Abdullin Ruslan

ru.abdullin@innopolis.university

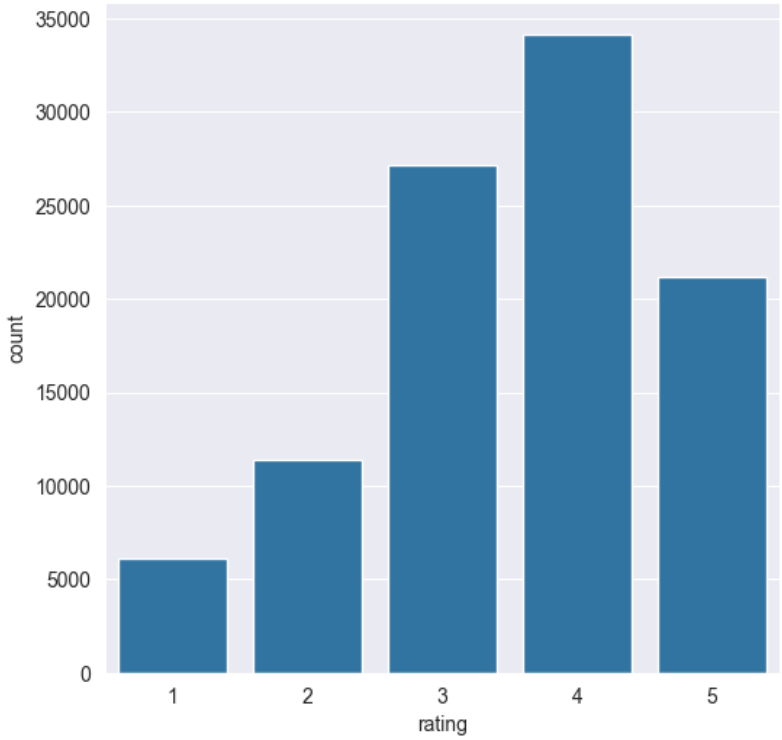
AAI BS21

Introduction

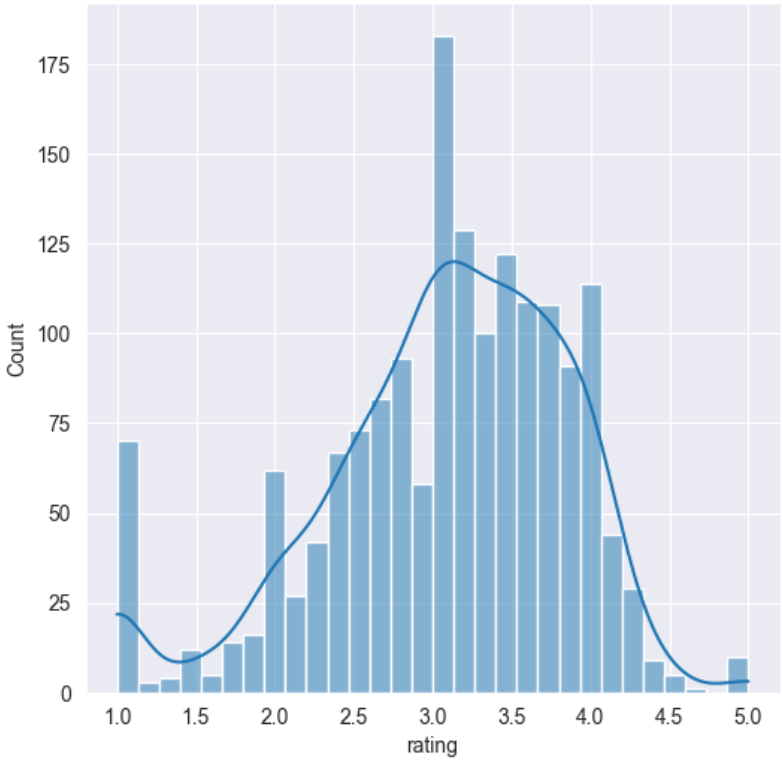
In this project, we aimed to develop a movie recommender system leveraging the MovieLens 100K dataset. The system is designed to suggest movies to users based on their demographic information and their favorite movies. Our approach involved experimenting with various machine learning models to find the most effective one for our needs.

