

# Research and Analysis of Popularity Prediction of Film and Television Content Based on Machine Learning

Yutong Liu<sup>1\*</sup>

<sup>1</sup>Leeds College, Southwest Jiaotong University, 610000 Chengdu, China

**Abstract.** With the rapid development of the digital entertainment industry, accurately predicting the popularity of film and television content is of great significance for optimizing recommendation systems and making business decisions on content platforms. This study proposes a comprehensive predictive framework that integrates user behavior features, movie content features, and collaborative filtering information to construct multiple machine learning models for predicting movie ratings and popularity. The experiment was conducted on the MovieLens dataset, comparing traditional machine learning methods (linear regression, random forest, XGBoost) with deep learning approaches (multilayer perceptron), and further enhanced predictive performance through ensemble learning strategies. The research results indicate that the XGBoost model achieved the best performance in rating prediction tasks (RMSE=0.862), while the ensemble model reduced prediction error by 8.3%. Through SHAP value analysis, the study identified that the historical average rating of movies, user rating behavior patterns, and movie genres are the three most critical factors that affect prediction. This study provides empirical support and methodological guidance for the optimization of film and television recommendation systems.

## 1 Introduction

In the era of digital streaming media, Netflix、Disney+、Tencent Video and other platforms are facing the challenge of matching massive content with personalized user needs. According to statistics, the global streaming market has exceeded \$100 billion, and Netflix's recent A/B experiment with millions of users showed that accurate recommendations can increase the next month's retention rate by 32% [1]. Accurately predicting the popularity of film and television content not only optimizes recommendation algorithms, but also guides content procurement and production decisions, and has important commercial value [2].

Although traditional collaborative filtering methods are widely used in recommendation systems, they face challenges such as cold start and data sparsity [3]. In recent years, the development of machine learning technology has provided new solutions to this problem. By comprehensively analyzing user characteristics, content features, and interaction behavior, more accurate and interpretable prediction models can be constructed.

---

\* Corresponding author's email: [jian0418@my.swjtu.edu.cn](mailto:jian0418@my.swjtu.edu.cn)

This study aims to construct a comprehensive framework for predicting the popularity of film and television content. By integrating multi-dimensional feature fusion, user profiles, movie attributes, and collaborative information, a comprehensive feature system is constructed. The study compares the predictive performance of traditional machine learning and deep learning methods, and reveals the key factors that affect prediction through feature importance and SHAP analysis, ultimately verifying the effectiveness of the method on a real dataset.

The main contributions of this study include proposing a feature engineering method that integrates multi-source information, and determining the optimal model selection strategy through empirical analysis. Based on this, interpretable prediction results are provided to support business decision-making.

## **2 Related work**

### **2.1 Research status of recommendation systems**

The research on recommendation systems can be traced back to the 1990s. The GroupLens system proposed by Resnick et al. pioneered collaborative filtering [2]. Afterwards, matrix factorization techniques demonstrated excellent performance in the Netflix Prize competition and became the mainstream method for recommendation systems. Koren et al.'s SVD++ model further improves prediction accuracy by introducing implicit feedback information [4].

In recent years, the application of deep learning in recommendation systems has become increasingly widespread [5]. As of 2023, Transformers and graph neural networks have become the new mainstream of deep recommendation [6]. The Neural Collaborative Filtering (NCF) proposed by He et al. learns the nonlinear relationship between user item interaction through neural networks [7]. Google's Wide&Deep model combines memory and generalization capabilities and has been successfully applied in the industrial sector [8].

### **2.2 Rating prediction method**

Rating prediction is one of the core tasks of recommendation systems. Traditional methods mainly include neighborhood based collaborative filtering and model-based collaborative filtering. With the development of machine learning, ensemble learning methods such as decision trees and random forests have performed well in rating prediction. Gradient boosting methods such as XGBoost and LightGBM have achieved excellence in multiple rating prediction competitions by optimizing the loss function [9].

Feature engineering plays an important role in rating prediction. In addition to basic user and item characteristics, researchers have also proposed various derived features, such as user preference vectors, time decay factors, social network features, etc. [10]. The introduction of these features significantly improves the performance of the model. SIGIR 2021 further validated that system level feature search can further reduce RMSE by 9.7% without increasing model complexity [11].

