# Subject: Recommender System Laboratory (DJS22DSL6012)

## (A.Y. 2024-2025)

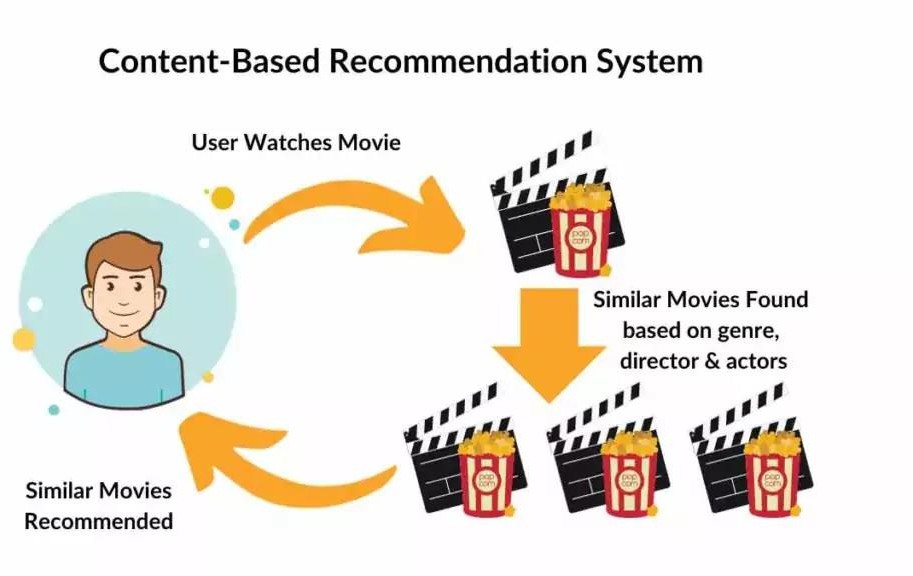
**Experiment 2**

**Name : Soham Walam Roll No : D 102**

**Aim:** Implement Content based Recommender System on an appropriate dataset.

**Theory:**

Content-based recommender systems are a subset of recommender systems that tailor recommendations to users by analyzing items’ intrinsic characteristics and attributes. These systems focus on understanding the content of items and mapping it to users’ preferences. By examining features such as genre, keywords, metadata, and other descriptive elements, content- based recommender systems create profiles for both users and items.



This enables the system to make recommendations matching user preferences with items with similar content traits. Content-based systems operate independently, which makes them particularly useful in scenarios where user history is limited or unavailable. Through this personalized approach, content-based recommender systems play a vital role in enhancing user experiences across various domains, from suggesting movies and articles to guiding users in choosing products or destinations.



The concepts of Term Frequency (TF) and Inverse Document Frequency (IDF) play a crucial role in information retrieval systems and content-based filtering mechanisms, such as content

A screenshot of a white paper with black text

AI-generated content may be incorrect.

A screenshot of a test

AI-generated content may be incorrect.

based recommenders. These concepts help determine the relative importance of a document, article, news item, movie, etc.

All TF means is how often a given word occurs in a given document so within one web page one Wikipedia article, how common is a given word within that document , what is the ratio of that word occurrence rate throughout all the words in that document that’s it. TF just measures how often a word occurs in a document. A word that occurs frequently is probably important to that document’s meaning.

DF is how often a word occurs in an entire set of documents, i.e., all of Wikipedia or every web page. this tells us about common words that just appears everywhere no matter what the topic, like ‘a’, ‘the’, ‘and’, etc. Word with high TF and DF both might not be important measure relevancy of a word to a document.

The cold-start problem essentially consists of how a system handles new users or new items. Both pose a problem in collaborative filtering because it recommends items by grouping users according to inferred similarities of behavior and preference. New users do not have an evidenced similarity with others, however, and new items do not have enough user interaction (for example, ratings) for recommending them. While content-based filtering struggles with new users, it nevertheless adeptly handles incorporating new items. This is because it recommends items based on internal or metadata characteristics rather than past user interaction.

Content-based filtering enables greater degree of transparency by providing interpretable features that explain recommendations. For example, a movie recommendation system may explain why a certain movie is recommended, such as genre or actor overlap with previously watched movies. The user may therefore make a more informed decision on whether to watch the recommended movie.

