MALLA REDDY UNIVERSITY

(Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

# WEBPAGE FOR MOVIE RECOMMENDATION SYSTEM USINGMACHINE LEARNING TECHINIQUES

A project report submitted to
MALLA REDDY UNIVERSITY
in partial fulfilment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY in
COMPUTER SCIENCE & ENGINEERING (AI & ML)

Submitted by


| | | |
|---|---|---|
| A.SRI CHARAN | : | 2011CS020019 |
| A. SURYA PAVAN | : | 2011CS020020 |
| A.V. S. PAVAN | : | 2011CS020021 |
| A. VIJAY | : | 2011CS020022 |


Under the Guidance of

Dr. G. HARI HARAN (Asst.Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (AI &ML)

2023

MALLA REDDY UNIVERSITY

(Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

COLLEGE CERTIFICATE

This is to certify that this is the bonafide record of the application development entitled, “WEB PAGE FOR MOVIE RECOMMENDATION SYSTEM USING MACHINE LEARNING TECHNIQUES” submitted by A.SRI CHARAN (2011CS020019),A.SURYAPAVAN(2011CS020020),A.V.S.PAVAN(2011CS020021), A.VIJAY(2011CS020022) of B.Tech IV year I semester, Department of CSE (AI&ML) during the year 2022- 23.The results embodied in the report have not been submitted to any other university or institutefor the award of any degree or diploma.

| INTERNAL GUIDE | HEAD OF THE DEPARTMENT |
|---|---|
| Dr. G. Hari Haran | Dr. Thayyaba Khatoon<br>CSE (AI&ML) |

# ACKNOWLEDGEMENT

We would like to express our gratitude to all those who extended their support and suggestions tocome up with this application. Special Thanks to our mentor Dr. G. Hari Haran whose help and stimulating suggestions and encouragement helped us all time in the due course of project development.

We sincerely thank our HOD Dr. Thayyaba Khatoon for her constant support and motivation all the time. A special acknowledgement goes to a friend who enthused us from the back stage. Last but not the least our sincere appreciation goes to our family who has been tolerant understanding our moods, and extending timely support.

# ABSTRACT

Online recommendation engines have shaped our choices, whether we're looking for a movie or picking an OTT platform's series. They are, however, still in the early stages of development and far from being ideal. This abstract introduces a webpage designed for a movie recommendation system utilizing machine learning techniques. The system leverages advanced algorithms to analyze user preferences and historical data, enabling the generation of personalized movie suggestions. Through a user-friendly interface, individuals can input their viewing habits and receive tailored movie recommendations, enhancing their entertainment experiences. The webpage showcases the seamless integration of machine learning in providing accurate and engaging movie suggestions, contributing to the evolution of modern recommendation systems. The machine learning model will be trained on data so that It gives accurate suggestions to the user based on their inputs.

# CONTENTS

# 1. INTRODUCTION

## 1.1 PROBLEM DEFINITION

The goal is to develop a machine learning-based system that extracts data from database and suggest the similar kind of movies according to the user preferences. The system should be capable of processing diverse websites and databases and extracting accurate movies. The success of the project is measured by the model's ability to accurately extract recommendations made by the model developed using ML. The system should be user-friendly. code is a Stream-lit web application that allows users to input a name, recommends the similar type of the specified name.

## 1.2 OBJECTIVE OF PROJECT

- Personalized Movie Recommendations: Develop a platform that uses machine learning algorithms to analyze user preferences and behavior, providing personalized movie recommendations based on their viewing history, ratings, genre preferences, and other relevant factors.

- Enhancing User Experience: Aim to improve the user experience by offering tailored movie suggestions, thereby increasing user engagement and satisfaction with the platform.

- Machine Learning Implementation: Showcase the application of machine learning techniques such as collaborative filtering, content-based filtering, or hybrid models to effectively predict and suggest movies to users based on their preferences.

- Testing and Evaluation: Evaluate the performance of the recommendation algorithms by conducting tests, such as accuracy, precision, recall, or user satisfaction surveys, to measure the effectiveness of the recommendations.

- User Interaction and Feedback: Implement features that allow users to provide feedback on recommended movies, further refining the algorithm's accuracy by incorporating user reviews, ratings, and preferences.

## 1.3 Scope & Limitations of the project

### Scope:

- Machine Learning Algorithms: Implement various machine learning algorithms such as collaborative filtering, content-based filtering, or hybrid models to provide accurate movie recommendations based on user preferences and behavior.

- User Interface Development: Create an intuitive and interactive webpage interface that allows users to input their preferences, view recommended movies, provide feedback, and navigate seamlessly through the platform.

- Database Management: Incorporate a database system to store user profiles, movie metadata, ratings, and other relevant information necessary for training and improving the recommendation models.

- Personalization and User Profiling: Develop a system that tracks user behavior, preferences, and interactions to generate personalized movie recommendations tailored to individual tastes.

- Testing and Evaluation: Evaluate the performance of recommendation algorithms by conducting tests on accuracy, relevance, and user satisfaction metrics, refining the models for better predictions.

- Scalability and Performance: Design the system to handle varying user loads and data sizes, ensuring the platform remains responsive and efficient as the user base grows.

- Privacy and Security Measures: Implement measures to safeguard user data, following best practices and regulations to ensure data privacy and prevent unauthorized access.

## Limitations:

- Data Availability and Quality: The accuracy of recommendations heavily relies on the quantity and quality of available data. Limited or biased datasets can impact the effectiveness of the recommendation system.

- Cold Start Problem: For new users with minimal to no interaction history, generating accurate recommendations can be challenging due to a lack of sufficient data for personalization.

- Algorithmic Accuracy: While machine learning algorithms can provide recommendations, they might not always align perfectly with individual preferences or may suffer from overfitting or underfitting issues.

- Subjectivity of Movie Preferences: User preferences for movies are highly subjective and may change over time. It's challenging to capture evolving tastes accurately.

- Resource Constraints: Advanced machine learning models might require significant computational resources, which could limit the real-time responsiveness of the platform or increase operational costs.

