Content Recommendation System

Presented by **Nidhish Sharma**

Student code : iitrprai_24082677

nidhishsharma060504@gmail.com

Minor in AI

IIT Ropar x Masai

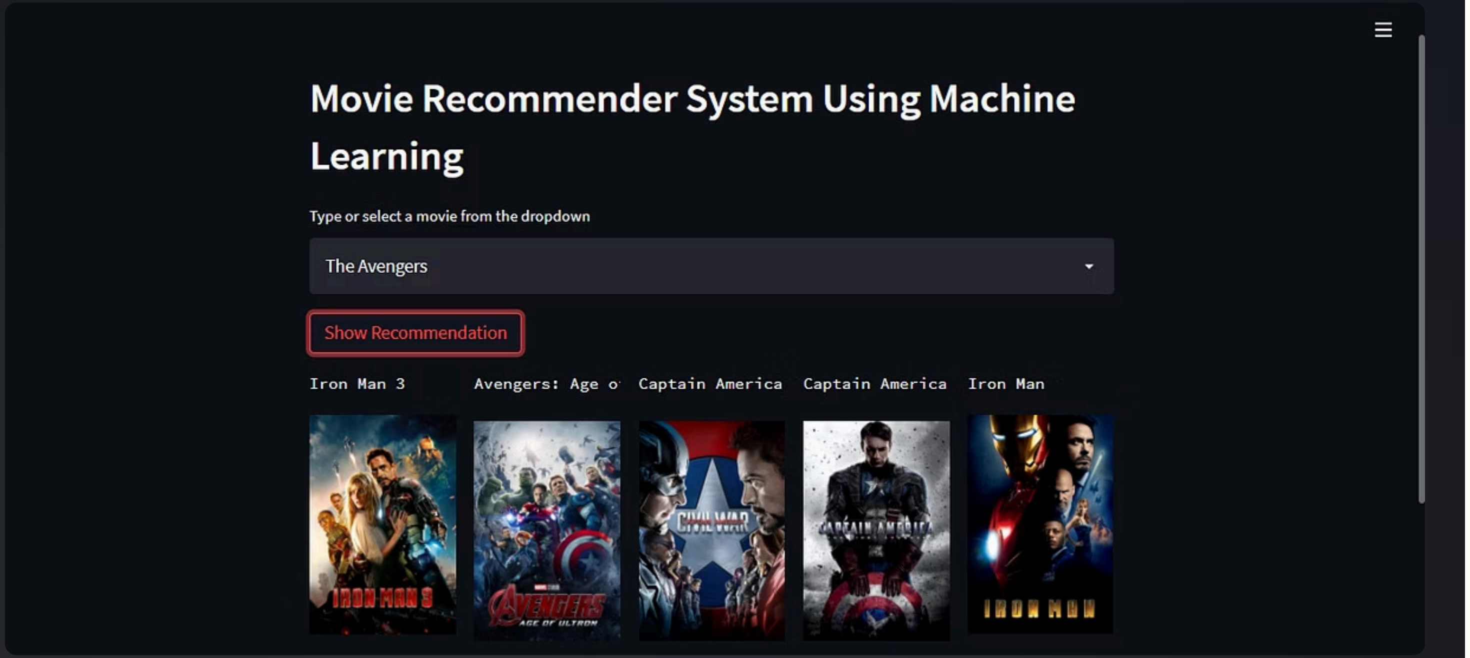
Module E

Problem Statement

Finding the right movie is challenging due to the large number of options available.
Users often waste time searching and may not get personalized suggestions.
This project develops a **Content-Based Recommendation System** that recommends movies based on genres, cast, crew, keywords, and overview to provide relevant, user-specific recommendations.

Why Movie Recommendation is Needed?

- Abundance of available movies overwhelms users
- Difficulty in discovering personalized, relevant content
- Generic recommendation leads to poor user satisfaction
- Content-based recommendations leverage movie attributes
- Improves user engagement by matching preferences to content



Movie Recommender System Using Machine Learning

Type or select a movie from the dropdown

The Avengers

Iron Man 3 | Avengers: Age o | Captain America | Captain America | Iron Man

Project Outcome

A fully functional **Content Recommendation System** web application has been developed using Python and Streamlit. Users can search for a movie and receive a list of similar movies along with their posters. The system effectively analyzes movie features like genres, cast, crew, keywords, and overview to provide personalized recommendations.