One chief disadvantage of content-based filtering is feature limitation. Content-based recommendations are derived exclusively from the features used to describe items. A system’s item features may not be able to capture what a user likes however. For instance, returning to the movie recommendation system example, assume a user watches and likes the 1944 movie Gaslight. A CBRS may recommend other movies directed by George Cukor or starring Ingrid Bergman, but those movies may not be similar to Gaslight. If the user rather relishes some specific plot device (for example, deceptive husband) or production element (for example, cinematographer) not represented in the item profile, the system will not present suitable recommendations. Accurate differentiation between a user’s potential likes and dislikes cannot be accomplished with insufficient data.

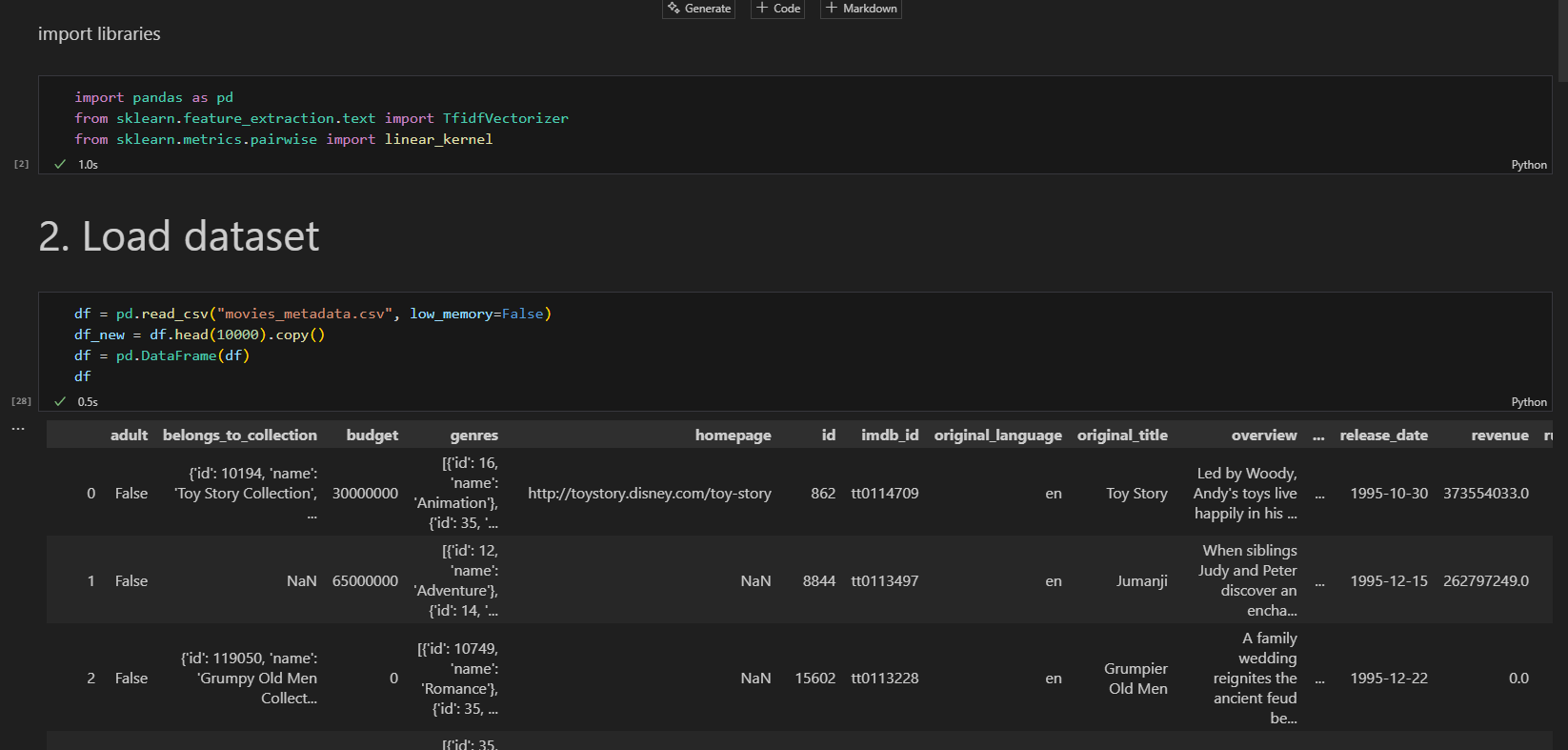
Because content-based filtering only recommends items based on a user’s previously evidenced interests, its recommendations are often similar to items a user liked in the past. In other words, CBRSs lack a methodology for exploring the new and unpredicted. This is overspecialization. In contrast, because collaborative-based methods draw recommendations from a pool of users who have similar likes to one given user, they can often recommend items that a user may have not considered, appears with different features than a user’s previously liked items but that retain some unrepresented element that appeals to a user type.

