

Flixalit: Movie Recommendation System based

¹Sagar Patel, ²Madhav Ajwalia, ³Aneri Adani

¹Assistant Professor, ²Assistant Professor, ³B. Tech Student

^{1,2,3,4,5}Smt. Kundaben Dinsha Patel Department of Information Technology,

^{1,2,3,4,5}Chandubhai S. Patel Institute of Technology, Charotar University of Science and Technology, Changa-388421, Gujarat, India.

¹sagarpatel.it@charusat.ac.in,

²madhavajwalia.it@charusat.ac.in, ³17it001@charusat.edu.in

Abstract--This project mainly focuses on recommending movies based on users preference using Machine Learning. It has always been a difficult task for what to watch next for it analyzes the contents (storyline, genre, cast, director etc.) of the movie to find out other movies which have similar content. Then it ranks similar movies according to their similarity scores and recommends the most relevant movies to the user. Researchers have always been highly interested in this particular research area. It takes dataset of movie-lens and TMDB5000 dataset to recommend movies based on users preference. In this project, recommendation is done using Pearson's correlation prediction and cosine Similarity a optimum approach to recommend movies. The implementation of model is done using Flask framework for content based filetring Dataset of movies is taken from <https://www.kaggle.com/tmdb/tmdb-moviemetadata>

Keywords—Flask, cosine similarity, Pearson's correlation

Introduction

According to popular estimates, the amount of movies online have increased therefore to find a movie matching your interest for a system using existing technologies are very hard, For, this reason, the users want a system that can suggest the movie to them and best technology about these is recommendation system.

Content Based Recommendation System is used is used This is often one of the straightforward approach of suggesting items or substance to the client. This sort of suggestion frameworks, takes in a movie that a client as of now likes as input. At that point it analyzes the substance (storyline, sort, cast, direction etc.) of the movie picture to discover out other movie which have comparative features using cosine similarity.

With content Based filtering is not enough because of cold start problem Hence collaborative filtering is used to recommend movies. By introducing Pearson's correlation coefficient we standardize the rating. The idea of collaborative filtering is finding users with same interests . for instance if two users rate same or almost same items then they tend to have similar taste for movie genres. Such group of user are so-called "neighborhood". A user gets recommendations for those items that user hasn't rated before but was positively rated by users in his/her neighborhood. These is implemented by using Pearson's correlation coefficient. Collaborative Filtering gives used far better results than content based systems as it takes user rating in consideration. Hence this will help us in recommending movies based on.

Implementation

The entire project is prepared on Microsoft Azure notebooks ,cloud service. The Anaconda(open-source distribution of the Python) used for building UI for Building Flask framework.

1. This project is based on recommending movies based on cosine similarity and Pearson's correlation .
2. The content based filtering the movie lens data set is processed based on features to recommend movies.
3. The dataset for this project called tmdb.csv for GUI in Flask for recommendation is available on www.kaggle.com/tmdb/tmdb/tmdb-moviemetadata The project has data from 1995 . Cleaning data for Its features

Are ratings, tagline ,cast, average votes ,genre, director.

4. Using Cosine similarity we find similarity between movie user enters to movie from data set using Cosine similarity .The cosine similarity calculates the distance in terms of $\text{Cos}(\theta)$.

hence making recommendation comparatively easier.

5. Implemented cosine similarity and built model using libraries pandas, NumPy, from sklearn used text count Vectorizer for building similarity matrix based on features using sklearn cosine similarity.

$$\text{Cosine Similarity : } \text{Sim}(u_i, u_k) = \frac{r_i \cdot r_k}{|r_i||r_k|} = \frac{\sum_{j=1}^m r_{ij}r_{kj}}{\sqrt{\sum_{j=1}^m r_{ij}^2 \sum_{j=1}^m r_{kj}^2}}$$

