

# **MOVIE RECOMMENDATION SYSTEM**

Alok Sinha, asinha\_be22@thapar.edu

Jyotish Yadav, jyadav\_be22@thapar.edu

## **1. Introduction:**

### **A. Technical Background of the Project:**

Recommendation systems are widely used to predict user preferences and suggest items such as movies, music, or products. The project leverages cosine similarity, a mathematical approach for determining the similarity between two vectors, to create personalized movie recommendations.

### **B. Technical Concepts Used:**

- Cosine similarity for vector similarity measurement
- Vector representations of movie features
- Dataset handling and preprocessing
- Python for implementation
- Libraries: NumPy, pandas

### **C. Motivation:**

With the abundance of movie content available today, users often struggle to decide what to watch. A recommendation system simplifies this process by filtering options based on user preferences.

### **D. Problem Statement:**

To develop an efficient and scalable movie recommendation system that provides users with personalized suggestions based on similarity metrics.

### **E. Area of Application:**

This system can be applied to online streaming platforms, e-commerce websites, and personal movie libraries.

### **F. Dataset and Input Format:**

The system uses a publicly available movie dataset containing features like movie titles, genres, ratings, and user reviews. Input is structured in tabular format, suitable for processing with Python data libraries.

## **2. Objective**

### **A. Main Objective:**

To build a movie recommendation system using cosine similarity to calculate similarity scores between movies.

### **B. Sub-Objective:**

- To preprocess and transform movie data into vector form.
- To optimize the similarity algorithm for faster and accurate recommendations

### 3. Methodology

#### A. Steps:

1. **Dataset Preparation:** Import and clean movie data.
2. **Feature Engineering:** Convert movie features like genres and ratings into numerical vectors.
3. **Cosine Similarity Calculation:** Compute pairwise similarity scores.
4. **Recommendation Generation:** Recommend movies based on similarity rankings.

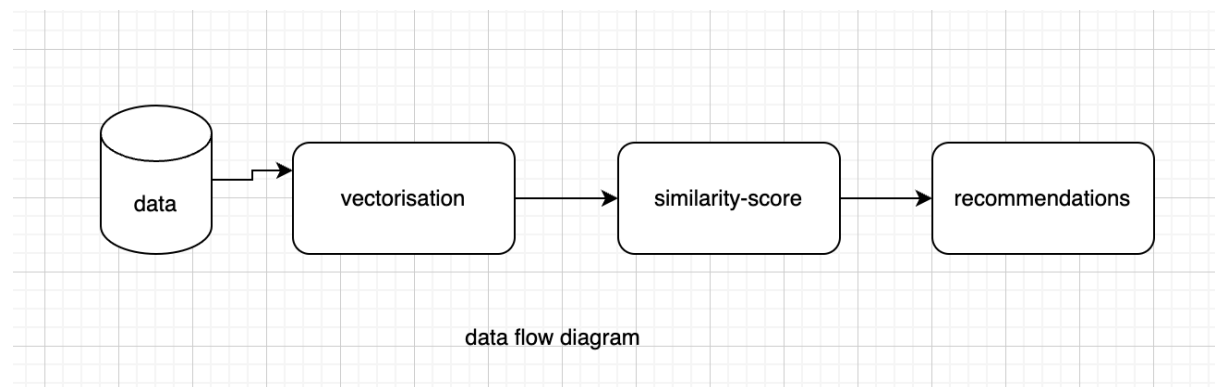
#### B. Deliverables of Each Step or Phase:

- **Phase 1:** A cleaned and structured dataset.
- **Phase 2:** Feature vectors representing movies.
- **Phase 3:** A similarity matrix.
- **Phase 4:** A working recommendation system.

### 4. Working Model

#### A. Technical Diagram:

Includes a flowchart showing:



- Input data -> Feature vectorization -> Similarity calculation -> Recommendations.

#### B. Working Module:

##### Data Preprocessing Module:

- Cleans the dataset by handling missing values and duplicates.
- Encodes features (e.g., genres) and normalizes data.

##### Feature Vectorization Module:

- Converts movie features (e.g., genres, descriptions, ratings) into numerical vectors.

##### Similarity Computation Module:

- Uses cosine similarity to calculate the closeness between movie vectors.

##### Recommendation Module:

- Ranks movies based on similarity scores and generates top-N recommendations for users.