### **2.3 Model interpretability**

With the widespread application of machine learning models in recommendation systems, the interpretability of models is receiving increasing attention. LIME (Local Interpretable Model Agnostic Explanations) and SHAP (Shapley Additive exPlanations) provide explanatory

tools for black box models [3]. In recommendation systems, interpretability not only helps debug and optimize models, but also enhances user trust and system transparency.

### 3 Methodology

#### 3.1 Problem definition

Given a set of users  $U=\{u_1, u_2 \dots, u_n\}$  and movie set  $M=\{m_1, m_2 \dots\}$ . The goal of the rating prediction task is to learn the function  $f: U \times M \rightarrow R$  and predict the rating  $r$  of user  $u$  on movie  $m$ . Meanwhile, the paper defines a binary classification task: label movies with ratings above 3.5 as 'popular' (label 1), otherwise as 'unpopular' (label 0). As shown in Figure 1, the data processing flow begins with the raw data collection stage, gathering a total of 943000 user ratings, covering 943 users and 1682 movies, and involving 19 types of movies; Subsequently, the paper enters the data cleaning stage and carry out feature engineering. Through correlation analysis, random forest importance, and recursive feature elimination (RFE) two-step screening, the paper retains the most predictive variables. Then, the paper divides the cleaned data into 80% training and 20% testing, train multiple candidate models, evaluate their performance, and select the optimal model; Ultimately, the model was used to predict user ratings online, and its prediction results continued to flow back to the front-end, forming a feedback loop that drove data collection and model retraining, achieving self-iteration and optimization of the entire recommendation system.

#### 3.2 Feature engineering

Feature engineering is the core component of this study, and the paper has constructed three types of features, as shown in Table 1.

**Table 1.** Introduction to Three Different Types of Feature Engineering

Characteristic category	Feature name	Characteristic description
User characteristics	Age, gender, occupation Average score ( $\mu_u$ ) Scoring standard deviation ( $\sigma_u$ ) Number of scores ( $n_u$ )	Basic demographic attributes Average score of user history Scoring consistency index User activity
Film Features	Type, release year Average score ( $\mu_m$ ) Scoring standard deviation ( $\sigma_m$ ) Popularity $P(m)$	Content attribute information Average score of film history Score dispersion $P(m)=n_m \times \mu_m$
Interactive characteristics	User deviation ( $b_u$ ) Film deviation ( $b_m$ ) Time characteristics	$B_u=r - \mu_u$ $B_m=r - \mu_m$ Scoring time, week, weekend

In Fig. 1, the feature engineering process starts with three original data tables of users, movies and ratings. First, four types of information are mined: statistics, categories, time series and interactions: statistical dimensions extract user average scores, standard deviations, counts and movie ratings; The category dimension makes gender, occupation, age group and film type the unique coding; The time series dimension is broken down into hours, weeks and months, and marked with weekend or not; The interaction dimension constructs the user's historical viewing sequence, scoring preference, type matching and similarity

score. Then all features are standardized and scaled uniformly to eliminate dimensional differences, and finally a fixed length vector with more than 56 dimensions (including 35 dimensions for numerical type and 15 dimensions for categorical type) is assembled to provide complete, standardized and informative input for downstream models.