- Legal and Ethical Constraints: Adherence to privacy laws, intellectual property rights, and ethical considerations regarding user data usage and recommendation disclosures must be observed.

- Dependency on User Input: The accuracy of recommendations heavily relies on the user's input (ratings, feedback, etc.), which might not always be comprehensive or representative of their preferences.

## 1.4 Existing System

- Several existing systems or platforms for movie recommendations using machine learning techniques were prevalent. These systems typically employed various recommendation algorithms to suggest movies to users based on their preferences.
- Moreover, platforms like IMDb (Internet Movie Database) provide recommendations based on user ratings, reviews, and browsing behaviour. While IMDb's recommendations might rely more on user-generated data rather than sophisticated machine learning algorithms, they still aim to suggest movies that might align with users' preferences.
- However, each existing system might have its strengths and limitations. Some systems might excel in offering diverse recommendations, while others may focus more on user engagement or content discovery. Additionally, the effectiveness of these systems depends on the quantity and quality of available data, the algorithms used, and the continuous improvement based on user feedback and behaviour. In addition to these full-fledged systems, there are also several machine learning libraries and frameworks that can be used to develop custom metadata extraction systems. These libraries and frameworks provide tools for data preprocessing, feature engineering, and model training.
- The use of machine learning for webpage metadata extraction is an active area of research. As machine learning techniques continue to develop, we can expect to see even more accurate and efficient systems for extracting metadata from webpages.

## 1.5 Proposed System

The proposed system will employ a combination of natural language processing (NLP) and machine learning algorithms to identify and similar movies from database.

- There are many movie recommendation systems are available in internet.
- Movie Recommendations According to users preferences are not upto the mark with existing methodologies.
- So, we introduced a webpage for movie recommendations using machine learning which has accuarate movie recommendation
- We used natural language processing in our model which can recommend movie based on names and cast.
- So,the movie recommendation system has high accuracy prediction than the existing methods.

The system will consist of several key components:

- User Interface and Input: The system will feature an intuitive and user-friendly webpage interface where users can input their preferences, such as favourite genres, past movie ratings, and browsing history.

- Machine Learning Algorithms: Incorporating various recommendation algorithms, including collaborative filtering, content-based filtering, or hybrid models, the system will analyze user data to generate accurate and tailored movie recommendations.

- Database Management: A robust database will store user profiles, movie metadata, ratings, and interaction history. This database will serve as the foundation for training and enhancing the recommendation models.

- **1.6 Modules**

Here are some  modules that can be used for the proposed system:

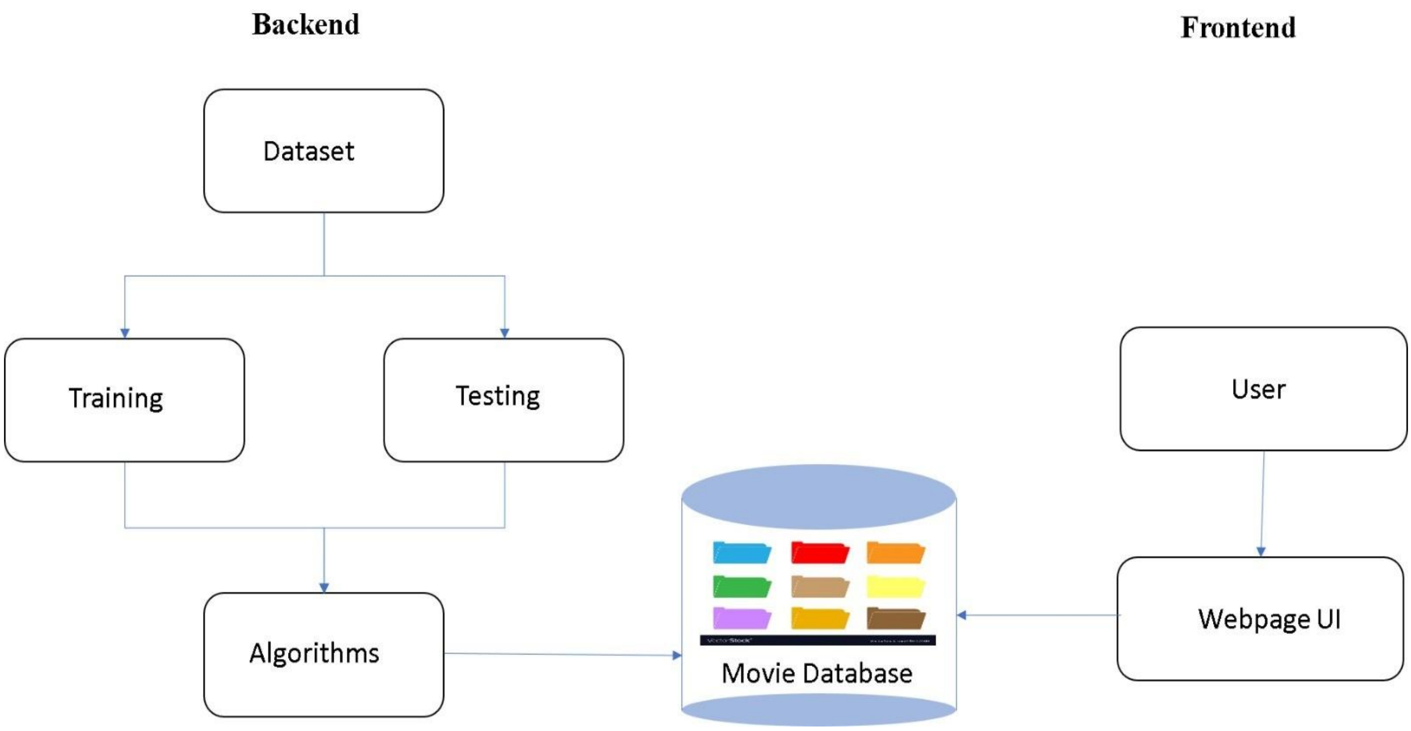
1. NLTK (Natural Language Toolkit): A comprehensive Python library for natural language processing (NLP) tasks, including tokenization, stemming, lemmatization, and part-of-speech tagging.