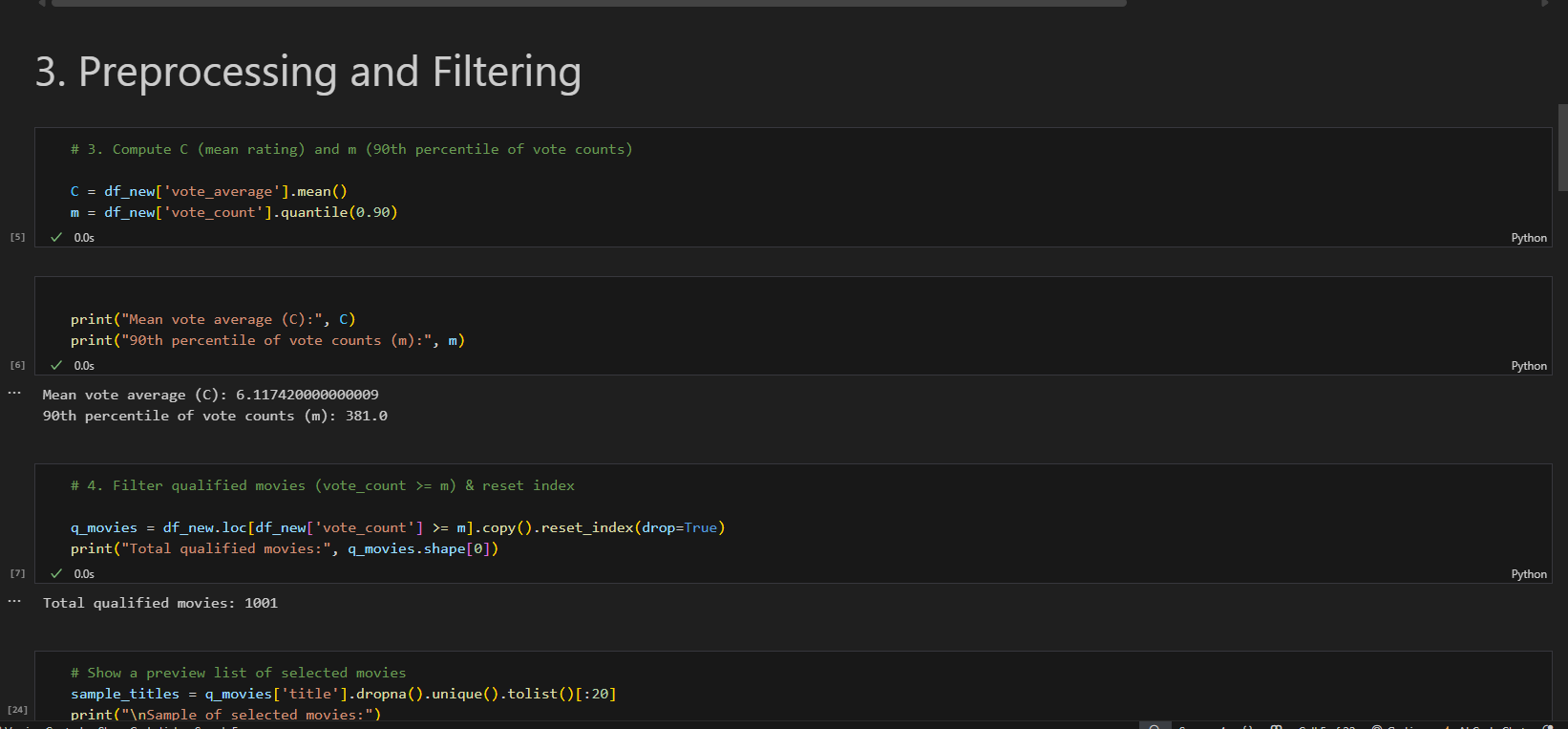
Dataset Link: [https://drive.google.com/drive/folders/1mFpbV1SmZT57bKvWUY-](https://drive.google.com/drive/folders/1mFpbV1SmZT57bKvWUY-FzP3YSuM1pFJd?usp=drive_link) [FzP3YSuM1pFJd?usp=drive\_link](https://drive.google.com/drive/folders/1mFpbV1SmZT57bKvWUY-FzP3YSuM1pFJd?usp=drive_link)

