**Movie Recommendation System:**

**1. NumPy (import numpy as np)**

* **Purpose:** Handles numerical computations efficiently.
* **Usage:**
  + Storing and processing similarity scores in arrays.
  + Handling mathematical operations like distance metrics.
  + Managing large-scale numerical data structures.

**2. Pandas (import pandas as pd)**

* **Purpose:** Handles movie datasets efficiently.
* **Usage:**
  + Storing movie metadata (titles, genres, descriptions, etc.) in a **DataFrame**.
  + Performing **data manipulation** (filtering, sorting, and merging).
  + Reading datasets from files (CSV, Excel, JSON, etc.).

**3. Difflib (import difflib)**

* **Purpose:** Finds close matches between user input and movie titles.
* **Usage:**
  + Helps **handle typos or variations** in user queries.
  + Used for **matching user-input movie names** to actual movie titles in the dataset.
  + Example: If a user types *"Intersteller"*, it can match it to *"Interstellar"*.

**4. TfidfVectorizer (from sklearn.feature\_extraction.text import TfidfVectorizer)**

* **Purpose:** Converts text data (movie descriptions, genres) into numerical vectors.
* **Usage:**
  + Extracts **important keywords** from movie descriptions or genres.
  + Transforms movie plots into **TF-IDF vectors**, which allow the system to measure movie similarity.
  + Helps find movies with **similar themes and content**.

**5. Cosine Similarity (from sklearn.metrics.pairwise import cosine\_similarity)**

* **Purpose:** Measures similarity between movies based on their TF-IDF vectors.
* **Usage:**
  + Finds movies that are **most similar** to a given movie.
  + Ranks recommended movies based on how closely their descriptions match the user's input.
  + Example: If a user searches for *"Inception"*, cosine similarity can suggest movies with similar sci-fi or dream-related themes like *"Interstellar"* or *"The Matrix"*.