Data Analysis

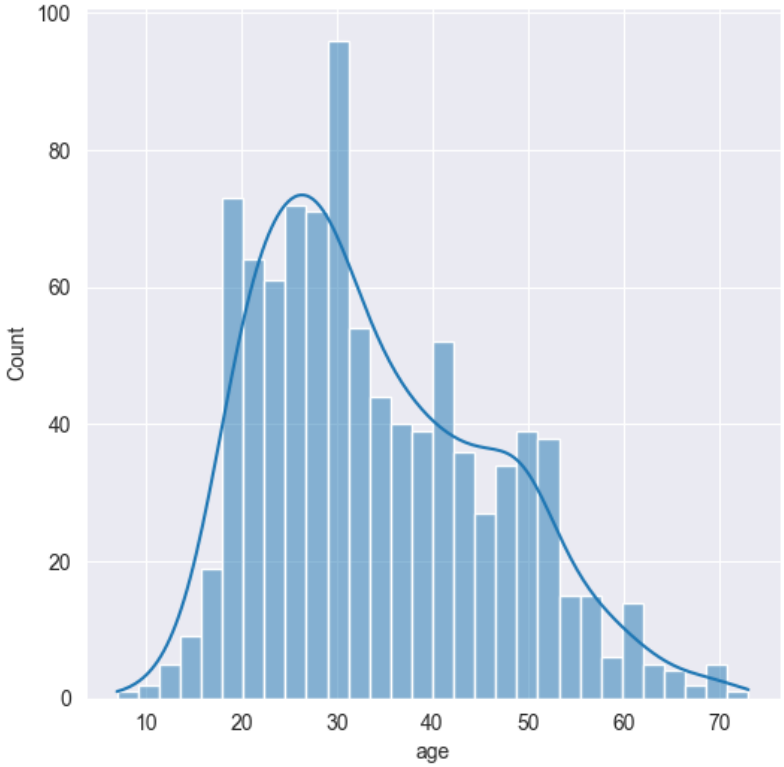
The MovieLens 100K dataset consists of 100,000 ratings from 943 users on 1,682 movies. The data analysis phase involved exploring these ratings along with demographic information (age, gender, occupation, zip code) of users. In this section, we explored the distribution of user ratings, the age and gender demographics of users, the number of ratings per user, and the popularity of different movie genres.



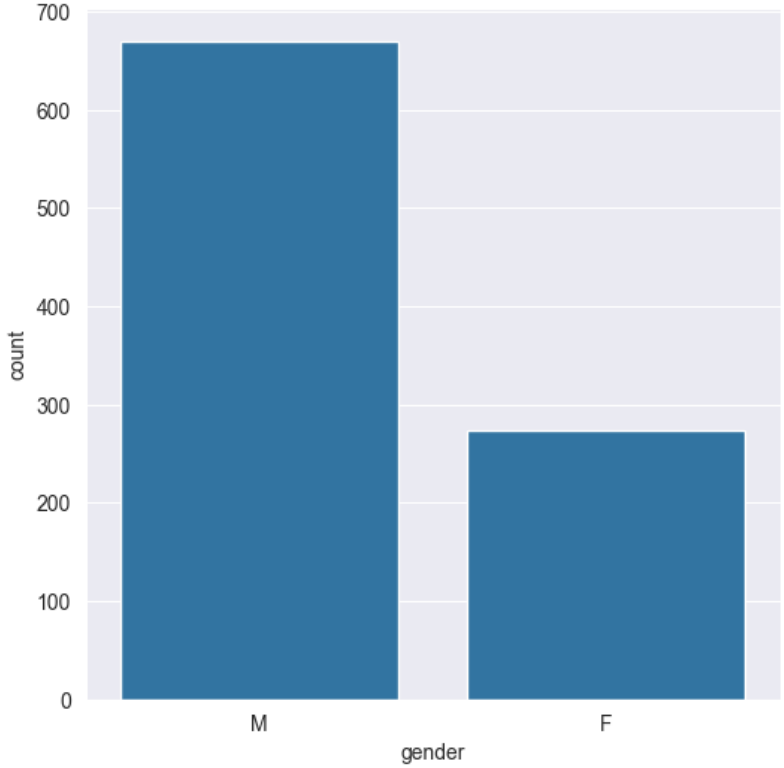
This histogram shows the distribution of ratings across the dataset, with a fitted line indicating the general trend.