## Lab Assignments to complete:

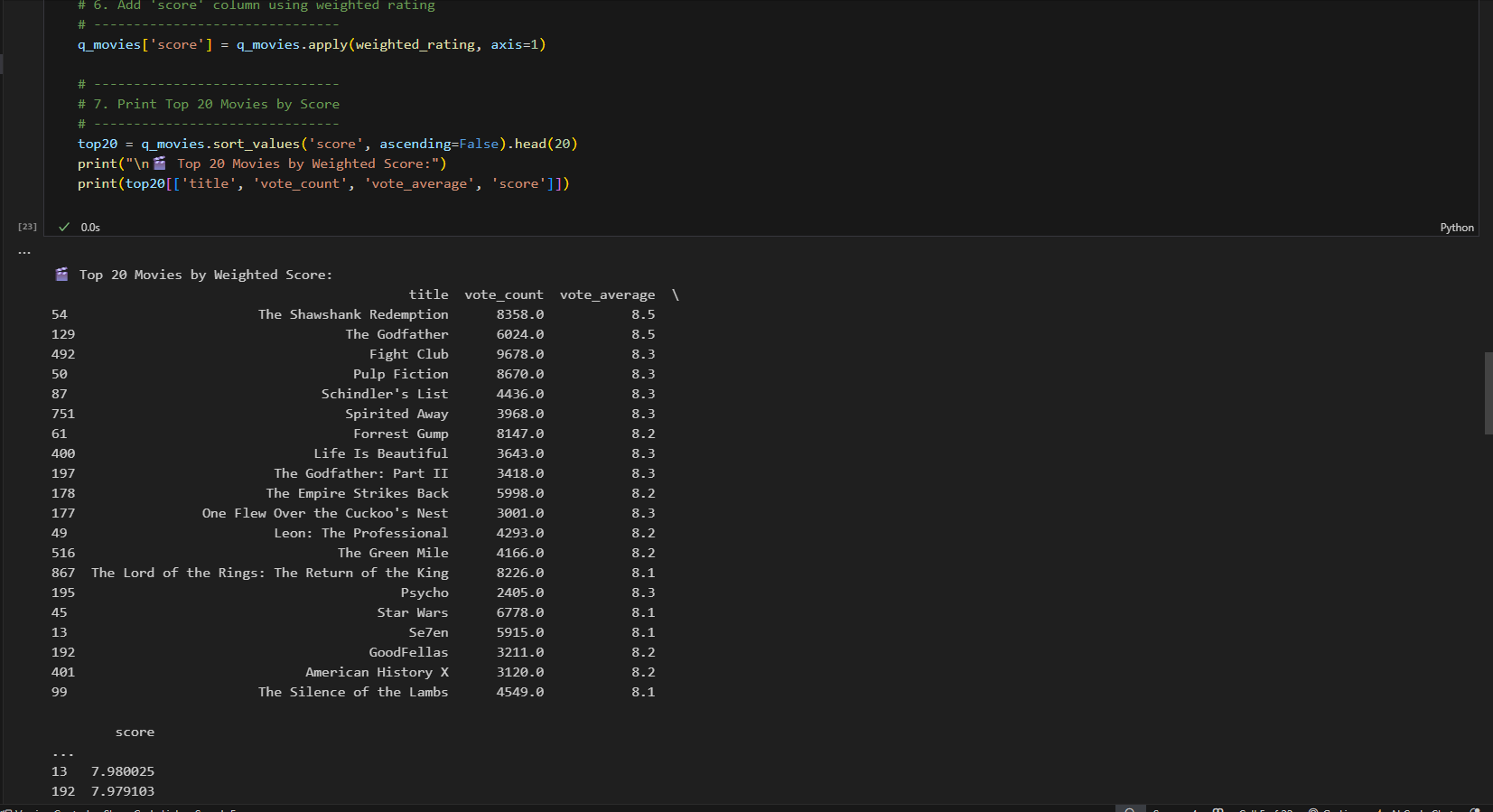
Perform the following tasks on the **movies\_metadata.csv** dataset:

