

**Movie Recommendation System**.

**REPORT ON**

**Data Science and Big Data Analytics (SEMESTER -VI)**

*Submitted by:*

Mrudula Jambhalkar(31426)

Kedar Koshti (31435)

*Under the guidance of:*

Prof. R. S. Paswan



**THIRD YEAR ENGINEERING**

# Society for Computer Technology and Research’s

**PUNE INSTITUTE OF COMPUTER TECHNOLOGY**

# DHANKAWADI, PUNE – 43

**A.Y. 2022-23**



**Title**: Developing a Movie Recommendation Model Using the Scikit-Learn Library in Python

**Problem Definition**:

Develop a movie recommendation model using the scikit-learn library in python. Refer dataset <https://github.com/rashida048/Some-NLP-Projects/blob/master/movie_dataset.csv>

# Learning Objectives:

1. Develop a content-based movie recommendation model using scikit-learn library to provide personalized movie suggestions based on the attributes and preferences of users.
2. Evaluate the performance of the model by measuring accuracy, coverage, and diversity of recommendations, using appropriate evaluation metrics such as precision, recall, or NDCG.
3. Explore enhancements to the model, including hybrid approaches or incorporating user feedback, to further improve the relevance and effectiveness of the movie recommendation system.

# Learning Outcomes:

Students will be able to:

1. understand the need of dataset and develop a content-based movie recommendation model.
2. understand the working of movie recommendation model.

# Requirements:

* Computer System with:
* I5 processor, 256 GB SSD, 8GB RAM.
* Jupyter Notebook
* Python with numpy, pandas, matplotlib.

# Motivation:-

* Personalized User Experience: Movie recommendation systems offer users a personalized and tailored movie-watching experience. By analyzing user preferences and movie attributes, the model can suggest relevant movies that align with individual tastes and interests, enhancing user satisfaction and engagement.
* Information Overload: With the vast amount of movie options available, users often face difficulty in discovering movies that align with their preferences. A recommendation system helps users overcome information overload by providing curated suggestions, saving time and effort in searching for movies manually.
* Business Benefits: Movie recommendation systems have become a crucial component for streaming platforms, as they can drive user engagement, increase user retention, and boost revenue. By providing accurate and personalized movie recommendations, these platforms can improve user satisfaction and loyalty, leading to higher customer acquisition and retention rates.
* Exploring New Movies and Genres: Recommendation systems expose users to a wider range of movies and genres they might not have considered before. By suggesting relevant but lesser-known movies, the model encourages users to explore and discover new cinematic experiences, broadening their movie preferences.
* Learning and Applying Machine Learning Techniques: Developing a movie recommendation model using the scikit-learn library offers an opportunity to learn and apply machine learning techniques. It allows for the practical implementation of data preprocessing, feature extraction, similarity computation, and model evaluation, providing hands-on experience in building real-world machine learning applications.

Overall, building a movie recommendation model presents an exciting opportunity to create personalized experiences for users, tackle information overload, benefit businesses, foster exploration of new movies, and gain practical experience in machine learning techniques.

# Abstract:-

In this assignment, we develop a movie recommendation model using the scikit-learn library in Python. By employing content-based filtering, we extract movie attributes such as genre, cast, director, and plot to generate personalized recommendations based on user preferences. The model's performance is evaluated using precision, recall, and diversity metrics, ensuring accurate and diverse movie suggestions. Additionally, we explore enhancements, including hybrid approaches and user feedback incorporation, to further improve the relevance and effectiveness of the recommendation system. Through this assignment, we aim to provide users with a personalized movie-watching experience while acquiring practical knowledge in machine learning and recommendation systems.

# Hardware Requirements:-

* 1. Processor: Intel Dual Core, i5, i7..
  2. Ram: 8 Gb
  3. SSD: 512 Gb

# Software Requirements:-

1. Operating System: Windows 11
2. Front End: Anaconda, Jupyter Notebook

# Project Background:-

Recommendation systems are information filtering systems that suggest items to users based on their preferences, interests, or behavior. These systems are widely used in various domains, such as e-commerce, social media, music streaming, and movie streaming platforms, to help users discover relevant and personalized content.

# Dataset:

# The dataset used for this assignment is called "movie\_dataset.csv" and it is available on GitHub. This dataset contains information about various movies such as the movie title, the cast, the genre, the plot, etc.

# The dataset consists of the following columns:

# movie\_title: The title of the movie.

# cast: The names of the actors in the movie.

# genre: The genre of the movie.

# director: The name of the director of the movie.

# plot\_keywords: The keywords related to the plot of the movie.

# movie\_imdb\_link: The link to the movie's page on IMDb.

# language: The language of the movie.

# country: The country where the movie was produced.

# content\_rating: The content rating of the movie.

# budget: The budget of the movie in dollars.

# title\_year: The year in which the movie was released.

# imdb\_score: The IMDb score of the movie.

# Common Challenges

# In this project, three of the above recommender systems were build using different algorithms. Following challenges were encountered while building these recommender systems:

* Data sparsity
* Popular bias (how to recommend products from the tail of product  distribution)
* User Inconvenience with spelling while giving movie names as input (can't  expect to give exact movie name every time)
* Scalability (computation grows as number of users and items grow)
* Pool relationship between like-minded yet sparse users.

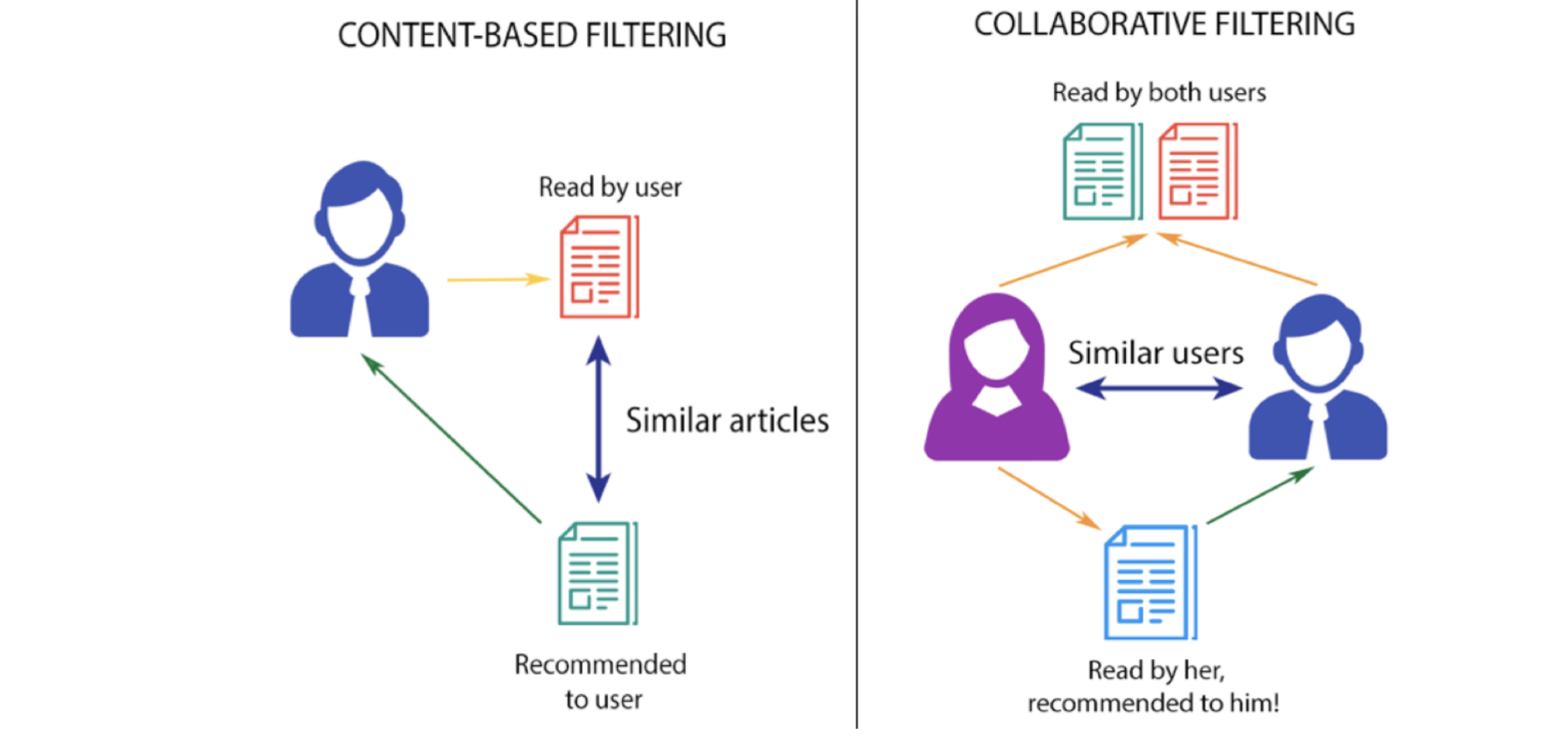
# Methodology:

There are several techniques for building a movie recommendation system, including collaborative filtering, content-based filtering, and hybrid filtering. In this assignment, we will use the content-based filtering technique.

