Statement of Work

SOW for Movie Recommendation System

Date	By:	For:
2020-12-01	Muhammad Faizan Ali 100518916	AI Algorithms 1 - AI Analysis, Design and Implementation

Rationale/Problem

Now days the trend is for everything to be available on our computers and phones that can be accessed from anywhere at any time. But with so much content available in the palm of our hands, it has become harder and harder to find what you are looking for. Due to the sheer amount of content available, It has become necessary to have some way of sorting through all the data to find what you are looking for easily.

One thing that we all love to do in our free time is watching movies or tv shows, and as with all things now days you have easy access to almost all the movies and tv shows ever made with the click of a button. The problem is how do we find something we are interested in with so much content available. That is where a recommender system comes in, I will be developing a recommender system that will help you find movies to watch from a list of thousands of movies.

In this project I will be recommending movies to a certain user based on movies previously watched by that user as well as movies liked by other people with similar taste in movies as the user. The movie recommender system will consider features such as the rating of the movie, and the number of people that gave a rating to the movie when making recommendations to a certain user.

Data Requirements/Assumptions

1. List of movies
    - The genre of the movie
    - The id of the movie
2. Ratings given to a movie by users
    - Ideally ratings by at least 10 users per movie for it to be considered reliable
    - Ideally each user has at least 5 movies liked to make better predictions
3. There are at least 10,000 movies available to make recommendations from
4. There is data for at least 10,000 users

Data Source/Description

Data Source:

MovieLens 25M Dataset: https://grouplens.org/datasets/movielens/25m/

Data Source Info:

MovieLens is run by GroupLens, which is a research lab at University of Minnesota. This data is publicly available for download and use for non-commercial purposes.

I do not have permission to redistribute the data so will include the link to the data source in my Github. Read the Readme file at the link above for more info on usage of the data.

Data Info

This data consists of Six CSV file related to the movies and the ratings/tags assigned by the users to those movies.

6 CSV Files:

1. movies.csv
    - List of movies
    - Movie ID, Movie title, Movie genres
2. ratings.csv
    - Rating assigned to a movie by users
    - User ID, Movie ID, Rating, timestamp
3. tags.csv
    - Tags assigned to a by users
    - User ID, Movie ID, Tags, timestamp
4. links.csv
    - Links to the movie on other sources
    - movieId, imdbId, tmbId
5. genome-tags.csv
    - IDs of tags assigned by users
    - tagId, tag
6. genome-scores.csv
    - movie ID, tag id, relevance

Acknowledgement

F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19. https://doi.org/10.1145/2827872

Test Process

I will split the data into three categories, training dataset, validation dataset and test dataset. The model will be trained using the training data and then evaluate it using the validation data. Based on accuracy of the validation data I will make changes to hyperparameters of the model and retrain the model until I get an acceptable accuracy (90%). Once an acceptable accuracy is reached I will test the model using the test data, and make sure the accuracy is still similar to the validation data accuracy.

I will pre reserve ~10% of the data as test data(holdout dataset) before I split the remaining data into training and validation data. I will then use K-fold cross validation to find the optimal split for training and testing data so I can get the best accuracy.

EDA – Statistical Analysis Observations

The original data contained a lot of files but we are interested in just 3 of them: ratings, movies, and links. The EDA was performed on these 3 files in order to identify the main characteristics of the data. Movies file contains the list of movies, there unique id and genres. Ratings files contains the ratings given to each movie by a user. Links contains external links to imdb and tmdb for each movie.

After analyzing the 3 files we were able to identify that the data is not missing any values in the two main files,ratings.csv and movies.csv. The links.csv file is missing links to some of the movies for tmdb but all the ones are present for imdb, so we will just remove the data for tmdb and use imdb only.

All the data that we need is numerical data in all files except for movie title and genre in the movies.csv file. But since each movie is associated with a unique numerical variable movieId we do not need to do anything to the movie title and can just use that instead for the model.

Stats:

- Total movies in database: 62,423
- Total Ratings: 25M+
- Unique Users: 162,541
- Most movies had a rating between 3.0 and 5.0
    - 3.0/5.0 and 4.0/5.0 being the most common
- ~7% of the movies had a rating of 1 or less
- On average each movie had ratings from 423 unique users
- There are 107 movie links for the tmdb page in the links.csv file
    - No other files is missing any values

Data Cleaning

The tmdb webpage links in the 'links.csv' file are missing for some of the movies in the data. We have links for all movies in the imdb website so need to try and correct the links for tmdb we can just delete them all.

Some movies have ratings by only 1 user, which is not enough to be considered to make accurate recommendation. I will remove all user ratings where less than 10 users voted before training the model.

Feature Engineering

Categorical variables are not good for computing and since we have a lot of data(25M+ total ratings) categorical variables will slow down our model training considerably. We need to convert our categorical variable 'genre' for movies into a numerical variable, which I will do using One Hot Encoding. This will create new features for each unique genre available, assigning a value of 1 to features present and 0 to features not present.

The title of the movies currently also contains the year they were released, so I will sperate the movie name from the title and add it as a separate feature to the movies dataset. This can be useful for the model to help recommend movies to a certain user as some users may not like older movies due to the lower picture quality and bad graphics.

Making these changes to the data will help speed up the training of them model as well as help increase the accuracy of the model when we are training it.