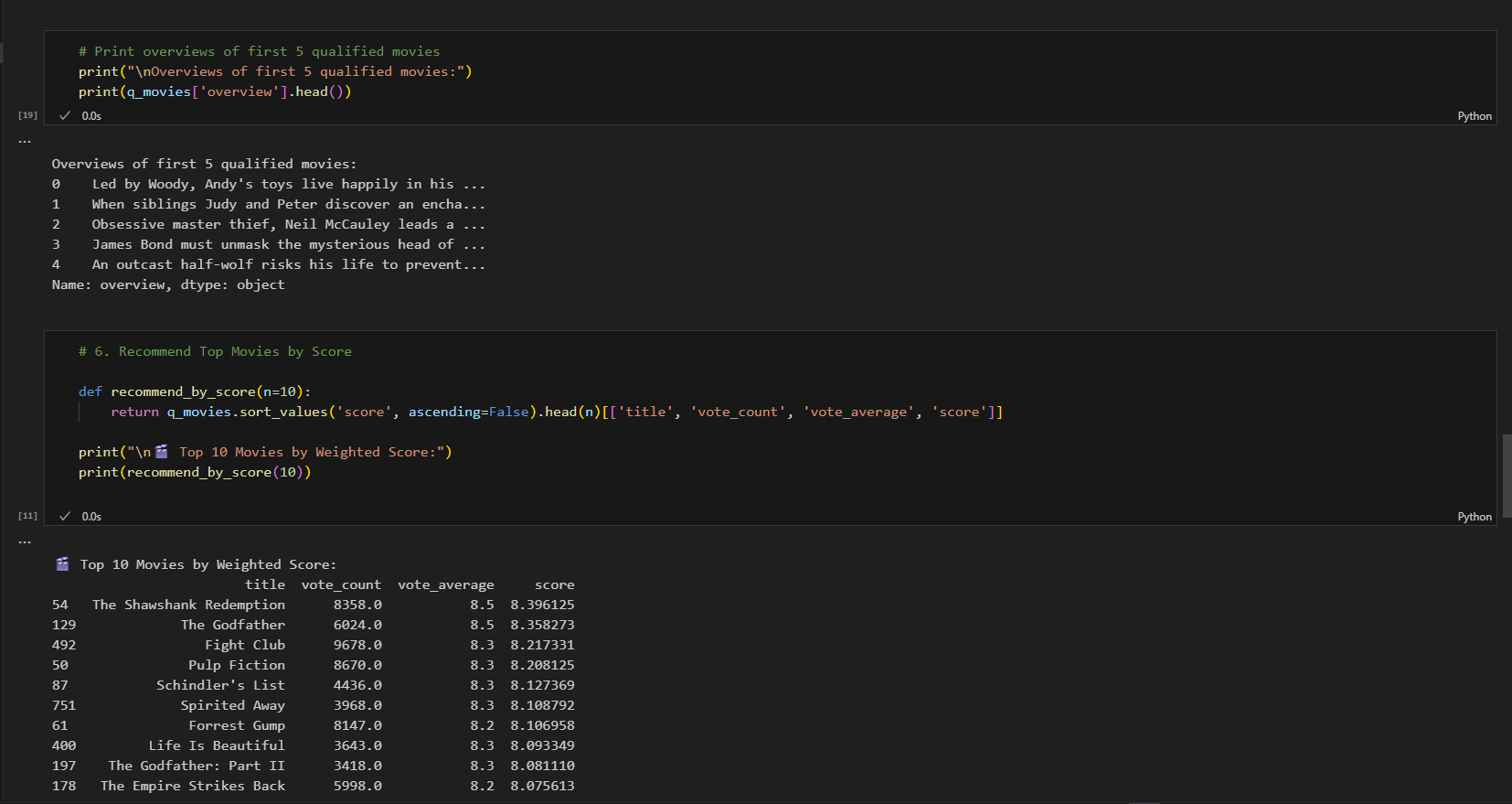
1. Calculate mean of vote average column.
2. Calculate the minimum number of votes required to be in the chart.
3. Compute the weighted rating (score) of each movie.
4. Apply TF-IDF method for fitting and transforming.
5. Compute the cosine similarity matrix.
6. Recommend appropriate movies based on similarity scores.