Types of Recommendation Systems:

1. **Collaborative Filtering**: Collaborative filtering recommends items based on the past behavior and preferences of similar users. It assumes that users who have similar tastes in the past will have similar tastes in the future. Two common approaches in collaborative filtering are user-based and item-based filtering. User-based filtering finds users with similar preferences to the target user and recommends items they have liked. Item-based filtering finds similar items based on user preferences and recommends them to the target user. Collaborative filtering does not require explicit item features and can capture complex user-item interactions. However, it suffers from the cold-start problem (when there is insufficient user or item data) and the scalability issue with large datasets.
2. **Content-Based Filtering**: Content-based filtering recommends items based on the characteristics and features of the items themselves. It focuses on the attributes of items and user preferences for those attributes. In the case of movie recommendation, content-based filtering analyzes features such as movie genre, cast, director, plot, and keywords. It recommends movies with similar attributes to the ones the user has liked in the past. Content-based filtering does not rely on user data, making it suitable for new users. However, it may lead to a limited scope of recommendations and struggle to capture diverse user interests. Content-based filtering recommends items to users based on their previous choices. In the case

of a movie recommendation system, it recommends movies based on the similarities between the movies that a user has watched and the movies in the dataset.



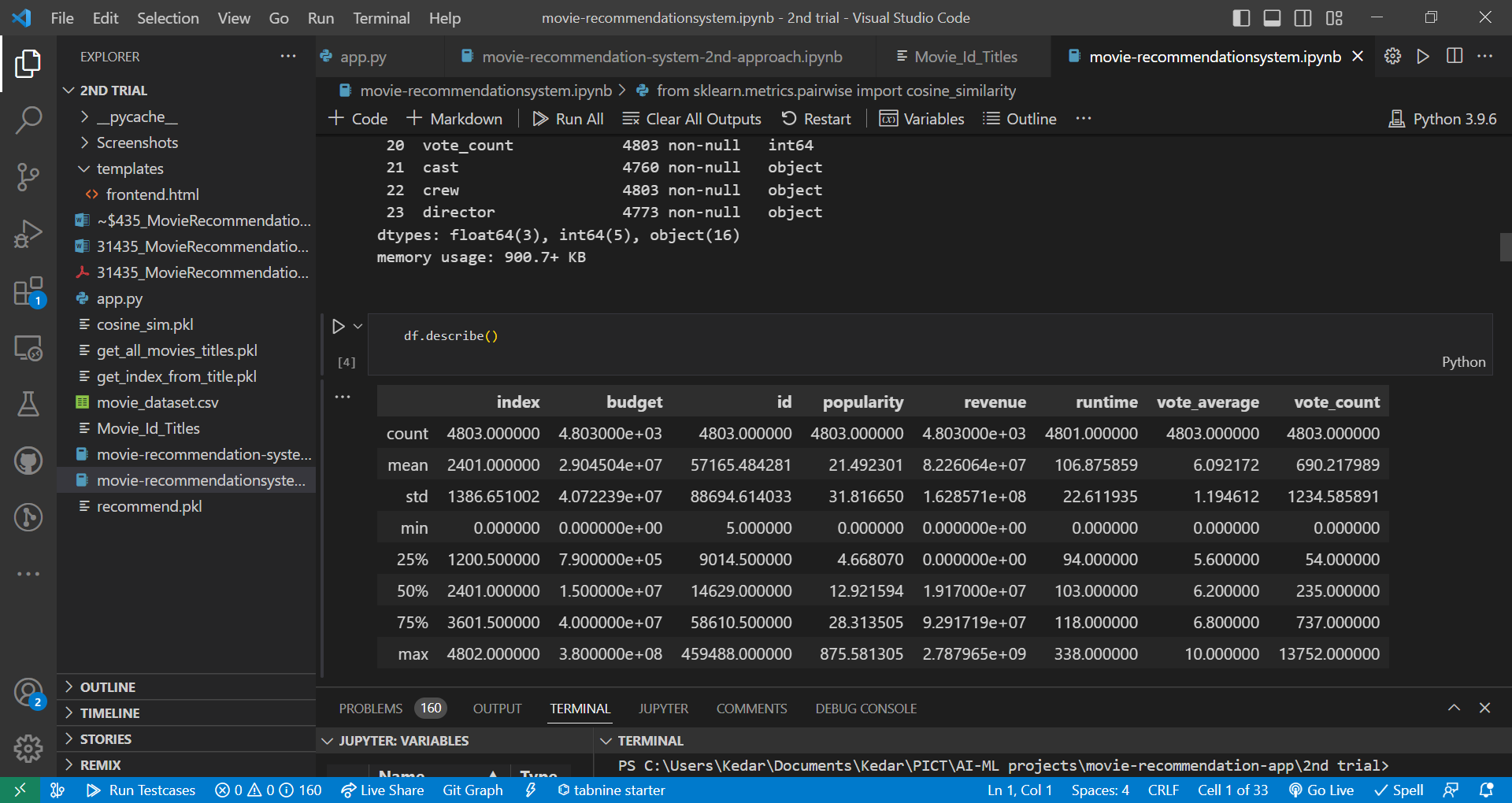
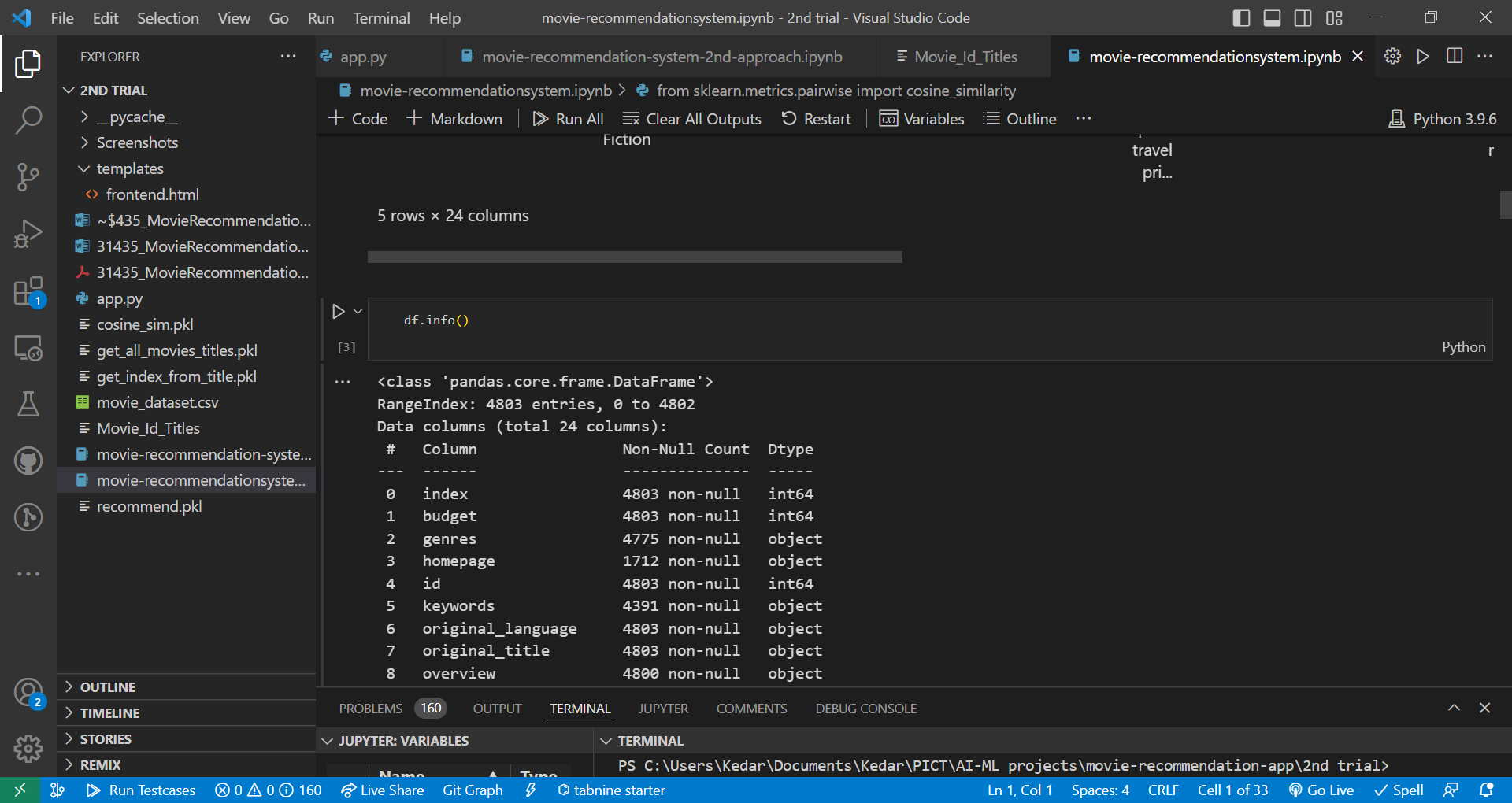
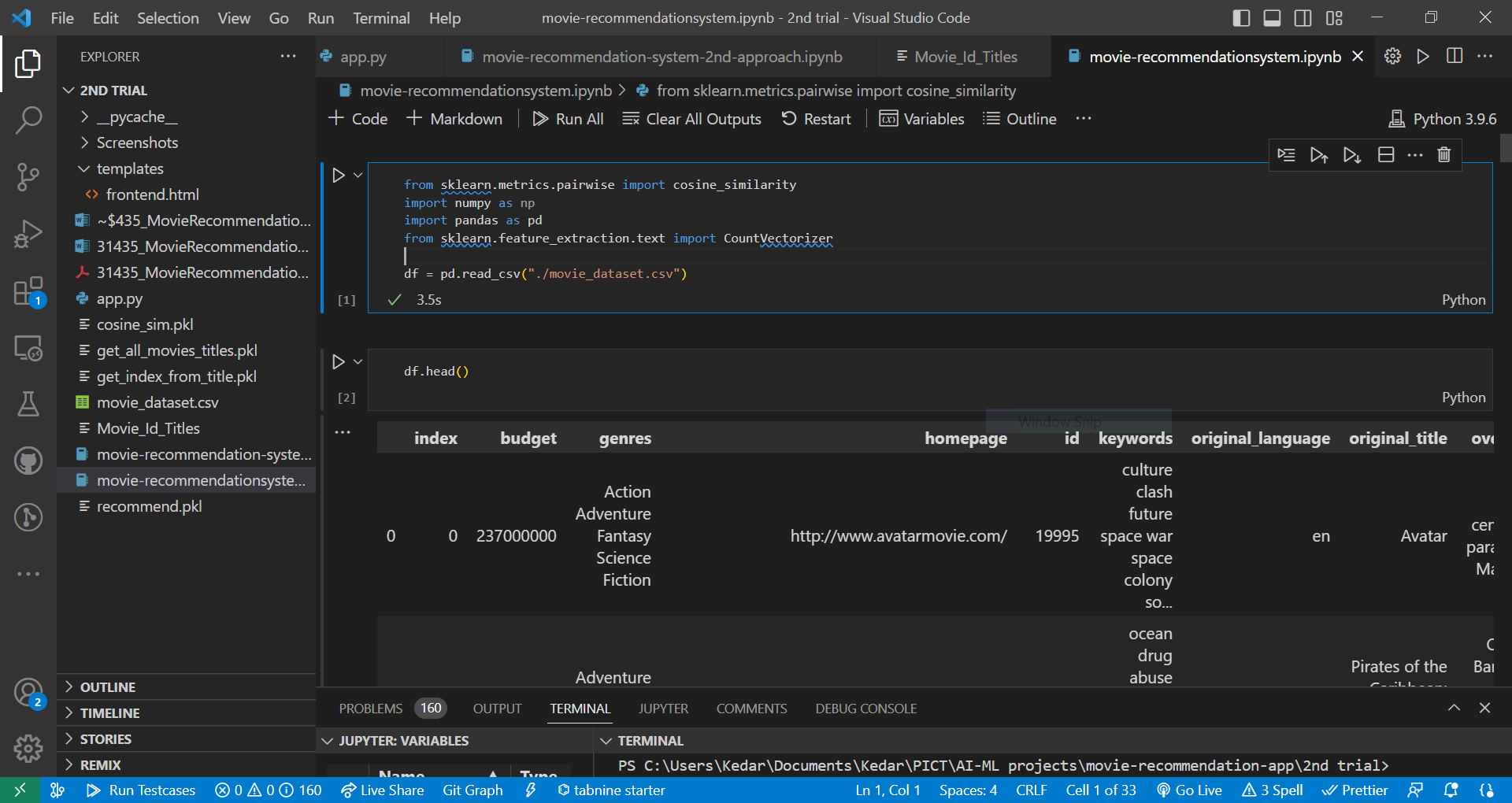
1. **Hybrid Filtering**: Hybrid filtering combines multiple recommendation techniques to leverage their strengths and overcome their limitations. It combines collaborative filtering and content-based filtering or other methods to provide more accurate and diverse recommendations. Hybrid models can be designed in various ways, such as using a content-based model to pre-filter items before applying collaborative filtering or using a weighted combination of multiple recommendation algorithms. Hybrid models aim to address the weaknesses of individual techniques and improve recommendation accuracy and coverage.
2. **Evaluation of Recommendation Systems**: Evaluation of recommendation systems is essential to measure their effectiveness and compare different approaches. Here are some common evaluation metrics: Precision and Recall: Measure the proportion of relevant recommendations and the proportion of relevant items retrieved, respectively. Mean Average Precision (MAP): Calculates the average precision for different cutoff points and averages them. Normalized Discounted Cumulative Gain (NDCG): Measures the ranking quality of recommended items. Root Mean Squared Error (RMSE): Evaluates the predicted ratings against the actual ratings in collaborative filtering.