Project Objectives

Develop Recommendation System

Create a content-based movie recommendation system tailored for personalized suggestions.

Data Preprocessing

Clean and transform raw movie dataset including cast, crew & genres for analysis.

Recommend Movies

Generate recommendations based on similarity of movie content attributes.

Display Posters

Show movie posters alongside recommendations for enhanced user experience.

Dataset & Tools

The project utilizes the **TMDB 5000 Movies Dataset**, containing extensive data on:

1) 5,000 movies with detailed metadata

2) Cast, crew, genres, keywords, and movie overviews

Tools and Libraries:

Python

Pandas

NLTK

Scikit-Learn

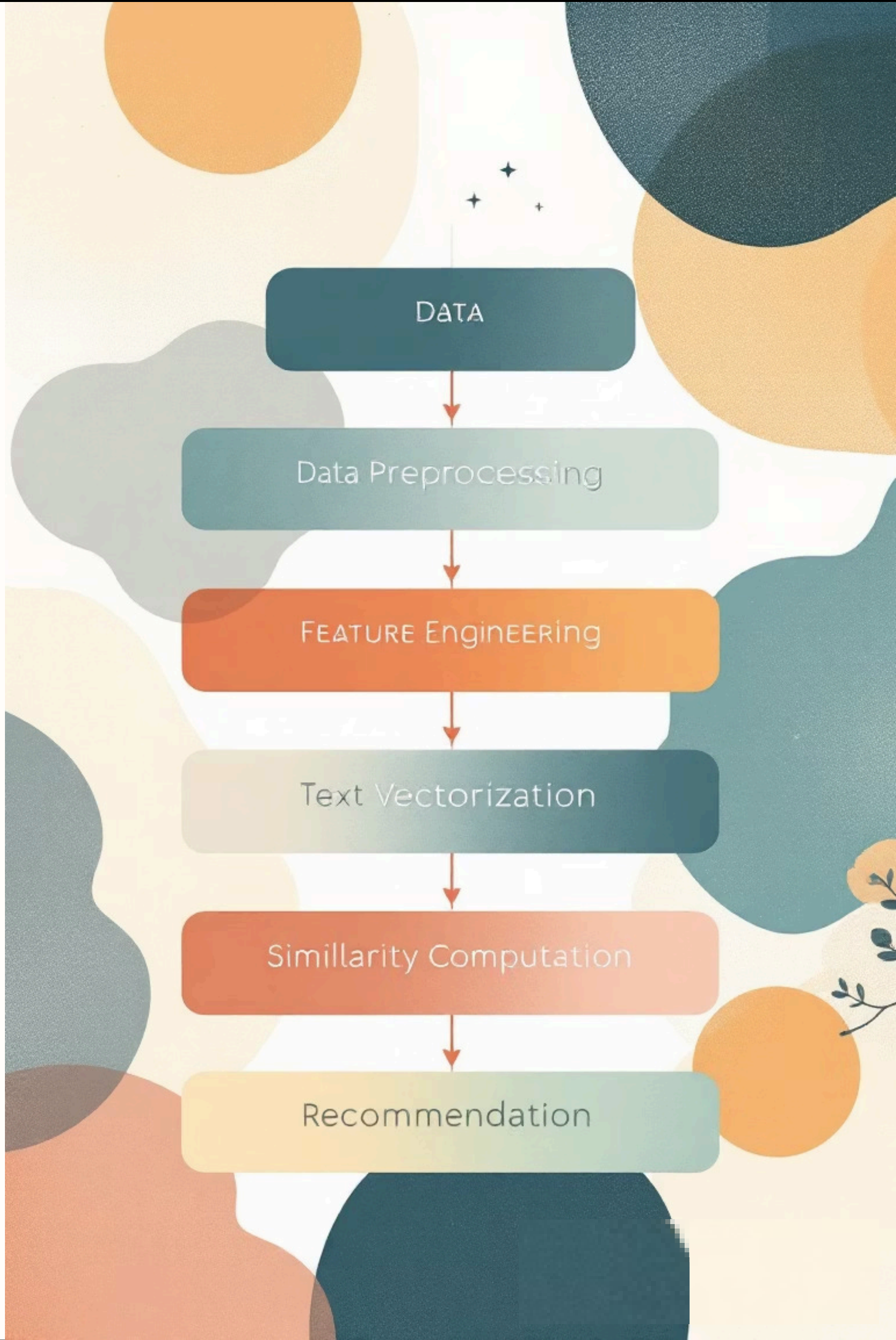
Streamlit

Dataset Size: ~5,000 movies

Methodology & Workflow

