

## VNR Vignana Jyothi Institute of Engineering and Technology (Affiliated to J.N.T.U, Hyderabad)

**Bachupally(v), Hyderabad, Telangana, India.**

**Movie Recommendation System**

A course project on Machine Learning Laboratory submitted incomplete requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**

IN

### Computer Science & Engineering-(AIML & IoT)

Submitted by

B. Priyanka 22071A6609

B. Rama Raju 22071A6610

O. Ajay 22071A6645

G. Tharun 22071A6657

Under the guidance of

## Dr. N. Sandhya Professor

**Department of Computer Science & Engineering-(AIML & IoT)**



## VNR Vignana Jyothi Institute of Engineering and Technology (Affiliated to J.N.T.U, Hyderabad)

**Bachupally(v), Hyderabad, Telangana, India.**

**CERTIFICATE**

This is to certify that B.Priyanka 22071A6609, B.RamaRaju 22071A6610, O.Ajay 22071A6645, and G.Tharun 22071A6657 completed their course project work at Department of Computer Science & Engineering-(AIML&IoT) of VNR VJIET, Hyderabad entitled **Movie Recommendation System** in complete fulfillment of the requirements for the award of B.Tech degree during the academic year 2024-2025. This work is carried out under my supervision and has not been submitted to any other University/Institute for award of any degree.

**Dr.N.Sandhya**

Professor

Dept of CSE-(AIML&IoT)

### Dr.Y.Sagar

Professor and Head of Department CSE- (AIML&IoT)

# DECLARATION

This is to certify that our project report titled **"Movie Recommendation System"** submitted to Vallurupalli Nageswara Rao Institute of Engineering and Technology in complete fulfillment of requirement for the award of Bachelor of Technology in Computer Science and Engineering- (AIML & IoT) is a bonafide report to the work carried out by us under the guidance and supervision of **Dr. N. Sandhya**, Professor, Department of CSE- (AIML & IoT), Vallurupalli Nageswara Rao Institute of Engineering and Technology. To the best of our knowledge, this has not been submitted in any form to other university or institution for the award of any degree or diploma.

|  |  |  |
| --- | --- | --- |
| Roll Number | Name | Signature |
| 22071A6609 | B. Priyanka |  |
| 22071A6610 | B. Rama Raju |  |
| 22071A6645 | O. Ajay |  |
| 22071A6657 | G. Tharun |  |

# ACKNOWLEDGEMENT

Over a span of two years, VNRVJIET has helped us transform ourselves from mere amateurs in the field of Computer Science into skilled engineers capable of handling any given situation in real time. We are highly indebted to the institute for everything that it has given us. We would like to express our gratitude towards the principal of our institute, **Dr. Challa Dhanunjaya Naidu** and the Head of the Department, Computer Science & Engineering-(AIML & IoT), **Dr. Y. Sagar**, for their kind cooperation and encouragement which helped us complete the project in the stipulated time. Although we have spent a lot of time and put in a lot of effort into this project, it would not have been possible without the motivating support and help of our course coordinator of Software Engineering, **Dr. N. Sandhya**, Professor, CSE-(AIML & IoT). We thank her for the guidance, constant supervision and for providing necessary information to complete this project. Our thanks and appreciations also go to all the faculty members, staff members of VNRVJIET, and all our friends who have helped us put this project together.

# ABSTRACT

This project presents a novel approach towards movie recommendation systems by proposing a hybrid model that amalgamates content-based recommendation and collaborative filtering techniques. In contrast to conventional market practices that predominantly rely on either of these methods independently, our model synergistically leverages both methodologies to enhance recommendation accuracy and diversity. The system encompasses four integral stages: data processing, model development, website integration, and deployment. Through meticulous implementation and integration of these stages, our endeavor aims to furnish a robust and efficient movie recommendation platform poised to cater to diverse user preferences and interests.

**Keywords**—Movie recommender system, content-based recommendation, Python, scikit-learn, pandas, NumPy, vectorization, data processing, model development, website integration, deployment.