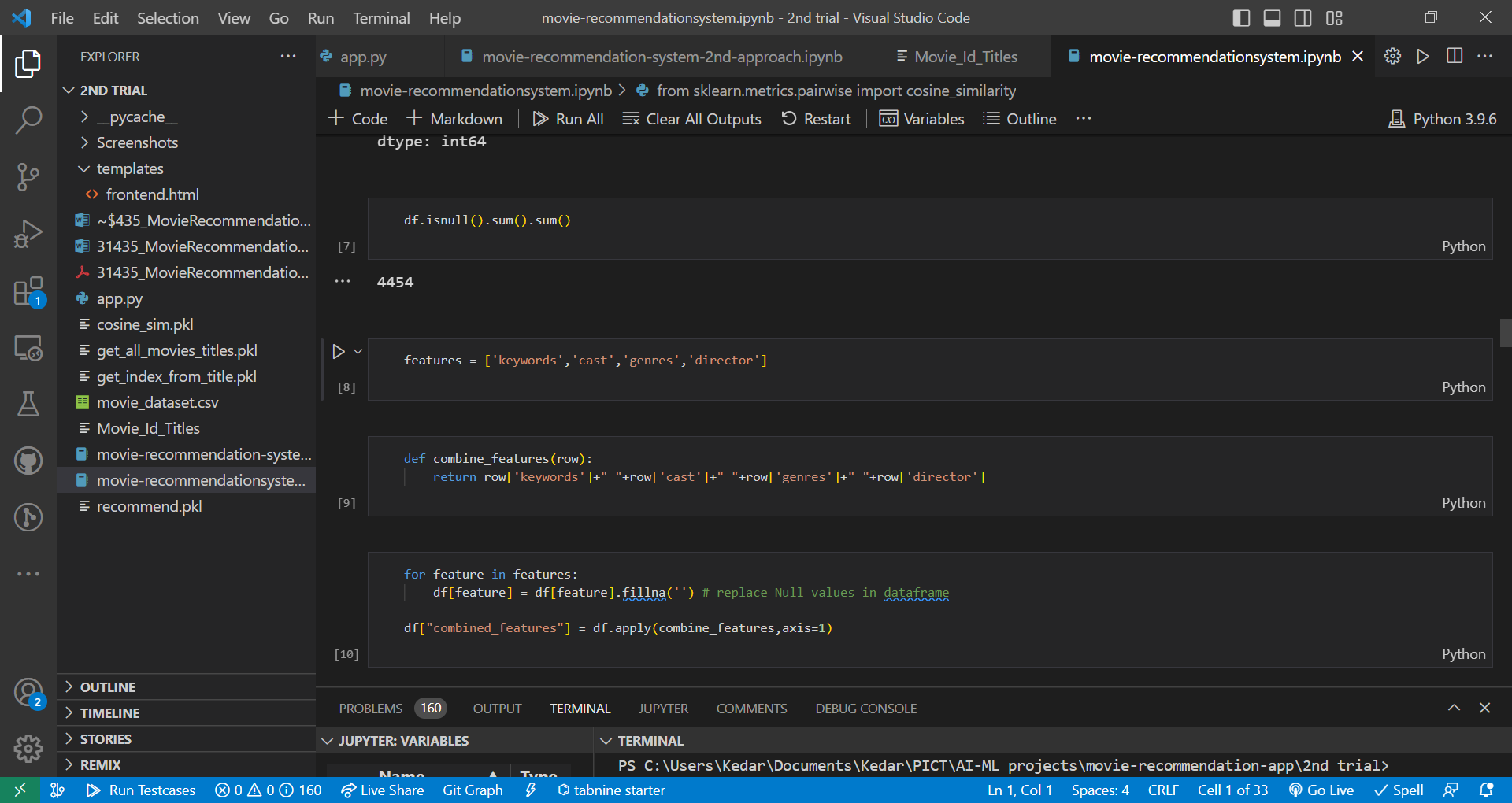
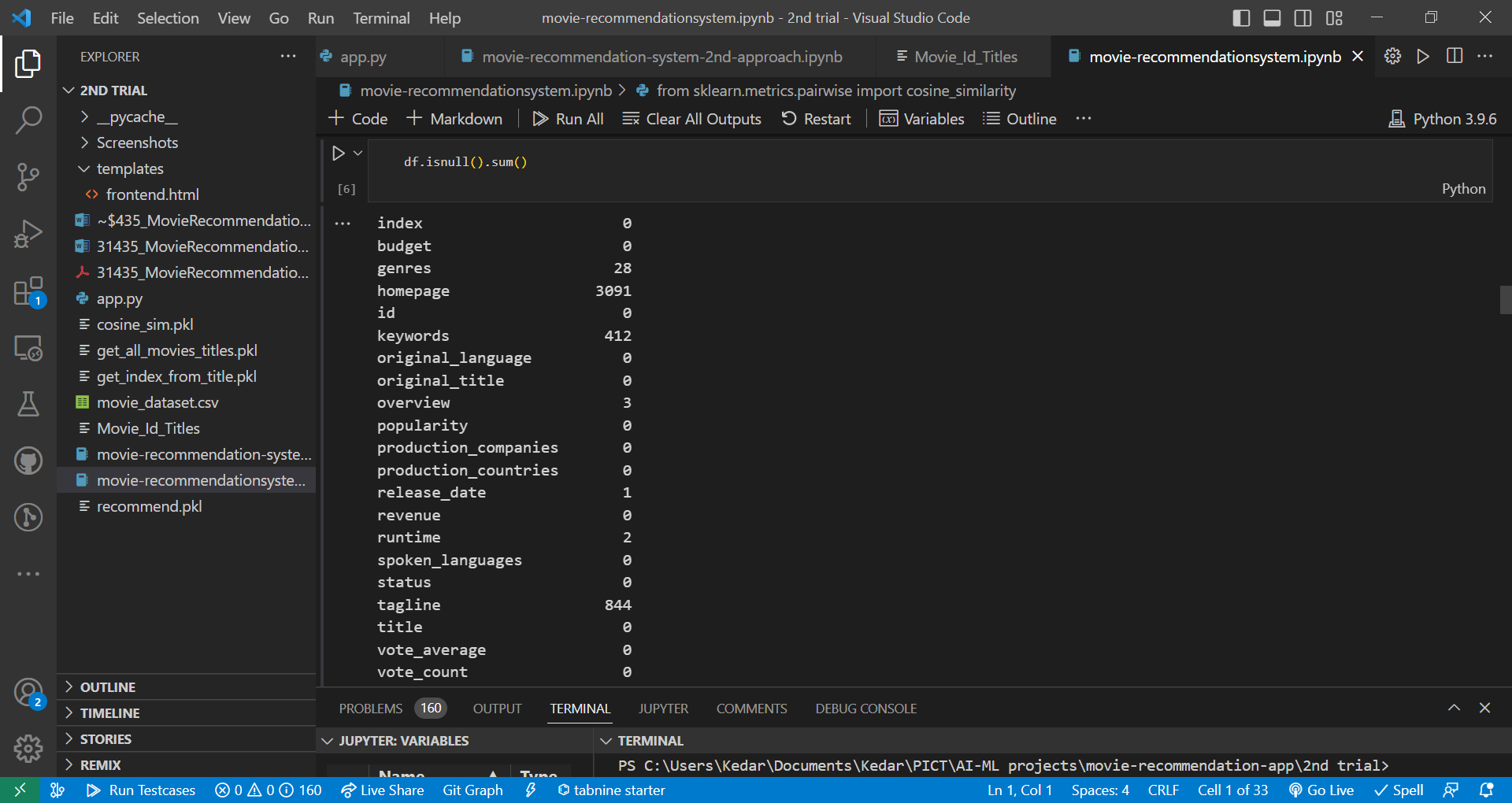
The process of building a content-based movie recommendation system involves the following steps:

