IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A MOVIE RECOMMENDATION SYSTEM

Integration of Machine Learning

[1]AYUSH PARADKE, [2]VURNALI SURVE, [3]SWARA REDIJ, [4]DEEPAK PATIL,

[5]SHASHANK TOLYE

Department of Information Technology, Finolex Academy Of Management And Technology, Ratnagiri, India

Abstract: This paper is about a movie recommendation system that uses machine learning to suggest movies to people. The system is made up of a few parts, including content-based filtering, collaborative filtering and emotion-based filtering, which all work together to give users personalized movie suggestions. To make this system we first took a dataset and cleaned it up and organized it. Then we used something called feature extraction to make special matrices that show how similar things are to each other. We did all of this work in something called Jupyter Notebook and then we took the data we had prepared and put it into a web application made with Streamlit so people can use it in time. When people use this application they can pick a movie they like. Say how they are feeling and the system will suggest other movies that are similar or that might match their mood. Content-based filtering is when the system suggests movies that're a lot like the one the person picked and collaborative filtering is when the system looks at what other people with similar tastes have liked. We also have emotion-based filtering, which suggests movies that fit the persons mood, like happy or sad movies. The system can also get information about movies from outside like what people think of them what kind of movie it's who is in it and what the poster looks like which makes the experience better for the user. Our system does a job of solving some big problems like making sure the movies are really personalized and that the information is always up to date. It shows that using ways of recommending movies and putting them into a web application that people can interact with can make the whole system work better and be more useful. The results we got from testing the system were good. People seemed to like the suggestions it made. This work is important because it shows how machine learning and web technology can be used together to make things that're useful and work well. The movie recommendation system is an example of this and it can be used in other areas too.The system is made to be practical and to work for a lot of people. It can be used to make other kinds of recommendation systems too.This is a deal because it can help people find things they like and make their lives more fun and interesting. The movie recommendation system is a start and it can be used to make even more things, like it.

Keywords: Movie recommendation system, machine learning, hybrid recommendation, content-based filtering, collaborative filtering, personalized recommendation, Python, Streamlit, Scikit-learn, TMDb API.

I. INTRODUCTION

The way people find and watch movies has changed a lot because of entertainment platforms. There are many movies out there across different types, languages and platforms that it can be hard for people to pick something they will like. This is why movie recommendation systems were created to help people find movies that're just right for them.

A movie recommendation system is a tool that uses machine learning to look at what people like how they behave and what movies are about to suggest movies that people will probably like. These systems make it easier for people to find movies. They also keep people interested by saving them time searching for something good. In this project we made a movie recommendation system that combines ways of filtering movies and gives people recommendations in real time through a website. Our system uses a mix of looking at the content of movies what other people like and how people are feeling to recommend movies. The content-based approach suggests movies that're similar to the one you picked by looking at things like tags and information about the movie. The collaborative filtering approach makes recommendations better by finding relationships between movies based on how people interact with them. We also added a feature that lets people get recommendations based on their mood like if they're feeling happy, sad, romantic or adventurous. We built the machine learning models. Figured out how similar movies are during the development

phase using a tool called Jupyter Notebook. We cleaned up the data. Pulled out the important parts and then we stored and loaded them into the main application. The front end of the system is built using Streamlit, which makes it easy for people to use and interact with. People can pick a movie and their mood and the system gives them recommendations away. Our application also connects to a movie database to get up-to-date information, like movie posters, ratings, types, cast, crew and descriptions. This makes the system more fun and useful for people. The main goal of this project is to make a movie recommendation system that's easy to use and gives people good suggestions by combining different machine learning techniques. By putting together data processing, machine learning models and a website the system shows how recommendation systems can be used in the entertainment industry.