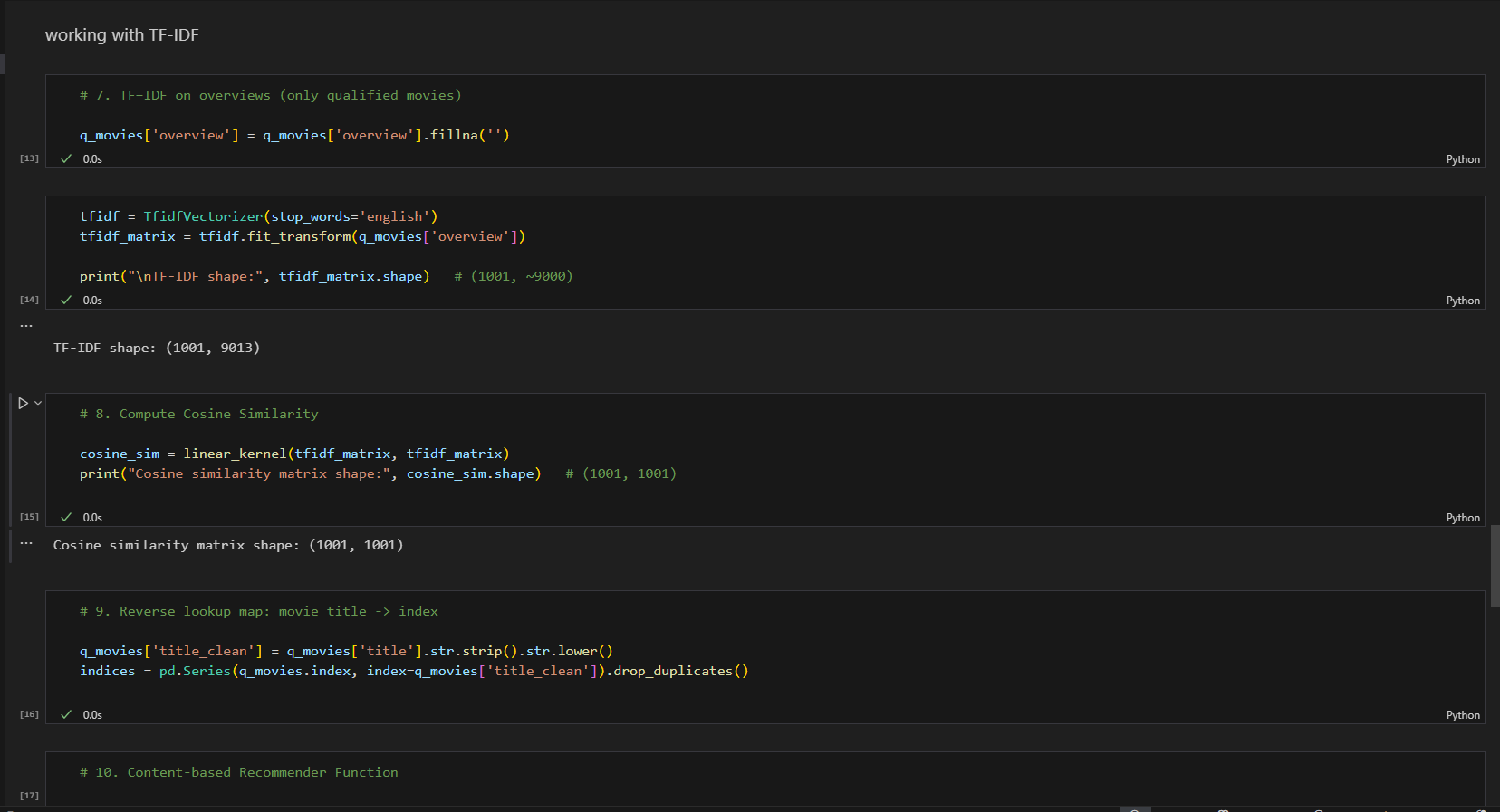


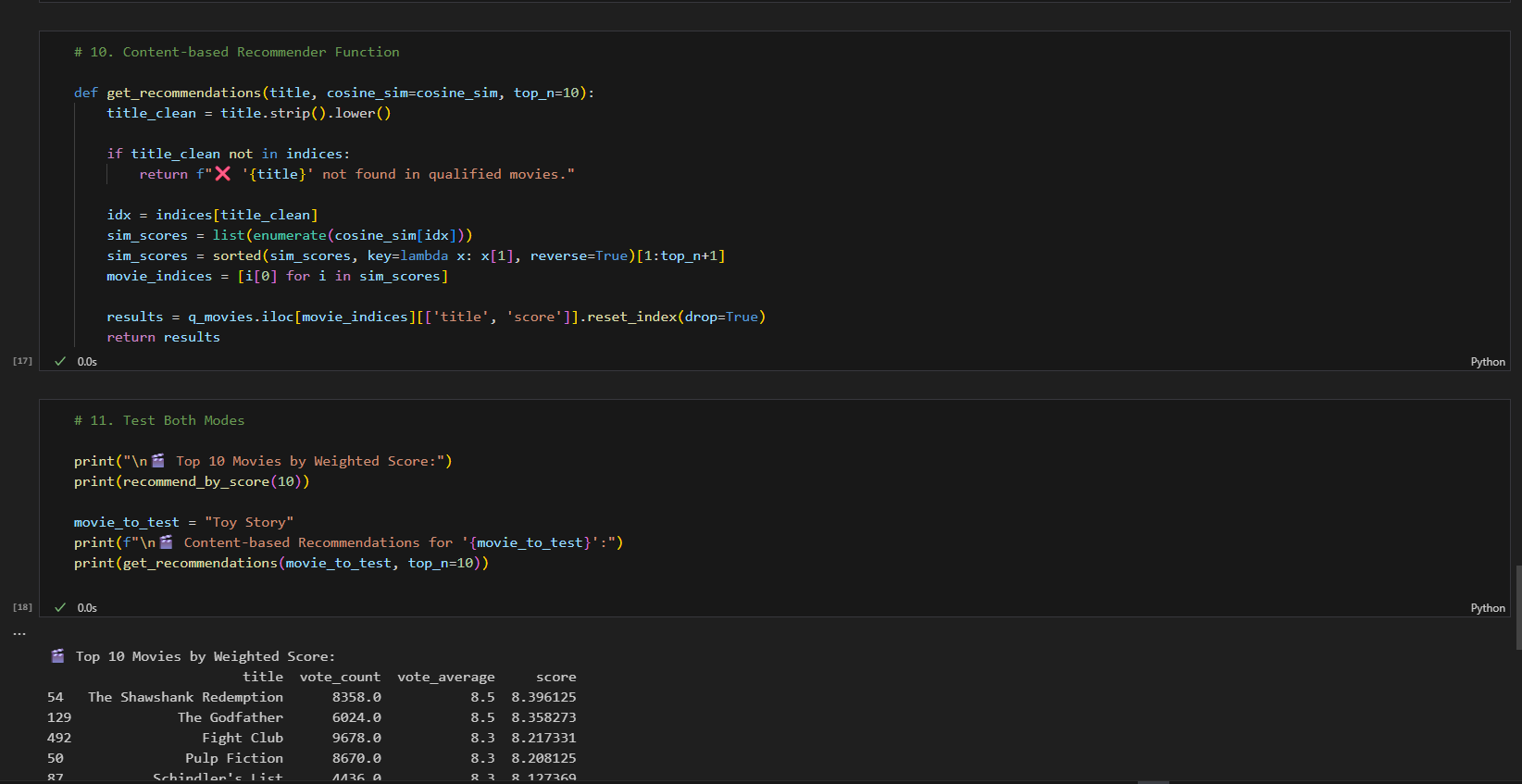