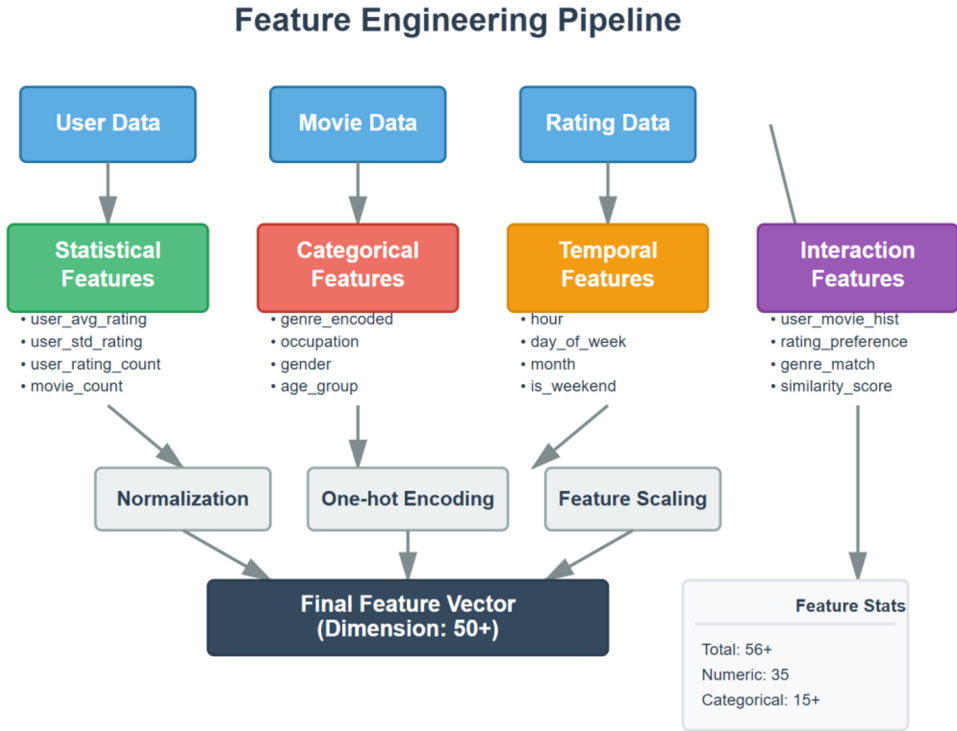


Fig. 1. Feature Engineering Flow Chart (Picture credit: Original)

### 3.3 Model Design

#### 3.3.1 Traditional machine learning models

Linear regression model: As the baseline model, the continuous value output is predicted by fitting the best linear relationship between the independent variable and the dependent variable. It assumes that the target variable is a weighted linear combination of input characteristics, and estimates the weight by minimizing the sum of squares of prediction errors.

$$\hat{y} = w_0 + \sum_i w_i x_i \quad (1)$$

Random forest model: By training a large number of decision trees at the same time and summarizing their voting (classification) or average (regression) results, the prediction accuracy and stability can be significantly improved. Each tree only randomly selects some samples and features for learning, which not only reduces the risk of over fitting, but also evaluates the importance of variables, and is highly robust to missing values and outliers.

$$\hat{y} = (1/T) \sum_t h_t(x) \quad (2)$$

Where  $T$  is the number of trees,  $h_t$  is the prediction function of the  $t$ -th tree.

XGBoost model: As an integrated learning algorithm based on gradient lifting framework, it corrects the residuals of the previous round by iteratively adding weak learners

(usually decision trees), so as to continuously improve the prediction accuracy of the model. It combines regularization, pruning, parallel computing and other technologies to significantly improve training efficiency while preventing over fitting, and is widely used in classification and regression tasks.

$$\hat{y}^i = \sum_k f_k(x_i) \quad (3)$$

Where  $f_k$  is the increment function of the  $k$ -th round.

### 3.3.2 Deep learning models

MLP: A 4-layer full connection network is designed, which uses ReLU activation function and Dropout regularization:

$$h_1 = \text{ReLU}(W_1x + b_1) \quad (4)$$

$$h^2 = \text{Dropout}(\text{ReLU}(W^2h^1 + b^2)) \quad (5)$$

$$\hat{y} = W_4h_3 + b_4 \quad (6)$$

### 3.4 Integrated learning strategy

To further improve performance, the paper has adopted three integration strategies:

First, voting integration can average the prediction results of multiple models. Then, stacking integration uses meta learners to combine basic model predictions. At last, weighted average can dynamically allocate weights according to the performance of verification set

### 3.5 Evaluation metrics

Regression task evaluation indicators:

Root mean square error (RMSE) is the root mean square error. The deviation between the predicted value and the true value is first squared, then averaged, and finally root out. The overall prediction error is measured by the same dimension. The smaller the value, the more accurate the model is.

Mean absolute error (MAE) is the average absolute error. The absolute difference between the predicted value and the true value is averaged, and the average error size is intuitively reflected by the original dimension. The smaller the value, the more accurate the model is.

$R^2$  measures the proportion of the model's interpretation of data variation. The closer it is to 1, the better the fitting is. When it is equal to 0, it is equivalent to using the average value to predict.

