Study on Movie Recommendation System

Ketha Jagadhish

Abstract

The study introduces a Movie Recommendation System that uses classification techniques based on user ratings to offer movie suggestions. It works with a dataset that includes user interactions, with movies like ratings and user IDs. Initially the dataset is prepared by encoding variables and grouping ratings into categories like 'low' 'medium' and 'high'. After preprocessing the data is divided into training and testing sets. Feature scaling is applied for normalization. Three classification algorithms. Random Forest, Logistic Regression and Decision Tree. Are used to build models using the training data. Hyperparameter tuning is carried out through GridSearchCV to enhance model performance. The metrics such as accuracy and F1 score are employed to evaluate how well each model predicts movie ratings. The findings highlight the effectiveness of classification based methods, in recommending movies based on users preferences improving the movie recommendation system experience.

Keywords: Classification, Hyperparameter tuning

1 Introduction

In today's world of online streaming, choosing a movie can be overwhelming. That's where recommendation systems come in. They use smart computer programs to suggest movies based on what you've watched before and what others like you enjoy. Traditionally, recommendation systems worked by looking at what others liked or by matching up details of movies you enjoyed. But now, there's a better way called classification-based methods. These methods simplify things by putting movies and your preferences into different groups, making it easier to find what you'll like. Our study introduces a new movie recommendation system that uses this classification technique, focusing on how you rate movies. Instead of just numbers, we group ratings into 'low', 'medium', and 'high' categories. This helps our system understand your preferences better. Our goal is to make your movie nights more enjoyable by giving you suggestions that truly match what you like. For that we have taken rating as main parameter i.e if a person watches a movie of a particular rating the model will suggest the movie in that particular range movie. So we have taken a movielens dataset that contains different movie titles and rating and we merge those different datasets and made into particular dataset.

2 Related work

Many recommendation systems have been developed over the past decades.
These are the systems use different approaches like collaborative approach, content based approach, a utility base approach, hybrid approach so on.
Looking at the purchase behavior and history of the shoppers, Lawrence et al.. To refine the recommendation collaborative and content based filtering approach were used. To find the potential customers most of the recommendation systems today we use ratings given by previous users. These ratings are further used to predict and recommend the item of one's choice. Weng used the MD recommendation model (multidimensional recommendation model) for this purpose only.

3. RESEARCH METHODOLOGY

3.1 **Preprocessing Step**

I collected data from two different sources, one containing movie deatails and the other containg user ratings. To analyze them together, I merged them based on a unique identifier called "movieId." Then, I prepared the data for analysis. This involved converting non-numeric information, like user and movie IDs, into numbers so the computer could understand them better. I also adjusted the scale of the data to make sure all features were treated equally during analysis. Once the data was prepped, I split it into two parts: one for training and one for testing. This separation helps me evaluate how well my models can predict new, unseen data. It's like giving the model a practice test before the real exam.

3.2 **Data Description**

It contains the attributes

1.movieId
2.title
3.genres
4.rating
5.timestamp

By completing these steps, I set the foundation for the next phase of my analysis, where I'll use various machine learning techniques to uncover insights or make predictions based on the prepared data. Different attributes have different weights. In our research we have found that the most of the appropriate recommendations that can be generated should be based on the ratings given to the movies by previous users, therefore we have given more importance to the rating attribute than other attributes. These ratings have been taken from the rating given to these movies by a large number of different users from different

	movieId	title	genres	userId	rating	timestamp
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1	4.0	964982703
1	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	5	4.0	847434962
2	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	7	4.5	1106635946
3	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	15	2.5	1510577970
4	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	17	4.5	1305696483

parts of the world. Another important parameter in our proposed model is total number of votes received by a particular movie.

4. Methodology

4.1 **Classification**

After preparing the data, I used the 'rating' attribute as the target variable for my analysis. This means I wanted to predict or classify the rating of a movie based on other features like user ID and movie ID. To do this, I employed three different machine learning algorithms: Random Forest, Navie Bayes, Logistic Regression, and Decision Tree.

Random Forest is an ensemble learning method that constructs multiple decision trees during traing and output the mode of the classes (classification) or mean prediction (regression) of the individual trees. Logistic Regression is a linear model used for binary classification tasks, which predicts the probability of occurrence of an event by fitting data to a logistic function. Decision Tree, on the other hand, is a simple tree-like structure where an internal node represents a feature or attribute, the branch represents a decision rule, and each leaf node represents the outcome.