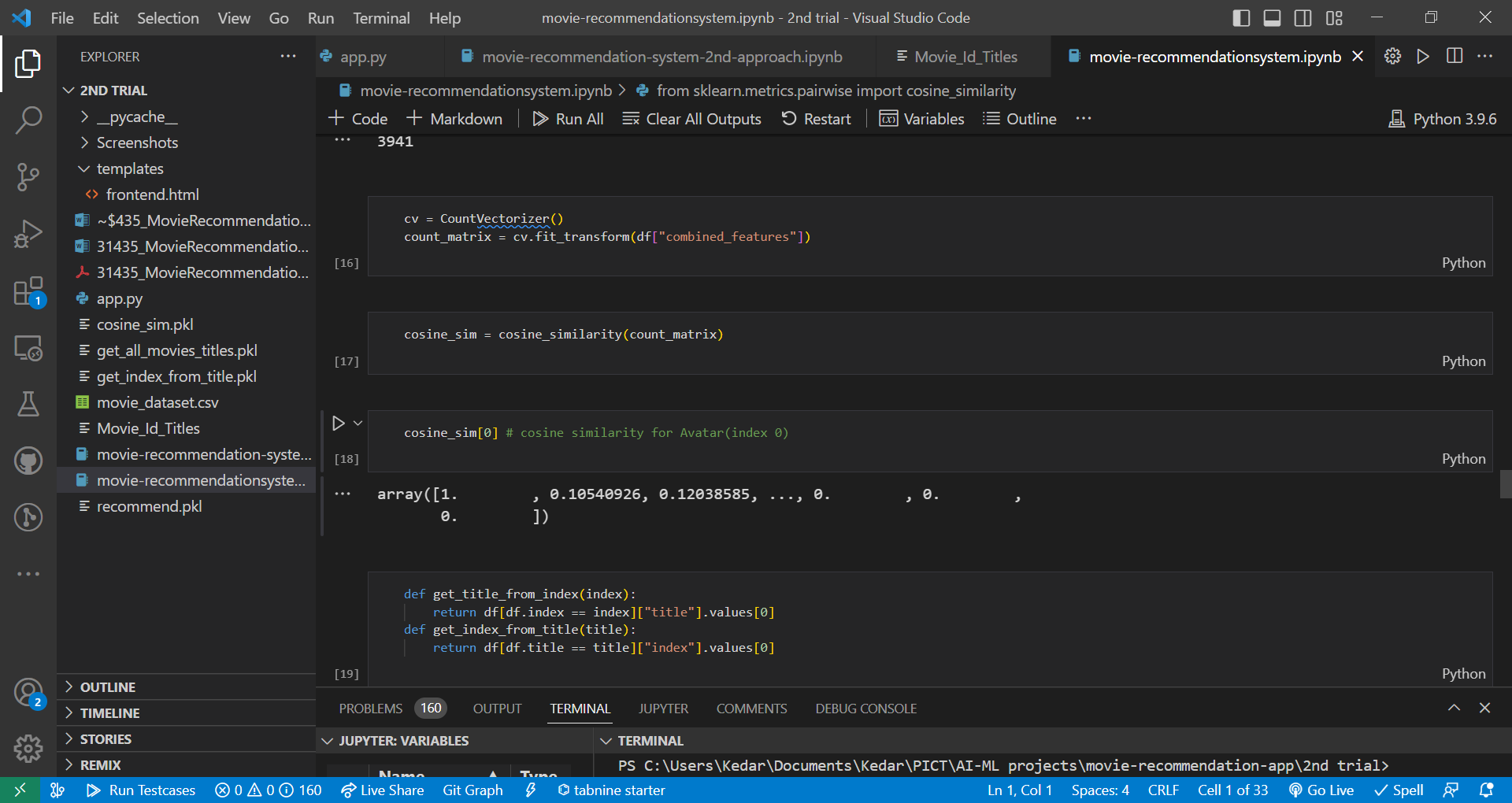
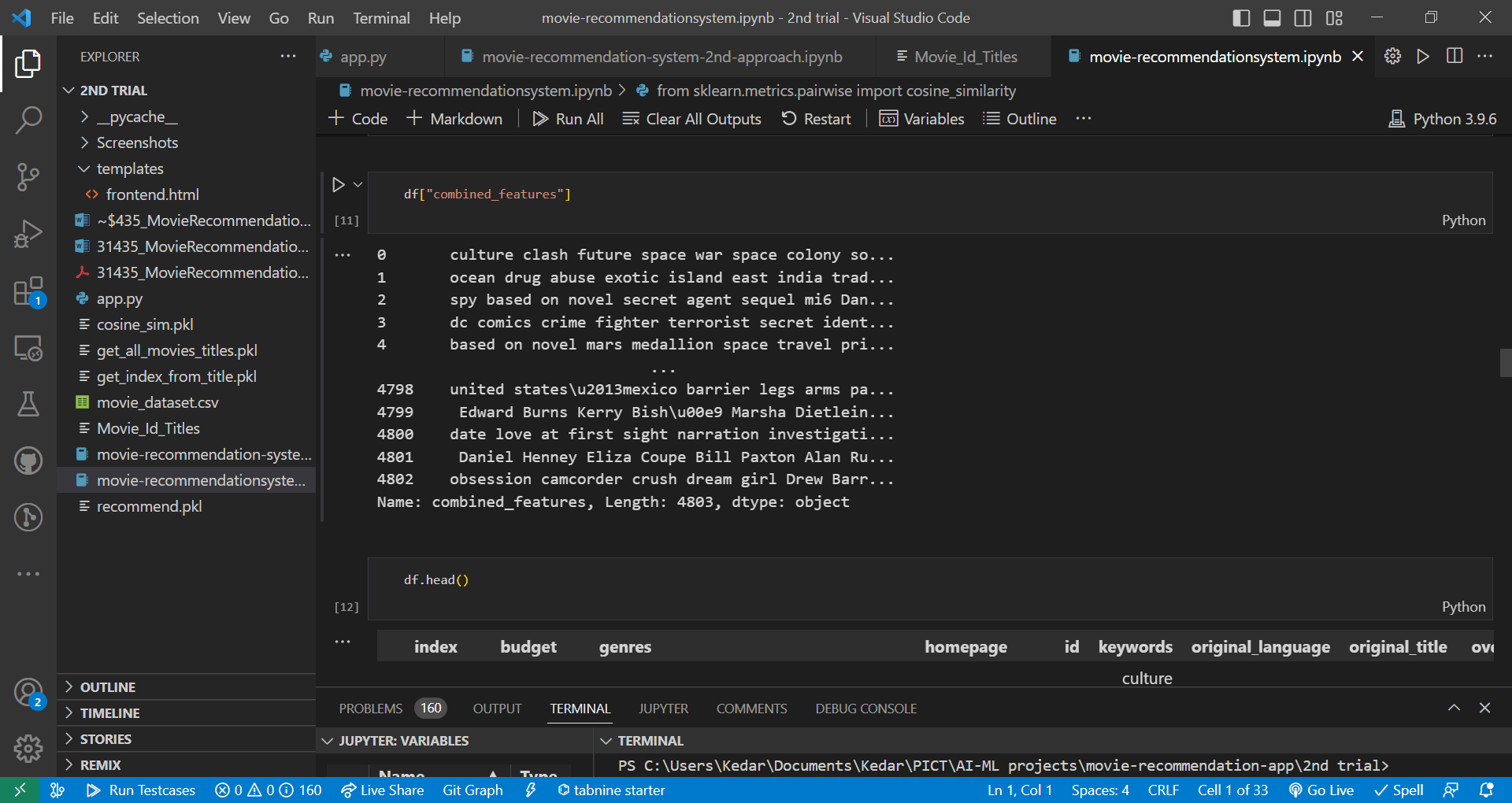
1. **Data preprocessing**: The first step is to preprocess the dataset by removing any irrelevant columns and filling in missing values.
2. **Feature extraction**: The next step is to extract features from the dataset. In the case of a movie recommendation system, the features could be the plot, the cast, the director, the genre, etc.
3. **Feature transformation**: Once the features have been extracted, they need to be transformed into a numerical format that can be used by the machine learning algorithm.
4. **Similarity computation**: The next step is to compute the similarity between the movies based on their features. This can be done using various similarity measures such as cosine similarity, Jaccard similarity, etc.
5. **Recommendation generation**: Finally, based on the similarities between the movies, the system can generate recommendations for the user.