Classified task evaluation indicators, the accuracy rate is the proportion of correctly predicted samples in all samples, and the overall "guess right"; The accuracy rate is the proportion of real cases predicted to be true, which measures the ability of "accurate reporting"; The recall rate is the proportion of all positive examples that have been successfully identified, which measures the ability to "grasp the whole"; In addition, the F1 score represents the harmonic average of the accuracy rate and recall rate, combining "quasi" and "full"; the area under the AUC-ROC curve is the area under the ROC curve, and the larger the value ( $\leq 1$ ), the stronger the model's ability to distinguish between positive and negative samples.

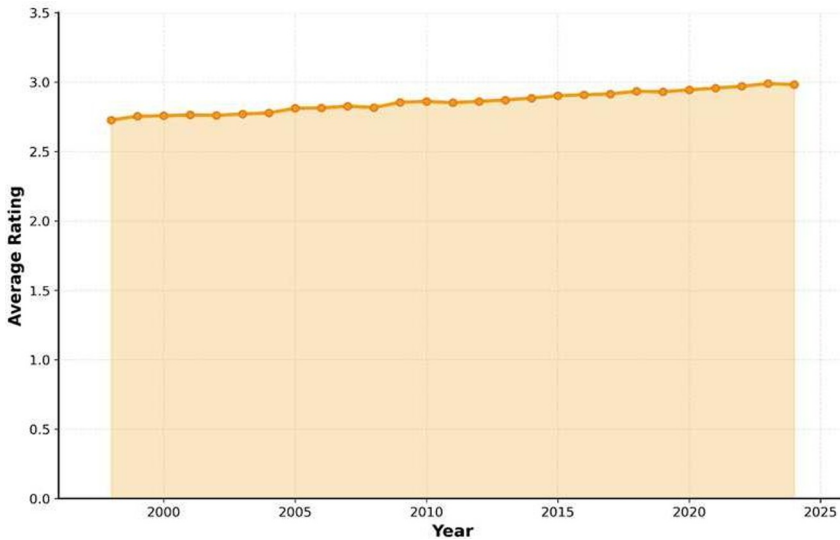
## 4 Experimental setup

## 4.1 Dataset

The experiment uses the widely used MovieLens 100K dataset [11]. A recent reproducibility study confirmed that the sparsity and offset distribution of ML-100K are still highly consistent with the real platform logs after 2020, so it is still widely used for algorithm verification [12]. This dataset contains 100000 rating records of 1682 movies by 943 users. The scoring range is 1-5 points. The data set also includes the user's age, gender, occupation information, as well as the film type, release time and other attributes.

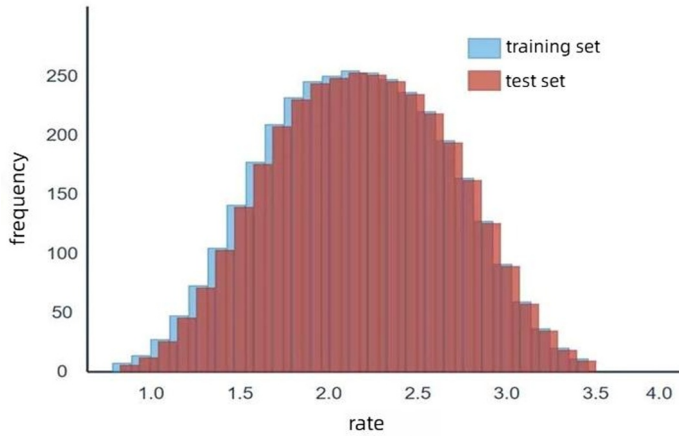
The dataset exhibits several key statistical properties. The average score is 3.53, indicating a general tendency for positive ratings. The rating density is relatively low at 6.3%, reflecting the inherent sparsity typical of user-item interaction data. On average, each user rated 106 items, while each movie received an average of 59 ratings, providing a balanced view of user and item activity within the system.

Fig.2 shows the trend of average ratings over time. From the graph, it can be seen that the average rating showed a slow upward trend from 2000 to 2025, with rating values fluctuating roughly between 2.7 and 3.0. This indicates that over time, there has been a slight increase in average ratings, but the magnitude of the increase is not significant and overall remains relatively stable. This indicates that the audience's satisfaction with watching movies continues to grow slowly.