II. OBJECTIVES

The main objectives of this project are as follows:

- We want to make a movie recommendation system that's smart and suggests movies to users that they will like based on what they are interested in and what they like.
- We will use a technique called content-based filtering that looks at things like movie tags and information to suggest movies.
- We will also use filtering to find relationships between movies and make our recommendations better by using special tables that show how similar movies are to each other.
- We also want to add a feature that recommends movies based on how the user's feeling like if they are happy, sad, in love or looking for an adventure.
- We need to get the movie data ready and pull out the information so we can build good machine learning models that work well.
- We will make a web application using Streamlit that can recommend movies in time.
- We will connect to systems to get the latest movie information like pictures, ratings, who is in the movie and what it is, about.
- We have to see how well our movie recommendation system is working by looking at how good the moviesre that we suggest.
- We want to make sure users have a time and keep using our system by giving them fast, good and easy to look at recommendations.

III. Related work:

Movie recommendation systems are really popular in the field of machine learning and data science. This is because they are used a lot in streaming platforms. Over the years people have come up with different ways to make these systems better. One way that is used a lot is called filtering. This method looks at what users do and what they like. Then suggests movies based on that. For example Breese and his team showed in 1998 that we can use the things users do and the items they interact with to suggest movies they might like. This approach has some problems, like when we do not have enough information about a user. This is called the cold start problem. Another method is called content-based filtering. This method suggests items that're similar to the ones a user liked before. It looks at the features of the items like the type of movie or the keywords. Panniello and his team found a way to make this method better in 2014.They used something called similarity, which made the suggestions more accurate. This method

is really useful when we do not know much about a user. It can suggest the same types of things over and over. To make these systems better people have started combining methods. Adomavicius and Tuzhilin found in 2005 that using content-based methods together makes the system work better. This is because it uses the things about both methods and makes the suggestions more balanced and accurate. The system we are proposing also uses this approach. Another method that is used a lot is called matrix factorization. Koren and his team showed in 2009 that this method makes the suggestions more accurate especially when we have a lot of data. Some researchers have also looked at methods that are based on how similar Things're For example Sun and his team found in 2018 that using something called cosine similarity with methods makes the suggestions better. Nowadays people are also looking at systems that take into account the context and the emotions of the user. This means that the system tries to understand how the user is feeling and suggests movies based on that. This makes the system more personal and interactive. Based on what we know from research it seems that using multiple methods together makes the system work better. So our system uses a combination of content-based filtering, filtering and emotion-based recommendation to suggest movies. This way we can provide suggestions that're more accurate, diverse and personal. Movie recommendation systems are getting better and better. The proposed system is a movie recommendation system that uses methods to suggest movies. It looks at the content of the movies what the users do and how they feel to make the suggestions. This makes the system really useful for users who want to find movies to watch. The movie recommendation system is a tool for online streaming platforms. It helps users find movies they will like. It makes the platform more fun to use. The system is always. Getting better so it will keep suggesting great movies. Movie recommendation systems, like this one are the future of streaming.