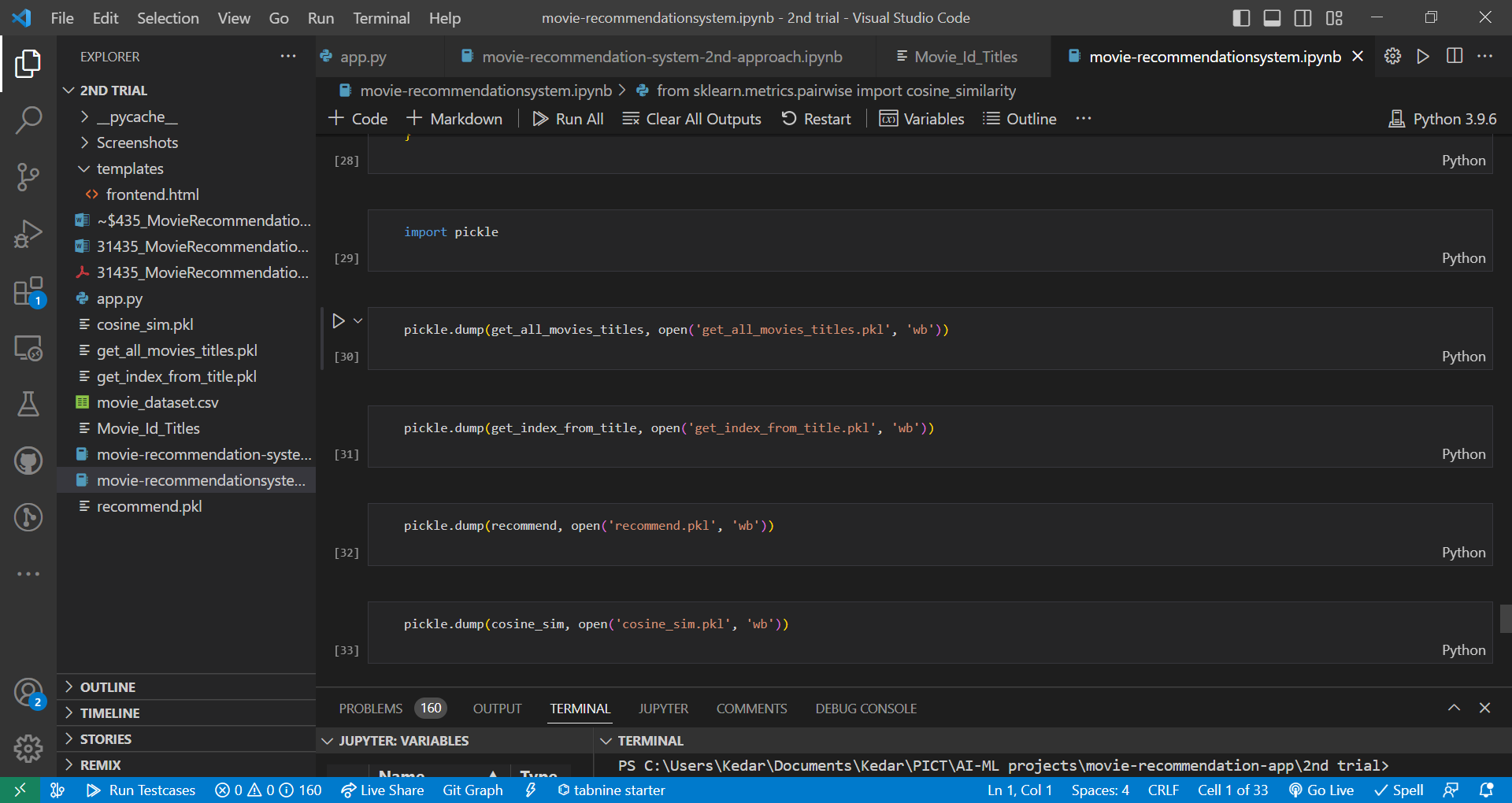
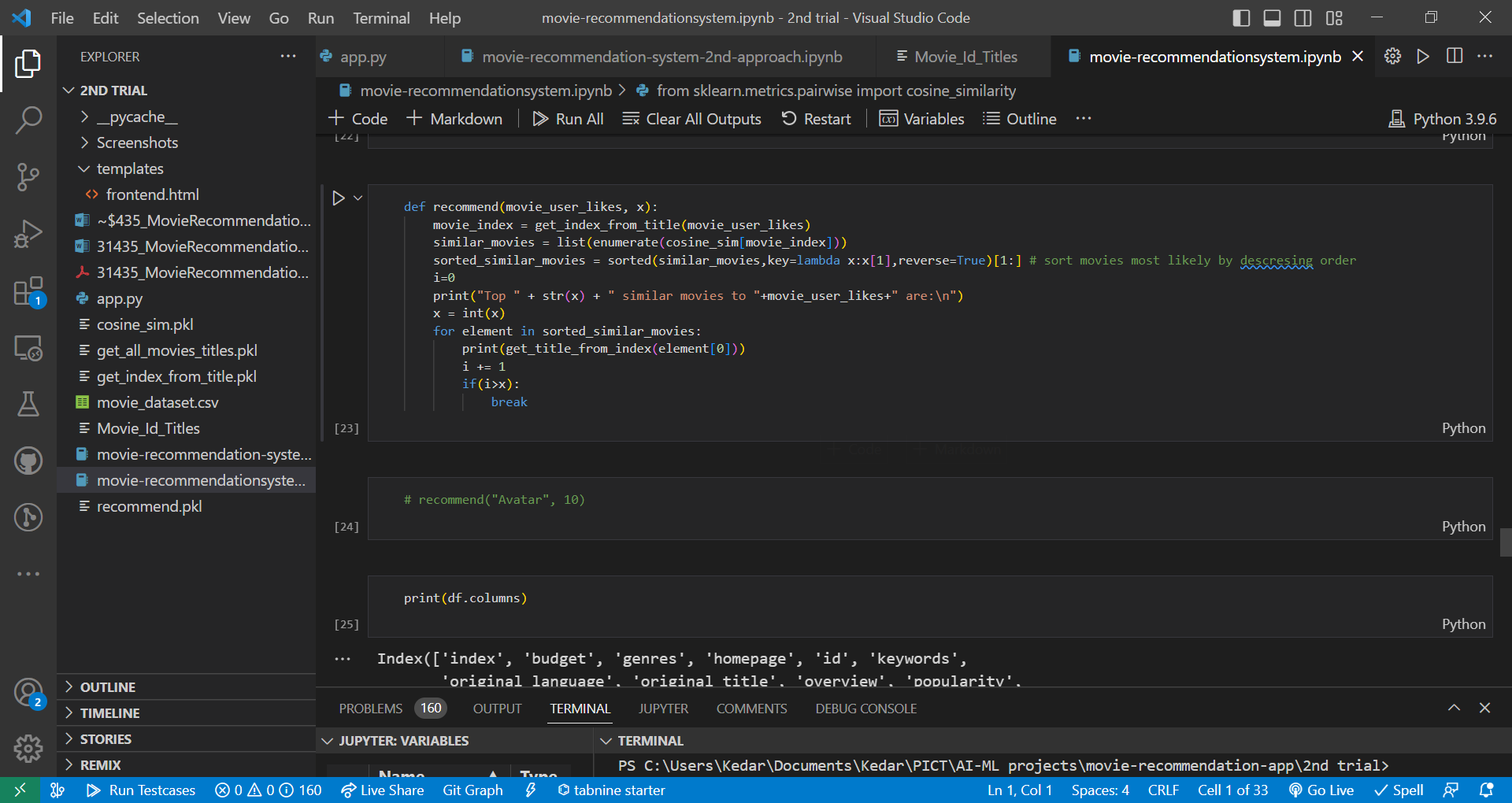
**Scikit-learn library**: Scikit-learn is a Python library that provides various machine learning algorithms and tools for data preprocessing, feature extraction, feature transformation, and model evaluation. It is widely used in the industry and academia for building machine learning models.