The step-by-step workflow for the content-based recommendation system:

1. **Data Preprocessing:** Clean dataset and extract genres, cast, crew, keywords, and overview
2. **Feature Engineering:** Combine attributes into a consolidated ‘tags’ column
3. **Text Vectorization:** Use CountVectorizer to convert tags into numeric vectors
4. **Similarity Computation:** Calculate cosine similarity between movie vectors
5. **Recommendation:** Identify and output movies similar to the user’s selected movie



Implementation

Data Collection: Gather movie metadata and details from TMDB.
Datasets:- TMDB 5000 Movies & Credits

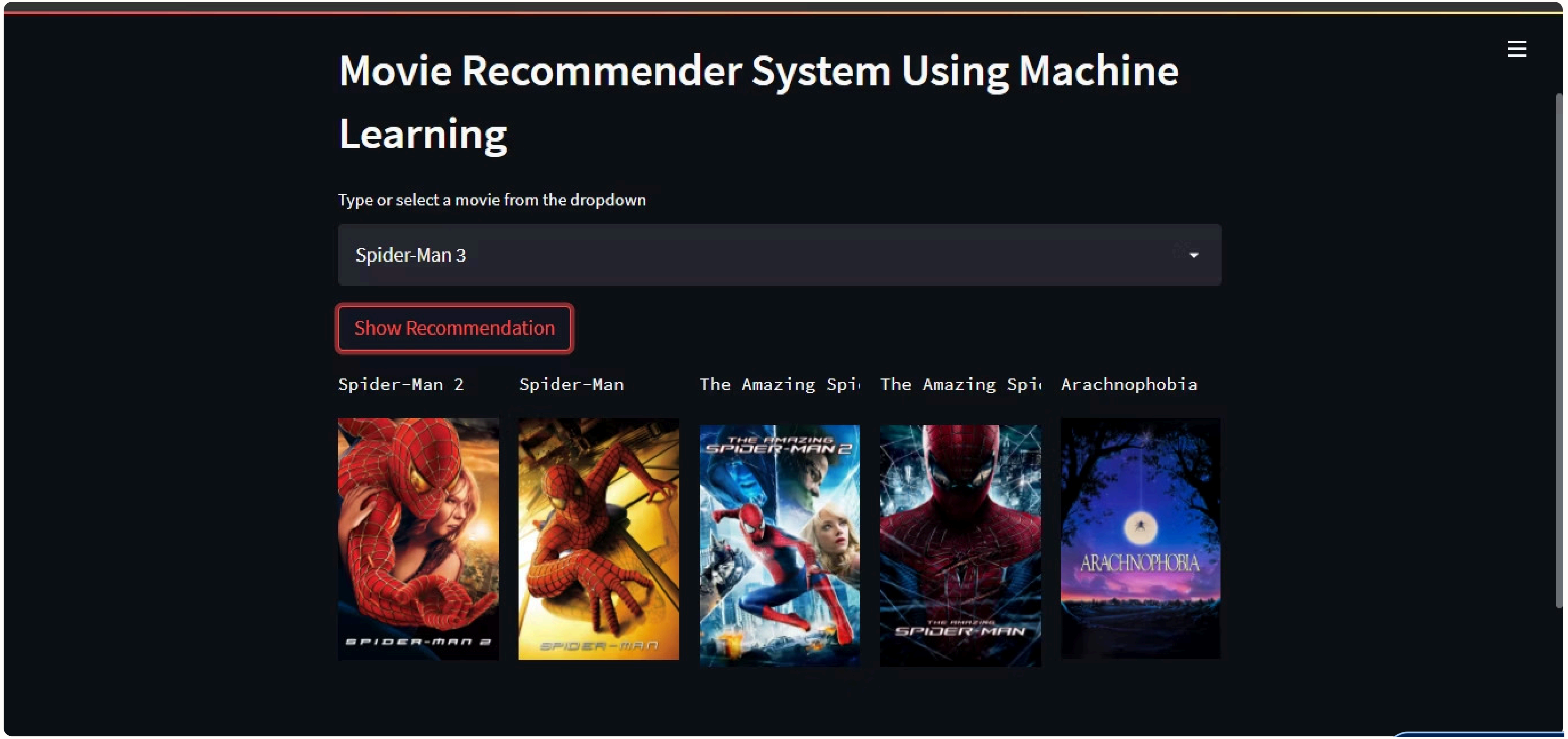
Data Preprocessing: Clean, tokenize and merge relevant fields into tags. Extracted genres, cast, crew, keywords, and overview.

