UNIVERSITÀ DEGLI STUDI DI MILANO
FACOLTÀ DI SCIENZE E TECNOLOGIE

DIPARTIMENTO DI INFORMATICA
GIOVANNI DEGLI ANTONI

Algorithms for massive datasets

RECOMMENDER SYSTEMS (COLLABORATIVE FILTERING)

Project Owner:
Stefano Capelli
Matr. Nr. 47269A

Academic Year 2023 - 2024

Contents

Introduction

This project presents a recommendation system for a movie dataset using collaborative filtering based on the actors.
This paper aims to present the project and explain the reasoning behind the decision taken and the code written.
The dataset used is Letterboxd from Kaggle (last updated around summer 2024) and all the code is contained in the .ipynb file developed using Google Colab (Python3).
The code is commented in the notebook, so here there will be no screenshots or direct references to it.

Chapter 1

Database Analysis

1.1 Structure

The dataset includes several files but for the porpuse of this project the focus is only on:

1. movies.csv : The main file containing all the informations about the movie
2. actors.csv : An association of actors to the movies they have starred in
3. genres.csv : An association of movies to their genres

The main focus of the project is on movies and actors. Genres has been considered and tested, but in the end dropped as explained later in the paper.

1.2 Data Organization and Preprocessing

All the data has been divided into chucks before being imported, in order to maintain a stable input and manage the big dataset and, with the same porpuse of optimization, the code defaults to a subsampled portion of the initial dataset (around 2%).
There is the possibility to modify the size of the samples or fully disable the subsampling option.

1.2.1 Movies

The initial analysis of the movie dataset showed an high percentage of missing data, in particular for the columns

- Rating : the rating of the movie
- Tagline : the slogan of the film
- Description : a brief description of the movie
- Minute : the duration of the movie
- Date : the release year of the movie

'Tagline' and 'Description' has been considered optional and not important to the project, so the decision has been to drop them.
For the other missing data, the approach has been different.
The columns have been renamed in order to be more clear.

Data Management

Name and IDs of the movies weren't missing, but to be sure every record missing values has been dropped, while minutes and rating has been filled with the average value of the non missing records.
Minutes and especially rating are important in order to recommend movies, so the decision to drop the missing records would have been terrible, since most of the rating were missing.
Minutes had some big outliers containing the likes of the world record longest movies, so the option to drop them has been considered, but in the end not taken in order to keep the integrity of the dataset. So, the decision has been to normalize the minutes.
Finally, the date has been checked for validity and converted into int32 format, since the year wouldn't need much space to be stored and could be easily optimized.

1.2.2 Actors

Actors contains only the movies ID and the relative actors name and role, each of them renamed.
The role has been dropped and the focus has been on the missing values (dropped) and the duplicates, this means that in the dataset there were multiple instances of the same movie ID and actor (with different role).
After removing the duplicates we could see the distribution of the actors and the numbers of movies they participated in.

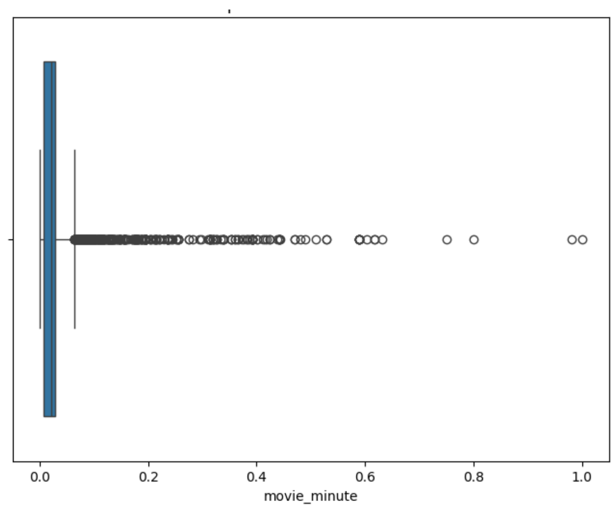


Figure 1: Boxplot

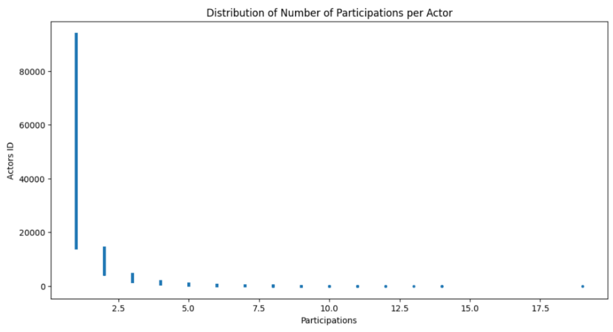


Figure 2: Participation

1.2.3 Genres

The genres has been renamed and checked for missing values.
A movie could have multiple genres, this is one of the thing that influenced the slow development of this part, so the genres has not been studied further.