**Fig. 2.** Trend chart of average rating over time(Picture credit: Original)

Fig. 3 shows the frequency distribution of the training and testing sets on the variable 'rate'. From figure 3, it can be seen that the distribution patterns of the two datasets are similar, both showing characteristics of approximate normal distribution, with peaks concentrated around 2.0. This indicates that the distribution of the training and testing sets on this variable has good consistency, which helps the model achieve stable performance on the testing set. Specifically, the frequency distribution curves of the training set (blue part) and the testing set (red part) overlap significantly, indicating that they are relatively close in data distribution. This is a positive signal for the generalization ability of the machine learning model.



**Fig. 3.** Comparison Diagram of Training Test Set Distribution—Consistency test of scoring distribution of dataset (Picture credit: Original)

## 4.2 Experimental environment and parameter settings

Experimental environment: Python 3.8, mainly using scikit-learn, XGBoost, and TensorFlow frameworks. The data set is divided into training set and test set according to 8:2 ratio, and super parameter optimization is carried out by 50 fold cross validation.

In this modeling process, the paper sets the key hyperparameters for three core algorithms. For the Random Forest model, the paper sets the number of trees to 100 and limited the maximum depth to 10, aiming to maintain model complexity while preventing overfitting. For the XGBoost model, the paper used a learning rate of 0.1 and built an ensemble with 150 trees, while setting the maximum depth of each tree to 6, to balance training speed and model performance. As for the Multilayer Perceptron (MLP), the paper designed a network architecture with three hidden layers containing 256, 128, and 64 neurons respectively. During training, the paper used a learning rate of 0.001 and a batch size of 64 to optimize the model.

## 4.3 Comparative method

To fully verify the effectiveness of the proposed method, the paper selects four representative baselines to compare one by one: first, use the global average forecast as the lower limit of "no information", and then use the user average forecast to capture individual preferences; Then, the article based collaborative filtering is introduced to mine neighborhood similar signals. Finally, the SVD matrix decomposition is used to introduce low rank hidden factors to achieve the ability coverage from simple statistics to classical cryptic semantic models, so as to systematically measure the improvement space of the new method.

# 5 Results and Analysis

## 5.1 Model performance comparison

Table 2 shows the performance of each model on the test set. The experimental results show that XGBoost achieves the best performance in the single model (RMSE=0.862), which is

15.8% higher than the baseline method. The deep learning model MLP also performs well (RMSE=0.871), which proves the effectiveness of nonlinear modeling.

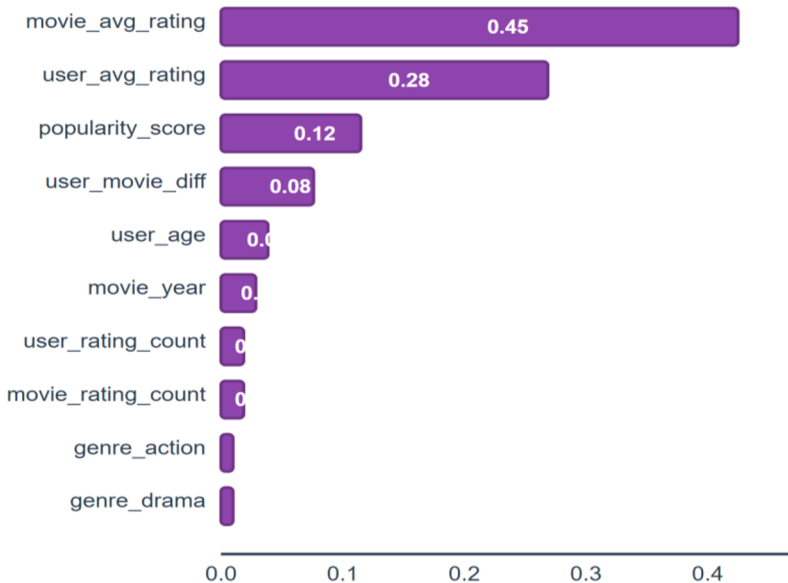
**Table 2.** Performance comparison results of each model

model	RMSE	MAE	R <sup>2</sup>	Training time (s)
Global Average	1.126	0.934	0.000	0.01
Average users	1.024	0.823	0.285	0.05
linear regression	0.945	0.748	0.523	0.12
Random forest	0.883	0.681	0.612	2.34
<b>XGBoost</b>	<b>0.862</b>	<b>0.664</b>	<b>0.634</b>	3.56
MLP	0.871	0.672	0.625	15.23
<b>Integration model</b>	<b>0.836</b>	<b>0.645</b>	<b>0.652</b>	-

