# Movie Clustering Based On Ratings

Project is submitted in fulfilment of the requirements of the course

**INFORMATION RETRIEVAL (CSE 466)**

Submitted by

Pasupuleti Divya

AP22110010969)


Under the Guidance of

**Dr Bala Venkateswarlu**

**Department of Computer Science and Engineering**
**SRM University–AP**
**Neerukonda, Mangalagiri, Guntur**
**Andhra Pradesh – 522 240**

# TABLE OF CONTENTS

# 1. DECLARATION

This is to certify that the work present in this Project entitled “Movie Clustering Based On Ratings”has been carried out by P.Divya. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology in School of Engineering and Sciences.

Supervisor

(Signature)

Dr.Bala Venkateswarlu

Assistant Professor,

Department of CSE

## 2. ACKNOWLEDGEMENT

I would like to acknowledge the guidance and support provided by my mentor and advisor Dr. Bala Venkateswarlu. His wisdom and experience helped steer me in the right direction, and his feedback was instrumental in refining my ideas and approach throughout the project. I extend my sincere thanks to my friends and well-wishers for their motivation and support during the course of this work. Their encouragement greatly contributed to the successful completion of this project.Furthermore, I am thankful for the resources and opportunities provided by my university, which enabled me to carry out this project successfully.

## 3. NOVELTY OF THE PROJECT

Unlike traditional movie recommendation systems that depend on user preferences or genres, this project groups movies purely based on rating patterns using unsupervised learning. The novelty lies in automatically discovering hidden rating-based groups without any labeled data, which helps in improving recommendation strategies and audience analysis.

## 4. ABSTRACT

This project focuses on clustering movies based on their **average user ratings** using the **K-Means clustering algorithm**, which is an unsupervised machine learning technique. The **MovieLens dataset** is used as the primary data source, as it contains real-world user rating information along with movie details. Initially, the dataset undergoes **data cleaning** to remove duplicates, handle missing values, and filter invalid rating entries. After that, **Exploratory Data Analysis (EDA)** is performed to understand the overall distribution of ratings, user activity, and movie popularity patterns.

Once the data is prepared, the **average rating of each movie** is calculated and used as the main feature for clustering. Since K-Means is a distance-based algorithm, **feature scaling** is applied using Standard Scaler to ensure accurate and fair distance calculations. The K-Means algorithm is then applied to group movies into different clusters, where each cluster represents a distinct rating level such as low-rated, moderately rated, and highly rated movies.

The outcome of this project helps in clearly identifying **movie popularity trends** and audience preferences. It organizes large movie datasets into meaningful groups, which can be effectively used in **future recommendation systems, movie analytics, and decision-support applications**. This project demonstrates how unsupervised machine learning can be successfully applied to real-world datasets to discover hidden patterns and insights without the need for lable data.

# 5. INTRODUCTION

Movies are one of the most popular forms of entertainment in today's digital world. With the growth of online streaming platforms such as Netflix, Amazon Prime, and IMDb, millions of users watch and rate movies every day. These ratings generate a large amount of data that reflects user preferences, movie quality, and popularity. Analyzing this data helps in understanding audience behavior and improving movie recommendation systems. However, manually analyzing such huge datasets is not practical. Therefore, machine learning techniques are used to automatically discover patterns in the data.

This project focuses on clustering movies based on user ratings using the K-Means clustering algorithm, which is an unsupervised learning method. Clustering helps group movies with similar rating patterns into the same category. By doing this, movies can be classified into groups such as high-rated, medium-rated, and low-rated movies.

The MovieLens dataset is used in this project, as it contains real user ratings for a large number of movies. The project involves data preprocessing, data analysis, feature scaling, and K-Means clustering. The final result helps in understanding movie popularity and forms a strong base for future movie recommendation systems and business analytics in the entertainment industry.

## 5.1 Background / overview

Nowadays, movies are watched by millions of people on online platforms like **Netflix, Amazon Prime, and IMDb**. Users give ratings to movies based on their experience. Every day, a huge amount of rating data is generated. Analyzing this large data manually is very difficult and time-consuming.

To handle this problem, **machine learning techniques** are used. These techniques help in finding hidden patterns from large datasets automatically. One such technique is **clustering**, which groups similar data without using predefined labels.

In this project, **K-Means clustering** is used to group movies based on their **average user ratings**. This helps in identifying **high-rated, medium-rated, and low-rated movies**. The project uses the **MovieLens dataset**, which contains real user ratings. This analysis is useful for improving **movie recommendation systems** and understanding **movie popularity trends**.

