Chatbot Sessions Links:

- Main one I used: https://chatgpt.com/share/68f17b7a-70f4-8005-ae9e-5080565e63cc
- https://chatgpt.com/share/68f17b98-1318-8005-b8b8-23200cf2caa2

Chatbots used:

I chose to use ChatGPT for the entire assignment because I am most familiar with it.

Case-sensitivity handling:

When I was writing the program, the only part of the data to be processed in a case-sensitive manner was the genre name. This is because when I was running the program, I realized that when the user goes to input the genre they want to see for "Show top N movies in a genre", they have to input it in a way that matches exactly how it appeared in the data file ('Comedy', 'Adventure', etc.). If I tried to type 'comedy' in all lowercase, the function would not recognize this input. For all other inputs, the user is only typing in integers to denote 1 of 3 things: the option they are choosing from the menu, the number of movies/genres they want to be displayed (N), or the user ID of the a specific user to return their favorite genre or 3 movie recommendations. The chatbot did not automatically handle the case-sensitivity when inputting the name of the genre, but it was able to resolve this issue with a little bit of prompting and a few tweaks in the code, by stripping and converting all user inputs to lowercase and comparing them to the lowercase genres to see if they match.

Unclear code/chatbot explanations:

I was able to understand almost all of the code that ChatGPT was giving me as it was pretty straightforward logic. That isn't to say that I would have been able to write the entire program myself without the help of AI, but I found reading and understanding the code and chatbot explanations to be rather simple. One built-in python feature that was not covered in class that ChatGPT implemented was setdefault(), so I asked ChatGPT about it and it explained it in a generic sense with simple examples in addition to showing me how this method works in my movie recommender program.

Input data issues handling:

The AI did a good job handling the basics of input data issues, like checking for file existence and empty file handling. It also did a good job automatically using strip() to handle issues with blank spacing and poor formatting. Outside of that, however, there were still many issues left to be handled.

Issues not handled by AI, code fixes:

There were a plethora of issues not automatically handled by AI that I had to keep adding in long after I thought I was done. Some of these include:

- Ensuring the user loads in both files before using the other menu features
- Incorrectly/Partially correctly formatted files
- Duplicate entries
- Missing fields
- Invalid ratings (outside of the 0-5 range)

As I kept running the program and playing around with different inputs, I would see the gaps in the code to check for these types of errors, and proceed to prompt the chatbot. It was able to eventually handle all these issues with ease once I brought these issues to its attention.