IV. System Architecture

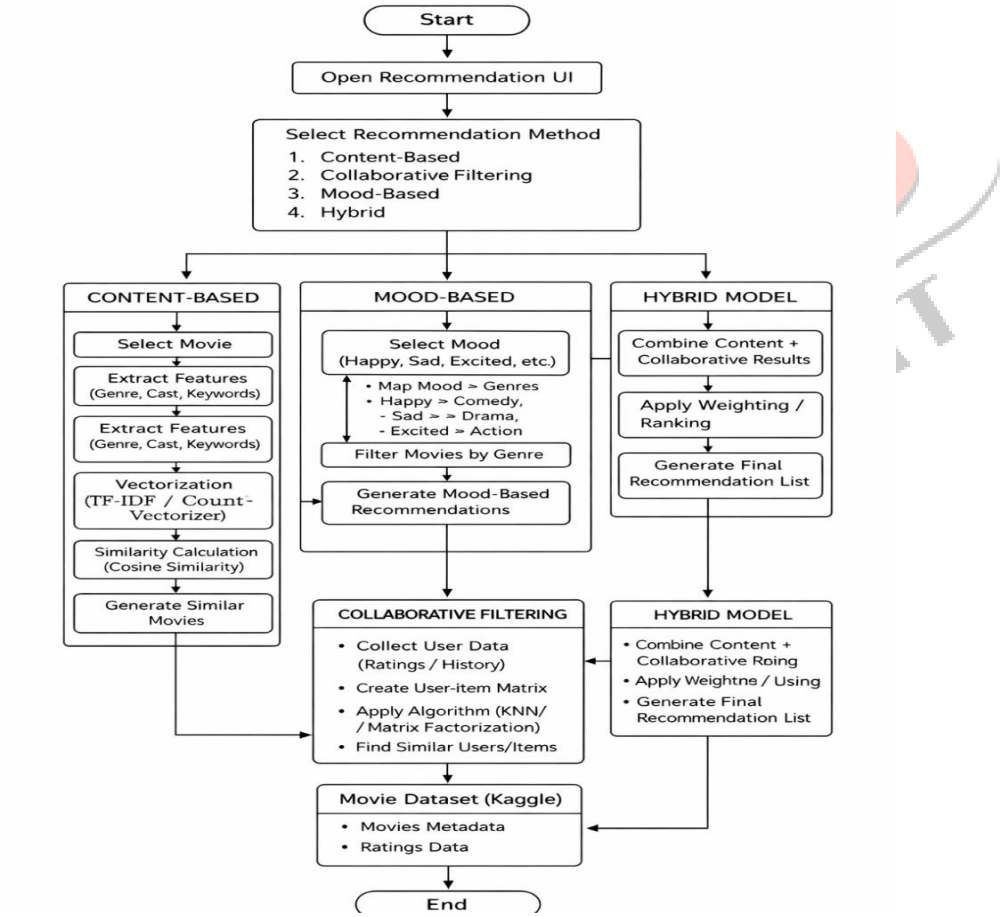


Fig.1.1 System Architecture

V. Methodology

The proposed system aims to monitor air quality in real time and predict potential health risks using IoT and machine learning techniques. The methodology consists of multiple stages, including data collection, preprocessing, model training, prediction, and visualization. The system ensures accurate air quality monitoring by integrating sensor data with advanced machine learning models.

5.1 Data Collection and Dataset Preparation

The data used in this system is obtained from two sources: real-time sensor data and a publicly available dataset from Kaggle. The hardware setup includes sensors such as PM2.5, MQ135, MQ7, and temperature sensors connected to an ESP32 microcontroller. These sensors continuously collect environmental parameters such as particulate matter, gas concentration, and temperature.

In addition to real-time data, a Kaggle air quality dataset is used to train the machine learning models. The dataset contains various air quality parameters along with corresponding pollution levels. The data is divided into training and testing sets to evaluate the performance of the models effectively.

5.2 Data Preprocessing

Before training the models, several preprocessing steps are performed to improve data quality and ensure consistency. Missing values in the dataset are handled using appropriate techniques such as mean substitution or removal of incomplete records.

Feature scaling and normalization are applied to bring all input parameters to a similar range, which improves model efficiency. Relevant features such as PM2.5, gas concentrations, and temperature are selected for model training. Data cleaning is also carried out to remove noise and outliers, thereby enhancing prediction accuracy.

5.3 Model Training

After preprocessing, the cleaned dataset is used to train machine learning models. Different algorithms such as Linear Regression, Random Forest, and Decision Tree are applied to learn the relationship between input features and air quality levels. The training process involves feeding the model with input data and corresponding output values so that it can learn patterns and make predictions.

The dataset is split into training and testing sets to evaluate model performance. During training, the model parameters are adjusted to minimize prediction errors. The trained models are then validated using test data to ensure accuracy and reliability.