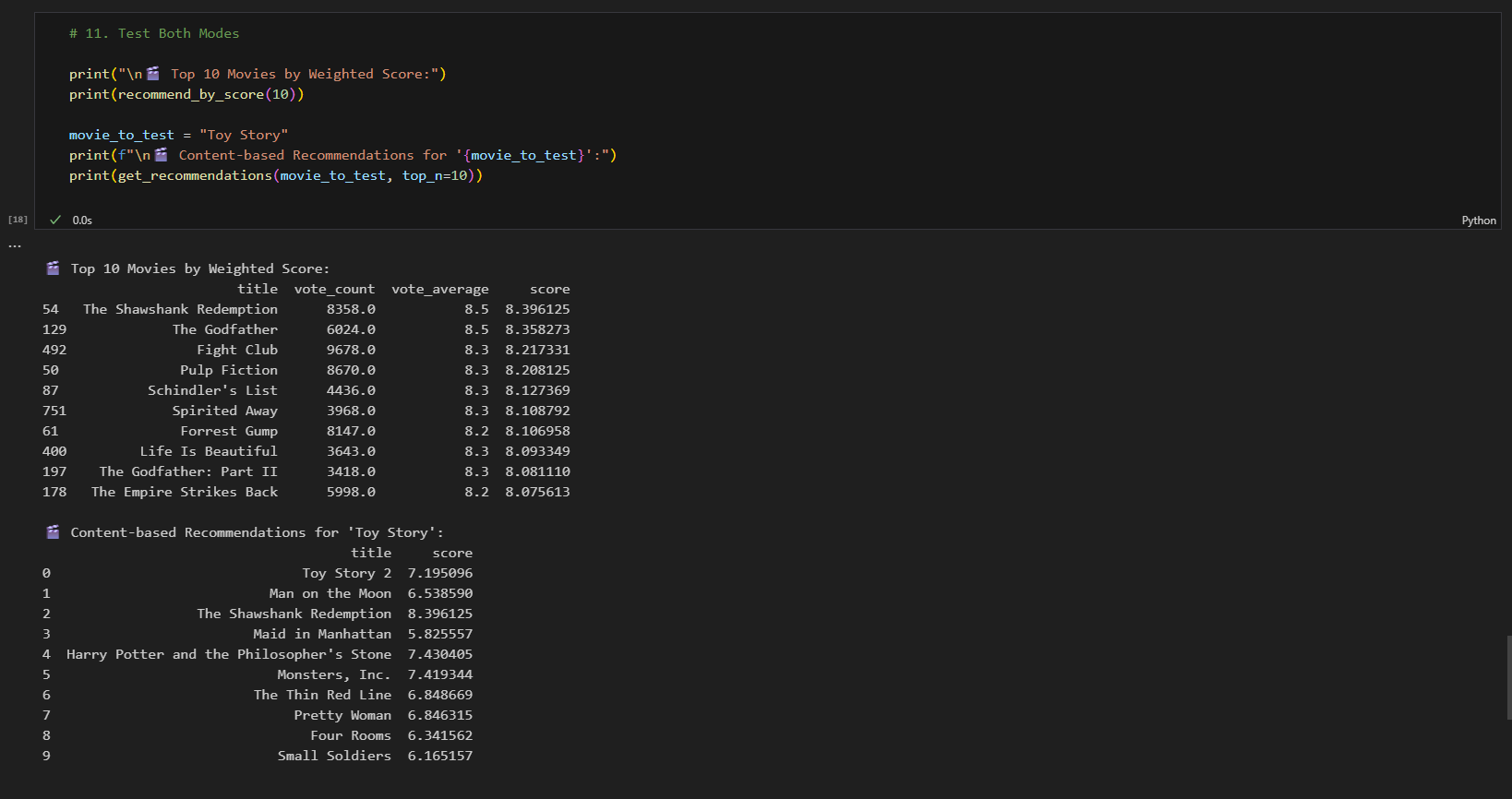










