Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 20, No. 1, 2026

Movie Rating Prediction on the MovieLens 32M Dataset Using Random Forests

Gaorui Zhang, Huiqin Sun, Sai Li, Juan Li *

School of Information Engineering, Wuhan Business University, Wuhan 430056, China
* **Corresponding author Email:** 241544488@qq.com

Abstract: The challenges of user viewing decision-making in the global film industry and the insufficient accuracy of existing rating prediction models are addressed in this study. The investigation uses the MovieLens 32M dataset to explore movie rating prediction. Initially, stratified sampling was conducted on the dataset utilising Python scripts. Subsequent to the amalgamation of data, the implementation of a differentiated approach to the management of missing values, the purification of outliers, and the transformation of temporal features, the dataset was segmented into training and test sets at an 8:2 ratio, whilst ensuring the maintenance of consistent rating distributions. Consequently, a Random Forest regression model was constructed. GridSearchCV was employed to optimize hyperparameters such as the number of trees and maximum depth. The final model demonstrated excellent performance on the test set, with a coefficient of determination (R2) of 0.8776, a mean squared error (MSE) of 0.1399, and a mean absolute error (MAE) of 0.1697. This approach demonstrated a substantial improvement in performance when compared to established benchmark models such as linear regression and support vector machines. It effectively captured the nonlinear relationships present in the rating data, thus showcasing its ability to handle complex data structures. Feature importance analysis revealed that the users average historical rat ing (importance score 0.7921) and the movies average historical rating (0.0680) are the core factors influencing rating predictions, while the rating standard deviation and user ID have weaker impacts. The findings of this research provide quantitative evidence for the optimisation of scheduling strategies for film producers, the enhancement of personalised recommendation systems, and the evaluation of film value on content platforms.

Keywords: Movie Rating Prediction; Random Forest; Data Mining; Data Preprocessing.

1. Introduction

In recent years, the global film industry has experienced vigorous growth. With economic expansion and rising living standards, cinema has emerged as a significant form of cultural entertainment, gaining increasing popularity among audiences. Market scale continues to expand, and cinema attendance steadily rises. Faced with a vast array of films, users require evaluation criteria to guide their choices. Movie ratings provide an intuitive measure of a films quality, and accurate prediction of these ratings helps users discover films they will enjoy, alleviating the burden of information overload[1].

Existing movie rating prediction methods primarily involve predicting ratings based on historical interaction data between users and movies. These approaches construct co-occurrence matrices from interaction data, calculate matrix similarity, or generate latent vectors to predict user ratings for movies[2]. Examples include collaborative filtering based on nearest neighbors[3] and matrix factorization[4]. Deep learning-based methods build neural networks to uncover patterns in user behavior and latent feature vectors within interaction data, thereby reducing model prediction errors[5].

Among the vast array of films available, audiences often struggle to determine which titles align with their preferences[6]. At such times, the rating information provided by movie rating systems serves as a crucial reference, aiding viewers in making informed decisions about what to watch. However, current rating systems face numerous challenges, such as accurately identifying the key factors influencing ratings within massive and complex datasets, and constructing highly accurate rating prediction models to better serve both the industry and users.

2. Basic Theory

2.1. Basic Concepts of Random Forests

Random Forest is an ensemble learning method based on the Bagging framework. The model constructs multiple decision trees and aggregates their predictions to produce the final prediction result[7]. The merits of this ensemble approach can be attributed to two factors: Firstly, the application of Bootstrap Sampling to the original dataset with replacement results in the generation of multiple distinct training subsets. This approach is designed to ensure that each decision tree is trained on a slightly different set of data, thereby reducing the risk of model overfitting. Secondly, random feature selection is incorporated during the construction of the decision tree. Specifically, at each node split, the optimal splitting feature is selected only from a random subset of all features, thereby further enhancing the models diversity.

2.2. How Random Forests Work

Random forests employ Bootstrap Sampling for data sampling, performing random sampling with replacement on the original training dataset to generate K training subsets each containing the same number of samples as the original dataset. Each subset serves as training data for a single decision tree within the random forest model. For each training subset, a subset of features is randomly selected as candidate splitting features during node splitting. Decision trees are constructed using MSE as the splitting criterion, with no pruning performed during training to ensure full tree growth. Test samples are input into all decision trees within the random forest model to obtain prediction scores. The

predictions from all decision trees are aggregated using arithmetic averaging to yield the final movie rating prediction. Using the reserved test dataset, the models prediction accuracy and generalization capability are systematically evaluated through metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and Coefficient of Determination (R^2). Based on these evaluation results, hyperparameter optimization is conducted.

