**Introduction**

This project is a Movie Recommender System designed to suggest movies based on user input. It uses a hybrid approach combining **content-based filtering** and **collaborative filtering** to provide accurate and personalized movie suggestions.

**Abstract**

The recommender system works in two major phases:

1. **Content-Based Filtering**: It processes movie titles using **TF-IDF vectorization** and identifies similar titles using **cosine similarity**.
2. **Collaborative Filtering**: It finds users who rated the given movie highly, then looks at what other movies those users liked.

An interactive search widget dynamically fetches results as the user types. The project merges both techniques to deliver reliable movie suggestions.

**Tools Used**

* **Python**
* **Pandas**
* **NumPy**
* **scikit-learn**
* **Regular Expressions (re)**
* **ipywidgets** (for live interaction in Jupyter/Colab)

**Steps Involved in Building the Project**

1. **Data Loading**  
   Load movies.csv and ratings.csv files which contain metadata and user ratings respectively.
2. **Data Cleaning**  
   Clean movie titles by removing special characters using regular expressions.
3. **TF-IDF Vectorization**  
   Use TfidfVectorizer with bi-grams to vectorize the cleaned movie titles.
4. **Cosine Similarity**  
   Measure similarity between input title and all other titles based on TF-IDF vectors.
5. **Interactive Search with Widgets**  
   Add real-time search box using ipywidgets to display similar titles as the user types.
6. **Collaborative Filtering**  
   Use user ratings to find users who liked a movie and recommend what else they liked.
7. **Display Results**  
   Merge and display top results based on a combined score of both filtering methods.

**Conclusion**

The Movie Recommender System successfully combines both content-based and collaborative filtering methods to deliver relevant and intelligent movie recommendations. By utilizing text processing (TF-IDF) for understanding movie titles and analyzing user preferences from ratings data, the system demonstrates how hybrid models can enhance user experience. The use of interactive widgets makes the application intuitive and responsive, making it suitable for real-world deployment or further enhancement with deep learning-based recommenders.