## 5.2 Problem definition

Online movie platforms receive millions of user ratings every day, resulting in large volumes of data. However, this data is mostly unstructured and difficult to analyze manually. Without proper analysis, it becomes challenging to:

- Understand movie popularity trends
- Identify high-rated and low-rated movies
- Support recommendation systems

  Traditional methods are slow, inefficient, and cannot handle large datasets effectively. Therefore, there is a need for an automatic and efficient machine learning-based approach to group movies based on their rating patterns.

## 5.3 Objectives

The main objectives of this project are:

- To collect and preprocess movie rating data from a reliable dataset.
- To analyze user ratings and compute the average rating for each movie.
- To apply the K-Means clustering algorithm to group movies based on their average ratings.
- To classify movies into different rating-based clusters such as low-rated, medium-rated, and high-rated movies.
- To visualize the clustered results for better understanding and analysis.
- To evaluate the effectiveness of clustering in identifying movie popularity patterns.
- To provide insights that can be used for movie recommendation systems in the future

## 5.4 Scope of the project

- The project is limited to clustering movies based only on user ratings and does not include content-based features such as genre, cast, or storyline.
- It uses the MovieLens dataset as the primary data source for analysis.
- The project focuses on applying K-Means clustering, an unsupervised machine learning algorithm.
- The system groups movies into low, average, and high-rated clusters to understand rating patterns.
- The project is designed for academic and learning purposes and not for direct commercial deployment.
- It provides analytical insights only and does not generate real-time personalized recommendations.
- The results can be extended in the future by adding more features like genres, user demographics, and review text.

## 5.5 Brief description of all modules

**1. Data Collection**

This module is responsible for collecting the dataset required for the project. The MovieLens dataset is used, which contains movie details and user ratings. The dataset is uploaded into the system for further processing.

**2. Data Preprocessing**

In this module, the raw data is cleaned and prepared for analysis. It removes missing values, duplicate records, and invalid ratings. The cleaned data is then formatted properly to make it suitable for machine learning.

**3. Exploratory Data Analysis (EDA)**

This module analyzes the dataset to understand the distribution of movie ratings, number of users, and number of movies. Graphs and summaries are generated to study patterns and trends in the data.

**4. Feature Extraction**

This module calculates the average rating of each movie from all user ratings. The average rating is selected as the main feature for clustering.

**5. Feature Scaling**

Since K-Means works based on distance, this module scales the average ratings using Standard Scaler. Scaling improves clustering accuracy and performance.

**6. K-Means Clustering**

This is the core module of the project. The K-Means algorithm is applied to group movies into clusters based on their average ratings. Each movie is assigned a cluster representing its rating category.

**7. Cluster Analysis & Visualization**

This module visualizes the formed clusters using scatter plots. It helps in understanding how movies are grouped based on ratings and how cluster centers are positioned.

**8. User Query & Recommendation**

This module allows users to enter a movie name and view its:

- Average Rating
- Cluster Number
- Similar Rated Movies from the same cluster
- Movie Poster URL

**9. Result Storage & Export**

This module saves the final clustered movie results into a CSV file. The output file can be downloaded and used for future analysis or reporting**.**

## 6. OBJECTIVES

**1. Explore and Understand the Movie Rating Data (EDA):**

The first objective of this project is to analyze the movie ratings dataset to understand its structure and characteristics. This includes studying the distribution of ratings, identifying highly rated and poorly rated movies, analyzing the number of ratings per movie, and visualizing rating trends using charts and graphs. Exploratory Data Analysis helps in understanding how users rate movies and reveals important patterns in the data.

**2. Compute Average Ratings for Movies:**

The next objective is to compute the average rating for each movie using the user rating data. This conversion from user-level ratings to movie-level statistics provides a compact and meaningful feature for clustering. It helps in representing each movie using a single numeric rating value.

**3. Apply Machine Learning for Movie Clustering:**

This project aims to apply the K-Means clustering algorithm to group movies based on similar average ratings. Movies with comparable popularity levels are placed in the same cluster. This unsupervised learning approach allows automatic grouping without predefined labels.

