

Computer Science Internal Assessment

Criterion A: Planning

Word Count: 399

Movie Recommendation System

Defining the Problem - Finding an interesting movie to watch

Client Mr Khilan Shah is my neighbour; we live in the same building. Since the lockdown began, he had a lot of spare time and binge-watched movies on Netflix. After finishing the movies on his list, he referred to his recommendation list. However, he did not like any of them.

Once, when we spontaneously made a plan to meet, I asked him for some movie recommendations, and he told me about his problem. After giving it some thought, I realised that it would be an interesting idea for my Computer Science Internal Assessment to create a recommendation algorithm for Mr Khilan.

While I knew that my algorithm would not be more efficient than Netflix's, I realised that Netflix did not suggest movies based on what my client liked but rather what he watched, making the suggestions flawed in this scenario.

Therefore, I later decided to set up a formal interview, which can be accessed through Appendix A, with Mr Khilan. We discussed the specifications demanded and required by him, including his genre preferences, whether he wanted a website or an application, and the number of movies he wanted in a single suggestion list. Later, in our next meetings, we also discussed the user interface, i.e. pages and sub-pages on the application, the background and aesthetics—specific designs, icons, colours— and also the required functionality aspects of the program.

Rationale for Proposed Solution

An application through a Python program can solve this problem as it will be easy to import, view, edit and iterate the MovieLens dataset from Kaggle datasets archive, which is useful because it has information on ratings and each individual movie, including the genre and date released. The matrix-factorisation algorithm will be adopted from, and referenced appropriately, Scikit-learn, a python library with effective tools and functions. To create an

accurate recommendation system, I plan to use the nearest neighbour algorithm to predict users' ratings for all movies in the dataset. This method will address the issue of sparse data and thus increase the speed of the algorithm, also allowing for enhancing the accuracy of the recommendations.

The client will have to first create an account by entering necessary details, and then enter a movie based on which the algorithm will output recommendations.

I decided to use a Python program because I believe it will be highly employable due to following factors:

1. Easy to import, read, edit, reiterate large amount of data, which is useful because a new, comprehensive ratings matrix must be derived.
2. The above will be aided through the use of the Panda library because it simplifies iteration, data manipulation and data wrangling.
3. The Numpy library in python is also effective as it consists of in-built mathematics functions and can be used while implementing the matrix-factorisation algorithm of the large ratings matrix
4. Lastly, the matrix factorisation algorithm can be imported from the Scikit-learn open-source library in python

Success Criteria

The Program will:

1. Recommend to Client a personalised and sorted list of movies to watch on Netflix.
2. Recommend particular number of movies that the client wants
3. Display the name and year each suggested movie
4. Display an 'error' message if the requirements for the algorithm are not met
5. Guide the client with the use of sign-posts and instructions
6. Secure the client's password and other important details
7. Verify the user's log-in information with a secure database

The Client will:

1. be able to log in
2. Be able to sign up
3. Be able to navigate through the login and signup page smoothly
4. be able to navigate through tabs, input boxes and filters smoothly
5. Be satisfied with the recommendations given

Citations

Source: Small Movie Lens Dataset from Kaggle Datasets, 2018: *Movie Lens Small Latest Dataset*, Shubham Mehta,
<https://www.kaggle.com/datasets/shubhammehta21/movie-lens-small-latest-dataset>