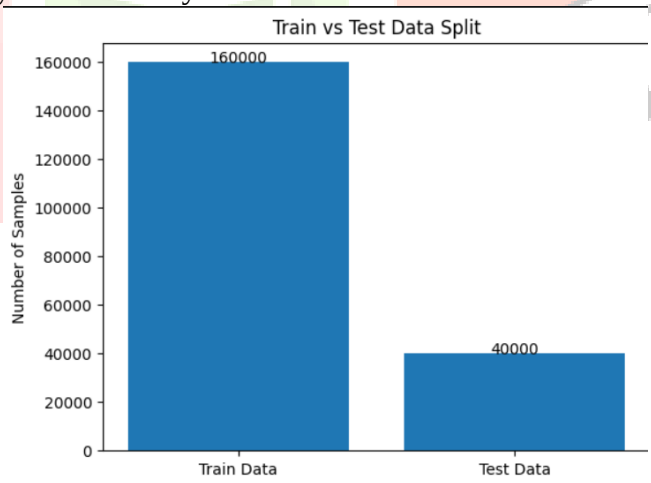


Fig.1.2 Train and Test Data

5.4 Prediction

Once the model is trained, it is used to make real-time predictions. The sensor data collected from the ESP32 is sent to the system, where it is processed and passed to the trained model. The model predicts the air quality level based on input parameters such as PM2.5, gas concentration, and temperature.

The predicted values help in identifying pollution levels and potential health risks. The system can provide alerts if the air quality crosses safe limits, enabling users to take necessary precautions.

5.5 Visualization

The system includes a visualization module to display air quality data in an understandable format. Graphs and charts are used to represent real-time and historical data, making it easier for users to analyze trends.
The user interface shows current air quality levels, predicted values, and health risk indicators. This visual representation helps users quickly understand the environmental conditions and make informed decisions.

VI .RESULT

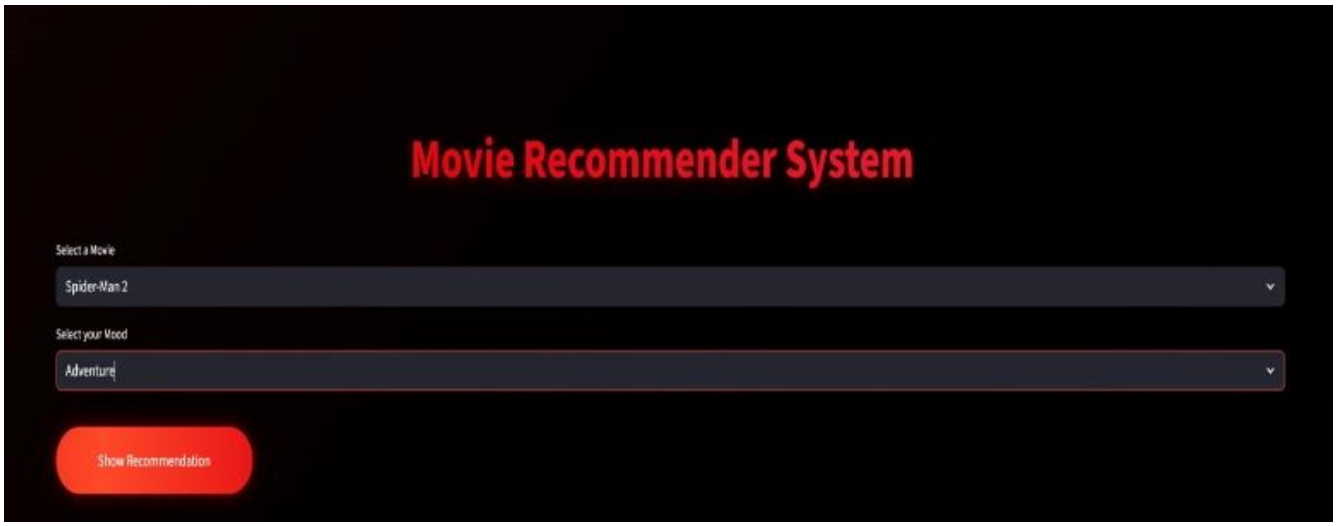
The movie recommendation system we built works well.
It uses three methods:
One that suggests movies like the one you choose
One that suggests movies based on what users like
One that suggests movies based on how you're feeling
The system gives you movie suggestions that're just right for you. You can interact with it easily.