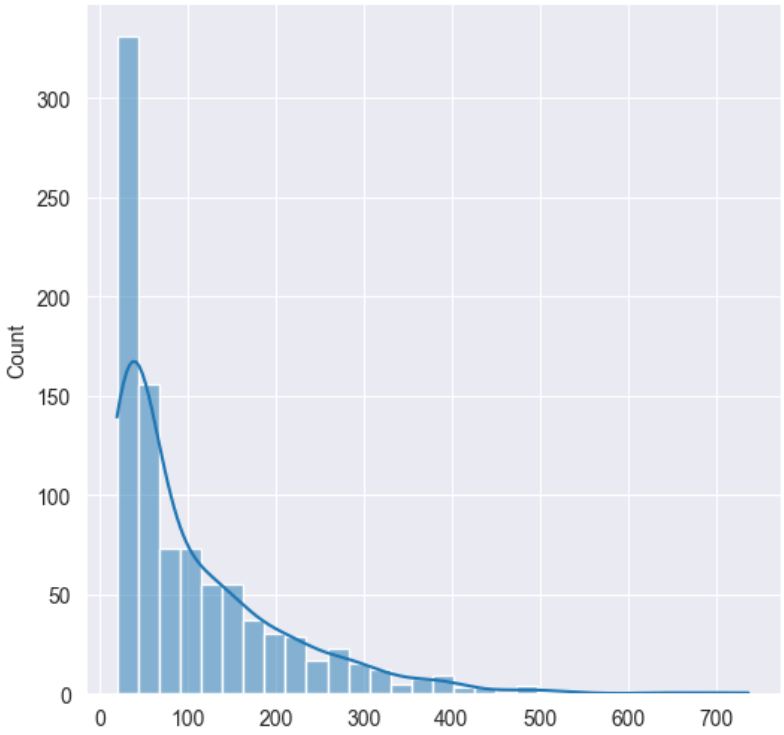
The plot indicates the average rating per movie, providing insight into the overall reception of movies by users.



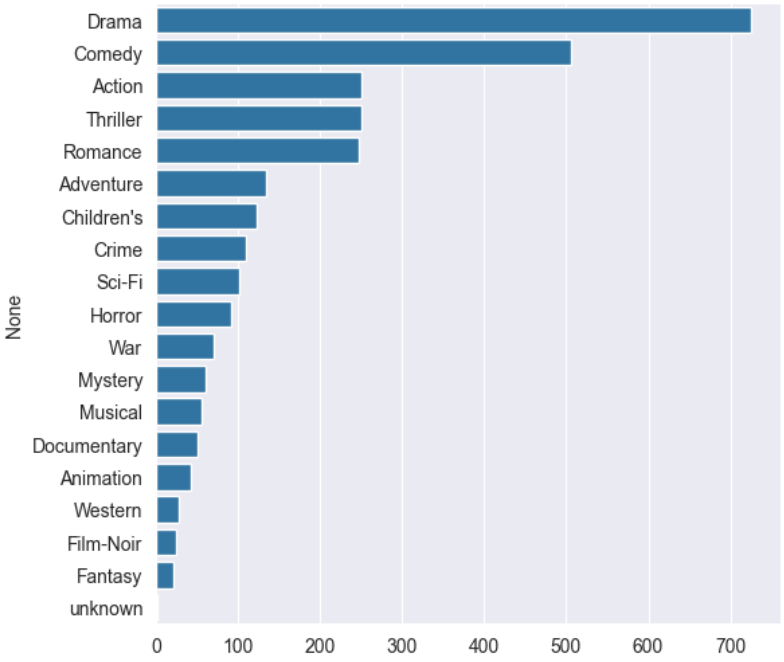
This histogram outlines the age distribution of the users, with the majority falling within a specific age range.



The bar chart illustrates the gender distribution of the dataset's users.



This visualization shows how active users are in terms of the number of ratings they have provided.



A bar chart representing the popularity of different movie genres among the users.