After training each model using the training data, I evaluated their performance using accuracy and F1 score metrics. Accuracy measures the proportion of correctly classified instances out of the total instances, while F1 score is the harmonic mean of precision and recall, providing a balanced assessment of a classifier's performance. By comparing these scores across the three models, I can determine which one performs best for predicting movie ratings

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

4.2 **Hyperparameter Tuning**

I optimised Random Forest, Logistic Regressor, Navie Bayes and Decision Tree Classifier using GridSearchCV for hyperparameter tuning. This involved in the systematically exploring hyperparameters to maximize accuracy. I defined parameter grids for each classifier, trained GridSearchCV objects with cross-validation, and selected the best parameters to fine-tune the models. This process ensured optimal configurations for accurate predictions on unseen data, improving performance in real-world scenarios.

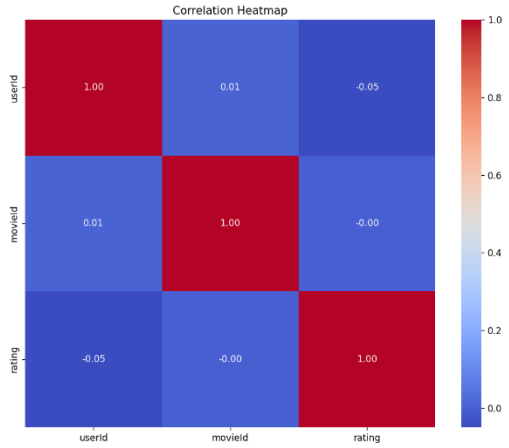
5. Result and Discussion

5.1 Dataset

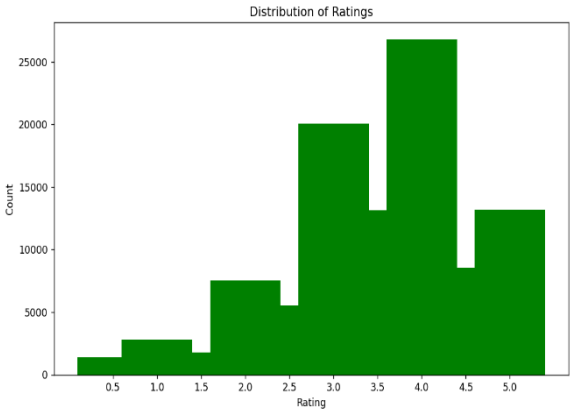
The Movie Lens dataset is widel used resource in the fiels of recommender systems and machine learning, offering a diverse collection of movie ratings provided by users. This dataset typically includes information such as user IDs, movie IDs, ratings, and timestamps. By discretizing the ratings into low, medium, and high categories, we aim to simplify the analysis process and make it more interpretable. This preprocessing step allows us to apply classification techniques, such as decision trees, support vector machines, or neural networks, to predict the likelihood of a user rating falling into one of these categories. Additionally, accuracy scores obtained from these classification models serve as valuable metrics for evaluating the effectiveness of our prediction algorithms.

5.2 **Result**

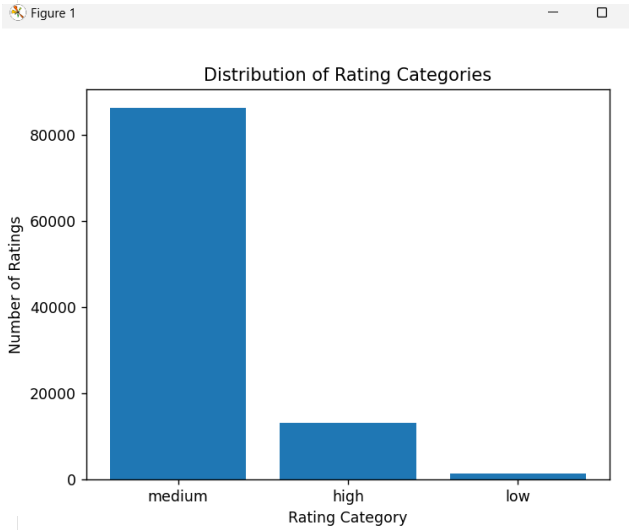
In the evaluation of four classifiers namely RandomForestClassifier,LogisticRegression, and DecisionTree, for a given task, we observed that Logistic Regression yielded the highest accuracy score among them. This outcome suggests that Logistic Regression performed most effectively in capturing the underlying patterns within the data and making accurate predictions. Further , to enhance the performance of these classifiers , we employed Hyperparameter tuning technique. Hyperparameter tuning allows us to systematically search for the optimal set of hyperparameters for each classifier, thereby maximizing its predictive power. Through this process, we aim to fine-tune the model parameters to achieve the best possible performance metrics, such as accuracy and f1 score. By evaluating this we not only aim to improve the overall accuracy of the classifiers but also to enhance their generalization capabilities, ensuring robust performance on unseen data. This iterative optimization process contributes to building more reliable and effective machine learning models for real-world applications