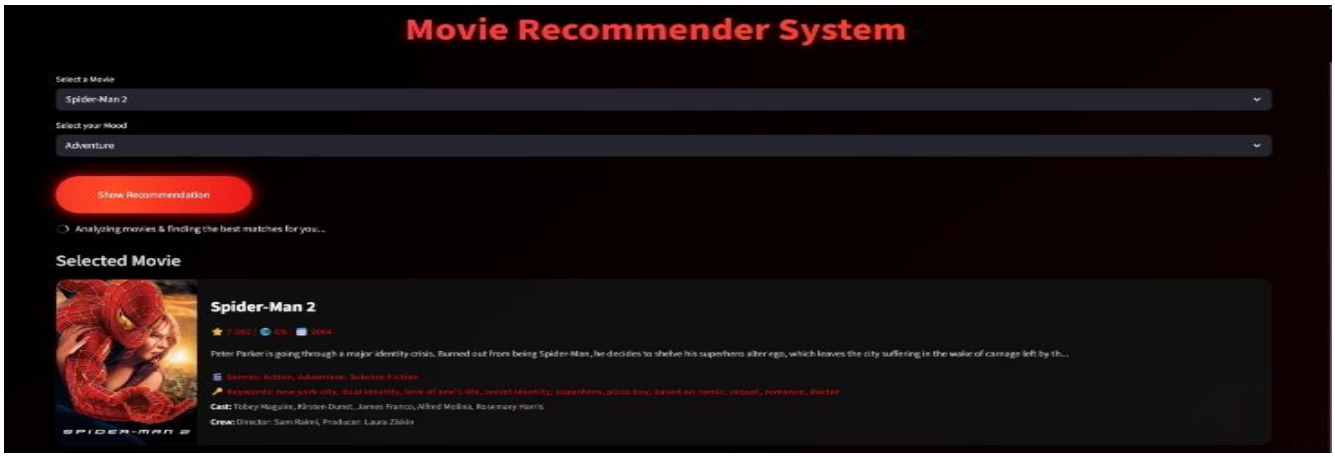
1. Recommendation Output
The system gives you three kinds of suggestions:

Content-Based Suggestions:
Movies that're similar to the one you picked.
They have genres, keywords, cast and story.

Suggestions based on what others, like:
Movies that are liked by users who like things.
This is based on ratings that users give.

Suggestions based on your mood:
Movies that match how you're feeling.
You can choose Happy, Sad, Romantic and more.
The system also shows you:
Movie poster
Rating
Release year
Language
Keywords
Cast and crew details
This makes it more fun to use. Helps you make a better choice.





