**Movie Recommendation System Using Similarity**

**-Based Filtering**

**A PROJECT REPORT**

**BY –**

Submitted by:

TEAM MEMBER 1- SWEETI RATHORE (E23CSEU1470)

TEAM MEMBER 2- HARSHIT PUNDIR (E23CSEU1468)

TEAM MEMBER 3- TANMAY JAIN (E23CSEU1442)

SUBMITTED TO:

Mrs. Sushmita Das

SCHOOL OF COMPUTER SCIENCE ENGINEERING AND TECHNOLOGY, BENNETT UNIVERSITY

GREATER NOIDA, 201310, UTTAR PRADESH, INDIA

NOVEMBER 2024

**INDEX**

|  |  |  |
| --- | --- | --- |
| Sr.No | Content | Page No |
| 1 | Abstract | 3 |
| 2 | Introduction | 4 |
| 3 | Related Work | 5 |
| 4 | Methodology | 8 |
| 5 | Software required | 9 |
| 6 | Experiment results | 9 |
| 7 | Future Scope | 10 |

**Abstract**

This project develops a movie recommendation system leveraging content-based filtering and cosine similarity. By analyzing metadata such as genres, actors, directors and reviews, the system computes similarity scores to recommend movies tailored to user preferences. The focus on cosine similarity ensures accurate measurement of similarity between movie vectors, making the system efficient and user-centric.

**INTRODUCTION**

Recommendation systems help users find content they might like, such as movies, books, or songs, by analyzing data. A **movie recommendation system** focuses on suggesting movies that match a user's preferences. This project uses a method called **content-based filtering**, which recommends movies based on their features, like genres, cast, and keywords. For example, if a user likes an action movie starring a certain actor, the system will suggest similar movies with the same genre or actor. This approach focuses on the content of the movies rather than user ratings or behavior.

To find similar movies, the project uses **cosine similarity**, a mathematical method to compare the features of two movies. It calculates how closely the attributes of one movie match another, providing accurate recommendations. The system is built on a dataset of movie details and allows users to input a movie they like to get suggestions of related movies. By making movie discovery easier, this system improves user experience and can be a useful tool for platforms like streaming services.

**Related work**

1. Movie Recommendation System Using Content Based Filtering and Cosine Similarity:

Authors: Aleena Joseph, Ms. Jetty Benjamin

Publisher: Amal Jyothi College of Engineering, Kanjirappally, Kottayam

Year: 2022

The paper talks about how recommendation systems can make users more interested by understanding what they like and giving them personalized suggestions. It focuses on using content-based filtering and cosine similarity to find and suggest movies with similar features. The goal is to show that these techniques can help in detecting movie similarities and providing better recommendations to users.

1. Intelligent Movie Recommendation System Using AI and ML:

Authors: D.S. Hirolikar, Ajinkya Satuse, Omkar Bhalerao, Pavan Pawar, Hrithik Thorat

Publisher: International Journal for research in Applied science & Engineer technology

Year: 2022

The paper talks about building a smart movie recommendation system using AI. It uses data from 5,000 movies and compares them with a method called cosine similarity to suggest movies that users might like.

1. Movie Recommendation System Modeling Using Machine Learning:

Author: Raja Marappan, S. Bhaskaran

Publisher: Scientific Publiscations

Year: 2022

This paper talks about the cosine similarity method that is more suitable for the movie recommendation system and to classify the movie reviews we can use the SVM classifier that is much more better than the NB classifier.

1. Latent Factor Models for Web Recommender Systems:

Authors: Bee-Chung Chen, Deepak Agarwal, Pradheep Elango, Raghu

Ramakrishnan

Publisher: Yahoo! Research & Yahoo! Labs

This paper discuss about the model that uses feature-based regression to predict the initial point for online learning, and reduces the dimensionality of online learning Rapidly update online models once new data is received.

Fast learning: Low dimensional and easily parallelizable Online selection for the best dimensionality

**Methodology**

**1.Data Collection**

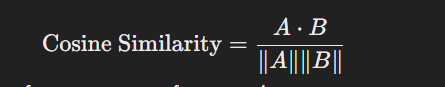
* **Movie Metadata**: Gather attributes of movies such as genres, cast, directors, keywords, etc.
* **User Preferences**: Collect user ratings or interactions to personalize recommendations.

**2. Feature Extraction**

* **Text Vectorization**: Convert textual attributes like movie genres, plot keywords, or descriptions into numerical representations.
  + Example: Use **TF-IDF (Term Frequency-Inverse Document Frequency)** or **Count Vectorizer** to represent textual data as vectors.
* **Feature Matrix**: Create a feature matrix where each row represents a movie, and columns correspond to extracted features.