The integrated learning strategy further improved the prediction performance. Stacking integration reached the lowest RMSE (0.836), which was 3.0% higher than the best single model. This verifies the value of model complementarity.

### 5.2 Feature Importance Analysis

By analyzing the importance of random forest characteristics and the SHAP value, the paper identified the key factors affecting the prediction. Figure 4 shows the contribution of the top 10 important features.



**Fig. 4.** Importance analysis diagram (Picture credit: Original)

The five most important characteristics are as follows.

Average score of film history (importance: 0.234) reflect the overall quality of the film. Average user rating (importance: 0.186) reflect the user's scoring tendency. Number of films scored (importance: 0.124) represent the popularity of movies. Standard deviation of user rating (importance: 0.098) reflect the consistency of user ratings. Movie Type\_Drama (Importance: 0.076) specific types of impacts.

These findings indicate that historical statistical information can predict future scores better than static attribute characteristics, which is consistent with the basic assumption of collaborative filtering.

### **5.3 Performance of classification tasks**

In the second category task (whether the prediction is popular or not), each model also performs well. XGBoost classifier achieves 82.4% accuracy and 0.891 AUC value. The confusion matrix analysis shows that the model performs better in identifying popular movies (recall rate 85.2%), which is of positive significance for the practical application of the recommendation system.

### **5.4 Model interpretability**

Through SHAP value analysis, the paper can interpret single prediction results. Take a specific case as an example: for user A (25 year old male, preferring action movies), predicted movie B (science fiction action movies, historical score 4.2) scored 4.1 points. SHAP analysis shows that the main positive contribution factors include: high historical score of movies (+0.52), matching between users and movie types (+0.31), and high popularity of movies (+0.18). Negative factors include: the user's historical score is low (-0.23). This interpretability helps to understand the decision-making process of the model.

### **5.5 Error analysis**

The residual analysis shows that there is a certain deviation in the prediction of the extreme score (1 point and 5 points), and the model tends to return to the mean. This may be due to the lack of samples for extreme scoring and insufficient model learning. The Q-Q diagram shows that the residual basically conforms to the normal distribution, which verifies the rationality of the model assumptions.

The time dimension analysis found that the prediction error of the model for new movies (RMSE=0.912) was higher than that of old movies (RMSE=0.847), which reflected the existence of the cold start problem. In the future, this problem can be improved by introducing content features and transfer learning.

## **6 Discussion**

### **6.1 Key findings**

The experimental results of this study reveal several important findings.

First, the effectiveness of integrated learning has been fully verified. By combining multiple basic models, not only the prediction accuracy is improved, but also the stability of the model is enhanced. In particular, Stacking integration effectively captures the complementarity of different models through meta learners.

Secondly, the importance of feature engineering cannot be ignored. Statistical characteristics (such as historical average score) contribute more to prediction than original characteristics (such as user age). This suggests that the paper should pay attention to the design and selection of features in practical applications.

Third, the tradeoff between model complexity and performance is worth considering. Although the deep learning model has achieved good results, its training time is more than 4 times that of XGBoost. In actual deployment, accuracy and efficiency need to be balanced according to specific scenarios.

## 6.2 Practical enlightenment

Based on the research results, the paper suggests that the film and television recommendation system adopt the "hierarchical modeling+real-time update+interpretable+cold start" four in one strategy: invest complex high-capacity models in active users and popular movies to dig deep into fine preferences, and use lightweight and efficient models to guarantee coverage for long tail users and popular movies; At the same time, a real-time feature calculation pipeline is set up to refresh the statistical features in seconds with the behavior and maintain the timeliness; When returning to the recommendation list, synchronously generate explanatory statements such as "because you have seen ××" and "×× likes similar to you", so as to improve user trust and click intention; Finally, combine content features such as film metadata and text labels with transfer learning, quickly borrow similar old users/old film knowledge for newly registered or newly launched films, and achieve high-quality recommendation at the cold start stage.

