



PRML Project

Movie recommendation system

Authors: Vatsal Dadhaniya, Prakash Nandaniya, Vekariya Sagar, Gadhiya Ronak,
Pareen Shah
Guided By: Anand Mishra

April 20, 2024

Abstract

Our aim is to develop a Movie recommendation system based on ML model to recommend relevant movies to user. In this Project we have created 4 models Based on Collaborative filtering 3 of which uses KNN, PCA, SVD and one Hybrid Model uses K means clustering and Random forest.

Keywords: Collaborative Filtering, KNN, PCA, SVD, Hybrid Model

Contents

1	Introduction	3
1.1	Background	3
1.2	Major Findings and Overview of our work	3
2	Approaches Tried	3
2.1	Collaborative filtering using KNN	3
2.2	Collaborative filtering using PCA	4
2.3	Collaborative filtering using SVD	4
2.4	Hybrid Model	4
3	Experiments and Results	5
3.1	About Dataset	5
3.2	Experimental setting and Comparing Results	5
4	Summary	7
A	Contribution of each member	7

1 Introduction

1.1 Background

Our project centers on exploring traditional machine learning (ML) models commonly used in movie recommendation systems. These models include collaborative filtering techniques such as user-based and item-based filtering, which leverage similarities between users or items to generate recommendations. User-based filtering identifies users with similar preferences and recommends movies based on their collective behavior, on the other hand item-based filtering suggests similar movies given one movie as an input.

By focusing on these established ML models, our objective is to develop a reliable movie recommendation system that provides personalized suggestions based on user behavior and preferences, without relying on deep learning architectures.

1.2 Major Findings and Overview of our work

We explored many different techniques to be used in the movie recommendation system. Majorly two types of methods are used, user-based collaborative filtering and item based collaborative filtering. User based collaborative filtering is implemented using 3 different models which uses KNN, PCA and SVD [1], similarly item based collaborative filtering is also implemented using the same models. After analysing the results of this model we were able to conclude it practically that PCA and SVD are similar techniques and hence they also gives similar results.

We have also created a hybrid model which uses K-means clustering and Random forest to generate recommendations based on user's activity as well as analysing various features of movies. By creating this model we were able to link different ML techniques in one model. This Model is discussed in details in following parts of this report.

These techniques play pivotal roles in enhancing user engagement, satisfaction, and retention on movie streaming platforms, enriching the user experience through personalized and contextually relevant movie recommendations.

2 Approaches Tried

‘ Following are the approaches used:

1. Collaborative filtering using KNN
2. Collaborative filtering using PCA
3. Collaborative filtering using SVD
4. Hybrid Model

2.1 Collaborative filtering using KNN

Using KNN we have implemented both user and item based collaborative filtering. Firstly, movie-user matrix is created to represent the ratings given by each user for a particular movie. By identifying the k nearest neighbors to a user's vector, we personalize movie recommendations to align with similar users' tastes. This aspect is particularly beneficial for streaming platforms, where tailored content suggestions on a user's homepage to recommend the movies on a general basis without any genre bias.

Now, this is further extended to give genre wise recommendations, for this we added only movies of particular genre in movie-user matrix and implemented KNN similarly.

On the other hand, our model gives item-based recommendations by identifying similar movies using KNN, instead of focusing solely on user preferences. This functionality

proves invaluable when users search for specific movies or upon completing a movie, where recommending related content enhances the overall user experience. Leveraging scalable algorithms like KNN allows for efficient handling of large datasets, contributing to enhanced recommendation accuracy and user retention in dynamic streaming environments.

2.2 Collaborative filtering using PCA

Similarly using this model also we have implemented three types, general and genre wise in user based filtering and item based filtering. To implement item based filtering, similar movie-user matrix is created. Next, we compute the covariance matrix from the normalized movie matrix to determine eigenvalues and eigenvectors. These eigenvalues and eigenvectors are pivotal in calculating cosine similarity scores. By leveraging cosine similarity, we identify the top "n" movies closely related to a given movie, thus providing tailored recommendations based on item similarities.

To implement user based filtering, for all the movies that were rated by user, item based filtering was applied and from each recommendations two were selected to give final recommendation. It was ensured that the movies rated by users were selected in descending order of rating to ensure that the given recommendations corresponds to some of the top rated movies by user. For genre specific recommendations movies were selected from the recommendations only if they belongs to that particular genre.