**4. Analyze Cluster Performance and Movie Similarity:**

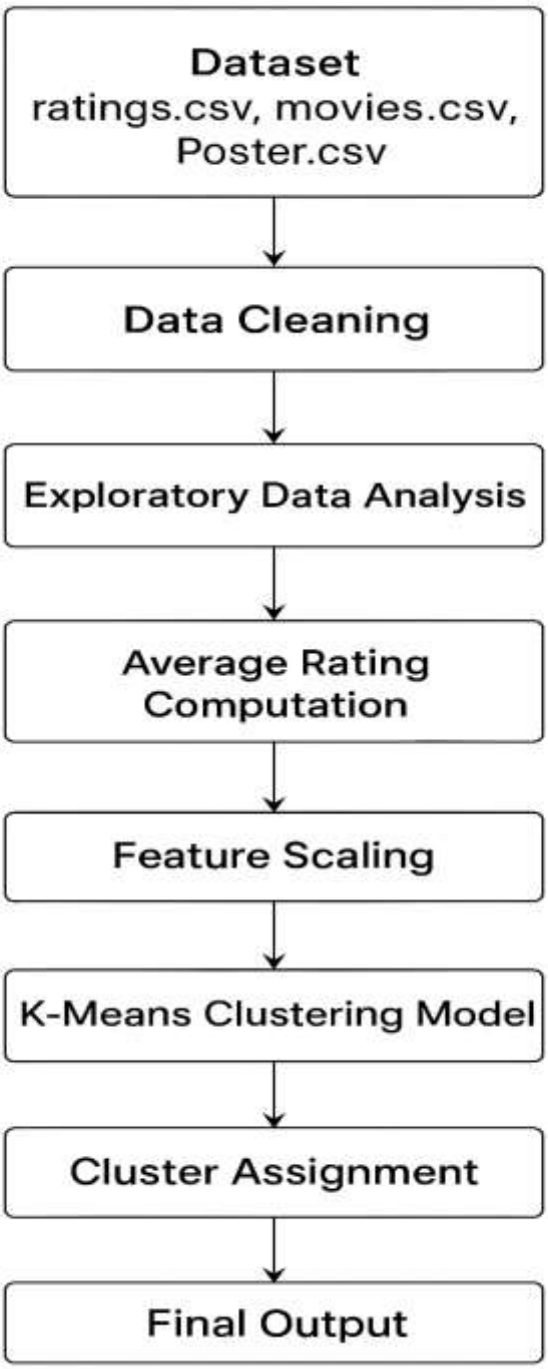
After clustering, the system analyzes movies within the same cluster to identify similar movies and across clusters to identify different popularity groups. This helps in understanding rating patterns and supports future recommendation applications.

**5. Develop a Usable Output for Analysis:**

The final clustered dataset is exported to CSV format including:

- Movie title
- Average rating
- Cluster number
- Poster URL

## 7. MODEL ARCHITECTURE



## 8. SYSTEM DESIGN

The main components of the system are:

**i. Data Loading and Cleaning**

The system loads three datasets:

- ratings.csv
- movies.csv
- poster.csv

Cleaning includes removing missing values, ensuring consistent movieId mapping across files, and removing duplicate entries.

**ii. Exploratory Data Analysis**

The cleaned data is analyzed to study:

- Rating distribution
- Frequency of ratings per movie
- Variation in movie popularity
  Graphs such as bar charts and histograms are used for visualization.

**iii. Feature Extraction (Average Rating Computation)**

The average rating of each movie is calculated by aggregating user ratings. This becomes the primary numerical feature for clustering.

**iv. Feature Scaling**

Since K-Means is distance-based, feature scaling is applied using **Standard Scaler** to normalize rating values.

**v. K-Means Model Training**

The scaled dataset is fed into the K-Means algorithm. The number of clusters k is selected, centroids are initialized, and movies are grouped based on distance similarity.

**vi. Cluster Assignment**

Each movie is assigned a specific **cluster label**. Movies within the same cluster have similar average ratings.

**vii. Visualization Module**

The clustered results are visualized using:

- 1D rating distribution plots
- 2D scatter plots for clustered separation

**viii. Output & Similarity Analysis Module**

The final clustered dataset is exported to CSV format including:

- Movie title
- Average rating
- Cluster number
- Poster URL

This allows identification of similar and dissimilar movies

## 9. IMPLEMENTATION

This chapter explains the step-by-step process followed to implement the movie clustering system based on ratings using machine learning. The implementation involves data collection, preprocessing, feature extraction, model training, visualization, and result generation.

**1. Data Collection**

Three datasets are used in this project:

- ratings.csv – contains user ratings for movies
- movies.csv – contains movie ID and movie titles
- poster.csv – contains movie ID and corresponding poster URLs

These datasets are loaded into the system using Python libraries such as **Pandas** and **NumPy**.

**2. Data Pre-processing and Cleaning**

To ensure high-quality input data, the following cleaning steps are applied:

- Removal of missing values from all datasets
- Elimination of duplicate records
- Validation of movieId across all datasets
- Filtering invalid ratings (outside 0–5 range)
- Merging ratings.csv and movies.csv using movieId
- Adding poster URLs from poster.csv after merging

After cleaning, a single unified dataset is created for analysis.

**3. Exploratory Data Analysis (EDA)**

EDA is performed to understand the structure of the dataset and rating patterns. The following analyses are conducted:

- Distribution of movie ratings
- Count of ratings per movie

- Identification of highly rated and poorly rated movies
- Visualization using histograms and bar charts

EDA helps in understanding trends and prepares the data for clustering.