Model Implementation

Three models were considered for this task: Singular Value Decomposition (SVD), LightGBM Regressor (LGBMR), and RandomForestRegressor. After experimentation and evaluation, RandomForestRegressor

emerged as the most suitable model for our system. The implementation involved preprocessing the data, splitting it into training and test sets, and training the RandomForestRegressor model.

Model Advantages and Disadvantages

- RandomForestRegressor (RMSE is 0.23392)

Advantages	Disadvantages
Handles non-linear data effectively	Computationally intensive
Robust to overfitting	Less interpretable compared to models like SVD
Good balance between bias and variance	

- Singular Value Decomposition (RMSE is 0.32343)

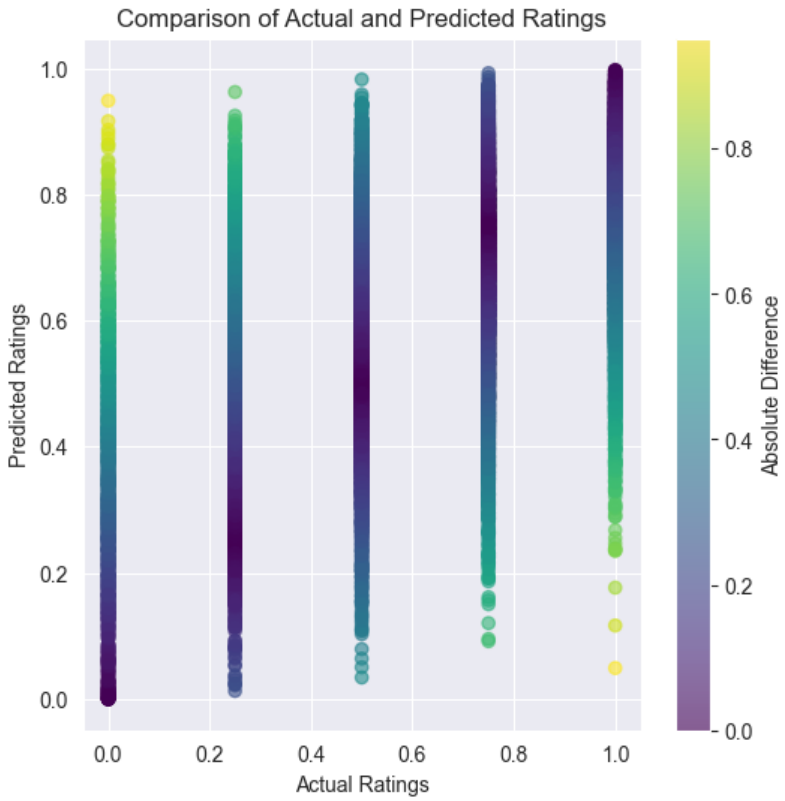
Advantages	Disadvantages
Effective in capturing latent factors in data	Assumes linear relationships between variables
Scalable to large datasets	Sensitive to sparse and noisy data
Good interpretability of results	Requires careful handling of missing values

- LightGBM Regressor (RMSE is 0.68323)

Advantages	Disadvantages
Fast training speed and efficiency	Can overfit on small datasets
Handles large-sized data and supports GPU learning	Less interpretable compared to linear models
Good performance on unbalanced data	Requires careful tuning of hyperparameters

Training Process

The model's performance was evaluated by comparing the actual ratings with the predicted ratings.



This scatter plot with color coding illustrates the comparison between actual and predicted ratings, highlighting the accuracy of the model.

Evaluation

The model was evaluated using the Root Mean Squared Error (RMSE) metric. This metric was chosen as it effectively captures the average error between the predicted and actual ratings, providing a clear measure of the model's accuracy.

Results

The RandomForestRegressor model achieved an RMSE of 0.2339, indicating a high level of precision in predicting movie ratings. This performance suggests that the model is well-suited for recommending movies to users based on their preferences and demographic information.

References

[1] F. M. Harper and J. A. Konstan, "The MovieLens Datasets: History and Context," ACM Transactions on Interactive Intelligent Systems (TiiS), vol. 5, no. 4, Article 19, pp. 1–19, 2016.

[2] K. Pearson, "On lines and planes of closest fit to systems of points in space," Philosophical Magazine, vol. 2, no. 11, pp. 559-572, 1901.

[3] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," Advances in Neural Information Processing Systems, pp. 3146-3154, 2017.

[4] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.