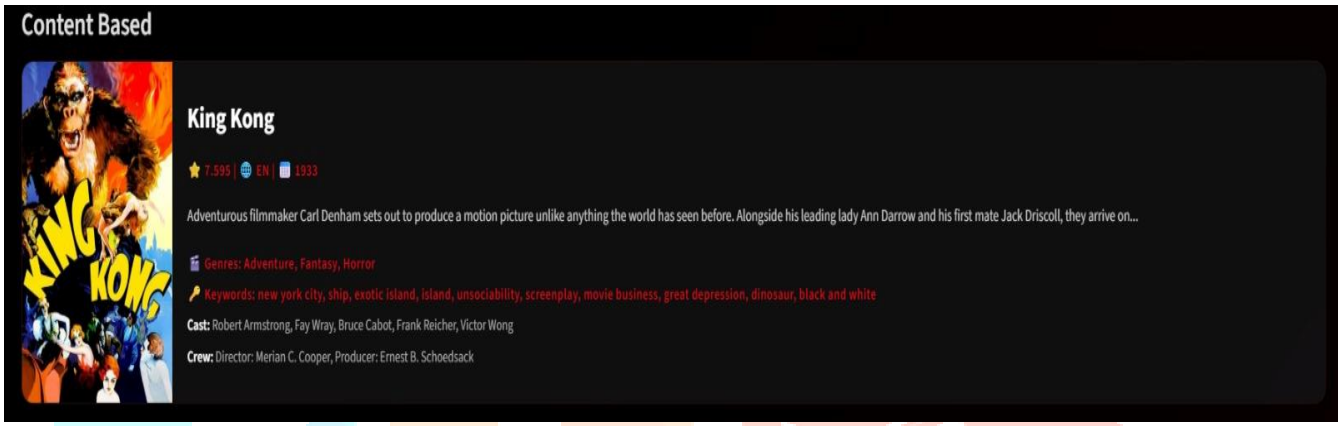


Fig.1.3 Content Based Filtering

Fig.1.4 Collaborative Filtering

Adventure-Mood

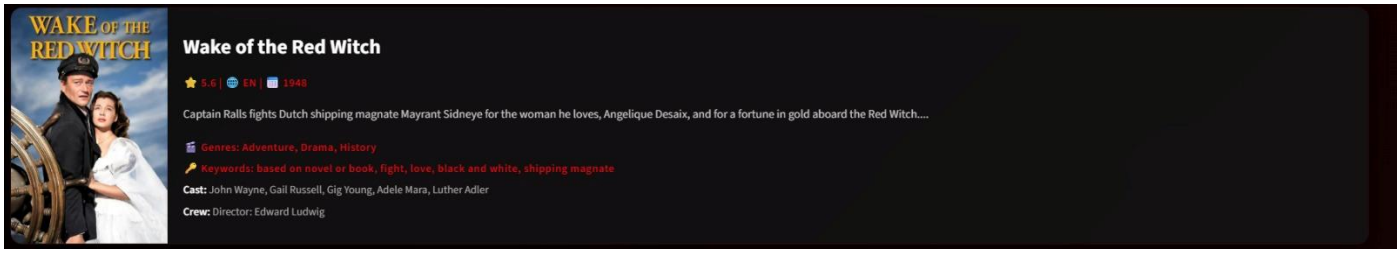


Fig.1.5 Emotion Basesd Movie Recommendation

2. Graph Analysis

The graph in this project shows how movies are related to each other based on how similar they're Each movie is a node in the graph. The lines between the nodes or edges show how similar the movies are. When movies are close together in the graph it means they are very similar. We can see groups of movies, which we call clusters.

What does it all mean?

Movies in the cluster are usually in the same type of movie or have similar ideas.If a movie is connected to a lot of movies it is probably a very popular movie or very similar to other movies. If a movie is not connected to other movies it is probably a unique movie or not very similar, to other movies. This graph shows us that our movie recommendation system is working well and grouping movies together.

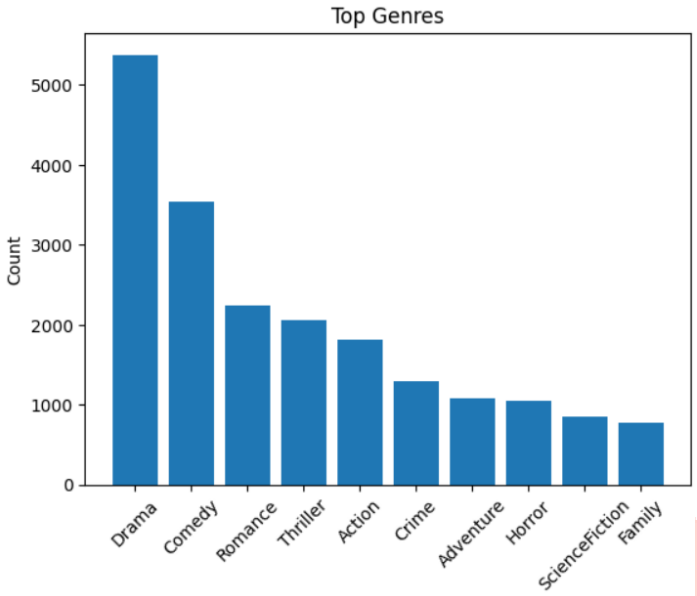


Fig.1.6 Movies Genres

3. How Well The Model Works

The model that uses filtering was tested with some metrics to see how it does. It was tested with regression and classification metrics.