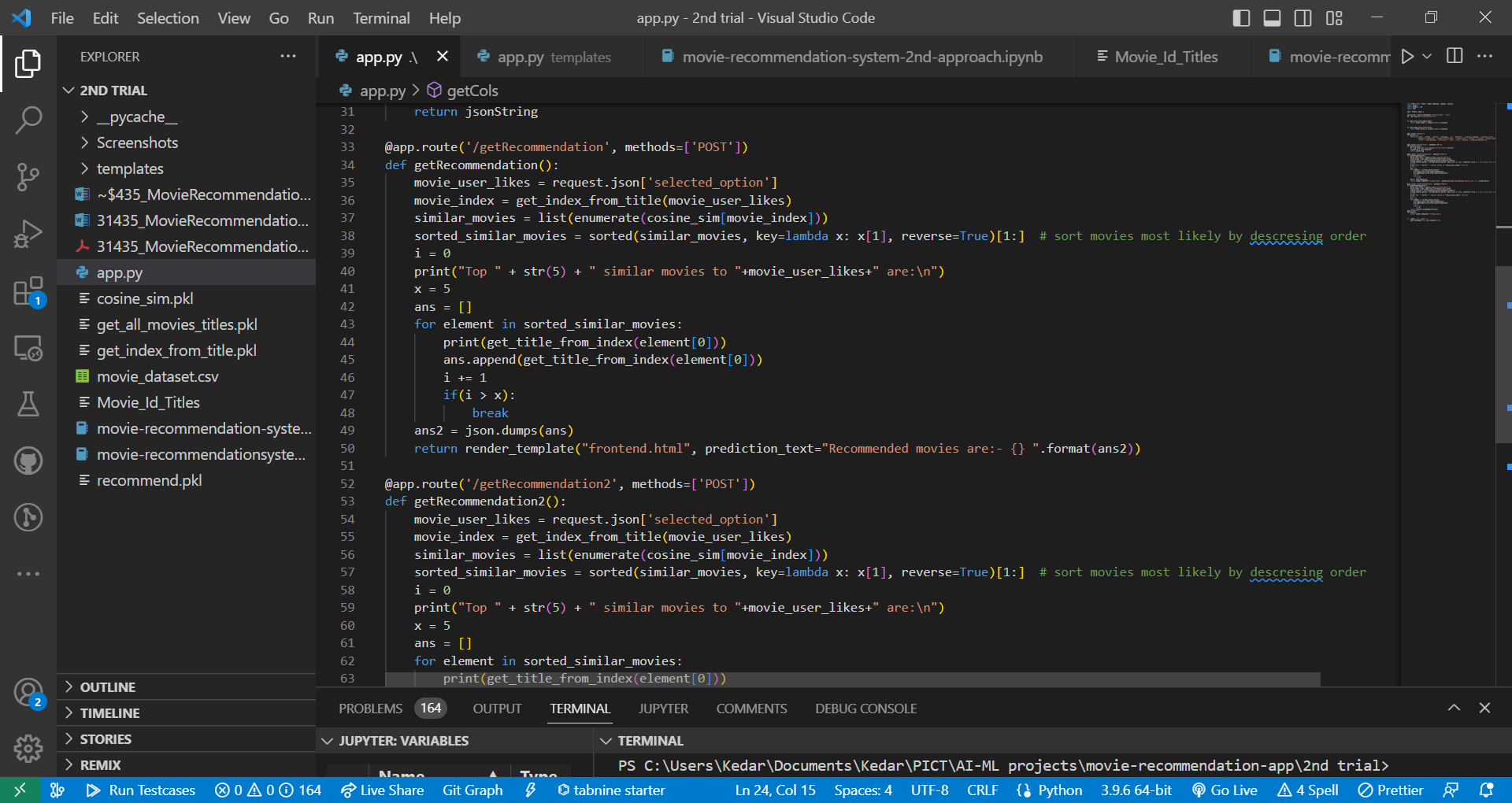
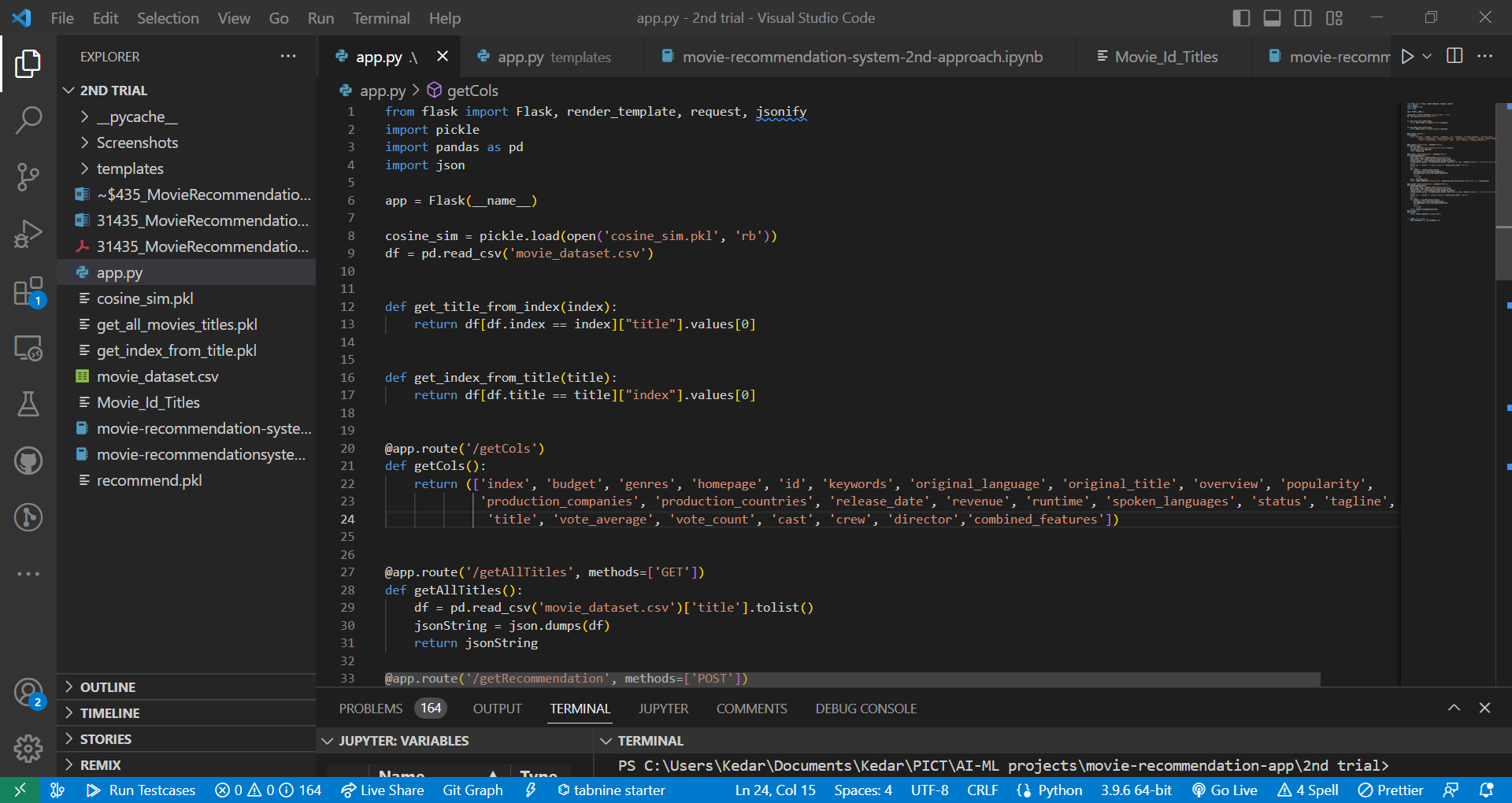
**Our Work:**