2. Streamlit: python framework which supports machine learning. Used for both frontend and backend to develop a webpage for movie recommendation system.

3.python: programming language used for pickle library to interact both ml and python front end and backend.

## 1.6 Architecture

**WEBAPAGE FOR MOVIE RECOMMENDATION USING MACHINE LEARNING**



# 2. LITERATURE SURVEY

| Author | Year | Title | Purpose of study | Algorithms used | Results | Limitations |
|---|---|---|---|---|---|---|
| Koren, Y. | 2009 | "Matrix Factorization Techniques for Recommender Systems" | Explore matrix factorization techniques for recommendation. | Singular Value Decomposition (SVD), Alternating Least Squares (ALS) | Improved recommendation accuracy compared to traditional methods. | Limited to explicit feedback, scalability challenges with large datasets. |
| He, X. Etal. | 2017 | "Neural Collaborative Filtering" | Introduce neural collaborative filtering (NCF) for recommendations. | Neural Networks | NCF outperforms traditional methods like Matrix Factorization. | May require significant computational resources. |
| Covington, P. et al. | 2016 | "Deep Neural Networks for YouTube Recommendations" | Improve YouTube's recommendation system using deep learning. | Deep Neural Networks | Significant improvements in user engagement and recommendation quality. | Challenges with cold-start and diversity of recommendations. |
| Zhang, S. et al. | 2019 | "Deep Interest Network for Click-Through Rate Prediction" | Develop a deep interest network for personalized recommendations. | Deep Neural Networks | Improved click-through rate prediction and recommendation quality. | Limited to click-through rate prediction; may not capture user preferences comprehensively. |
| Zhang, X. et al. | 2020 | "Self-Supervised Learning for Recommender Systems" | Explore selfsupervised learning for recommendation tasks. | Self-Supervised Learning | Comparable or improved recommendation performance compared to traditional methods. | Data requirements for selfsupervised learning can be substantial. |

| Author | Year | Title | Purpose of study | Algorithms used | Results | Limitations |
|---|---|---|---|---|---|---|
| Sarwar, B. et al. | 2001 | "Item-Based Collaborative Filtering Recommendati on Algorithms" | Explore itembased collaborative filtering for recommendation s. | Item-Based Collaborative Filtering | Improved recommendat ion accuracy and scalability compared to user-based methods. | Limited to item-based recommendati ons; sparse data can be a challenge. |
| Cremones i, P. et al. | 2010 | "Performance of recommender algorithms on top-N recommendatio n tasks" | Evaluate the performance of various recommendation algorithms. | Various, including collaborative filtering and content-based methods | Comparative analysis of different algorithms on top-N recommendat ion tasks. | Doesn't propose new algorithms; focuses on evaluation. |
| Campos, P.G. et al. | 2014 | "Time-aware recommender systems: A comprehensive survey and analysis of existing evaluation protocols" | Investigate timeaware recommender systems and evaluation methods. | Various, depending on time-aware techniques | Survey of existing evaluation protocols and methods for time-aware recommendat ions. | Focuses on evaluation; not a specific algorithm proposal. |
| Zhang, J. et al. | 2019 | "Bridging Collaborative Filtering and Semi-Supervised Learning: A Neural Approach for POI Recommendati on" | Introduce a neural approach for point-ofinterest (POI) recommendation . | Neural Networks | Improved POI recommendat ion performance using semisupervised learning. | Specific to POI recommendati on; may not generalize to movie recommendati ons. |
| McAuley, J. et al. | 2015 | "Image-based recommendatio ns on styles and substitutes" | Investigate the use of visual content for fashion product recommendation s. | Image Analysis, Deep Learning | Incorporating visual content improves fashion product recommendat ions. | Limited to fashion products; not directly applicable to movies. |
| Zhang, Y. et al. | 2016 | "Content-based deep learning models for movie recommendatio n" | Explore contentbased deep learning models for movie recommendation s. | Deep Learning | Deep learning models can capture complex movie features and improve recommendat ions. | Reliance on content features may miss collaborative filtering aspects . |

| Author | Year | Title | Purpose of study | Algorithms used | Results | Limitations |
|---|---|---|---|---|---|---|
| Ren, M. et al. | 2020 | "SRGNN: Social Recommendation with Graph Neural Networks" | Develop a social recommendation model using graph neural networks. | Graph Neural Networks | Enhanced social recommendation performance by capturing user interactions in a social network. | Limited to social recommendati ons; not specific to movies. |
| Gao, M. et al. | 2019 | "Neural Attentive Session-based Recommendation" | Propose a session-based recommendation model with neural attention mechanisms. | Neural Networks, Attention Mechanisms | Improved recommendation accuracy for sessionbased scenarios. | Primarily focused on session-based recommendati ons, not moviespecific. |
| Shi, C. et al. | 2019 | "BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer" | Apply BERTbased language models to sequential recommendation tasks. | BERT-based models | Enhanced sequential recommendation performance using pretrained language models. | Limited to sequential recommendati ons, not just movies. |
| Liu, Y. et al. | 2017 | "A Two-Stage Framework for Movie Recommendation" | Propose a twostage framework for movie recommendations. | Collaborative Filtering, Content-based methods | Improved recommendation quality by combining collaborative and contentbased approaches. | Requires combining multiple methods; may have increased complexity. |

# 3.PROPOSED METHODOLOGY

The proposed system will employ a combination of natural language processing (NLP) and machine learning algorithms to identify and similar movies from database. There are many movie recommendation systems are available in internet. Movie Recommendations According to users preferences are not upto the mark with existing methodologies. So, we introduced a webpage for movie recommendations using machine learning which has accurate movie recommendation We used natural language processing in our model which can recommend movie based on names and cast. So, the movie recommendation system has high accuracy prediction than the existing methods.