**INDEX**

|  |  |  |
| --- | --- | --- |
| S.NO | TOPIC | Page No. |
| 01 | Introduction | 07 |
| 02 | Data Set Description | 08 |
| 03 | Literature Review | 09 |
| 04 | Preprocessing Activities | 10 |
| 05 | Implementation | 11-15 |
| 06 | Conclusions | 16 |
| 07 | References | 17 |

**INTRODUCTION**

A movie recommendation system suggests films to users based on their preferences, viewing history, or relevant data, enhancing their movie-watching experience. It employs techniques like collaborative filtering, which identifies patterns among users, and content-based filtering, which analyzes movie attributes such as genre or cast. Hybrid systems combine these approaches for more accurate recommendations. By leveraging user data and advanced algorithms, these systems provide personalized suggestions, minimizing search time and becoming integral to streaming platforms and entertainment services.

### Existing System

A movie recommendation system suggests films to users based on their preferences, viewing history, or relevant data, enhancing their movie-watching experience. It employs techniques like collaborative filtering, which identifies patterns among users, and content-based filtering, which analyzes movie attributes such as genre or cast. Hybrid systems combine these approaches for more accurate recommendations. By leveraging user data and advanced algorithms, these systems provide personalized suggestions, minimizing search time and becoming integral to streaming platforms and entertainment services.

### Proposed System

Hybrid recommendation systems combine both content-based and collaborative filtering techniques to provide more accurate and diverse recommendations. This type of system uses a combination of user data, item data, and other contextual information to generate recommendations. The incorporation of deep learning techniques further enhances recommendation precision by extracting intricate patterns and representations from vast datasets. By utilizing deep neural networks, the system can discern complex relationships between users, items, and contextual features, facilitating more accurate predictions and a richer user experience.

## DATA SET DESCRIPTION

The dataset consists of metadata for 10,000 movies, encompassing various aspects such as genre, language, popularity, and user ratings. It has 9 attributes, each offering unique insights into the movies. This dataset integrates numerical, text, and categorical data, making it suitable for diverse analytical tasks, such as identifying trends in movie popularity, rating distributions, and genre-language relationships. With minimal missing data, it is well-prepared for exploratory data analysis (EDA), predictive modeling, and visualization tasks, offering a comprehensive foundation for understanding audience preferences and movie trends.

**General Overview:**

* **Total Records:** 10,000
* **Total Columns:** 9
* **Memory Usage:** Approximately 703 KB

**Column Descriptions:**

1. **id**: A unique identifier for each movie (integer, no missing values).
2. **title**: The title of the movie (text, no missing values).
3. **genre**: The genre(s) associated with the movie, separated by commas (text, 3 missing values).
4. **original\_language**: The language in which the movie was originally produced (text, no missing values).
5. **overview**: A brief description or summary of the movie's plot (text, 13 missing values).
6. **popularity**: A numeric score indicating the movie's popularity (float, no missing values).
7. **release\_date**: The date the movie was released (date in string format, no missing values).
8. **vote\_average**: The average user rating of the movie on a scale (float, no missing values).
9. **vote\_count**: The number of users who voted for the movie (integer, no missing values).

**Observations:**

* Some columns have minimal missing values (e.g., **genre** and **overview**).
* Key metrics like **popularity**, **vote\_average**, and **vote\_count** provide quantitative insights into user reception and engagement.
* Text fields such as **overview** and **title** offer descriptive information useful for content-based analyses.

**Introduction:**

## LITERATURE SURVEY

This survey explores key advancements in movie recommendation systems, focusing on collaborative filtering, content-based filtering, and hybrid approaches. It highlights methodologies, challenges, and innovations shaping personalized content delivery, providing a concise understanding of the field and future opportunities.

**Key Studies and Methodologies**

#### Trends in content-based recommendation:

**Lops, P., Jannach, D., Musto, C. (2019)**: The focus is on contrasting collaborative filtering and content-based filtering approaches, highlighting their respective reliance on preference patterns in user communities and past preferences of individual users. Additionally, the focus is on discussing the limitations of these approaches and the need for hybrid methods that combine the strengths of both.

#### Recommendation systems: Principles, methods and evaluation:

**Isinkaye, F.O., Folajimi, Y.O., Ojokoh, B.A. (2015):** This paper aims to delve into the diverse prediction techniques employed in recommendation systems, offering a comparative analysis of their unique characteristics and potentials. By doing so, it endeavors to serve as a valuable compass for both researchers and practitioners navigating the complexities of personalized content delivery in the digital age, thus alleviating the burden of information overload experienced by Internet users.

