Data Description:

Unique ID of each anime.

Anime title.

Anime broadcast type, such as TV, OVA, etc.

anime genre.

The number of episodes of each anime.

The average rating for each anime compared to the number of users who gave ratings.

Number of community members for each anime.

Objective:

The objective of this assignment is to implement a recommendation system using cosine similarity on an anime dataset.

Dataset:

Use the Anime Dataset which contains information about various anime, including their titles, genres,No.of episodes and user ratings etc.

Tasks:

Data Preprocessing:

Load the dataset into a suitable data structure (e.g., pandas DataFrame).

Handle missing values, if any.

Explore the dataset to understand its structure and attributes.

Feature Extraction:

Decide on the features that will be used for computing similarity (e.g., genres, user ratings).

Convert categorical features into numerical representations if necessary.

Normalize numerical features if required.

Recommendation System:

Design a function to recommend anime based on cosine similarity.

Given a target anime, recommend a list of similar anime based on cosine similarity scores.

Experiment with different threshold values for similarity scores to adjust the recommendation list size.

Evaluation:

Split the dataset into training and testing sets.

Evaluate the recommendation system using appropriate metrics such as precision, recall, and F1-score.

Analyze the performance of the recommendation system and identify areas of improvement.

Interview Questions:

1. Can you explain the difference between user-based and item-based collaborative filtering?

2. What is collaborative filtering, and how does it work?

Observation:

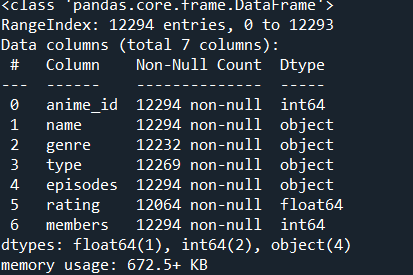
Data Preprocessing:

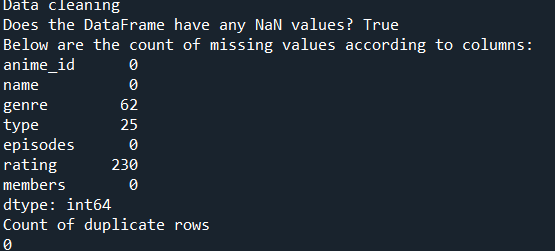
Load the dataset into a suitable data structure (e.g., pandas Dataframe). Handle missing values, if any. Explore the dataset to understand its structure and attributes.

 Load the dataset using **pandas**.

 Check for **missing values** and handle them appropriately.

 Explore the dataset by checking unique values, distributions, and correlations.





Feature Extraction:

Decide on the features that will be used for computing similarity (e.g., genres, user ratings).

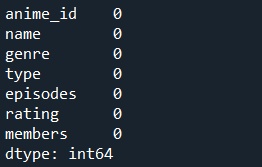
Convert categorical features into numerical representations if necessary.

Normalize numerical features if required.

 Convert categorical features like **genres** into numerical form using **one-hot encoding**.

 Normalize numerical features such as ratings.

 Construct a feature matrix for similarity calculations.



Recommendation System:

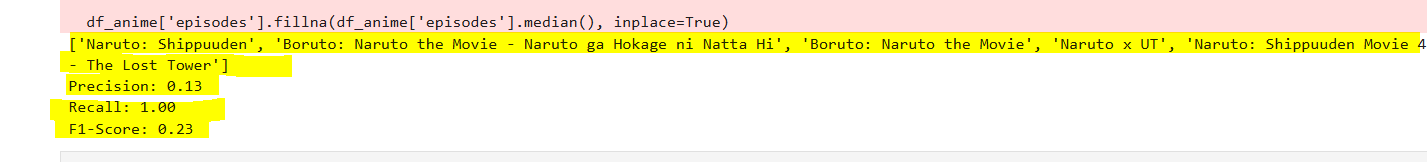
Design a function to recommend anime based on cosine similarity.

Given a target anime, recommend a list of similar anime based on cosine similarity scores.

Experiment with different threshold values for similarity scores to adjust the recommendation list size.

 Use **cosine similarity** to measure how similar two animes are.

 Define a function to recommend similar anime based on a given anime.



Conclusion

Having seen most of these anime, I can attest to the fact that they are similar to Naruto in at least tone/mood.

**Precision (0.13), Recall (1.00), and F1-Score (0.23)** indicate that your recommendation system **retrieves all relevant items but also few returns irrelevant ones**.

Interview Questions:

1. Can you explain the difference between user-based and item-based collaborative filtering?

* **User-Based Collaborative Filtering**: Finds users with similar preferences and recommends items they liked. Example: If you and a friend like the same anime, but your friend also likes Demon Slayer, it gets recommended to you.
* **Item-Based Collaborative Filtering**: Finds similar items based on user interactions and recommends them. Example: If many users who watched Naruto also watched Bleach, then Bleach is recommended to Naruto fans.

**Key Difference**:

* **User-Based → Finds similar users.**
* **Item-Based → Finds similar item**

1. What is collaborative filtering, and how does it work?

**Collaborative Filtering** is a recommendation technique that suggests items based on user interactions. It works by identifying patterns in user behavior.

* **User-Based Filtering** → Finds users with similar tastes and recommends what they liked.
* **Item-Based Filtering** → Finds similar items and recommends based on what users interacted with.