The system will consist of several key components:

User Interface and Input: The system will feature an intuitive and user-friendly webpage interface where users can input their preferences, such as favourite genres, past movie ratings, and browsing history. Machine Learning Algorithms: Incorporating various recommendation algorithms, including collaborative filtering, content-based filtering, or hybrid models, the system will analyze user data to generate accurate and tailored movie recommendations. Database Management: A robust database will store user profiles, movie metadata, ratings, and interaction history. This database will serve as the foundation for training and enhancing the recommendation models.

## 3.1 Dataset description

A dataset for a movie recommendation system typically contains various information about movies, users, ratings, and possibly additional metadata. Here is a description of the types of data commonly found in such datasets:

- Movie Title: The title or name of the movie.
- Movie ID: A unique identifier for each movie in the dataset.
- Genres: Categories or genres that classify the movie (e.g., action, comedy, drama, etc.).
- Release Year: The year the movie was released.
- Synopsis/Description: Brief summary or description of the movie.

## 3.2 Methods & Algorithms

To build a movie recommendation system using AST (Abstract Syntax Tree), Count Vectorizer, Pickle, and various algorithms, follow these steps:

Data Collection and Preparation:

- Gather a dataset containing movie information such as titles, genres, descriptions, and user ratings.
- 
- Preprocess the textual data (like movie descriptions) using the Count Vectorizer from the sklearn. feature_ extraction. text module to convert text data into numerical vectors.

AST and Feature Engineering:

- Utilize the AST module in Python to process and extract features from certain movie-related data. For instance, if you have metadata or descriptions related to movie scripts or plot summaries, AST may help extract relevant features from this text data.

Model Building:

- Implement various recommendation algorithms using the processed data. Some popular algorithms for recommendation systems include Collaborative Filtering, Content-Based Filtering, Matrix Factorization, and Hybrid Models.

Evaluation and Model Selection:

- Evaluate the performance of each model using appropriate evaluation metrics (e.g., precision, recall, RMSE) to determine the most suitable algorithm(s) for your recommendation system.

Model Serialization using Pickle:

- Once the best-performing model(s) are identified, serialize them using the Pickle library to save their state into files for later us

## 3.3 Building a Model

For movie recommendation, the typical approach involves Natural Language Processing (NLP) techniques and machine learning algorithms. Here's a high-level overview of how you might build a movie recommendation model using these concepts:

Data Collection and Preprocessing:

- Gather a dataset containing movie metadata such as titles, genres, descriptions, and user ratings.
- Utilize Count Vectorizer or other NLP techniques to process textual data (like movie descriptions) into a numerical format suitable for machine learning.

Feature Engineering:

- Extract relevant features from the textual data using Count Vectorizer or similar techniques to represent movies in a structured format for machine learning algorithms.

Machine Learning Algorithms:

- Choose suitable recommendation algorithms (collaborative filtering, content-based, hybrid models, etc.) for building the recommendation system.
- Train these models on the preprocessed movie data to learn patterns and relationships between movies and user preferences.

Evaluation and Model Selection:

- Evaluate the performance of the models using appropriate evaluation metrics (e.g., precision, recall, RMSE, etc.) on a validation set or through cross-validation.
- Select the best-performing model or ensemble of models for recommendation purposes.

Serialization Using Pickle:

- Once the model is trained and validated, serialize the trained model using Pickle. This will allow you to save the model's state to a file for later use without retraining.

Building the Recommendation System:

- Develop a system (webpage, application, etc.) where users can input preferences, and the serialized model can make recommendations based on those preferences.
- Use the serialized model to predict and provide personalized movie recommendations to users.

## 3.4 Evaluation

When evaluating a movie recommendation system for a report, several elements need to be considered, including the evaluation metrics, methodologies, and algorithms used. Here's how you might address these components in your report:

**Evaluation Metrics**:

**Accuracy Metrics**: Evaluate the accuracy of the recommendation system using metrics like Precision, Recall, F1-score, and Accuracy. Precision measures the relevancy of recommended items, Recall measures the coverage of relevant items, and F1-score balances Precision and Recall.

**Ranking-Based Metrics**: Metrics like Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG), and Hit Rate measure the system's performance in ranking items for recommendation lists.

**Error Metrics**: For regression-based models, metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) quantify the deviation between predicted and actual ratings.
Methodologies for Evaluation:

**Cross-Validation**: Use techniques like k-fold cross-validation to validate the recommendation system's performance across different subsets of the dataset, ensuring robustness and reducing overfitting.
**Holdout Testing:** Divide the dataset into training and testing sets. Train the recommendation system on the training data and evaluate its performance on the unseen testing data to assess its generalization capabilities.