**4. Average Rating Computation**

For clustering, user-level ratings are converted into movie-level features:

- Ratings are grouped by movieId
- The **average rating** of each movie is computed
- A new dataset is created containing:
    - Movie Title
    - Average Rating
    - Poster URL

This dataset becomes the main input for the clustering model.

**5. Feature Scaling**

Since K-Means clustering is distance-based, feature scaling is essential:

- **StandardScaler** is applied to normalize average ratings
- This ensures all values are on the same scale
- Prevents bias toward higher numerical values

**6. K-Means Clustering Model Training**

The scaled dataset is used to train the **K-Means clustering model**:

- The value of k (number of clusters) is selected
- Initial centroids are randomly selected
- Movies are assigned to the nearest centroid
- Centroids are updated iteratively
- The process continues until convergence

Each movie is assigned a **cluster label** representing its popularity group.

**7. Cluster Assignment and Interpretation**

After the model is trained:

- Each movie is assigned to a specific cluster
- Movies in the same cluster share similar average ratings
- Clusters represent different popularity levels:

- Low-rated movies
- Medium-rated movies
- Highly-rated movies

This helps in understanding grouping behavior.

**8. Visualization of Clusters**

Cluster results are visualized using graphs:

- Scatter plots show cluster separation
- Rating distribution per cluster is displayed
- Visualization helps in interpreting the effectiveness of clustering

**9. Final Output Generation**

The final clustered dataset is generated and exported as a CSV file including:

- Movie Title
- Average Rating
- Cluster Label
- Poster URL

This output can be used for:

- Movie recommendation systems
- Popularity analysis
- Academic research

**10. Tools and Technologies Used**

- Programming Language: Python
- Libraries: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn
- Machine Learning Algorithm: K-Means Clustering
- Dataset Source: MovieLens Dataset
- Development Environment: Jupyter Notebook / Google Colab

## 10. PROPOSED SOLUTION USING ML TECHNIQUES

To solve this problem, we use:

- Unsupervised Machine Learning
- K-Means Clustering Algorithm
- Deep Learning Algorithm

Solution Approach

1. Load movie and rating datasets.
2. Clean the data.
3. Compute average rating for each movie.
4. Standardize the rating values.
5. Apply K-Means clustering.
6. Assign each movie to a cluster.
7. Analyze cluster patterns

## 11. RESULTS

The movie clustering system successfully grouped movies into distinct popularity-based clusters. Movies with high average ratings appeared in the same cluster, while low-rated movies formed separate groups. The clustering output clearly represented different popularity levels of movies. Visualization graphs effectively displayed the separation among clusters. The system also enabled identification of similar movies based on rating behavior.

```
Enter a movie name: Toy Story

Movie Found: Toy Story (1995)
Average Rating: 3.91
Cluster: 0
Poster URL: https://originalvintagemovieposters.com/wp-content/uploads/2020/02/TOY-STORY-9045-scaled.jpg

Movies with Similar Ratings:
```

| title | avg_rating | posters_url_x |
|---|---|---|
| Johnny Mnemonic (1995) | 2.453846 | NaN |
| Candleheads (1993) | 2.592308 | NaN |
| Honey, I Shrunk the Kids (1989) | 2.626667 | NaN |
| Judge Dredd (1995) | 2.640845 | NaN |
| Ace Ventura: When Nature Calls (1995) | 2.642857 | https://tse2.mm.bing.net/th/id/OIP.u0GP7Nf0xWr4V3rSecnwcA&pid=Api&P=0&h=180 |
| Blair Witch Project, The (1999) | 2.650538 | NaN |
| City Slickers II: The Legend of Curly's Gold (1994) | 2.716667 | NaN |
| Cable Guy, The (1996) | 2.784615 | NaN |
| Mission: Impossible II (2000) | 2.793333 | NaN |
| Ace Ventura: Pet Detective (1994) | 2.849711 | NaN |


## 12. PERFORMANCE

- The K-Means algorithm efficiently grouped movies using average rating values.
- Feature scaling significantly improved clustering accuracy.
- The model showed stable and meaningful cluster formation.
- The system handled large datasets effectively without performance degradation

## 13. CONCLUSION

This project presents a complete machine learning-based solution for movie clustering using rating data. It includes data cleaning, exploratory analysis, feature extraction, normalization, model training using K-Means, and result visualization. The model successfully grouped movies into meaningful clusters based on popularity levels. The generated clusters can be used for recommendation systems, content analysis, and business decision-making. The system is simple, scalable, and effective for real-world movie data analysis.

## 14. FUTURE ENHANCEMENTS

- Use additional features such as genres, release year, and reviews for better clustering.
- Implement automated cluster selection using the Elbow Method.
- Integrate a recommendation system based on clusters.
- Develop a web or mobile application interface.
- Apply advanced clustering techniques like DBSCAN or Hierarchical Clustering.