract body	abstract heading (also in Bold)
10	level-1 heading (in Small Caps), paragraph		level-2 heading, level-3 heading, author affiliation
11	author name		
24	title		

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

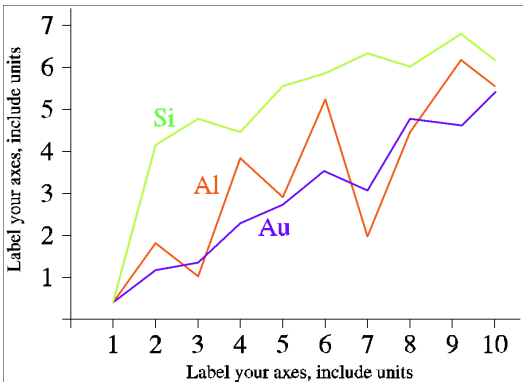

Fig. 1 Lorem ipsum

V. DISCUSSION

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

VI. CONCLUSIONS

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

REFERENCES

[1] Hsu, PY., Shen, YH., Xie, XA. (2014). Predicting Movies User Ratings with Imdb Attributes. In: Miao, D., Pedrycz, W., Ślęzak, D., Peters, G., Hu, Q., Wang, R. (eds) Rough Sets and Knowledge Technology. RSKT 2014. Lecture Notes in Computer Science(), vol 8818. Springer, Cham. https://doi.org/10.1007/978-3-319-11740-9_41

[2] Lee, K., Park, J., Kim, I. *et al.* Predicting movie success with machine learning techniques: ways to improve accuracy. *Inf Syst Front* **20**, 577–588 (2018). https://doi.org/10.1007/s10796-016-9689-z

[3] R. H. Singh, S. Maurya, T. Tripathi, T. Narula, and G. Srivastav, "Movie recommendation system using cosine similarity and KNN," *International Journal of Engineering and Advanced Technology*, vol. 9, no. 5, pp. 556–559, Jun. 2020. doi:10.35940/ijeat.e9666.069520

dataset: https://www.kaggle.com/datasets/gsimonx37/letterboxd?select=movies.csv

testing: https://letterboxd.com/ericmli/films/

if the user has random taste: so you're up for everything huh?

Trending Movies Dataset: 1990 to 2025