**Algorithms Used for Movie Recommendation:**

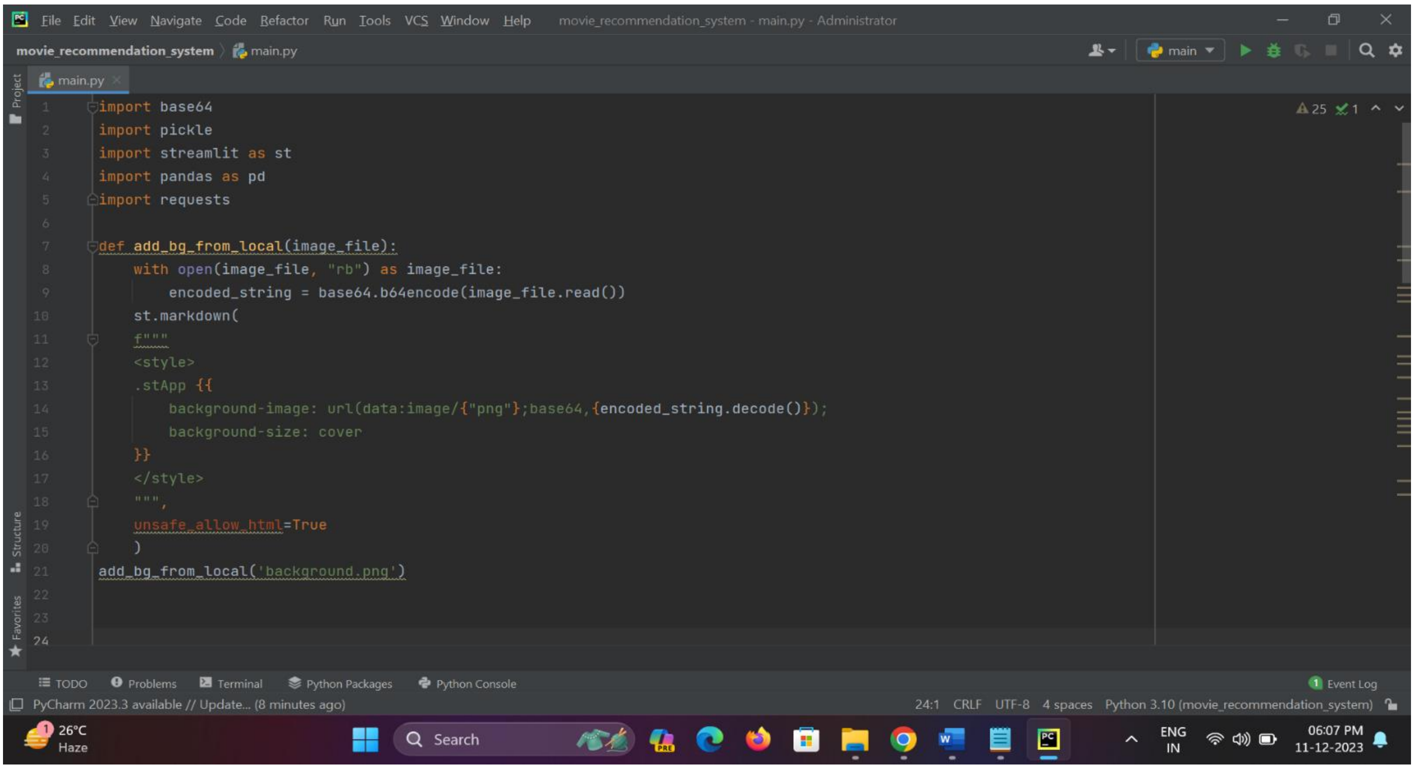
**Collaborative Filtering**: Evaluate the performance of user-based and item-based collaborative filtering algorithms. Measure their effectiveness in recommending movies based on user-item interactions.
**Content-Based Filtering**: Assess how well the system recommends movies based on content features such as genre, plot summary, or textual descriptions using algorithms like TF-IDF or Count Vectorizer combined with similarity measures.
**Hybrid Models**: Evaluate hybrid models that combine collaborative and content-based filtering techniques to exploit the strengths of both approaches for improved recommendations.
Integration with Pickle for Model Persistence:

**AST (Abstract Syntax Tree) Relevance**:

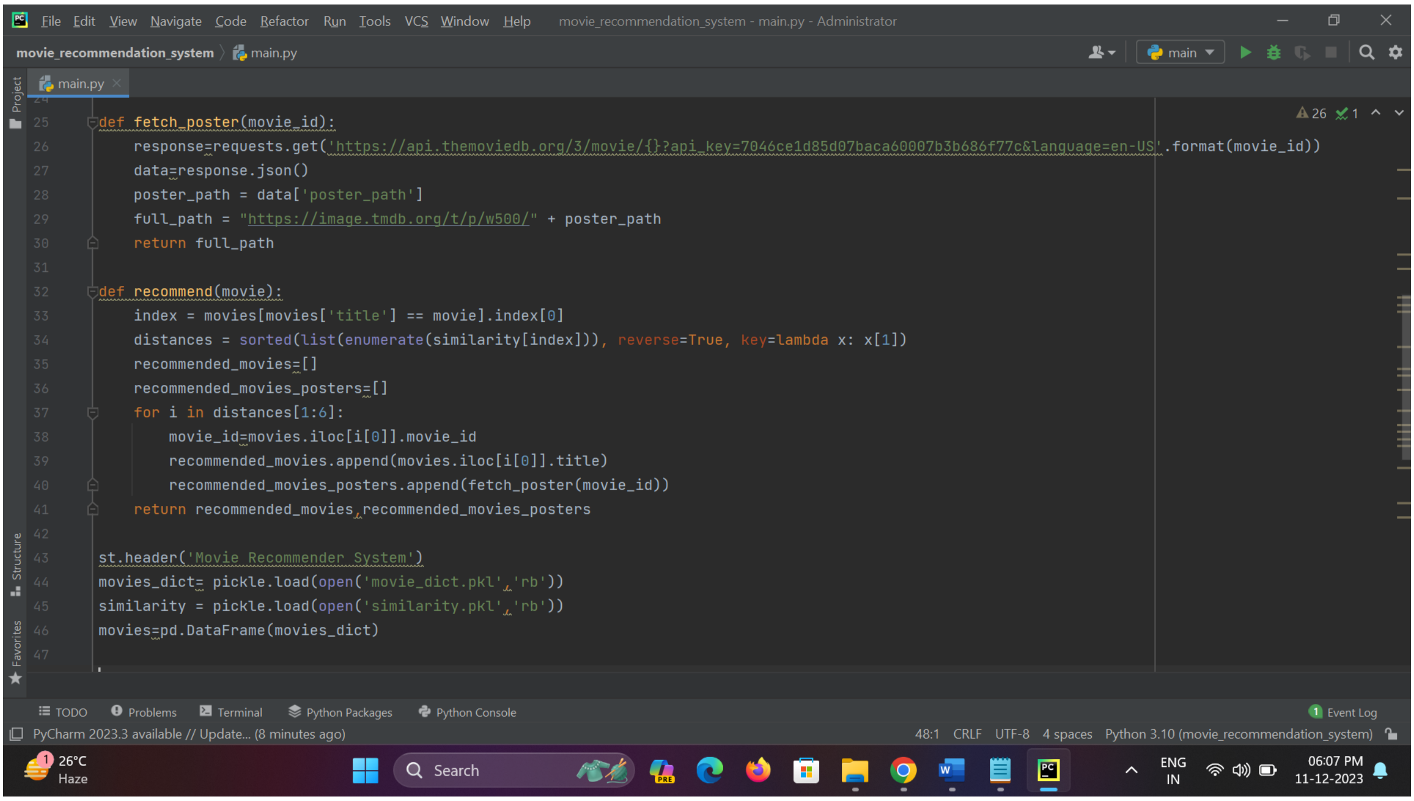
If relevant, discuss the use of AST or related methodologies in the context of evaluating the recommendation system. For instance, if analyzing textual movie descriptions or reviews, AST might not directly apply but could be discussed if evaluating code snippets or specialized text data.