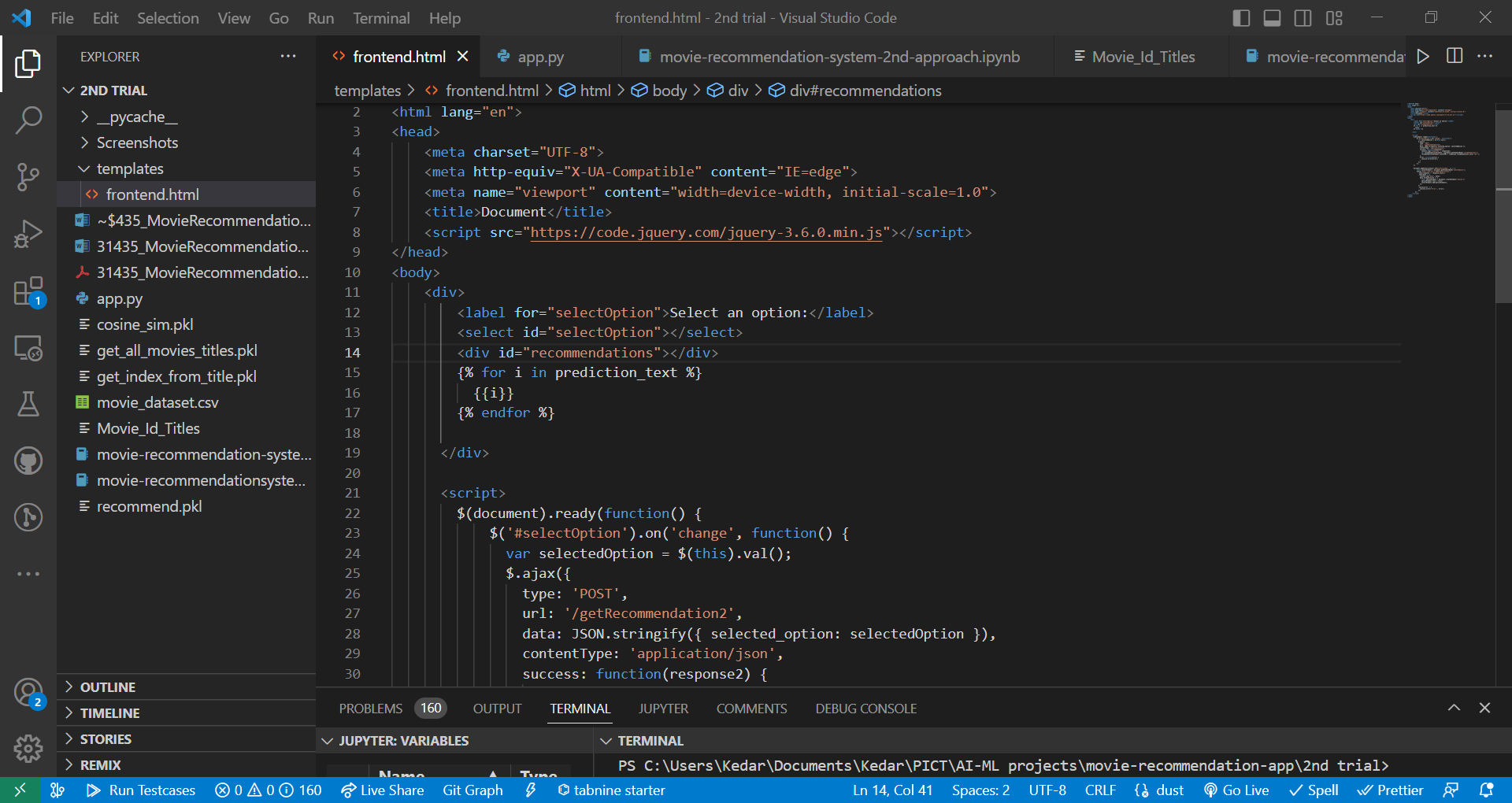
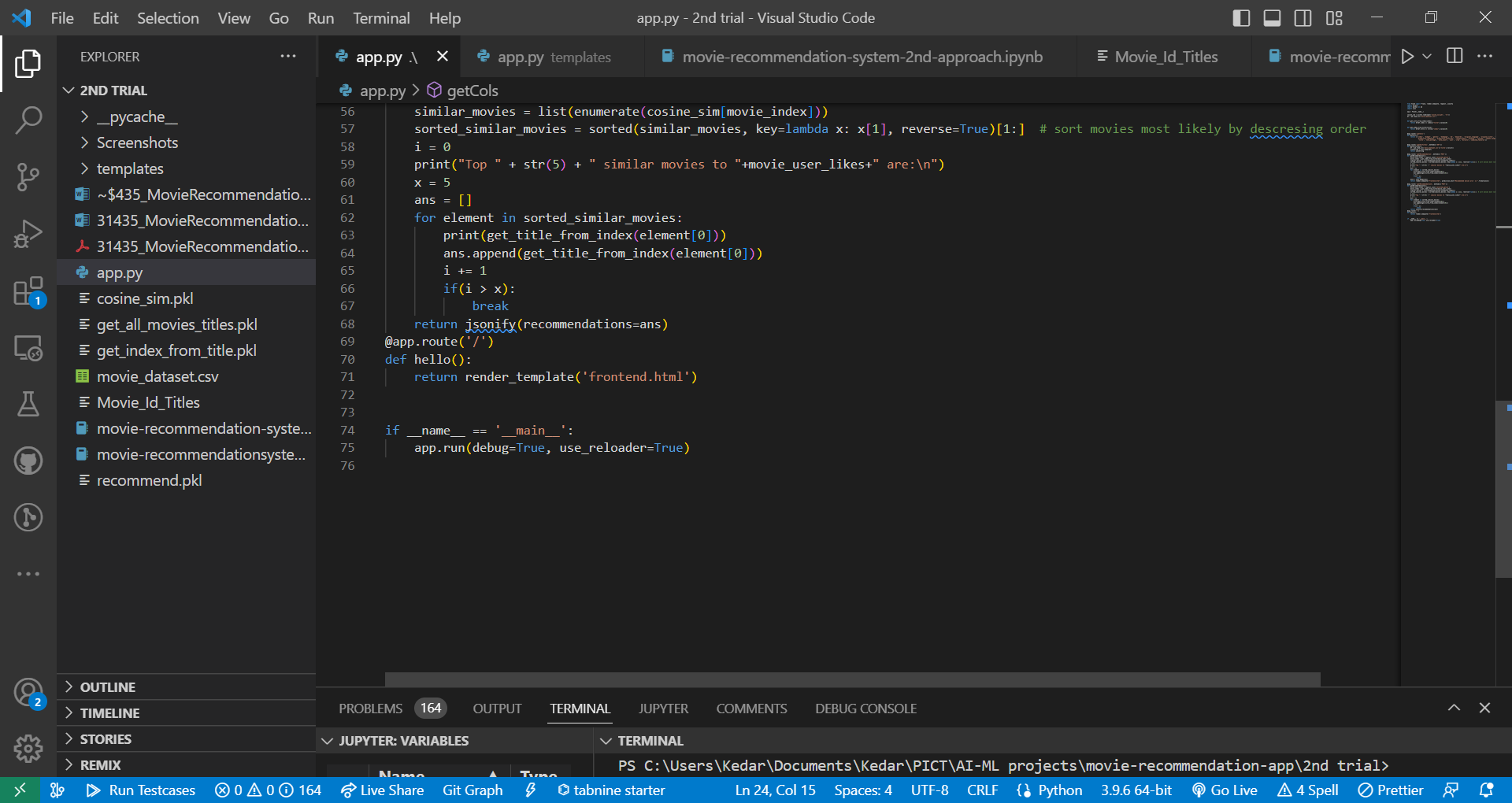


