UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

2023-2024

Machine Learning

Movie Ratings Predictions

Jintao Ma

Xianyun Zhuang

Supervisor

MARTA ARIAS VICENTE

June 2024

Contents

Abstract

In this report, we had implemented the Movie ratings prediction project ,according to the machine learning course content and some suggestions from the professor.It was a typical regression prediction task.It was more in line with the types of problems we had studied over the course.

(i) In Chapter1, we tried to understand related previous work and study different methods.

(ii) In Chapter2, We made exploratory data analysis of our dataset and implemented pre-processing work.Data cleaning was a critical step in this process.

(iii) In Chapter3, we drew a lot of diagrams to show relationships among many attributes and their own characteristics.Visualization work included box plots,histograms,correlation heatmap and scatter plots.

(iv) In Chapter4, we had selected key features.The dataset was split into training and test sets with an 80:20 ratio. Additionally, we performed five-fold cross-validation on the training set.Based on these, we used 4 common machine learning models to predict movie ratings. They were linear regression with and without regularization, SVMs, MLPs and random forests.

(v) In Chapter5, we compared key metrics from many different models,then chose the best model and made analysis of feather importance.

At last, we uploaded the code to Github: GitHub Repository

1 Related Previous Work

For our project, here are some related previous tutorials[Ng14] and works that can provide us with guidance and references:

1.1 Exploratory Data Analysis and Feature Selection

We found a GitHub project that delved into the preprocessing, exploratory data analysis (EDA), and feature selection of the TMDB-5000 movie dataset. This project employed various machine learning models such as logistic regression, decision trees, and XGBoost to predict movie success [Gos24].

1.2 Movie Popularity and Target Audience Prediction

On IEEE Xplore[Yan19], we came across a study that investigates predicting movie popularity and identifying target audiences using content-based features. This research highlighted the importance of features like genre, cast, crew, and budget in predicting a movie's success (cs229) .

1.3 Sentiment Analysis and Multimodal Data

The previous study also combined multimodal data, including textual features from movie synopses and visual features from posters, to predict movie ratings. This approach demonstrated the value of integrating different types of data to improve prediction accuracy.

2 Exploratory Data Analysis

2.1 Data Set

The goal of this project is to explore the Movie Data Set(TMDB) [1]to discover trends in the data and predict average rating based on some important attributes. The dataset has 4803 instances and 20 attributes that are a combination of categorical and real values, which provides room for experimentation with different models and approaches to data pre-processing. There are also some other relevant papers that make use of this dataset as identified. The dataset and all reproducible code can be also found in the GitHub [2] repository.

2.2 Pre-processing

2.2.1 Data Cleaning

First, we displayed the first few rows of the dataset to get an initial understanding of its contents. Next, we showed the basic information of the dataset, including each column's name, data type, and non-null value count, which helped determine the completeness of the data. Then, we generated a statistical summary of the numerical features, calculating metrics such as mean, standard deviation, minimum, and maximum values to gain a deeper insight into the data distribution. To ensure data type correctness, we also checked the data type of each column. After that, we identified and printed rows containing missing values, highlighting incomplete parts of the data. We then checked for and printed duplicate rows to identify and address data redundancy. Finally, we filled all missing values with 0 and removed all duplicate rows, ensuring the data was clean and consistent, and ready for subsequent data analysis and machine learning model training. Through

[1]https://www.themoviedb.org
[2]https://github.com/zhuangxianyun/ML

these steps, we cleaned and standardized the data, ensuring the foundational data quality for further processing.

2.2.2 Feather Selection

We are going to complete the movie rating prediction task. In the dataset, the minimum value of the movie average rating is 0.0, and the maximum value is 10.0.Then we just chose 'budget', 'runtime', 'popularity', 'revenue', and 'vote_count' as our key features. In the movie rating prediction task, processing and analyzing numerical data were crucial for several reasons. Standardization (making data have a mean of 0 and variance of 1) and normalization (scaling data to a range, typically 0 to 1) were important because they helped ensure that features contributed equally to the model, prevented large-scale features from dominating the results, and improved model stability and training speed. These preprocessing steps were particularly essential for algorithms sensitive to data scales, such as linear regression and neural networks. Additionally, they enhanced model performance by ensuring the model adapted better to the data distribution, reducing the risk of overfitting, and improving feature interpretability.