Chapter 2

Algorithm Implementation

2.1 Utility Matrix

In order to build a consistent and clear matrix, the actors' names has been converted to lower case and checked of eventual spaces.
It has been created an actor ID unique, in order to better handle the process. A mapping manages the conversion from actor ID and name, making sure to not create any duplicates.
The decision has been to still keep the actor's name for easier debugging.
For the same reasons as before, the movies ID in the actors data structure has been converted to a new ID starting from 0, with a proper mapping.
The mapping of the movies IDs to the new ones has been implemented with a permutation, in order to not always have a linear combination of (movie 1; actor 1)...
The sparse utility matrix has been created utilizing a CSR Matrix in order to optimize the space used, so the decision has been to create the numpy arrays containing the values of rows (movies), columns (actors) and data (1 if actor is present in the movie).
The utility matrix has been plotted for an easy check of its structure and the density has been calculated.

2.2 Recommendation

The Algorithm has been divided into dedicated functions.

- get random good movies: this function is used to get some movies with rating above average and different from the ones that the actor has participated in,

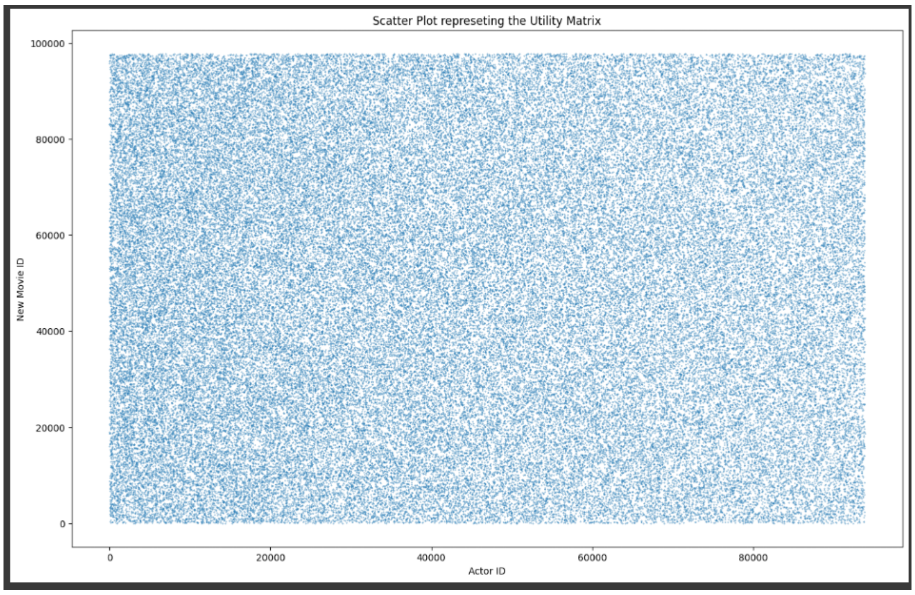


Figure 3: Plot of Utility Matrix

but it is called only if the algorithm is not able to properly recommend movies to the actor

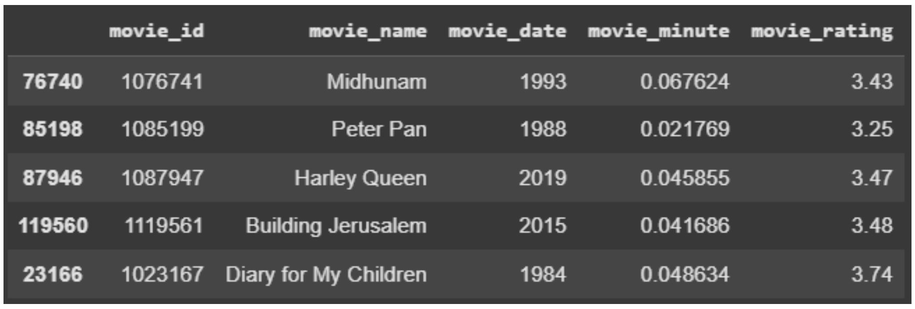
- actor recommend : it's the main function, utilizing the utility and similarity matrix to find similar movies to recommend, avoiding the ones the actor has been part of. The movies are ranked in a top 5 and returned
- get random recommendation: this function has been defined to pick a random actor and recommend the movies, if the specific recommendation is not possible, it calls the first explained function in order to get some good movies to recommend.

Chapter 3

Conclusion

3.1 Results

The Algorithms works best with a lot of data and with actors that has associations with a lot of movies, but works fine also in worse conditions.
In order to be sure to check the functionalities, it's possible to modify the random recommendation function in order to pick another actor is the personal recommendation is not possible.



	movie_id	movie_name	movie_date	movie_minute	movie_rating
76740	1076741	Midhunam	1993	0.067624	3.43
85198	1085199	Peter Pan	1988	0.021769	3.25
87946	1087947	Harley Queen	2019	0.045855	3.47
119560	1119561	Building Jerusalem	2015	0.041686	3.48
23166	1023167	Diary for My Children	1984	0.048634	3.74

Figure 4: Example of recommendation

3.2 Future Implementation

During the tests, the recommendation was ordered by genre, so that the more similar movies to the ones the actor has been part of would be showed first, but it was not fully developed and so not present.

3.3 Declaration

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.