IMDB MOVIE RECOMMENDATION SYSTEM

Data Collection (Web Scraping)

Tools Used: Selenium, Pandas
Approach:

- Scraped IMDb movie data based on different genres.
- Extracted movie details using tag names and class names.
- Converted extracted data into a Pandas DataFrame (pd.DataFrame()).
- Added a "genre" column to each movie for better categorization.
- Concatenated multiple dataframes and saved the final dataset as a .csv file.

Text Cleaning:

Tools Used: NLTK, SpaCy
Steps:

- Removed numbers, symbols, and special characters from movie descriptions.
- Applied stopword removal (e.g., "the", "is", "and").
- Used stemming/lemmatization to reduce words to their root forms.
- Converted all text to lowercase for consistency.

Text Representation (Feature Engineering)

Technique Used: TF-IDF Vectorizer (from Scikit-learn)
Why TF-IDF?

- Converts text into numerical values.
- Assigns higher importance to unique words in movie descriptions.

Dimensionality Reduction:

- Used **Principal Component Analysis (PCA)** to reduce feature space and make data visualization possible.

Cosine Similarity for Movie Recommendation

Algorithm Used: Cosine Similarity
Why Cosine Similarity?

- Measures similarity between movies based on their textual descriptions.

Model Deployment:

- Used Streamlit to build a user-friendly interface for movie recommendations.
- Allowed users to input movie names and retrieve similar movies based on cosine similarity.

Conclusion

- This project successfully scraped and processed IMDb movie data, cleaned textual information, applied machine learning techniques for similarity detection, and deployed a functional movie recommendation system. Future improvements may include incorporating deep learning models and expanding the dataset for better accuracy.