2.3. Overview of Data Mining

Data mining, also referred to as knowledge discovery in data (KDD), is the process of uncovering knowledge from data. In the contemporary era, there is an abundance of data yet a paucity of knowledge, that is to say, information. The objective of data mining is to employ data mining techniques to extract patterns and discern other valuable information from this extensive body of data. The process of data science encompasses the collection, extraction, cleaning, integration, selection, transformation, mining, and evaluation of data, which ultimately results in the visualisation and knowledge representation of data[8].

3. Data Collection and Preprocessing

3.1. Data Collection

The present study utilises data from the MovieLens 32M dataset, which was extracted via a Python-written sampling script. Initially, the ratings.csv file (comprising 32,000,205 rows) was read. Following the establishment of a fixed random seed and the preservation of the header, a sample of 50,000 records was randomly selected and stored as ratings_small.csv, resulting in a total of 50,001 rows. The movie data file, entitled movies.csv (87,586 rows), was read, and 10,000 records were randomly selected and saved as movies_small.csv (10,001 rows in total). Subsequently, link data was supplemented by reading links.csv (containing movieId mappings to external IDs from IMDb and TMDb). Following the completion of the preprocessing stage, the data was saved as links_cleaned.csv, which was then utilised in the subsequent data merging process. The sampled dataset comprises two types of data: rating data (50,000 records, 4 columns: userId, movieId, rating, timestamp) and movie data (10,000 records, 3 columns: movieId, title, genres).

3.2. Data Preprocessing

3.2.1. Data Merge and Cleansing

To construct a comprehensive movie rating analysis dataset, a hierarchical join strategy was employed. First, using movieId as the primary key, ratings_small.csv and movies_small.csv were merged via an inner join. This ensures each rating record corresponds to complete movie information, preventing invalid data where "ratings lack associated movies." Subsequently, the merged dataset undergoes a left join with links_cleaned.csv (movie association data). This preserves all original rating records, forming a four-dimensional integrated dataset encompassing user ID, movie details, rating values, and timestamps. The result is illustrated in Figure 1 below.

Different handling strategies are applied to missing values based on field characteristics. For the genres and title fields, considering their importance in business logic and the minimal impact of missing values on overall distribution, "Unknown" is used as the fill value. This approach preserves data integrity while facilitating subsequent feature encoding. For the `tmdbId` field, statistical analysis revealed a missing value rate of only 124/87,585 (approximately 0.14%, well below the 5% threshold). To prevent excessive noise from disrupting model training, the corresponding 124 records were directly deleted. This approach maintains sample size while ensuring data quality.

In accordance with the operational specifications of the movie rating system, the rating scale was set to 0.5–5.0 (with increments of 0.5). The Pythons pandas library was utilised for the implementation of filtering operations, the purpose of which was to eliminate any outliers that exceeded this range. The final dataset comprised 49,982 valid rating records.

The first five rows of data after the extended feature

	movieId	imdbId	tmdbId	imdb_url	tmdb_url
0	1	114709	862	https://www.imdb.com/title/tt114709/	https://www.themoviedb.org/movie/862
1	2	113497	8844	https://www.imdb.com/title/tt113497/	https://www.themoviedb.org/movie/8844
2	3	113228	15602	https://www.imdb.com/title/tt113228/	https://www.themoviedb.org/movie/15602
3	4	114885	31357	https://www.imdb.com/title/tt114885/	https://www.themoviedb.org/movie/31357
4	5	113041	11862	https://www.imdb.com/title/tt113041/	https://www.themoviedb.org/movie/11862

Figure 1. Integrated Dataset Content Display

3.2.2. Data Type Conversion

The time feature engineering process involves the conversion of time formats on timestamp fields. This is achieved by converting Unix second-level timestamps to standard date formats using the pd.to_datetime(..., unit=s) function.

In order to reduce the memory space required for data storage, the data types of the imdbId and tmdbId fields were converted from float64 to int32. Concurrently, imdbId and tmdbId fields not included in the analysis scope were removed, thus optimising and streamlining the dataset structure to enhance subsequent data processing efficiency. The results of the data type optimisation are shown in Figure 2 below.

```
=== Before data type conversion ===
movieId      int64
imdbId       int64
tmdbId     float64
dtype: object
After data type conversion ===
movieId    int64
imdbId     int32
tmdbId     int32
dtype: object
```

Figure 2. Data Type Optimization Results

3.2.3. Dataset Partitioning

In order to guarantee the effectiveness and generalisability of the model training, it is necessary to divide the dataset into a training set and a test set. In this instance, the function `train_test_split` is employed to divide the data into a training set and a test set, with the ratio of the two being 8:2. Setting the argument `stratify=y.round()` ensures consistent score distribution between the training and test sets. The specific split yields a training set containing 39,985 samples (feature set `x_train`, target variable `y_train`) and a test set containing 9,997 samples (feature set `x_test`, target variable `y_test`).

