

"Assignment 7"

Recommender System

Info

Name : Mohamed Mostafa Mohamed

Sec : 2

BN: 20

Introduction

This report will explain functions and algorithms used to build a movie recommendation system in python.

Part 1 - Data Preprocessing

Here's an overview about the taken steps to filter the data:

1. Read the ratings.csv and movies.csv files and turned them into pandas dataframe
 2. Got the index of the last row of user id = 200 which is the row number 29267, then removed all rows after it in order to keep only 200 users ratings as required.
 3. Merged the ratings and movies datasets together, then calculated how many ratings each movie has in order to keep only the most popular 200 movies (Movies that have more than 24 ratings, I still had 10 additional movies that I removed after this step).
 4. Merged the data again with only the most popular movies I got from the previous step.
-

Part 2 - Similarity Matrix

Here's an overview about the taken steps to calculate the similarity matrix:

1. Created a matrix with users IDs as rows and Movie Titles as columns, and filled the matrix with the given ratings data if given, otherwise filled it with Nan.
2. Normalized the matrix with the mean by subtracting the mean of each row divided by maximum data of row minus minimum data of row from each data inside the row.
3. Filled the NaN values in the matrix with zeros as the cosine similarity function doesn't deal with NaNs.
4. Applied Transpose transform on the previous matrix in order to switch rows and columns, then applied the cosine_similarity function on it to calculate the covariance between each movie and the other (gives back an array of arrays that holds the scores of how much each movie is similar to all other movies).

Part 3 - Getting Similar Movies

Here's an overview about how similar movies are recommended:

1. First we make sure that the given movie is inside the 200 movies list that were chosen from the data.
2. We get the index of the given movie in the cosine_similarity array to get its covariance with other movies, then we access that index and retrieve the data.
3. We sort the retrieved data descendingly, and take only the highest 10 items (we skip the first/highest item as it's actually the same given movie itself).
4. We map back those items to movie ids and titles, then print them out.

Part 4 - Recommend Movies To User

Here's an overview about how movies are recommended to a user:

1. We make another cosine_similarity matrix that calculates the covariance between each user and the others.
2. Just like the previous steps, we get the most 10 similar users to the given user, which is user number 200 in our case.
3. We drop down the movies that the given user has already watched from the data, Also the movies that none of the 10 similar users have watched.
4. We decide which movie to recommend to the target user by calculating movie_score for each movie left in the data. The recommended movies are determined by the weighted average of user similarity score and movie rating. Also the movie ratings are weighted by the similarity scores between the given user and each similar user so that the most similar users have a bigger effect on the results.
5. We sort the movies descendingly according to the calculated movie_score, then we print out the highest 3 movies and their scores.