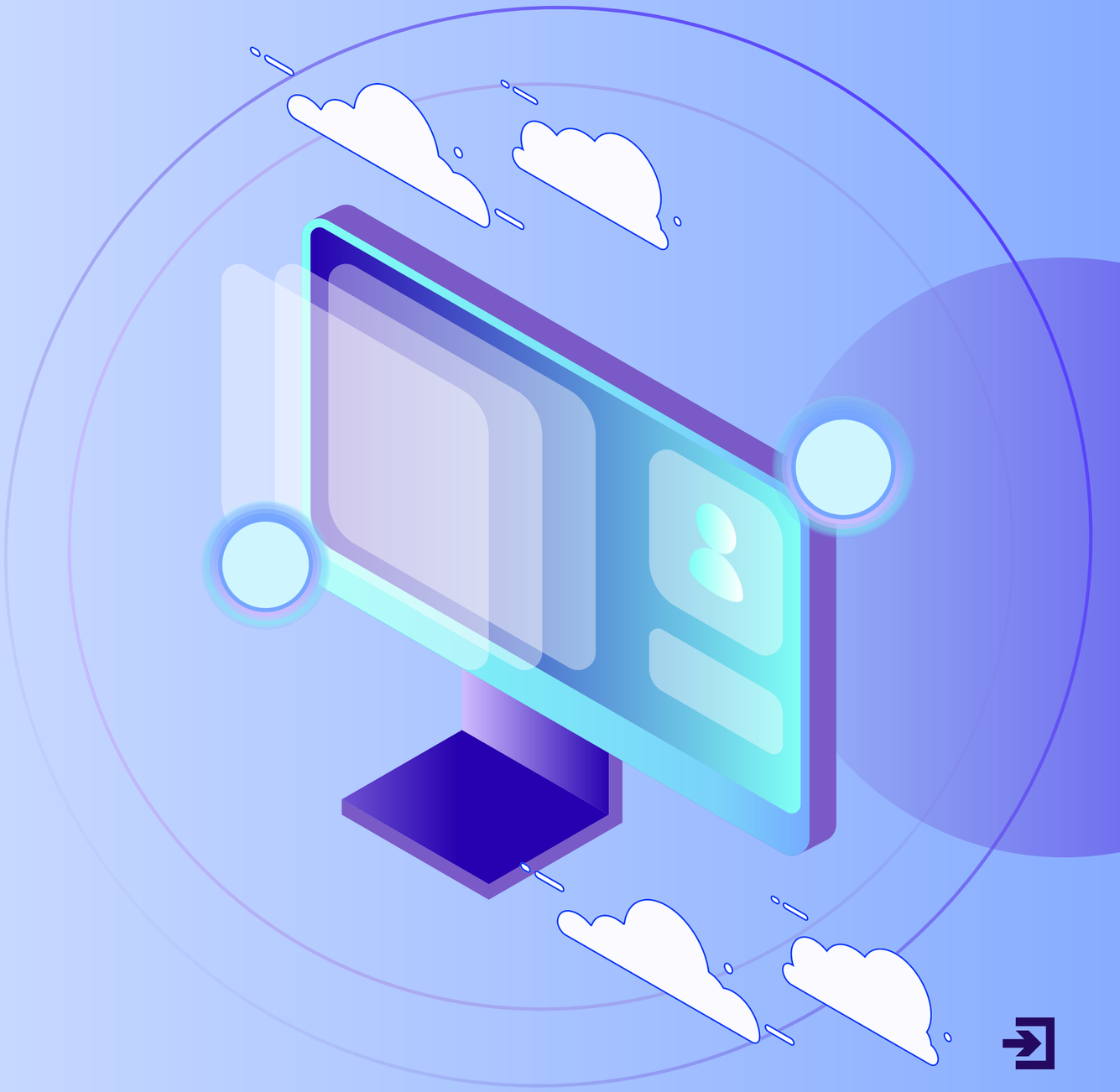
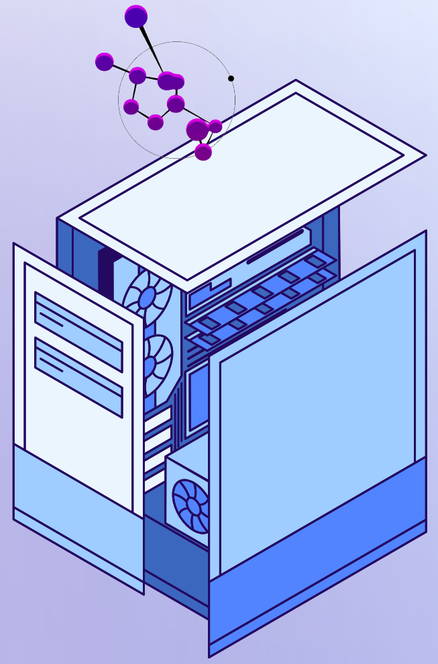
MOVIE RECOMMENDATION SYSTEM