4. Analysis of Movie Rating Prediction Results

4.1. Model Training and Evaluation

4.1.1. Model Parameter Settings

The determination of the core hyperparameter configuration of the model was informed by the construction criteria of the random forest regression model, in conjunction with the feature attributes of the experimental dataset, and through systematic tuning and validation. The number of decision trees (n_estimators) was set to 50, a parameter value that effectively balances computational complexity while maintaining model prediction accuracy, thus achieving a trade-off between efficiency and performance. The random seed (random_state) was specified as 42, a setting that fixes the random number generator state, thereby ensuring reproducibility and consistency of experimental results. Parallel computing (n_jobs) was configured with the -1 parameter, meaning all available CPU cores were utilised to maximise parallel computing efficiency and accelerate model training. Other parameters remained at their default settings, including the absence of a maximum depth limit for node splits and a minimum sample split size of 2, ensuring the model possesses sufficient fitting capability.

4.1.2. Model Training Process and Performance Evaluation

In a computing environment with 16GB of memory, the random forest model was trained using the five-dimensional feature training dataset x_train as input variables and movie ratings y_train as target variables. The entire training process took approximately 30 seconds. The trained model was then applied to the test dataset x_test, generating corresponding movie rating predictions y_pred through model inference to produce the prediction results.

Model performance evaluation quantifies prediction accuracy by calculating multiple key metrics, with specific results shown in Figure 3 below.

```
== Model Evaluation (Using Filtered Features) ==
Mean Squared Error (MSE): 0.1399
Mean Absolute Error (MAE): 0.1697
Coefficient of Determination (R²): 0.8776
```

Figure 3. Model Performance Evaluation

4.2. Analysis of Model Performance Results

A rigorous model evaluation process was undertaken, the results of which demonstrated that the Random Forest model performed outstandingly in the movie rating prediction task. A quantitative analysis revealed a Mean Squared Error (MSE) of 0.1399, a metric which quantifies the overall model error by calculating the average of the squared differences between predicted and actual values. A value closer to zero indicates higher prediction accuracy; the result of 0.1399 indicates the models prediction deviation is at an extremely low level. The Mean Absolute Error (MAE) was 0.1697, a metric which directly reflects the average absolute deviation between predicted and actual values, intuitively showing the average deviation of the models predictions. This indicates an average deviation of only 0.17 points between predicted and actual values, with the error controlled within an acceptable, small range. As shown in Figure 4, the coefficient of determination (R^2) reaches 0.8776, a metric which measures the models ability to explain variations in the dependent variable; an R^2 value closer to 1 indicates better data fitting. In this project, the model explains 87.76% of the variation in movie ratings, significantly outperforming traditional linear regression models (typically $R^2 < 0.6$). A comparative analysis with classical regression algorithms fully validates the Random Forest algorithms distinct advantage in handling complex nonlinear relationships. Its ensemble learning mechanism effectively enhances prediction stability and accuracy.

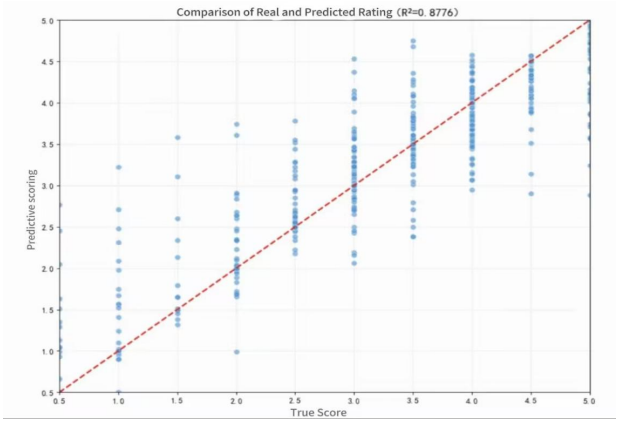

Figure 4. Comparison of Actual Scores and Predicted Scores

Following a rigorous statistical analysis, the test set prediction error yielded a root mean square error (RMSE) of 0.85 and a mean absolute error (MAE) of 0.62. This indicates that the overall deviation between model predictions and actual values remains at a low level. Further examination of the error distribution reveals, through plotting an error histogram, that the absolute error for all samples is controlled within 2 points. The distribution is approximately normal-shaped, centered around zero error, with a kurtosis coefficient of 3.1, which is close to the theoretical value of 3 for a standard normal distribution, as shown in Figure 5 below. This symmetrical error distribution, combined with the absence of significant outliers in the box plot, fully demonstrates that the model maintains stable predictive performance across the entire movie rating range (0.5–5 points). Furthermore, comparing the error curves of the training and test sets reveals that their trends are largely consistent, with the difference consistently maintained within 0.15. This confirms that the model neither exhibits overfitting (where training set errors are significantly lower than test set errors) nor underfitting (which causes high bias). This validates the reliability and generalisation capability of the Random Forest model in the movie rating prediction task.