2.3 Collaborative filtering using SVD

In collaborative filtering, Singular Value Decomposition (SVD) breaks down the user-item matrix into smaller, more manageable parts. This process involves splitting the matrix into three simpler matrices: one representing the relationship between users and latent factors, another representing the importance of each latent factor, and the third representing the relationship between items and latent factors. By simplifying the matrix, SVD helps in making predictions about user preferences for items they haven't interacted with yet, enabling recommendation generation based on these predictions. However, despite its popularity, SVD-based collaborative filtering encounters challenges with large datasets and sparse matrices, leading to the development of alternative techniques like stochastic gradient descent and Alternating Least Squares (ALS).

2.4 Hybrid Model

In this model, we initially employed the K-means clustering algorithm to group users based on their given ratings, this ensures that user with similar test are grouped in same clusters. For each cluster, we then focused on movies watched by at least one user within that cluster. Subsequently, we utilized the Random Forest algorithm to recommend movies to individual users that they have not yet watched. This two-step approach ensures that recommendations are personalized within user clusters while also leveraging the predictive power of Random Forest to enhance recommendation accuracy. Such a model is well-suited for streaming services, as it can efficiently manage large datasets and provide tailored recommendations on users' homepages, contributing to improved user satisfaction and engagement.

3 Experiments and Results

3.1 About Dataset

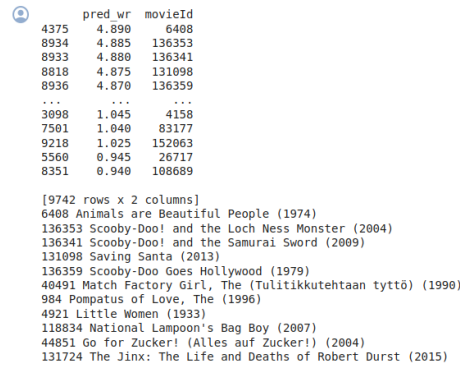
The dataset used in this project describes 5-star rating and free-text tagging activity from MovieLens, a movie recommendation service. It contains 100836 ratings and 3683 tag applications across 9742 movies. These data were created by 610 users between March 29, 1996 and September 24, 2018. This dataset was generated on September 26, 2018. Users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. Each user is represented by an id, and no other information is provided. It has 4 files: links.csv, movies.csv, ratings.csv and tags.csv. Moreover, we have created two extra files: imdb.csv and tmdb.csv by scraping data from respective sites.

The ratings dataset contains 100,836 rows and 4 columns, including unique identifiers for users ('userId') and movies ('movieId'), movie ratings ('rating' out of 5), and timestamps ('timestamp'). The tags dataset comprises 3,683 rows and 4 columns with user and movie identifiers, descriptive tags ('tag') for movies, and timestamps. The movies dataset includes 9,742 rows and 3 columns with movie identifiers, titles ('title'), and genres ('genres'). The TMDB dataset, also with 9,742 rows, features movie identifiers, user scores ('user_score' out of 100), and languages. The IMDB dataset, matching the TMDB dataset in size, contains movie identifiers, IMDB and TMDB identifiers ('imdbId', 'tmdbId'), IMDB ratings ('Ratings' out of 10), popularity metrics based on page visits ('Popularity'), counts of user and critic reviews ('User Reviews', 'Critic Reviews'), and a Metascore ('Metascore') representing a weighted average of critic ratings.

3.2 Experimental setting and Comparing Results

1. Collaborative filtering using KNN

- In this unified Model we can get recommendation based on user preference as well as based on movie
- We also have two types of user based recommendation: one is generalized recommendation while the other is genre based.
- The result for the user based recommendation for user270 of the MovieLens dataset 2a
- Result for item based recommendation part for the movie 'Iron man' are shown in 2b.



```

4375    4.890    6408
8934    4.885    136353
8933    4.880    136341
8818    4.875    131098
8936    4.870    136359
...      ...      ...
3090    1.045     4158
7501    1.040     83177
9218    1.025    152063
5560    0.945     26717
8351    0.940    108689

[9742 rows x 2 columns]
6408 Animals are Beautiful People (1974)
136353 Scooby-Doo! and the Loch Ness Monster (2004)
136341 Scooby-Doo! and the Samurai Sword (2009)
131098 Saving Santa (2013)
136359 Scooby-Doo Goes Hollywood (1979)
48491 Match Factory Girl, The (Tulitikkutehtaan tyttö) (1990)
984 Pompatus of Love, The (1996)
4921 Little Women (1933)
118834 National Lampoon's Bag Boy (2007)
44851 Go for Zucker! (Alles auf Zucker!) (2004)
131724 The Jinx: The Life and Deaths of Robert Durst (2015)
```

