

MOVIE RECOMMENDATION SYSTEMS

Name : Disha Sunil Nikam

Name : Hitesh Bishnoi

Name : Jayasimha Sathya

SRN: PES1UG20CS133

SRN: PES1UG20CS164

SRN: PES1UG20CS177

Abstract and Scope

- The project aims at using content based and collaborative based models to recommend similar movies to the user.
- The covid enhanced digital era has created a large population of OTT users worldwide. The project aims at recommending movies to users which will be useful in this scenario.
- The project will also aid movie production companies to make well informed decisions based on the insights from the EDA performed.



About the dataset

We have used TMDb 5000 Movie Dataset from Kaggle.

“tmdb_5000_movies.csv” contains-


homepage, budget, genres, keywords, id, original_title, original_language, revenue, runtime, overview, popularity, title, vote_count, production_companies, production_countries, release_date, spoken_languages, status, tagline, vote_average.

“tmdb_5000_credits.csv” contains-

movie_id, title, cast, crew



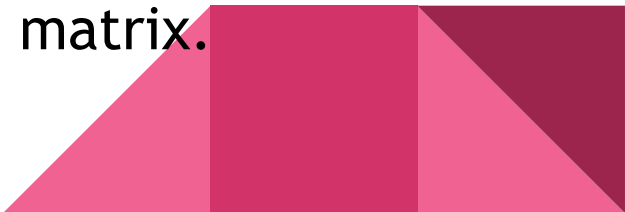
About the dataset continued

- 'Rumour' value is dropped from 'status' attribute as rumoured movies may/may not be released. Similarly 'post production' shouldn't be considered as the movies aren't released yet.
 - 'homepage', 'tagline', 'runtime', 'release_date', 'overview' are attributes with many null values .
 - The null values of the 'runtime' attribute is replaced with the median value.
 - The rows with 0 as their value are dropped for the attribute 'popularity'.
 - The 'homepage' and 'tagline' attributes are dropped as they do not contribute to the analysis.
- 

Implementation Details

Model 1- Content based filtering (TFIDF)

[recomm_movies_content.py]

- The values in the 'overview' and 'genres' column are combined to form a new column called 'statement'.
 - A TfidfVectorizer() object is created by importing sklearn.feature_extraction.text from TfidfVectorizer.
 - fit_transform is called on the new 'statement' column and that represents the tfv matrix.
 - Sigmoid_kernel() method is applied to this matrix.
- 

Model 1- Continuation


- Recommendations are found by first getting the index corresponding to the movie titles, retrieving pairwise similarity scores, sorting the movies based on the scores and finding the scores of 10 most similar movies.
- The top ten movies along with their ids are returned as recommendations to the user.



Implementation Details

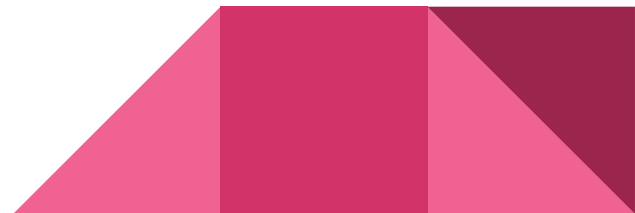
Model 2- Collaborative based filtering (KNN)

[recomm movies knn.py]

- The required columns for this model are 'genres', 'cast', 'director' and 'keywords'.
 - For each of the columns the unique value lists are obtained.
 - Next step is binary encoding.
 - For each row in the dataset the values present in that row for a unique value of 'genre' is represented as 1 in a list and 0 otherwise.
- 

Model 2- Continuation

- For each row in the dataset the values present in that row for a unique value of 'cast' is represented as 1 in a list and 0 otherwise.
- For each row in the dataset the values present in that row for a unique value of 'director' is represented as 1 in a list and 0 otherwise.



Model 2- Continuation

- For each row in the dataset the values present in that row for a unique value of 'keyword' is represented as 1 in a list and 0 otherwise.
- On passing two movie ids to the Similarity() method, genreDistance is calculated using `spatial.distance.cosine()`.



Model 2- Continuation

- The scoreDistance is calculated similarly using the cast attribute encoded values.
- The directDistance is calculated similarly using the director attribute encoded values.
- The wordDistance is calculated similarly using the 'keyword' attribute encoded values.



Model 2- Continuation

- Sum of these distances is returned in the Similarity function.
- Another function `predict_score()` is defined.
- The distance is calculated for each movie from the current movie entered using the Similarity function calls. This distance calculation takes place in the `getNeighbours()` method within `predict_score()`.



Model 2- Continuation

- The distances are sorted and the top 10 closest movies(Neighbours) are retrieved.
- The movie names,ratings and genres similar to the searched movie are returned by the predict_score() method as recommendations to the user.



Project Demonstration

The project is deployed in a Flask environment (Webapp)

- 1) A text box that retrieves the movie title via the user and runs a collaborative based KNN model with required features from the dataset and prints a table with recommended movies.
- 2) A second text box that retrieves the movie title via the user and runs a content based filtering model using TF-IDF, required dataset features and prints a table with recommended movies.



Movie Recommendation System

[About Us](#)


Enter the Movie Name to get Recommendations for

Search for similar movies using knn

Enter the Movie Name to get Recommendations for

Search for similar movies using content based filtering

Conclusion and Future Work

- 1) Broadening the dataset to provide better results.
 - 2) Using different set of features that may improvise on the current results of the project.
 - 3) Using other refined methods to calculate features like 'popularity', 'ratings' and trending movies.
 - 4) Further analyze the dataset to obtain various new insights that will be fruitful to the production companies and OTT platforms as well.
- 

References

- Costin Chiru Movie “Recommender system using the user's psychological profile” September 2015 Conference: 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)
- Jianjun Ni, Yu Cai, Guangyi Tang, Yingjuan Xie “Collaborative Filtering Recommendation Algorithm Based on TF-IDF and User Characteristics” 2021 Multidisciplinary Digital Publishing Institute
- Urszula Kuźelewska “Clustering Algorithms in Hybrid Recommender System on MovieLens Data” January 2014 STUDIES IN LOGIC, GRAMMAR AND RHETORIC 37 (50) 2014



References

- Rujhan Singla; Saamarth Gupta; Anirudh Gupta; Dinesh Kumar Vishwakarma “FLEX: A Content Based Movie Recommender” Published 1 June 2020 Computer Science 2020 International Conference for Emerging Technology (INCET)
- Triyanna Widiyaningtyas 1, Muhammad Iqbal Ardiansyah 2 and Teguh Bharata Adji “Recommendation Algorithm Using SVD and Weight Point Rank (SVD-WPR)” Triyanna Widiyaningtyas 1, Muhammad Iqbal Ardiansyah 2 and Teguh Bharata Adji Big Data Cogn. Comput. 2022, 6, 121

Thank You!!