Theory Questions:

1. What is the purpose of using TF-IDF in a content-based recommender system?

TF-IDF (Term Frequency–Inverse Document Frequency) is used to transform movie descriptions into numerical representations that reflect how significant each word is within that description. Simple word counts would give too much weight to common words like *“the”* or *“man”* that appear in nearly all movies, making comparisons meaningless. TF-IDF corrects this by lowering the importance of such frequent words while giving higher importance to rare, descriptive terms such as *“prison,”* *“redemption,”* or *“galaxy.”* This allows the recommender system to highlight the unique themes of each movie and make more accurate content-based matches.

1. Why is cosine similarity used for comparing movie descriptions?

Cosine similarity is preferred because it measures the similarity between two text vectors by considering the angle between them, not their absolute size. This is crucial since movie overviews vary in length — some are just one line while others may be a full paragraph. Using cosine similarity ensures that two movies with similar themes and vocabulary are judged close, regardless of how many total words were used. This makes it especially effective with TF-IDF vectors, as it captures how closely two movies align in terms of descriptive content rather than sheer word frequency.

1. What does filtering movies by a vote count threshold (like 90th percentile) help achieve?

Applying a vote count threshold, such as considering only movies above the 90th percentile of votes, ensures that the recommendations are both meaningful and reliable. Without this, a little-known film with only a handful of votes could appear as a top-rated movie due to statistical flukes. By filtering for movies with a sufficient number of votes, the system balances quality and popularity, surfacing titles that not only have high average ratings but also a strong level of consensus among viewers. This is why widely appreciated classics like The Shawshank Redemption naturally rise to the top.

Conclusion: Hence, we have successfully implemented Content based Recommender System.