

GitHub: <https://github.com/jayaaas/326-Final-Project>

Movie Dataset File: <https://www.kaggle.com/danielgrijalvas/movies>

Presentation Video: <https://youtu.be/gpKuhvVRNIE>

Documentation:

What our Project is and What it Does:

Our project is a program that has two functions: a movie searcher and a movie recommender. The recommender gives the user a list of movies to choose from based on what they input as their preferences, and the movie searcher gives the user information about a movie in the database based on what movie title they request.

How to Run the Program:

To run our program, simply download the Python file and type “python final_project.py” into the command line. The program will then prompt you with questions and will give you a clear format to follow in your responses. Following these prompts will eventually take you all the way to the end of the program.

How to Use/Interpret the Output:

The output gives the user a list of movies that match what the user put in for their preferences or options. With the output, the user can make a faster decision on what they should watch. Also, with the recommender system, once they get the output from that, they can then search for one of the movies using the search feature. This way, they can get information about the director, what year it was released, and the score so the user can further compare and narrow down their decision.

Annotated Bibliography:

Bibliography

Pykes, Kurtis. *How to Use Pytest for Unit Testing*. 15 July 2022,

www.datacamp.com/tutorial/pytest-tutorial-a-hands-on-guide-to-unit-testing.

Pytest is a way to test functions in a program to make sure they are functioning properly especially if they aren't returning or printing anything. A popular way to debug is to add print statements, but adding assert statements and unit tests can tell you what the function would get versus what the correct value should be. We plan to use this source in addition to module 4 and exercise 4 to write unit tests for the project.

Sharma, Aditya. *Beginner Tutorial: Recommender Systems in Python*. 29 May 2020,

www.datacamp.com/tutorial/recommender-systems-python.

This source was helpful because it gave a little bit of a background as to what recommender systems usually consist of. While some databases contain information that others don't like votes or ratings, it is still a good source of information to what is usually included or compared to. It mentions how you can create a dataframe which we ended up doing in a different way by using pandas.

Srivani, Karpuram Dhanalakshmi. "Movies Recommendation System Using Python." *Analytics*

Vidhya, 28 Aug. 2022,

www.analyticsvidhya.com/blog/2022/08/movies-recommendation-system-using-python.

This source was a more advanced system that talked about comparing two different datasets, one with movies and the other with credits. They also included the cast in the dataset, and filtered many different ways like calculating popularity, similarity, and more.

If more time was granted, this would have been fun to try and figure out, but like the other sources, it provided us with information about what these systems usually look for. Vidiyala, Ramya. "How to Build a Movie Recommendation System - Towards Data Science."

Medium, 9 Aug. 2022,

towardsdatascience.com/how-to-build-a-movie-recommendation-system-67e321339109.

Just like the source directly above, this talks about a more advanced way of creating a movie recommender system. Instead of ours where it very specifically matches up with exactly what the user put in, this one finds exactly what the user put in but also finds movies that are similar to it. This way, the user can still get recommendations of something that they didn't exactly put in, but would still enjoy.