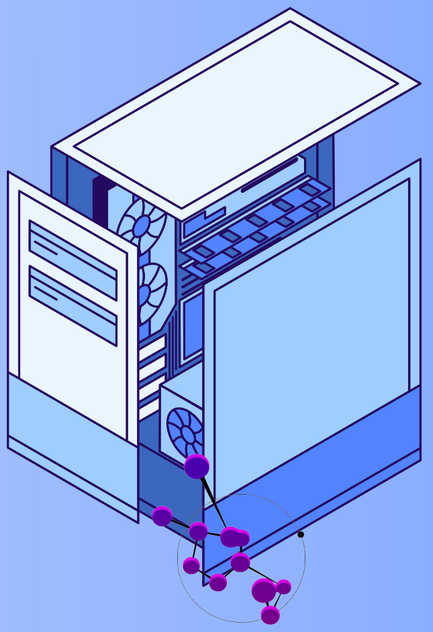


2206106 – PARNA BHATTACHARYYA
2206107 – PRATYUSH PRASOON

INTRODUCTION



- A Movie Recommendation System suggests movies to users based on their interests.
- Recommendation systems is widely used on platforms like Netflix, Prime Video, and YouTube analyze user preferences and find the best matches based on various algorithms.
- This project uses Content-Based Filtering, leveraging movie metadata such as genres, runtime, popularity, cast, crew, and director.
- Our system employs TF-IDF vectorization and Cosine Similarity to find similar movies.



PROBLEM STATEMENT

- With the vast number of movies available, finding ones that match a user's preferences can be challenging.
- This project aims to develop a movie recommendation system that suggests films similar to the user's favorite movie.
- The system should be capable of efficiently processing a large dataset while providing accurate and relevant recommendations.

DATASET AND PREPROSESSING

- **Dataset:** The system utilizes a movie dataset containing attributes such as title, genres, cast, director, and keywords to generate recommendations.
- **Data Cleaning:** Any missing values in these attributes are replaced with empty strings to ensure consistency in processing.
- **Feature Engineering:** Relevant movie metadata is combined into a single textual feature, making it easier to analyze similarities between films.

COMPONENTS USED

- **Programming Language:** Python
- **Libraries:**

1. Pandas – Used for handling and processing movie data from a CSV file.

2. NumPy – Used for mathematical computations and working with numerical data.

3. Scikit-learn:
   a. TfidfVectorizer – Converts text-based features (genres, keywords, cast, director) into numerical vectors.
   b. Cosine Similarity – Measures how similar two movies are based on their feature vectors.

4. Difflib – Helps find the closest match for the movie title entered by the user.

HOW IT WORKS ?

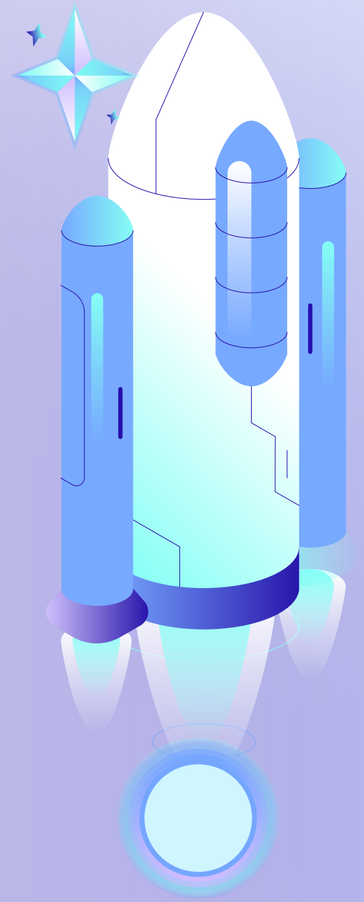
- **Load the Dataset** – The system loads a dataset with movie details like title, genres, cast, and director.
- **Process the Data** – Important features are selected and combined into a single text string for each movie.
- **Vectorization** – The text data is transformed into numerical vectors using TfidfVectorizer.
- **Find Similarity** – Cosine Similarity calculates how similar two movies are based on their features.
- **Recommend Movies** – The system finds the closest matching movie and suggests the top 30 similar ones.

RESULTS & RECOMMENDATIONS

- The system effectively recommends movies that are similar to the user's input by analyzing various movie features.
- Provides up to 30 relevant suggestions based on content similarity.
- The recommendation accuracy can be further improved by incorporating additional metadata, such as user ratings, reviews, or advanced filtering techniques.

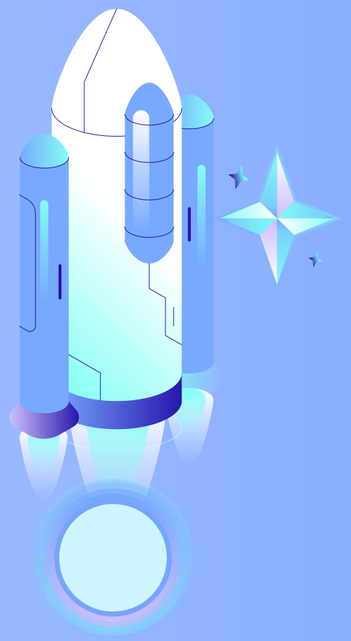
CHALLENGES

- **Data Cleaning** – Missing movie details (NaN values) were replaced with empty strings to avoid errors.
- **User Input Matching** – Difflib was used to handle partial or incorrect movie names and find the closest match.
- **Better Recommendations** – TF-IDF and Cosine Similarity improved accuracy by analyzing movie content.



FUTURE WORK

- **Enhance Accuracy –** Include user ratings and reviews for more personalized recommendations.
- **Advanced AI Models –** Use NLP transformers to improve content-based recommendations.
- **Hybrid Recommendation –** Combine content-based filtering with collaborative filtering for better results.
- **Deployment –** Convert the system into a web or mobile application for easy access.



THANK YOU!