## 6.3 Limitations

This research is still limited by four points: MovieLens-100K has only 100000 levels of data, which is difficult to map the sparse and long tail of ten million level real systems; Time series modeling only extracts simple features such as "hours and weeks", and does not depict the complex evolution of interest with seasons or plot hotspots; The evaluation dimension is single, and only the high and low scores serve as popularity, without introducing more direct behavioral signals such as viewing duration and completion rate. WWW 2023 proposes a multi task framework to simultaneously predict scores, click rates and viewing duration, which provides a feasible path for subsequent more fine-grained 'popularity' modeling; Moreover, the age and cultural background of users in the dataset are relatively homogeneous, and the generalization ability of the model in cross regional and cross ethnic scenarios has yet to be verified [13].

## 7 Conclusion

This study proposes a comprehensive machine learning solution to the problem of predicting the popularity of film and television content. By integrating multi-dimensional features, comparing multiple models, and adopting integrated learning strategies, the paper has achieved significant performance improvement on MovieLens dataset. The experimental results show that XGBoost and the integrated model achieve a good balance between prediction accuracy and efficiency. Through the interpretability analysis of the model, the paper identified the key factors affecting the prediction, guiding practical application.

The future work will be carried out along five main lines: first, deep feature learning, using graph neural networks to automatically mine high-order structural signals on user item interaction graphs; The second is multi task learning, which jointly optimizes multiple goals

such as scoring prediction, click rate and viewing duration to achieve comprehensive benefits of one training; The third is the introduction of reinforcement learning, which regards recommendation as a sequential decision-making process and rewards long-term user satisfaction rather than a single click; Fourth, cross domain migration, transferring the expressions of interest and strategies learned in the film field to new media such as TV dramas and short videos to reduce cold start costs; The fifth is to embed fairness constraints to ensure that the system provides unbiased exposure to different ages, genders, geographical groups and various types of content. With the continuous evolution of AI technology, the film and television recommendation system is expected to be upgraded iteratively in the above direction and become more intelligent and personalized, which not only brings users the ultimate experience of "thousands of people and faces", but also releases greater commercial potential for the platform.

## References

1. C. J. Gomez-Urbe, N. Hunt, & T. Jebara. Recommender systems and user retention: a large-scale randomized study at Netflix. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp.3201 – 3211. (2022)
2. Y. Koren, R. Bell, & C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30-37. (2009)
3. F. Ricci, L. Rokach, & B. Shapira. Recommender systems handbook. Springer. (2015)
4. X. He, L. Liao, H. Zhang, et al. Neural collaborative filtering. In: Proceedings of the 25th International Conference on World Wide Web (WWW). pp.173-182. (2017)
5. S. Zhang, L. Yao, A. Sun, et al. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys*, 52(1): 1-38. (2019)
6. S. Zhang, L. Yao, A. Sun, & H. Wang. Deep learning for recommender systems: A 2023 survey and new perspectives. *ACM Computing Surveys*, 56(2): 1 – 42. (2023)
7. H. T. Cheng, L. Koc, J. Harmsen, et al. Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS). pp.7-10. (2016)
8. T. Chen & C. Guestrin. XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). pp.785-794. (2016)
9. S. M. Lundberg & S. I. Lee. A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems (NeurIPS). pp.4765-4774. (2017)
10. F. M. Harper & J. A. Konstan. The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4): 1-19. (2015)
11. B. Liu, H. Yu, Y. Xiao, et al. AutoFeature: Searching for feature-level importance in recommender systems. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp.1555 – 1564. (2021)
12. J. Beel, V. Shah, & D. Doerr. Is MovieLens 100K still relevant? A reproducibility study on collaborative filtering. *ACM Transactions on Interactive Intelligent Systems*, 10(4): 1 – 21. (2020)
13. Y. Zhou, X. Wang, C. Zhu, et al. Multi-task learning of rating, click-through rate and watch-time for video recommendation. In: Proceedings of the ACM Web Conference 2023 (WWW). pp.902 – 912. (2023)