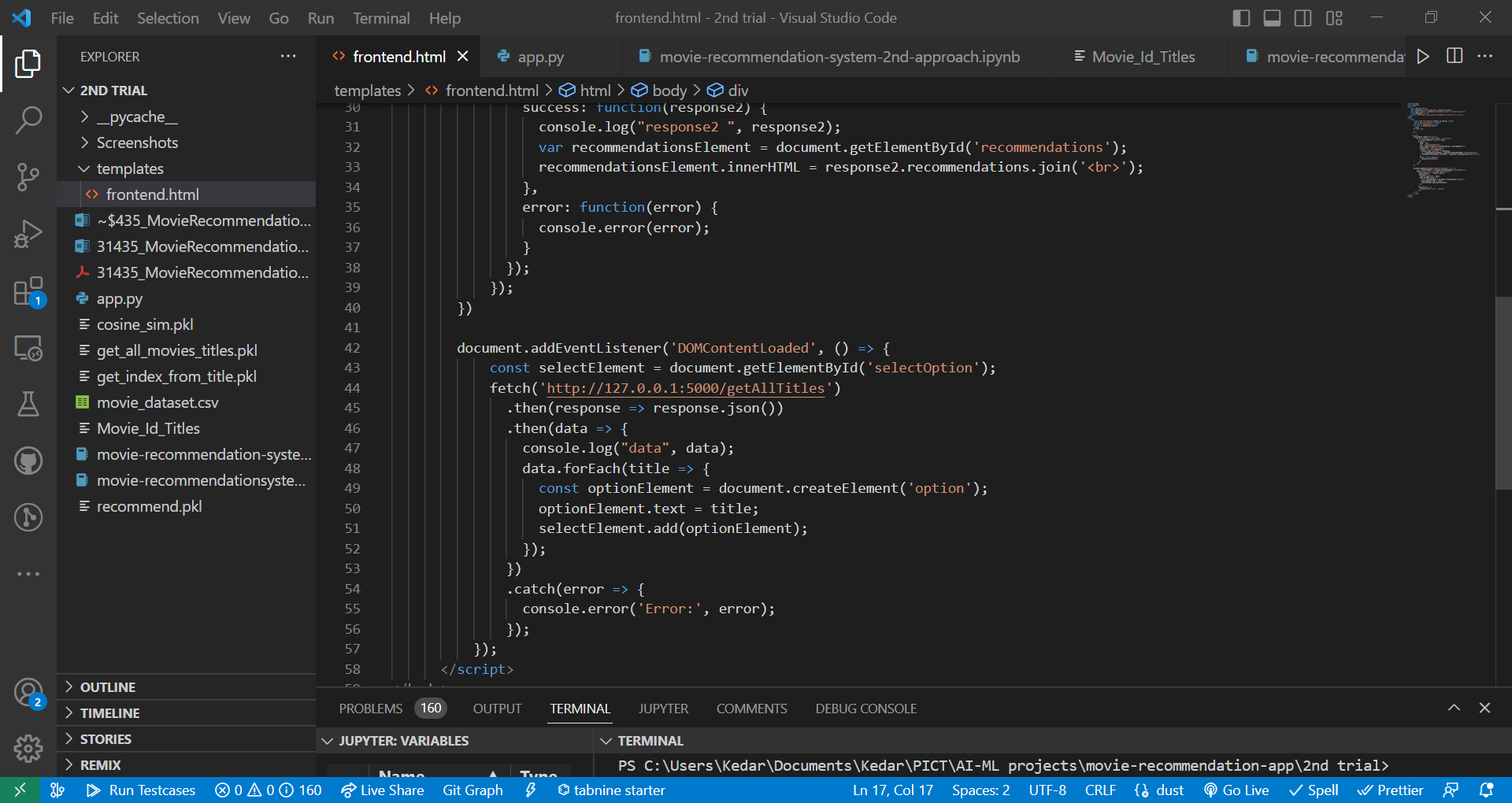




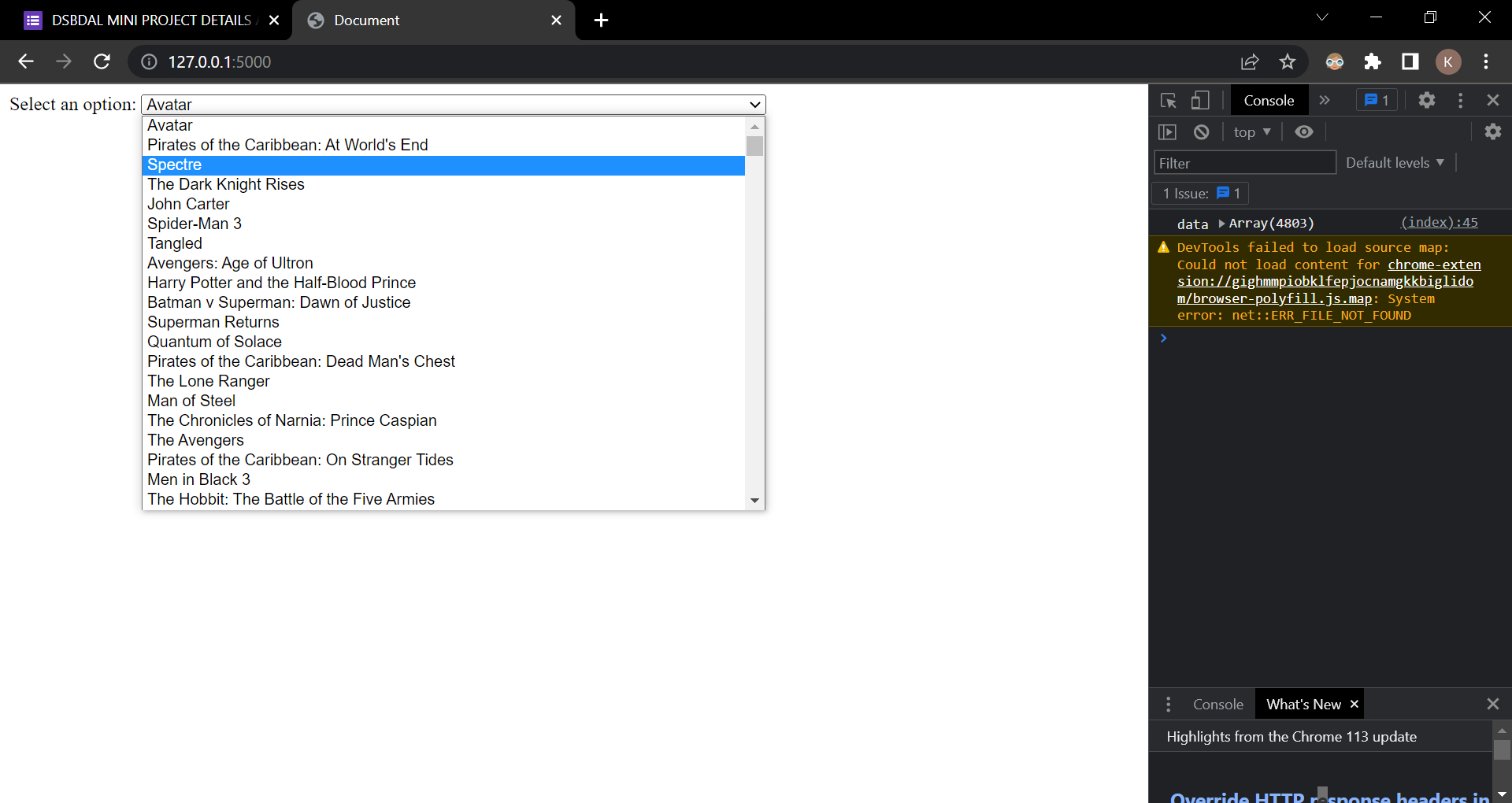


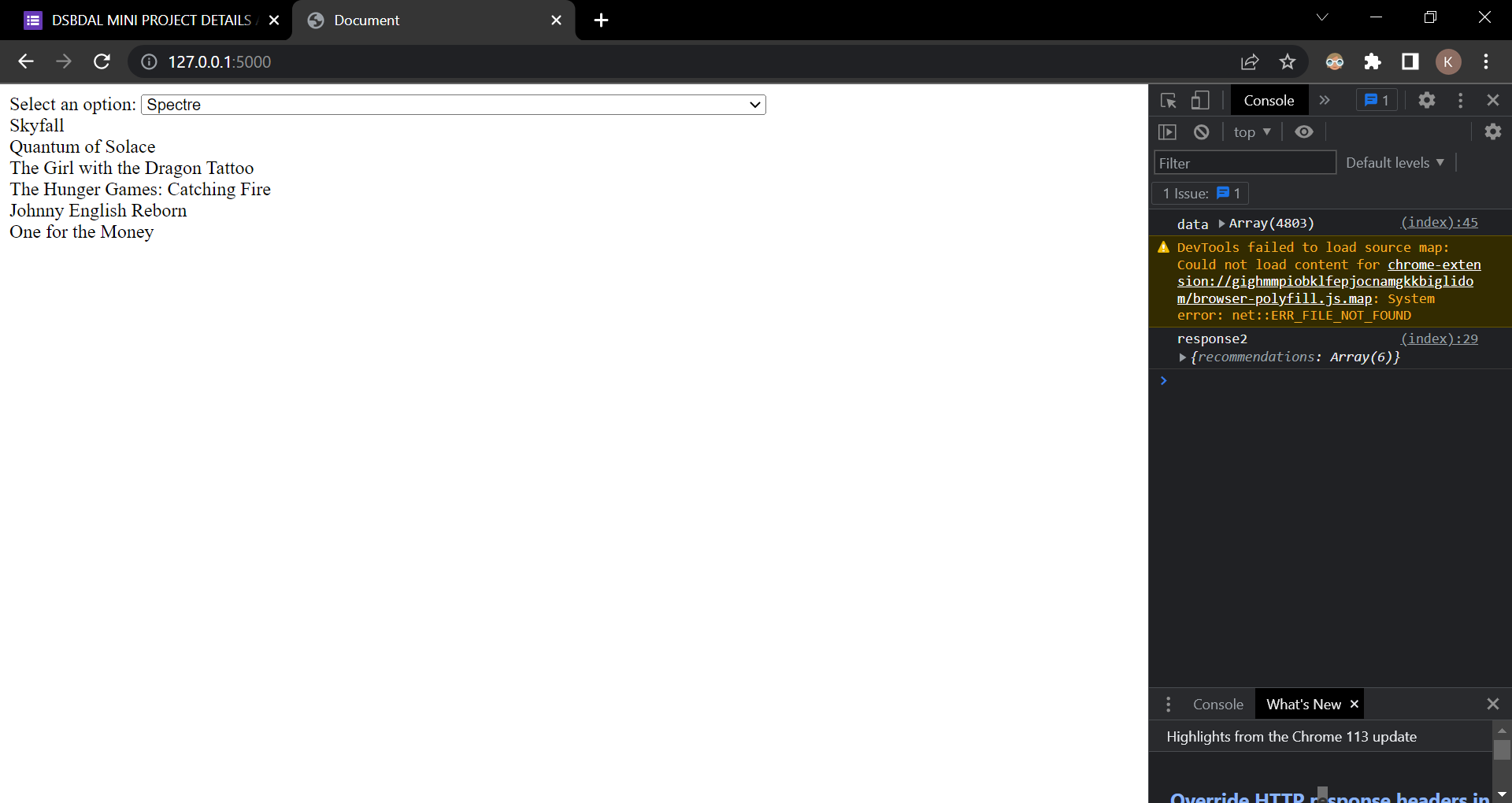






**Output Screenshots:**





# Future Work:

1. Incorporate temporal dynamics by considering the evolution of user preferences over time and adapting the recommendation model accordingly.
2. Explore the use of reinforcement learning techniques to optimize the movie recommendation system and further personalize recommendations.
3. Incorporate social network information to leverage the influence of social connections in generating more accurate and socially-aware recommendations.
4. Implement real-time recommendation updates to provide users with up-to-date and relevant movie suggestions based on changing trends and user preferences.
5. Extend the recommendation system to a mobile application or web platform to reach a broader user base and gather more diverse user feedback.

# Conclusion:

¬Cryptocurrencies such as BitCoin still have number of obstacles before they could totally

replace current currency systems.

¬No. of users is the biggest challenge to adopt the crypto as universal currency.

¬Prices is also one factor in adoption, as prices of cryptocurrencies varies time to time.

¬To adopt it one only need the internet connection, not dependent on institutions such as banks.

¬In the next 10-15 years Cryptocurrencies will have the potential to replace the Govt. backed

fiat currencies

 In conclusion, the developed movie recommendation model using the scikit-learn library provides personalized movie suggestions based on content-based filtering. The evaluation of the model demonstrates its effectiveness in delivering accurate and diverse recommendations. Future work can focus on incorporating temporal dynamics, reinforcement learning, social network information, real-time updates, and expanding the recommendation system to different platforms for an improved user experience and wider applicability. The movie recommendation system serves as a valuable tool for users to discover relevant movies and enhances their movie-watching journey.