**3. Compute Similarities**

* **Cosine Similarity**:
  + Cosine similarity measures the cosine of the angle between two vectors. It is computed as:



where A and B are the feature vectors of two movies.

* + Results in a similarity score between 0 (no similarity) and 1 (perfect similarity).
* **Similarity Matrix**:
  + Compute pairwise cosine similarities for all movies, resulting in a similarity matrix where entry (I , j)is the similarity score between movie i and movie j.

**SOFTWARE REQUIRED**

 Python (3.8+)

 Jupyter Notebook, VSCode, or PyCharm

 pandas

 numpy

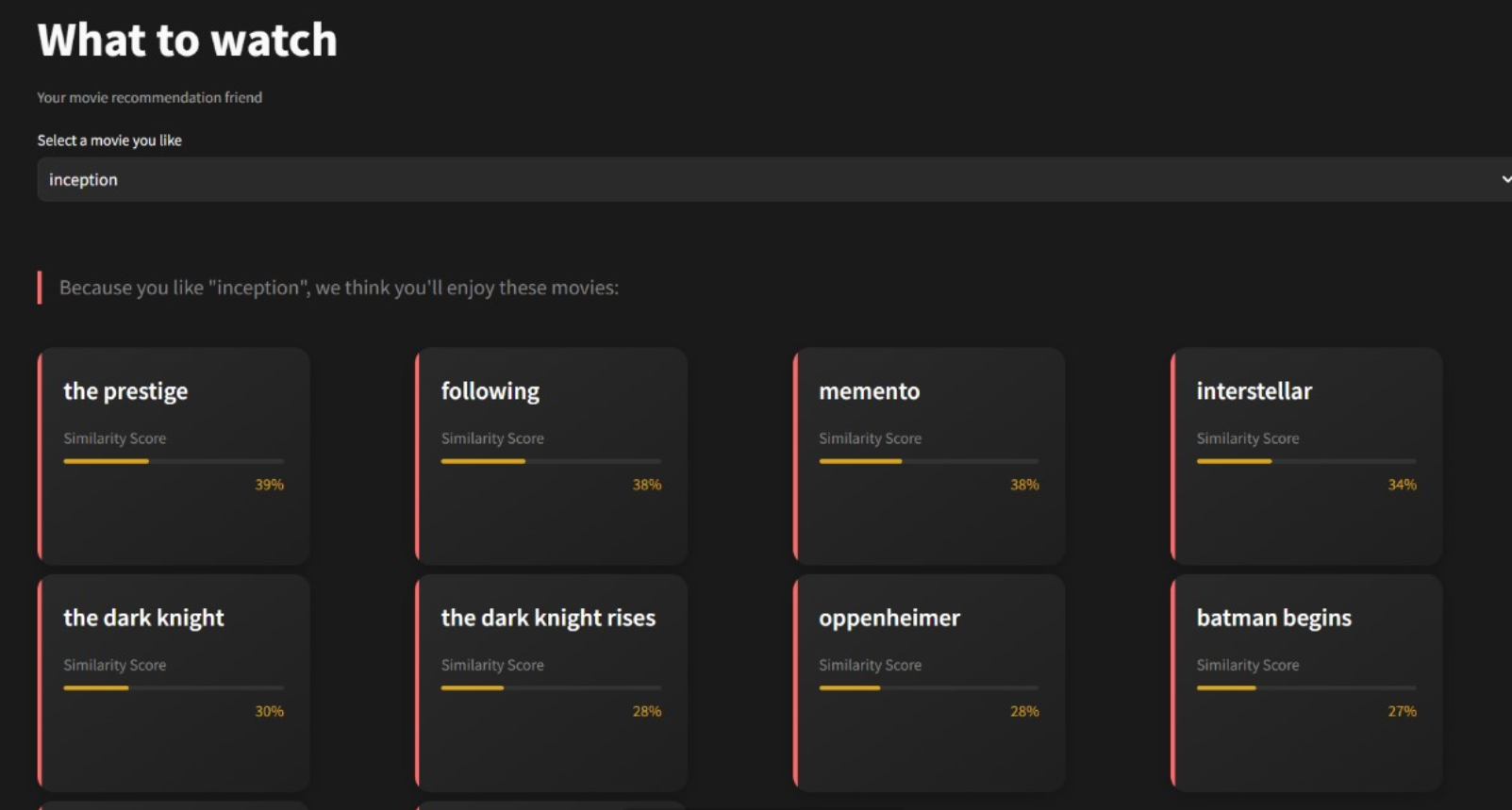
 scikit-learn

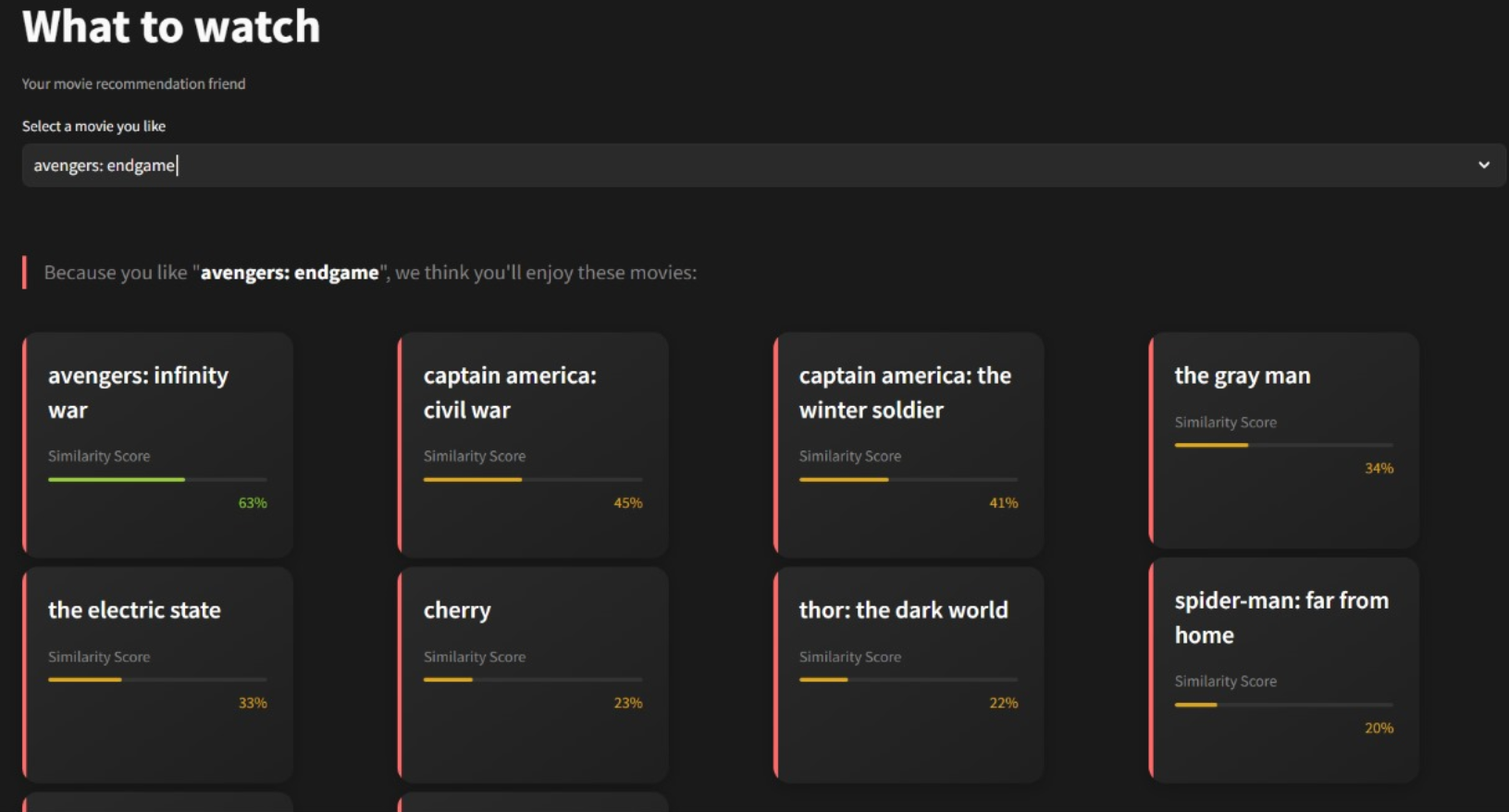
 matplotlib

 seaborn

Streamlit

**EXPERIMENTAL RESULTS**





**FUTURE SCOPE**

* **RAG based recommendation** : We will implement a RAG based recommendation system by using an LLM (gemma models or llama) which will search movies just by their description.
* **Neural Collaborative Filtering (NCF)** can combine user preferences and movie features using deep neural networks.

This allows for:

* + Capturing complex, non-linear relationships between users and items.
  + Better handling of sparse datasets where traditional content-based filtering struggles.
* **Explainable Recommendations**

Enhance user trust by explaining why a particular movie is recommended (e.g., "Similar to movies you liked: *Inception* and *Interstellar*").