### C. Attained Deliverable:

Preprocessed dataset ready for analysis.

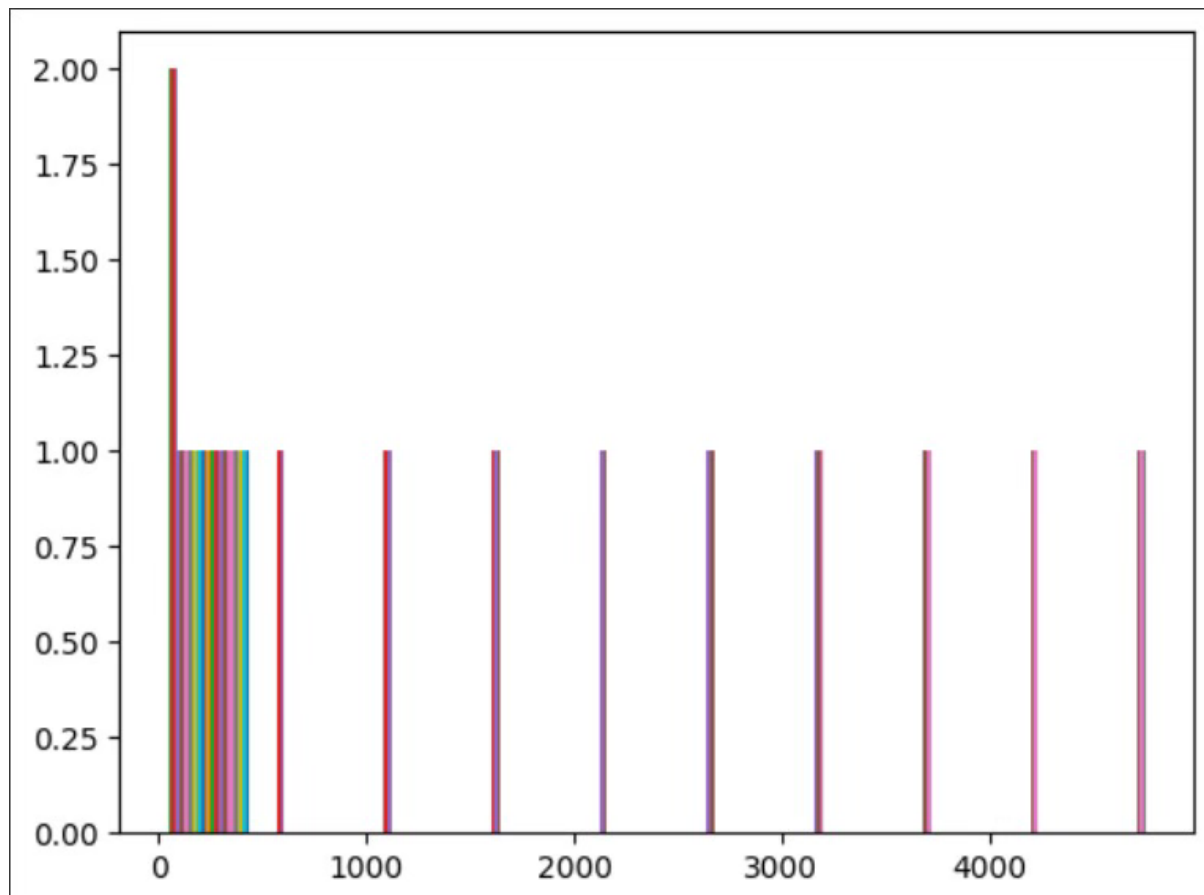
Feature vectors representing movie metadata and user preferences.

Similarity matrix containing pairwise cosine similarity scores for movies.

A functional system providing personalized movie recommendations.

## Results

### A. Outcome Graphs:



- Integration with user feedback for improved recommendations.
- Expansion to handle real-time queries.
- Incorporation of collaborative filtering for hybrid recommendations.

Project link- <https://github.com/MRwrong7/Movie-Recommendation-using-ML->

## 7. References

---

8. <https://www.kaggle.com/datasets/parasharmanas/movie-recommendation-system>

### Submitted by:

Alok Sinha  
102203580  
Jyotish Yadav  
102203632

### Submitted to:

Anjula Mehto  
TIET, Patiala