Figure 1: Recommendations by Hybrid Model

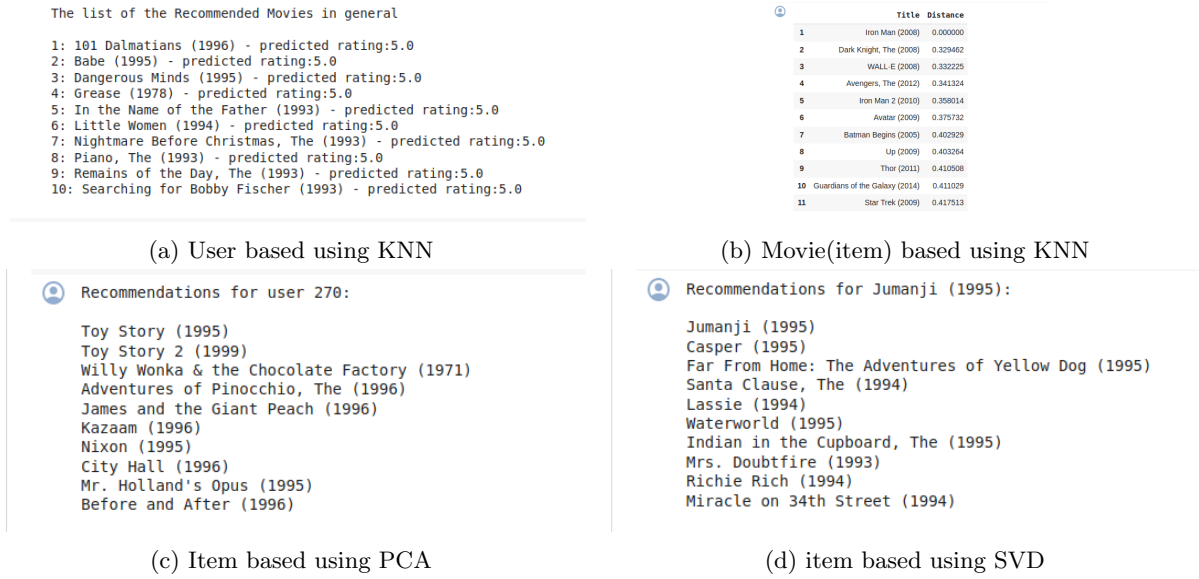


Figure 2: Recommendations



Figure 3: Recommendations

2. Collaborative filtering using PCA

- This Model also give recommendation based on both movies and user preference using PCA.
- Result of movie based recommendation for the movie 'Jumanji' using this model is shown in 2c
- Recommendations by this model for user 270 are shown in 3a

3. Collaborative filtering using SVD.

- Like previous Model this Model also give recommendation based on both movies using SVD.
- Result of movie based recommendation for the movie 'Jumanji' using this model is shown in 2d
- Recommendations by this model for user 270 are shown in 3b

4. Hybrid Model

- It is User based collaborative filtering model
- Recommendation by this model for user 270 are shown in 1

4 Summary

The PRML Project's draft document outlines a movie recommendation system based on traditional machine learning models, specifically focusing on collaborative filtering techniques like KNN, PCA, and SVD, as well as a hybrid model combining K-means clustering and Random Forest. The system aims to provide personalized movie suggestions by analyzing user behavior and preferences, leveraging a dataset from MovieLens containing ratings and tags across thousands of movies1.

References

- [1] Hervé Abdi. Singular value decomposition (svd) and generalized singular value decomposition (gsvd). URL https://www.cimat.mx/~alram/met_num/clases/Abdi-SVD2007-pretty.pdf.

A Contribution of each member

1. Ronak Gadhiya: Scrapped data from **IMDb** site for all movies in original dataset to get some other important features and created new dataset. Created Report of the project
2. Prakash Nandaniya: Scrapped data from the **TMDb** site for all movies in original dataset to get some other important features and created new dataset and created Report of the Project.
3. Pareen Shah: Created Project Website and presentation for the project.
4. Sagar Vekariya: Wrote a code base for the all Models.
5. vatsal Dadhaniya: Wrote a Code base for the all models.