#### Recommender systems: From algorithms to user experience:

**Konstan, J.A., Riedl, J. (2012):** The paper reviews the transformation of recommender systems from algorithm-centric to user-experience-centric approaches. It highlights the impact of embedding algorithms within user interactions, emphasizing the need for broader evaluation metrics. Identifying research gaps and challenges, it aims to enhance the real-world applicability of recommender systems.

#### Movie recommendation system using machine learning:

**F.Furtudo,A.singh(2020)**: The objective of this paper is to develop a recommendation system for movies that reduces the effort required by users to select films based on their interests. By combining content-based and collaborative approaches, the system aims to provide more explicit outcomes and broaden exploration opportunities compared to solely content-based systems.

The survey underscores the shift to hybrid approaches in recommendation systems, enhancing personalization and diversity. Despite progress, challenges like data sparsity and scalability persist, calling for advanced techniques like deep learning and real-time recommendations to further improve system effectiveness.

## PREPROCESSING ACTIVITIES

### Loading the Dataset:

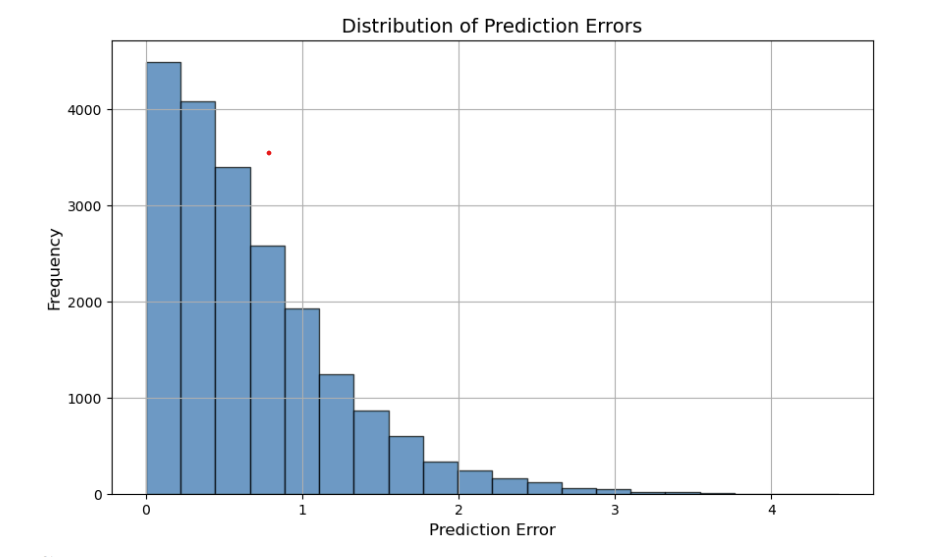
* Importing Libraries
* Reading the dataset

### Data cleaning:

* Identifying duplicates and replacing them with suitable values.

### Data Exploration and Visualization:

* Plotting pair plots
* Correlations between variables using heat maps
* Plotting a box plot to represent rents w.r.to each column



## IMPLEMENTATION

## import pandas as pd

## movies=pd.read\_csv('movies\_dataset.csv')movies.head(5)

## 

## movies.describe()movies.info()movies.isnull().sum()

## 

## movies.columnsmovies=movies[['id', 'title', 'overview', 'genre']]moviesmovies['tags'] = movies['overview']+movies['genre']movies new\_data = movies.drop(columns=['overview', 'genre'])new\_data

## from sklearn.feature\_extraction.text import CountVectorizercv=CountVectorizer(max\_features=10000, stop\_words='english')cv

## vector=cv.fit\_transform(new\_data['tags'].values.astype('U')).toarray()vector.shapefrom sklearn.metrics.pairwise import cosine\_similaritysimilarity=cosine\_similarity(vector)similarity

## 

## new\_data[new\_data['title']=="The Godfather"].index[0]distance = sorted(list(enumerate(similarity[2])), reverse=True, key=lambda vector:vector[1]) for i in distance[0:5]: print(new\_data.iloc[i[0]].title)

## 

## def recommand(movies): index=new\_data[new\_data['title']==movies].index[0] distance = sorted(list(enumerate(similarity[index])), reverse=True, key=lambda vector:vector[1]) for i in distance[0:5]: print(new\_data.iloc[i[0]].title)recommand("Iron Man")import picklepickle.dump(new\_data, open('movies\_list.pkl', 'wb'))pickle.dump(similarity, open('similarity.pkl', 'wb'))pickle.load(open('movies\_list.pkl', 'rb'))

