Movie Recommendation System

Team : Unknown

193050071, Rohit Kumar
193050003, Suraj Kumar
193059002, Rajershi Gupta

27th November, 2019

Contents

1 Introduction

Wikipedia states "A *recommender* system or a recommendation system (sometimes replacing system with a synonym such as platform or engine) is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item". The term recommendation is very abstract and therefore leads to subject to interpretation. Coming to our project of a Movie Recommendation System, we give the user one search box, wherein the user queries, whatsoever he/she remembers about a particular movie.Then, we do a token-wise match to our extended dataset. Thus, producing a result which would be the most probable set of movies, as per the user search information.

1.1 Prior Work

Recommender systems are made and used widely in different fields of engineering. There have been many blogs and post which discuss about various filtering techniques. Though for this project we used a simple filter which runs on the tokenized input to give the most probable output from a given dataset.

1.2 Dataset and Preprocessing

The IMDB dataset was taken from Kaggle from the following url:
https://www.kaggle.com/orgesleka/imdbmovies
The dataset has nearly 14000 movies each with attributes: *tid,title, wordsInTitle,url,imdbRating,ratingCount, duration,year,type,nrOfWins, nrOfNominations,Animation,Biography,Comedy, Crime,Documentary, Drama,Family,Fantasy,FilmNoir,GameShow,History,RealityTV,Romance, SciFi,Short,Sport,TalkShow,Thriller,War,Western*
The total number of attributes being 44.

The extended dataset is made which has the following attributes in addition to the Kaggle IMDB dataset: actor(s), character(s), title(English) and storyline. Also, certain attributes are discareded from the Kaggle dataset which are not required for the search and results. For example, nrOfWins,nrOfNominations, nrOfPhotos,nrOfNewsArticles,nrOfUserReviews,nrOfGenre etc. Finally, a total of 36 attributes were present in the extended dataset.

1.3 Objectives of the Project

We will make a movie recommendation system wherein the user queries, whatsoever he/she remembers, are put in the search box and we do a token-wise match to the extended dataset.

- To extend the initial Kaggle dataset(using Web scraping) which will include actor(s),character(s) and storyline and title,in English) from the available dataset. The extended dataset will also have discard the useless attributes which are not required.

- To make a web interface which will have input the user search query and henceforth show the required output.
- To give most logical output and do an efficient search on the search query.

2 Motivation

At times when we want to recommend a movie to one of our friends we somehow do not remember the name of the movie. Rather, something sticks with us about the movie, be it the actor or character etc. Therefore, we thought if the user queries the actor(s) or character(s) or title or story of the movie, then we can output the most probable movie from the dataset, with all its details in the dataset to the user.Adding to that, movies that are still relevant to the search query are also given with their names. Furthermore, this tool can also be used to recommend the movie to the user query according to the relevant search the user makes, to reccomend user certain similar kind of movies.(by Genre, Actor(s), Character(s), StoryLine etc.)

3 User Documentation

The extended dataset has the following attributes:

The following steps are required for the user to get the movie recommended by our system :

1. Clone the repositry from the git link given in readme.txt
2. Fetch the preprocessed csv file from the path: ../../web/preprocessing/dataset.csv
3. Clean the file or take the final MoviesData.csv file from ../../web/moviesData.csv
4. Install XAMPP and set the path to the source code "../web/"
5. Make the database from the given file createMovieDB.py, using the following command:
   python3 createMovieDB.py
   (prerequisite : **python3** and **SQLite** are already installed)
6. Launch **XAMPP** and start the server
7. open your favourite browser and put the following url : *localhost/moviePrediction.py*
8. In the search query given at the top-right corner of the website opened, the user can give the query. Note that the following few assumptions are made for the query:
   (a) If **query length <6**, then the search is made on all attributes of the extended dataset except storyline.

(b) **If query length >5**, then the search is made on the storyline only. The best matched entry is returned.

9. The webpage would redirect to the output page where the results can be seen

4 What all works and what not?

4.1 Usefulness of the project

This project can be used in two practical scenarios:

- When one has to recommend a movie to a peer and the name of the movie is long forgotten, the a search with whatever one remembers: the storyline or character,actor, title, release year etc. will give the most probable outcomes.
- When one has to see a certain kind of movies, then a search gives many movies that one can relate to with the queried input

The extended dataset is very proficiently build from the Kaggle dataset. It can be used for any future projects too. The search query works well for many tested cases. Additionally, the code for search query is written considering to output best results in mind and not optimised for the fastest results. Since, the difference in time of results will be insignificant in real life.

4.2 Limitations

Following are the listed drawbacks or pitfalls of the output:

- Movies are only present from 1890s to 2015, which may not include all the movies that were released in this time frame.
- The code for search query is optimized and is written considering practical cases in mind, which may not always result in the result the user expects.