6. The data flow of project is as below.

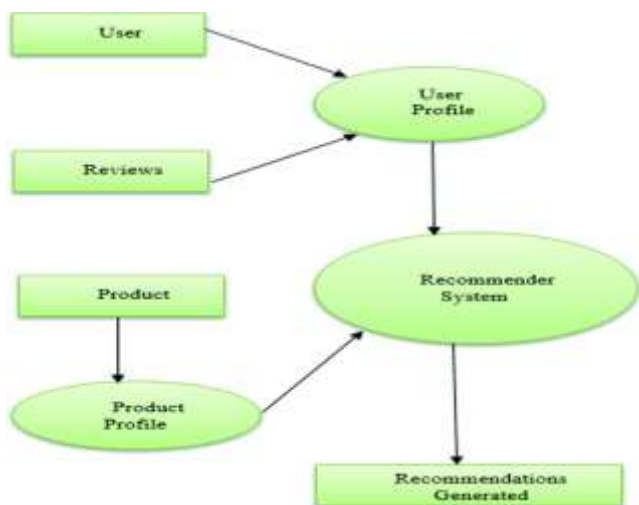


Fig1:flow of recommend system

7. The Flask framework is used to build GUI for model generating recommendation when movie is entered .The functionality for autocomplete the movie title and then top 9 movies are recommended to user, as below

8. The drawback of content based recommendation is that it may suggest nothing unexpected or surprising. Thus it recommends based on interests of user and but lacks the capability to expand users view. Hence, content based filtering is combined with other techniques in hybrid filtering.

9. The implementation of collaborative filtering is used for better recommendation. In, my project is used movie-lens dataset to study results of collaborative filtering.

10. In, the collaborative filtering the user need to rate the movie on basis of which of which (s)he wants recommendation. The ratings and movies data files are

merged along with standardizing user rating using on median value. The correlation matrix between users is generated using Pearson's correlation equation.

$$\text{Pearson Correlation : } \text{Sim}(u_i, u_k) = \frac{\sum_j (r_{ij} - r_i)(r_{kj} - r_k)}{\sqrt{\sum_j (r_{ij} - r_i)^2 \sum_j (r_{kj} - r_k)^2}}$$

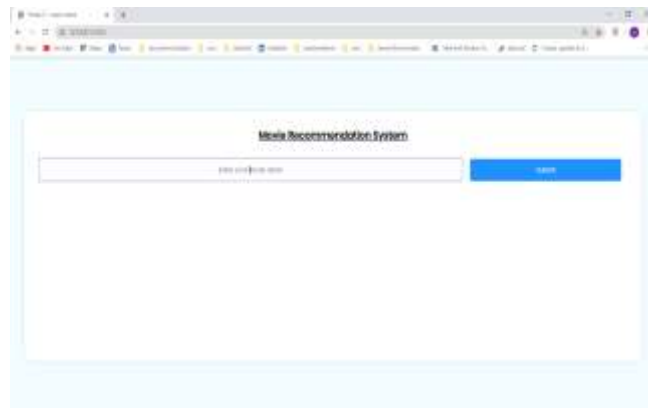


Fig2 Recommendation system module



Fig3 Suggestion based on entering movie title

	Movie Title	Movie Genre	Release Year
1	Avatar	Adventure, Fantasy, Action	2009-12-18
2	The Godfather: Part II	Drama, Crime, Thriller	1974-12-25
3	The Godfather: A Novel	Fiction, Mystery, Thriller	1954-01-01
4	The Godfather: A Novel	Fiction, Mystery, Thriller	1954-01-01
5	The Godfather: A Novel	Fiction, Mystery, Thriller	1954-01-01
6	The Godfather: A Novel	Fiction, Mystery, Thriller	1954-01-01
7	The Godfather: A Novel	Fiction, Mystery, Thriller	1954-01-01
8	The Godfather: A Novel	Fiction, Mystery, Thriller	1954-01-01
9	The Godfather: A Novel	Fiction, Mystery, Thriller	1954-01-01

Fig 4 Recommendation generated above for "Avatar movie".

Result:

This project results in providing assistance to anyone who wants to know what to watch next doesn't need any subscription for recommendation and watch history, and want to recommendation previous rates and patterns in detail instead of taking a blind shot at choosing what to watch.

Acknowledgement:

I express my deep sense of gratitude to our respectful and learned guide Prof Sagar Patel for his valuable time and efforts. I am thankful for his help, guidance and constant encouragement that helped to complete project. Also thankful to him for allowing to use all the necessary facilities of the institute.

References-

1. <http://dprogrammer.org/rnn-lstm-gru>
2. <https://medium.com/code-heroku/building-a-movie-recommendation-engine-in-python-using-scikit-learn-c7489d7cb145>
3. <https://towardsdatascience.com/introduction-to-two-approaches-of-content-based-recommendation-system-fc797460c18c>
4. <https://programminghistorian.org/en/lessons/creating-apis-with-python-an-flask>