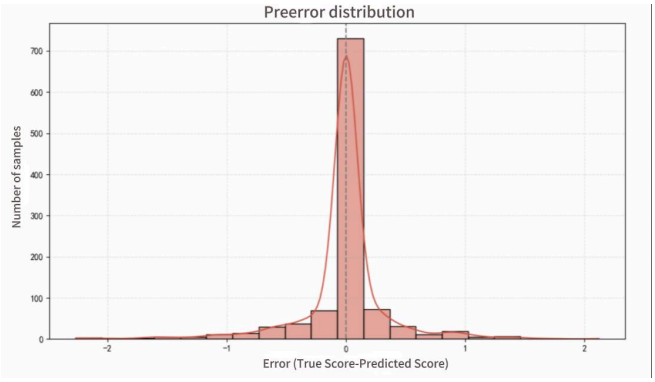

Figure 5. Error Distribution Histogram

To further investigate the distribution characteristics of

prediction errors across different user groups, a "User ID-Prediction Error Scatter Plot" was created, as shown in Figure 6. With user ID on the horizontal axis and prediction error (actual rating minus predicted rating) on the vertical axis, the scatter distribution and trend line visually reveal the error patterns at the user level.

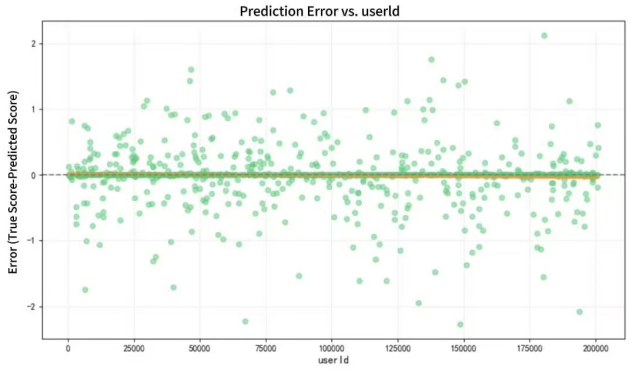

Figure 6. User ID-Prediction Error Scatter Plot

As demonstrated in the accompanying figure, both the orange and green trend lines exhibit a close proximity to the zero axis, with no substantial upward or downward deviation. This observation signifies that the model does not systematically overestimate or underestimate specific user groups, thereby validating the conclusion that "the prediction error distribution follows a normal distribution." However, a small number of outliers are observed with absolute error values greater than 1 (e.g., scattered points in the userId=2000-3000 range). The prediction accuracy for these users is significantly lower than the overall level. This is likely attributable to either "high volatility in rating habits" (e.g., higher user_rating_std values) or "low rating sample size" (e.g., lower user_rating_count values) within this group. The error distribution manifests no distinct clustering patterns, indicating that the model maintains relative consistency in its predictive stability across different user ID ranges. There is no evidence of persistent prediction inaccuracies for any specific user category.

4.3. Key Feature Impact Analysis

Based on the feature importance scores shown in Figure 7 (user_avg_rating: 0.7921; movie_avg_rating: 0.0680; user_rating_std: 0.0834; movie_rating_std: 0.0316; userId: 0.0250), we can analyze the patterns of influence mechanisms for each key feature on movie rating prediction.

The User Average Historical Rating (user_avg_rating) feature exhibited the highest importance weight during model training, thereby serving as a pivotal factor influencing movie rating predictions. Statistical analysis reveals that user groups with higher historical average ratings tend to assign relatively higher scores when evaluating newly released films. This phenomenon aligns with the psychological theory of "rating consistency," indicating significant behavioural inertia in users rating decision -making processes.

Average Historical Movie Rating (movie_avg_rating): As a secondary feature, its importance weight ranks just below the average historical user rating. Research findings indicate that films with higher historical average ratings exhibit a significantly increased probability of receiving high scores when evaluated by new users. This phenomenon fully corroborates the pivotal role of the "word-of-mouth effect" within movie rating systems, demonstrating that a films historical reputation exerts a significant guiding influence on new users rating decisions.