Vectorization: Convert text tags into numerical vectors using CountVectorizer

Similarity: Compute **Cosine Similarity** matrix for recommendation lookup and to measure closeness between movies.

Web App Development: Built an interactive **User-Interface** using **Streamlit** to display recommendations and movie posters.

Web-App Demo



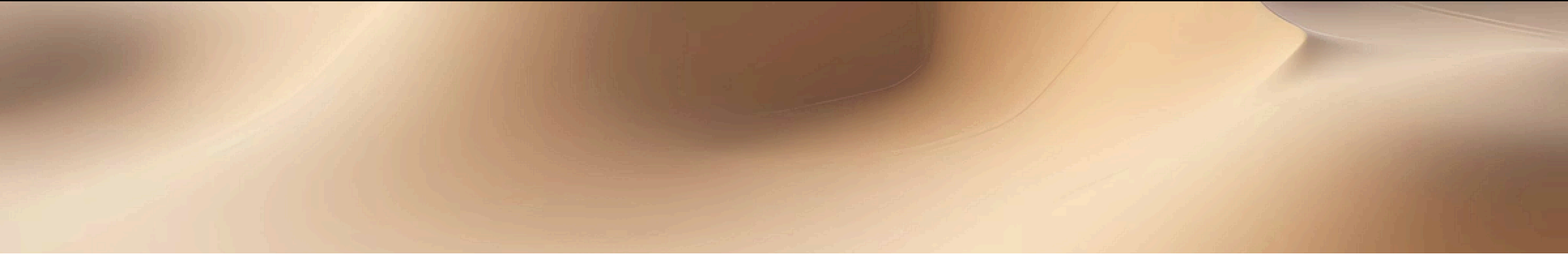
Movie Recommender System Using Machine Learning

Type or select a movie from the dropdown

Spider-Man 3

Show Recommendation

Spider-Man 2 | Spider-Man | The Amazing Spi | The Amazing Spi | Arachnophobia



Results & Observations

Key outcomes observed from system outputs:

- Relevant movie recommendations aligning with user preferences
- Effective handling of missing or unavailable posters (placeholders used)
- System performance is responsive with quick recommendation generation
- Limitations include reliance solely on content similarity without user behavior integration

Recommendation Example	Observations
Movie A similar to Movie B & C	Posters displayed; thematic similarity clear
Missing posters handled gracefully	Placeholder images maintain UI consistency

Conclusion & Future Work

The project successfully delivered a content-based movie recommendation system with an intuitive interface and accurate suggestions.

- Learnings: Importance of thorough preprocessing and feature engineering for quality recommendations
- Future Improvements:
  - Integrate collaborative filtering to incorporate user preferences and ratings
  - Enhance user interface with richer interactivity and filters
  - Expand dataset to include more movies and multimedia features

Thank You