## Python Execution

## import streamlit as st

## import pickle

## import requests

## # Load data

## try:

## df = pickle.load(open("movies\_list.pkl", 'rb'))

## similarity = pickle.load(open("similarity.pkl", 'rb'))

## except FileNotFoundError as e:

## st.write(f"Error loading files: {e}")

## st.stop()

## movies\_list = df['title'].values

## st.header("Movie Recommendation System")

## selectvalue = st.selectbox("Select movie from dropdown", movies\_list)

## def fetch\_poster(movie\_id):

## url = f"https://api.themoviedb.org/3/movie/{movie\_id}?api\_key=YOUR\_API\_KEY"

## data = requests.get(url)

## data = data.json()

## poster\_path = data.get('poster\_path', '')

## full\_path = f"https://image.tmdb.org/t/p/w500/{poster\_path}"

## return full\_path

## def recommend(movie):

## if movie not in df['title'].values:

## st.write("Movie not found in the list.")

## return [], []

## 

## index = df[df['title'] == movie].index[0]

## distance = sorted(list(enumerate(similarity[index])), reverse=True, key=lambda vector: vector[1])

## recommend\_movie = []

## recommend\_poster = []

## for i in distance[1:6]:

## movie\_id = df.iloc[i[0]].id

## recommend\_movie.append(df.iloc[i[0]].title)

## recommend\_poster.append(fetch\_poster(movie\_id))

## return recommend\_movie, recommend\_poster

## if st.button("Show Recommend"):

## movie\_name, movie\_poster = recommend(selectvalue)

## if movie\_name:

## col1, col2, col3, col4, col5 = st.columns(5)

## with col1:

## st.text(movie\_name[0])

## st.image(movie\_poster[0])

## with col2:

## st.text(movie\_name[1])

## st.image(movie\_poster[1])

## with col3:

## st.text(movie\_name[2])

## st.image(movie\_poster[2])

## with col4:

## st.text(movie\_name[3])

## st.image(movie\_poster[3])

## with col5:

## st.text(movie\_name[4])

## st.image(movie\_poster[4])

## 

## OUTPUT:

## 

## CONCLUSION

In this project, Despite its potential, the recommendation system faces notable challenges, starting with the **cold start problem**, where new users or items lack sufficient interaction data, leading to poor recommendation accuracy. Collaborative filtering is particularly affected as it relies on historical user data. Additionally, **data sparsity**, especially in large datasets, complicates the identification of meaningful correlations between users and items, reducing the system's overall effectiveness.

Another major hurdle is **scalability**. As the number of users and items grows, maintaining real-time processing and efficiently handling large datasets becomes increasingly difficult without degrading performance. Furthermore, the system can exhibit **bias in recommendations**, where overrepresented user preferences skew results, limiting the diversity of suggestions. Striking a balance between accuracy and diversity remains a persistent issue.

Finally, **evaluation and feedback** present challenges in refining the system. Accurate user feedback is often hard to obtain, as users may not provide explicit inputs. This limits the system’s ability to adapt and improve over time, emphasizing the need for more innovative ways to capture implicit feedback and enhance recommendation accuracy and personalization.

.

## REFERENCES

* Lops, P., Jannach, D., Musto, C. et al. Trends in content-based recommendation. User Model User-Adap Inter 29, 239–249 (2019). <https://doi.org/10.1007/s11257-019-09231-w>
* F.O. Isinkaye, Y.O. Folajimi, B.A. Ojokoh,Recommendation systems: Principles, methods and evaluation,Egyptian Informatics Journal,Volume 16, Issue 3,2015,Pages 261-273,ISSN 1110-8665,https://doi.org/10.1016/j.eij.2015.06.005.
* Konstan, J.A., Riedl, J. Recommender systems: from algorithms to user experience. User Model User-Adap Inter 22, 101–123 (2012). <https://doi.org/10.1007/s11257-011-9112-x>
* These references provide foundational knowledge on data preprocessing, machine learning techniques, evaluation methods, and insights into class imbalance and ensemble methods, aligning with the methodologies and approaches used in this project.