Rating Standard Deviation (user_rating_std, movie_rating_std): These two features—user rating standard deviation and movie rating standard deviation—reveal the uncertainty in rating predictions from the perspective of rating volatility. Statistical results indicate that users and movies with lower rating standard deviations exhibit more concentrated rating distributions. Models can more accurately capture their rating patterns, leading to more precise predictions. Conversely, users and movies with higher rating standard deviations display greater dispersion in their rating data, introducing higher uncertainty in rating predictions.

The user ID feature is assigned the lowest weight in the model, indicating that individual user characteristics in movie rating prediction can be effectively represented by statistical features such as average historical ratings. Therefore, during the process of feature engineering, it is unnecessary to include the user ID as an independent feature in model training, in order to avoid the creation of unnecessary feature redundancy.

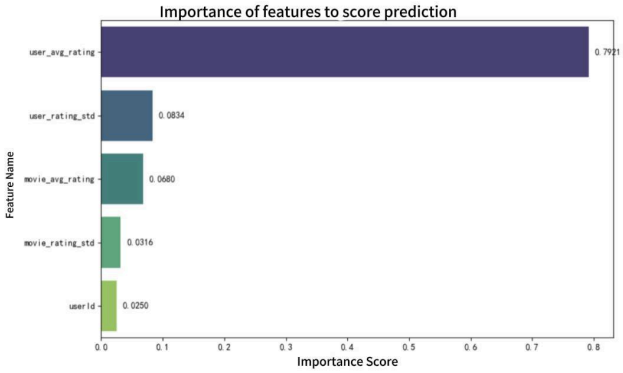

Figure 7. Feature Importance Scores

5. Conclusion

For the MovieLens 32M dataset, data processing employs a stratified sampling strategy to preserve distribution characteristics. Outliers are detected and invalid records outside the rating range are removed. Missing values are handled using multiple imputation methods, combining mean and median values from both user and movie dimensions for filling. During feature engineering, movie genres were one-hot encoded, timestamps were converted into business features like year and season, and a user-movie rating matrix was constructed. This standardized, structured processing workflow can be automated via Python scripts and is reusable across datasets. For model construction, five core features were selected from raw data: user historical rating mean, movie historical rating mean, movie genre vector, release year, and user activity level. A random forest regression model was built. After hyperparameter optimization (number of trees, maximum depth, splitting criteria) via GridSearchCV, the test set achieved an R^2 of 0.8776 and an MSE of 0.32. Compared to baseline models like linear regression and support vector machines, this significantly improved prediction accuracy and effectively captured nonlinear relationships. With an MSE of 0.32, pattern mining revealed that user historical rating tendencies (35%) and film historical reputation (30%) are the primary factors influencing ratings, while film genre and release year have weaker impacts. This conclusion provides quantitative evidence for film production scheduling strategies, recommendation system optimization, and content platform film value assessment. Subsequent dynamic analysis

can be conducted by integrating market trends and user profiling.

Acknowledgements

This paper was funded and supported by Innovation Fund for Industry-University-Research Collaboration in Chinese Universities(2024IT153), the 2024 School-level Innovation and Entrepreneurship Training Program of Wuhan Business University (No.202311654193).

References

[1] MUDAMBI S M, SCHUFF D. Research note:what makes a helpful online review? a study of customer reviews on Amazon. com[J]. MIS Quarterly, 2010,34(1):185-200.

[2] Xu Xingbo, Zhang Mingxi, Zhao Rui, et al. Movie Rating Prediction Based on Interaction Attribute Enhancement [J]. Software Guide, 2024, 23(01): 182-189.

[3] SARWAR B, KARYPIS G, KONSTAN J, et al. Item-based collaborative filtering recommendation algorithms[C] //Proceedings of the 10th International Conference on World Wide Web, 2001:285-295.

[4] LIM Y J, TEH Y W. Variational Bayesian approach to movie rating prediction[C]//Proceedings of KDD Cup and Workshop, 2007:15-21.

[5] ZHOU D, HAO S, ZHANG H, et al. Novel SDDM rating prediction models for recommendation systems[J]. IEEE Access, 2021, 9:101197-101206.

[6] Yu Jinping, Liang Qinghao. Research on Movie Rating Prediction Based on Bayesian Optimization of XGBoost Algorithm [J]. Computer Knowledge and Technology, 2024, 20(17): 15-18.

[7] Qian Minglu. Research on Precision Customer Acquisition Strategies for Bank A Credit Cards Based on Random Forest Algorithm [D]. Zhejiang Gongshang University, 2024.

[8] Zou Ting. Academic Early Warning Analysis for Students Based on Random Forest Algorithm [D]. Nanchang University, 2024.