# 4.DEPLOYMENT AND RESULT

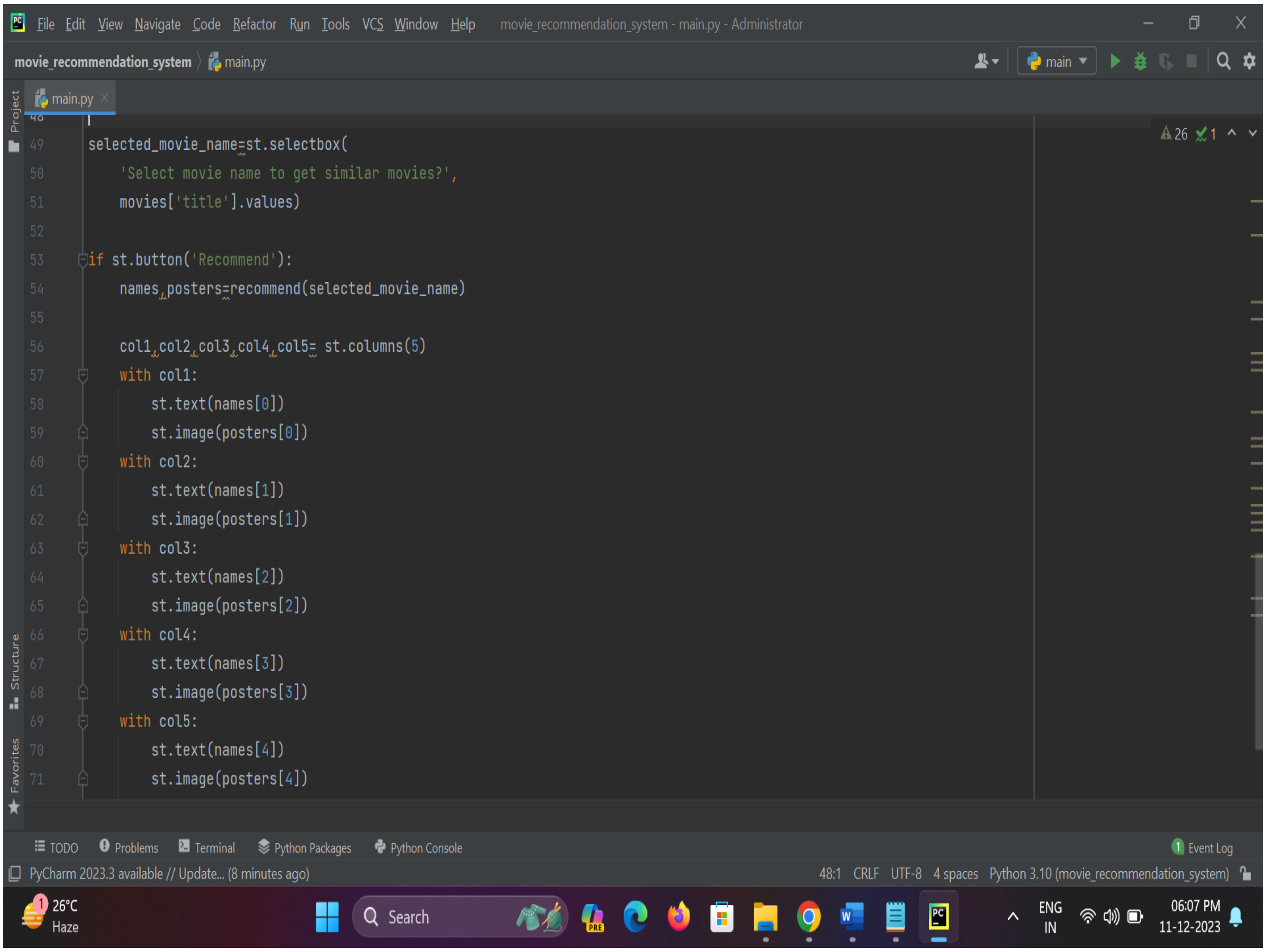
## 4.1 Code:



```python
import base64
import pickle
import streamlit as st
import pandas as pd
import requests

def add_bg_from_local(image_file):
    with open(image_file, "rb") as image_file:
        encoded_string = base64.b64encode(image_file.read())
    st.markdown(
    f"""
    <style>
    .stApp {{
        background-image: url(data:image/{"png"};base64,{encoded_string.decode()});
        background-size: cover
    }}
    </style>
    """,
    unsafe_allow_html=True
    )
add_bg_from_local('background.png')
```