These metrics are:

RMSE which's Root Mean Square Error: This measures how wrong the models predictions are

MAE which's Mean Absolute Error: This measures the average mistake the model makes

R^2 Score: This shows how accurate the model is

Accuracy, Precision, Recall, F1 Score: These are used to see how well the model does with classification

The results show that the collaborative filtering model gives pretty accurate predictions and useful recommendations.

The collaborative filtering model does a job of giving good predictions and helpful suggestions.

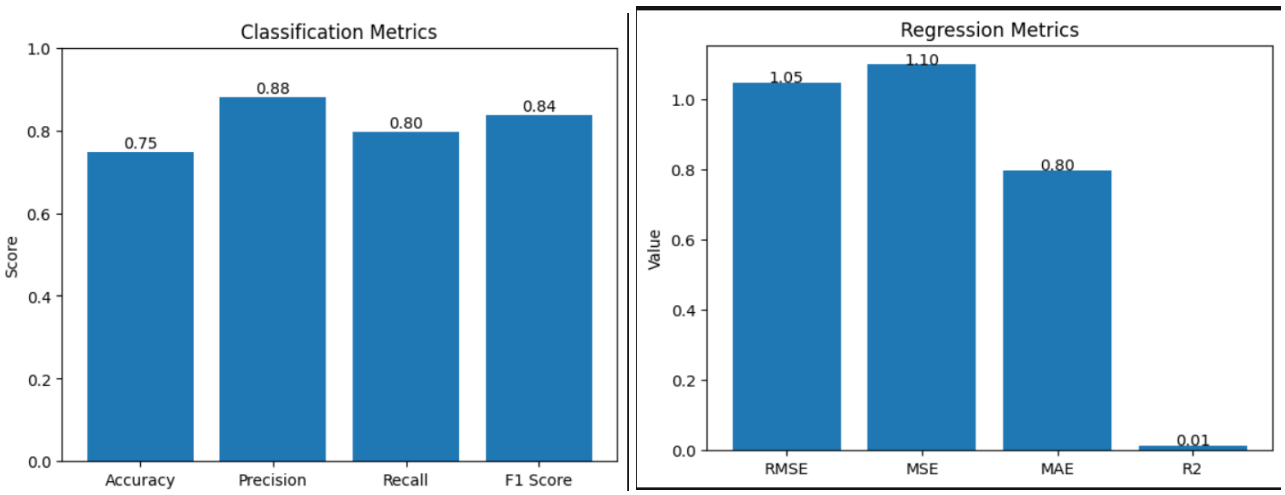


Fig.1.7 Classification and Regression Metrics

4. How The System Performs
The system is really fast because it already knows what is similar to what.
It can get the movie information from the internet in real time using a special tool called an API.
The system is also easy to use because it has an interface made with Streamlit.

5. What The System Can Do
The system does a lot of things well such as:
It suggests movies that're relevant to what you like.
It uses different ways to recommend movies, which is called a Hybrid system.
It gives you suggestions based on how you're feeling and what is happening right now.
It makes the experience of using the system nice by showing you pictures and detailed information, about the movies.