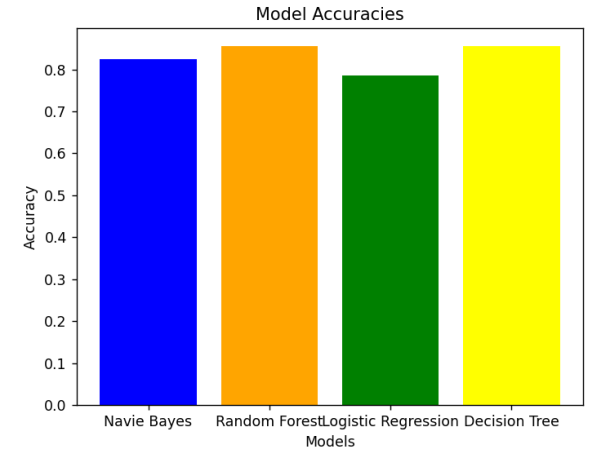
Heatmap for different attributes to check the dependency



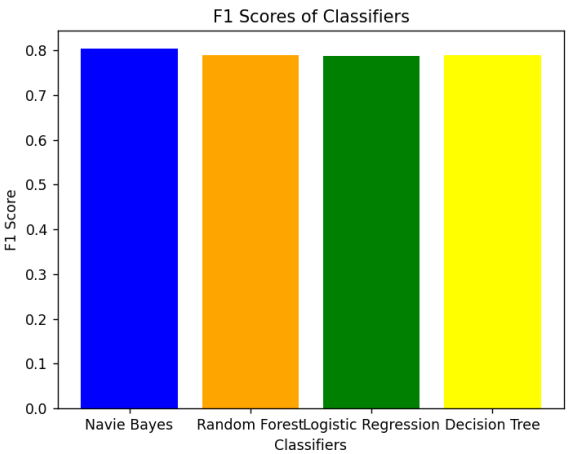
Bar graph of different ratings (continuous)



Bar graph of different ratings (Discrete)



Accuracy Scores



F1 Scores

6.Conclusion

In conclusion, through the exploration of a movie recommendation system, I have gained insights into various essential techniques in machine learning. Specifically, I learned about classification methodologies, hyperparameter tuning, and the transformation of continuous variables into discrete ones.

During this process, I observed that Logistic Regression emerged as the most accurate classifier when evaluating movie recommendations. Its ability to effectively classify user preferences based on discrete features made it a robust choice for this task. Moreover, by employing hyperparameter tuning techniques, I optimized the performance of the classifiers, ensuring they were finely tuned to the characteristics of the dataset. This iterative optimization process contributed significantly to improving the overall accuracy of the models.

Additionally, while Logistic Regression excelled in accuracy, Random Forest showcased superior performance in terms of F1 score. This highlights the importance of considering multiple evaluation metrics to gain a comprehensive understanding of model performance. While accuracy is crucial, F1 score provides insight into a model's ability to balance precision and recall, especially in scenarios with imbalanced class distributions.

Overall, this exploration underscores the importance of leveraging a combination of classification techniques, hyperparameter tuning, and thoughtful feature engineering to build effective recommendation systems. By integrating these methodologies, we can develop robust models capable of providing accurate and reliable movie recommendations, enhancing user experience and engagement.

References:

1. Thirunavu Karasu Va, Vishwa ravi, (February, 2024) movie recommendation system, IEEE Xplore.
2. Ricci, F, Rokach, L,& Shapira B,(2011).Introduction to recommender systems handbook Springer, Boston, MA
3. Simon Prananta Barus 2021 *J. Phys.: Conf. Ser.* **1842** 012008
4. Maher Maalouf Khalifa University, July 2011International Journal of Data Analysis Techniques and Strategies 3(3):281-299
5. Bahzad Taha Jijo, Adnan Mohsin Abdulazeez January 2021 Journal of Applied Science and Technology Trends 2(1):20-28
6. Bin Yu, Journal of Machine Learning Research 13 (2012) 1063-1095
7. G. Luo, A review of automatic selection methods for machine learning algorithms and hyper-parameter values, Netw. Model. Anal. Heal. Informatics Bioinforma. 5 (2016) 1–16. https://doi.org/10.1007/s13721-016-0125-6
8. Swayanshu Shanti Pragnya May 2018 International Journal of Computer Applications 180(45):30-35