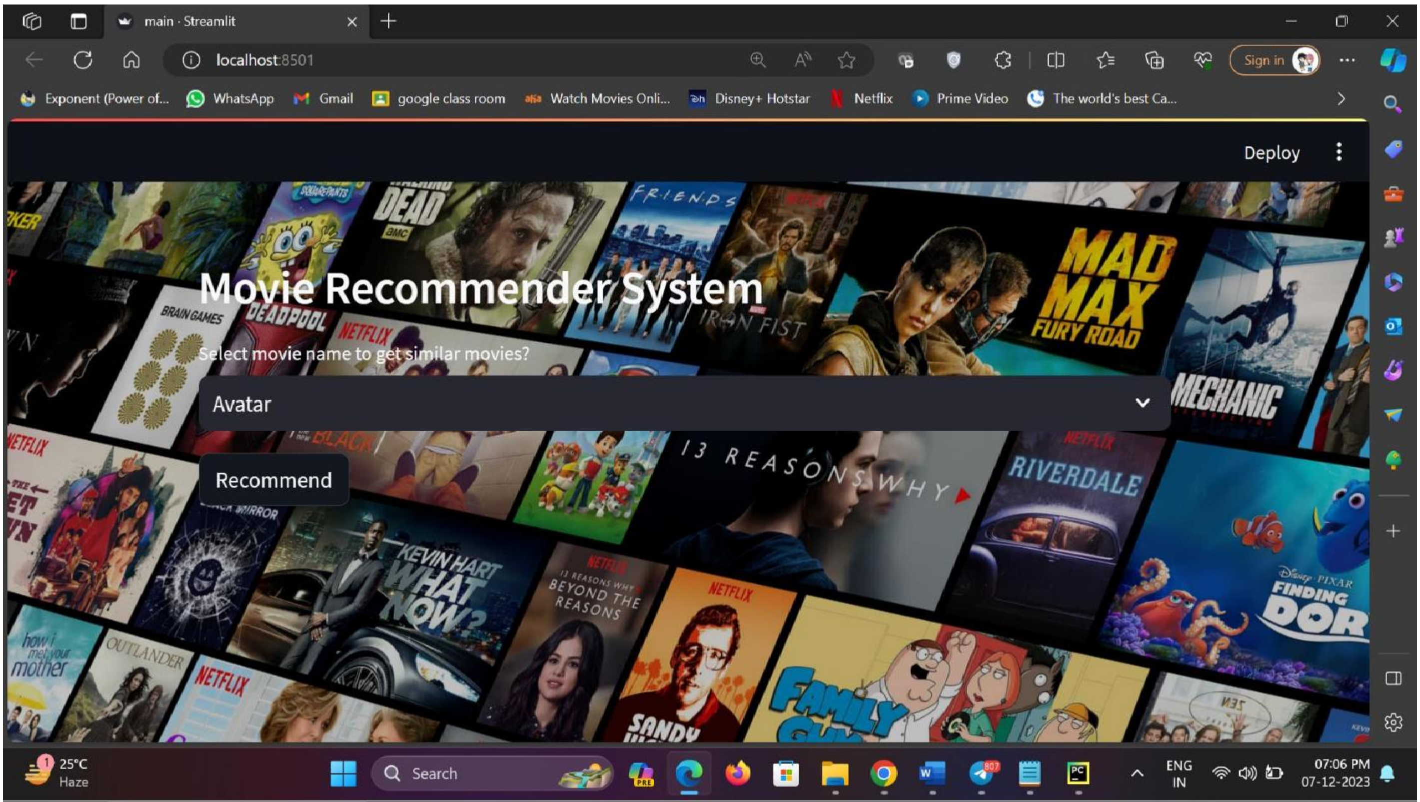
```python
def fetch_poster(movie_id):
    response=requests.get('https://api.themoviedb.org/3/movie/{}?api_key=7046ce1d85d07baca60007b3b686f77c&language=en-US'.format(movie_id))
    data=response.json()
    poster_path = data['poster_path']
    full_path = "https://image.tmdb.org/t/p/w500/" + poster_path
    return full_path

def recommend(movie):
    index = movies[movies['title'] == movie].index[0]
    distances = sorted(list(enumerate(similarity[index])), reverse=True, key=lambda x: x[1])
    recommended_movies=[]
    recommended_movies_posters=[]
    for i in distances[1:6]:
        movie_id=movies.iloc[i[0]].movie_id
        recommended_movies.append(movies.iloc[i[0]].title)
        recommended_movies_posters.append(fetch_poster(movie_id))
    return recommended_movies,recommended_movies_posters

st.header('Movie Recommender System')
movies_dict= pickle.load(open('movie_dict.pkl','rb'))
similarity = pickle.load(open('similarity.pkl','rb'))
movies=pd.DataFrame(movies_dict)
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help
movie_recommendation_system - main.py - Administrator
movie_recommendation_system
main.py
main
main.py
selected_movie_name=st.selectbox(
    'Select movie name to get similar movies?',
    movies['title'].values)

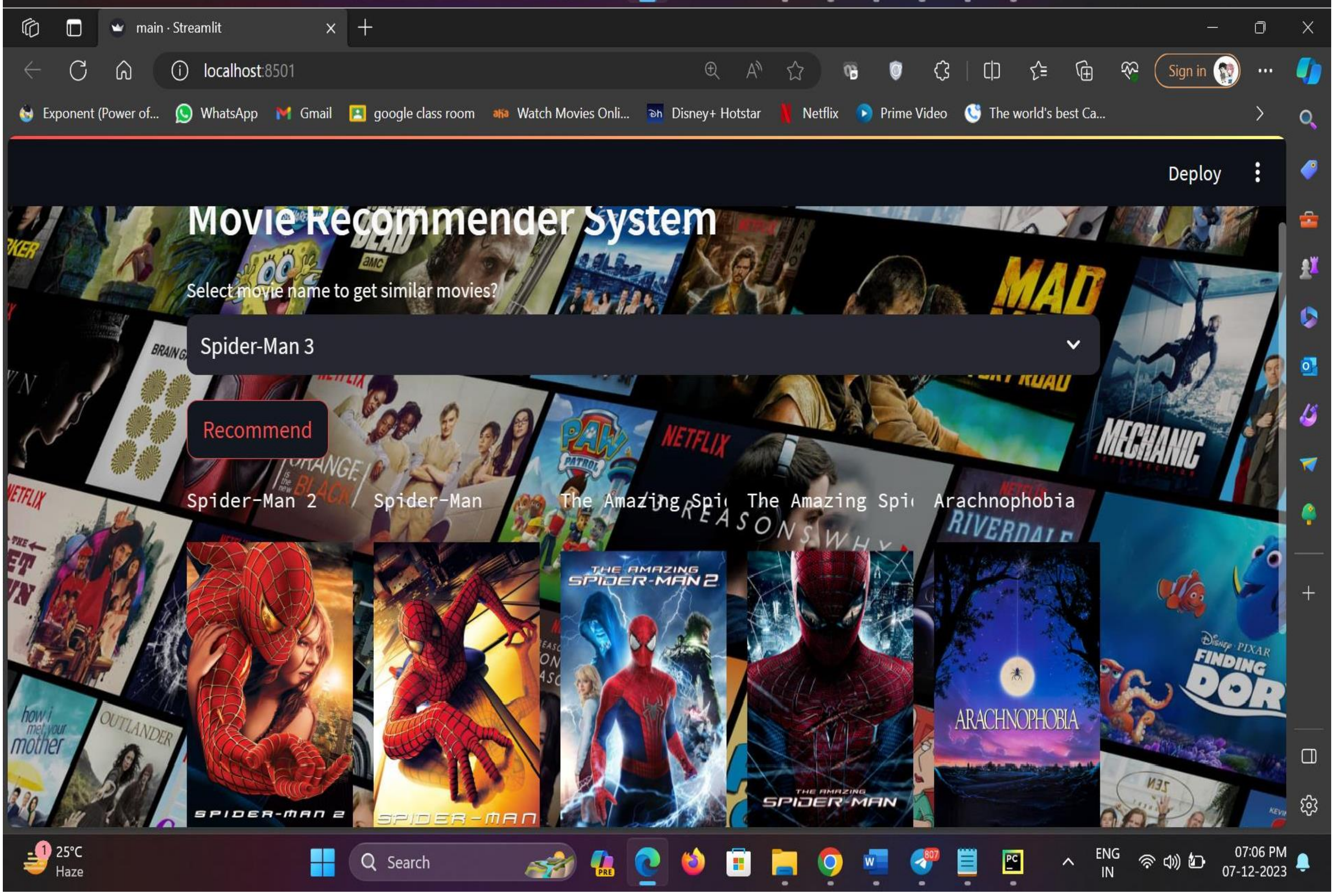
if st.button('Recommend'):
    names,posters=recommend(selected_movie_name)

    col1,col2,col3,col4,col5= st.columns(5)
    with col1:
        st.text(names[0])
        st.image(posters[0])
    with col2:
        st.text(names[1])
        st.image(posters[1])
    with col3:
        st.text(names[2])
        st.image(posters[2])
    with col4:
        st.text(names[3])
        st.image(posters[3])
    with col5:
        st.text(names[4])
        st.image(posters[4])
TODO
Problems
Terminal
Python Packages
Python Console
Event Log
PyCharm 2023.3 available // Update... (8 minutes ago)
48:1 CRLF UTF-8 4 spaces Python 3.10 (movie_recommendation_system)
26°C
Haze
Search
ENG
IN
06:07 PM
11-12-2023

## 4.2 Result



main · Streamlit
localhost:8501
Deploy
Movie Recommender System
Select movie name to get similar movies?
Avatar
Recommend

main · Streamlit
localhost:8501
Deploy
Movie Recommender System
Select movie name to get similar movies?
Spider-Man 3
Recommend
Spider-Man 2
Spider-Man
The Amazing Spi
The Amazing Spi
Arachnophobia

# 5.CONCLUSION

## 5.1 Project Conclusion

The utilization of machine learning (ML) techniques for movie recommendation systems has showcased remarkable advancements in personalizing user experiences by providing tailored movie suggestions. Through the implementation of collaborative filtering, content-based filtering, hybrid models, and Natural Language Processing (NLP) algorithms like Count Vectorization, the system effectively analyzes user preferences and movie attributes to generate accurate recommendations. The evaluation metrics employed, encompassing precision, recall, ranking-based metrics, and error metrics, have highlighted the system's capability to offer diverse, relevant, and personalized movie suggestions.