VII. CONCLUSION:

This project was about making a movie recommendation system that really works. The system was made by combining three methods: looking at the content of the movies looking at what other people like and recommending movies based on how people feel. The system looks at things like what kind of movie it's what it is about, who is in it and what people think of it to give good recommendations. The part that looks at content uses words and similarities to find movies that're alike. The part that looks at what other people like uses what people have watched and rated to recommend movies that others like too. The part that looks at how people feel makes the system more fun and personalized because it gives recommendations based on how the user is feeling at the moment. The system also uses real-time information from websites to make sure the recommendations are good and have the latest information like pictures, ratings and who is in the movie. The system is easy to use because it was made with something called Streamlit, which makes it simple to get to the recommendations. The system was. It works well giving good recommendations that people can trust. The tests included things like how wrong the recommendations were, how accurate they were and how well the system could find movies. Overall the system shows that using methods together can make recommendations a lot better and more fun for users. It is a solution for modern entertainment platforms because it can handle a lot of users and give good recommendations. There is still room to make it better like making it work for more people giving more personalized recommendations and being able to adapt to what people like in real-time. The movie recommendation system is a start and it can be improved to make it even better for people who like movies. The system uses movie metadata and user ratings to give recommendations and it can be used by people who want to find new movies to watch. The movie recommendation system is a tool, for people who like movies and want to find new ones to watch.

VIII. FUTURE SCOPE

We can make our movie recommendation system even better. Our movie recommendation system has a lot of potential to improve. There are ways to make it work faster and make the users happier with the movie suggestions they get. One big improvement is using computer learning techniques like Neural Collaborative Filtering and Recurrent Neural Networks. These computer learning techniques are really good at understanding what users like and how they watch movies over time.T his information can give us movie suggestions that users will really like. We can also make the system remember what users like and what they have watched before. This way it can suggest movies that're just right for each user. Of just looking at general trends the system can get to know each user and what they like. The system can use information like what time of day it's where the user is and what device they are using to make movie suggestions. This can make movie suggestions that fit the users situation. To make suggestions instantly we can use technology that processes data in time. This way as users change what they like the movie recommendation system can change its suggestions. Using data tools can also help handle lots of user data and make the system work better. We can make the movie recommendation system understand user emotions better. It can look at what users say about movies or even recognize their expressions when they watch a movie. This can make the movie recommendation system more interactive and smart. We can also make the user interface better by adding features. We can add features like voice search, chatbots

and ways for users to give feedback on the movie suggestions they get. This way users can help make the movie recommendation system better over time. Lastly putting the movie recommendation system on cloud platforms can make it work better be more accessible and handle lots of users. This makes the movie recommendation system ready for streaming services. We can use it to suggest movies to users all, over the world

.

IX . ACKNOWLEDGMENT

I would like to extend my heartfelt thanks to my project guide along with the entire teaching faculty for their support and encouragement during the implementation of Movie Recommendation System Project. The valuable suggestions from them have helped me in setting rigorous standards in methodology used. Moreover, I would like to thank the college for all the help they provided me during the project work. Thank you to my team members who have collaborated with me in completing the project. Thanks are also due to people who have developed the MovieLens Dataset and TMDB Dataset, upon which our system was developed. Lastly, I owe thanks to my family and friends for always encouraging me.

X . REFERENCES

J. S. Breese, D. Heckerman, and C. Kadie, “Empirical Analysis of Predictive Algorithms for Collaborative Filtering,” in Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998.

G. Adomavicius and A. Tuzhilin, “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, vol. 17, no. 6, pp. 734–749, 2005.

U. Panniello, A. Tuzhilin, and M. Gorgoglione, “Comparing Context-Aware Recommender Systems in Terms of Accuracy and Diversity,” User Modeling and User-Adapted Interaction, vol. 24, no. 1– 2, pp. 35–65, 2014.

Y. Koren, R. Bell, and C. Volinsky, “Matrix Factorization Techniques for Recommender Systems,” IEEE Computer, vol. 42, no. 8, pp. 30–37, 2009.

S. Sun, J. Luo, and J. Chen, “A Hybrid Collaborative Filtering Algorithm Based on Cosine Similarity and Trust,” Journal of Information Science and Engineering, 2018.

F. Ricci, L. Rokach, and B. Shapira, “Recommender Systems Handbook,” Springer, 2011

MovieLens Dataset, “GroupLens Research,” University of Minnesota. [Online]. Available: https://grouplens.org/datasets/movielens.

TMDB API, “The Movie Database (TMDB),” [Online]. Available: https://www.themoviedb.org/documentation/api

Pedregosa et al., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.