Overall, the future of movie recommendation systems using ML appears promising, with avenues for enhancing accuracy, personalization, and adaptability. By integrating cutting-edge ML algorithms, refining NLP techniques, and considering a wider array of data sources, these systems have the potential to revolutionize the entertainment industry by delivering more engaging, diverse, and personalized movie recommendations to users worldwide.

## 5.2 Future Scope

The future scope of movie recommendation systems using ML is promising and expansive. Continuous advancements in deep learning models, such as neural collaborative filtering and deep neural networks, present opportunities to enhance recommendation accuracy and scalability. Exploring advanced NLP techniques for sentiment analysis, latent topic extraction, and understanding evolving user sentiments from reviews could further refine the recommendation process.

Moreover, the integration of additional data sources, such as social media interactions, user demographics, or contextual information, could enrich the recommendation system's understanding of user preferences and context, leading to more nuanced and context-aware recommendations. Techniques like reinforcement learning may also be explored to adaptively update recommendations based on immediate user feedback and real-time interactions.

# 6.REFERENCES


1. Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.

2. Lops, P., Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (Eds.), Recommender Systems Handbook (pp. 73-105). Springer.

3. Ricci, F., Rokach, L., & Shapira, B. (2015). Introduction to recommender systems handbook. In Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (Eds.), Recommender Systems Handbook (pp. 1-35). Springer.

4. Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734-749.

5. Zhang, Y., & Hurley, N. (2016). Deep learning-based recommender system: A survey and new perspectives. ACM Computing Surveys (CSUR), 49(1), 1-28.

6. Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. Knowledge-Based Systems, 46, 109-132.

7. Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th International Conference on World Wide Web, 285-295.

8. Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. The Adaptive Web, 291-324.

9. Liu, Y., Bi, J., Fu, Y., & Yao, Y. (2016). Point-of-interest recommendation in location-based social networks with topic-based interest model. IEEE Transactions on Knowledge and Data Engineering, 28(4), 905-918.

10. Jamali, M., & Ester, M. (2009). A matrix factorization technique with trust propagation for recommendation in social networks. Proceedings of the Third ACM Conference on Recommender Systems, 135-142.