Attribute	**budget**	**runtime**	**popularity**
Description	Movie production cost	Movie duration	Level of popularity
Data Type	int64	float64	float64

Attribute	**revenue**	**vote_count**
Description	box office revenue	Number of ratings
Data Type	int64	int64

Table 2.1: Feature Selection

Additionally, we excluded many non-numeric features. The processing of non-numeric features might introduce noise and increase the risk of model overfitting. Non-numeric features could significantly increase the data dimensionality, leading to higher computational costs. For example, one-hot encoding of the 'production_countries' categorical feature would add a large number of new features. This may impact models such as linear regression.Moreover, some non-numeric features, such as 'keywords' and 'overview', required complex text processing steps, adding to the data processing complexity. Lastly, there were some highly subjective features that were not suitable for regression prediction, such as 'tagline' and 'original_title'. Ideally, we would incrementally introduce and test non-numeric features in future modeling processes to comprehensively evaluate their impact on model performance.

3 Data Visualization

Visualizations are utilized to identify potential trends and hypotheses for further testing. They also offer valuable insights into which models might be appropriate to test and which attributes are likely to have the most significant impact on predictions.

In addition to the initially selected five features, we also introduced the average movie rating/vote as part of the analysis. The average movie vote provided a baseline value, allowing us to assess the model's prediction bias. By analyzing the average movie vote, we could better understand the distribution of the rating data, such as the presence of significant skewness or outliers, to make appropriate adjustments accordingly. The average movie vote helped us evaluate the correlation between the selected features and the ratings, which facilitated feature selection and model optimization. Furthermore, by incorporating the average movie vote into the analysis, we were able to identify areas for model improvement by comparing the prediction errors of the model against the actual average ratings, thereby enhancing the model's accuracy.

3.1 Box Plots

We drew box plots as Figure3.7 to show the distribution of variables and their outliers. When encountering outliers in movie data, such as high budget and high ratings, we did not simply remove these points because they were reasonable and common phenomena in the film industry, reflecting real market trends and audience preferences. Retaining these data points enhanced the predictive power and practical relevance of the model, especially when using models that were robust to outliers (such as decision trees and random forests) or applying logarithmic transformations to mitigate the impact of high-value points, thereby better analyzing and predicting movie box office performance.

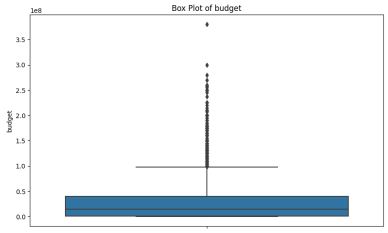

Figure 3.1: budget

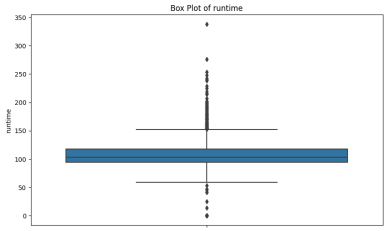

Figure 3.2: runtime

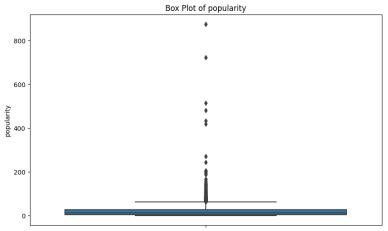

Figure 3.3: popularity

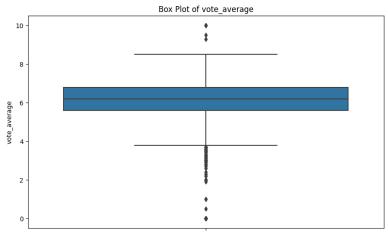

Figure 3.4: vote_average

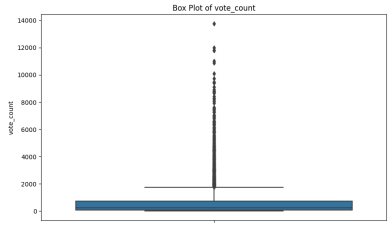

Figure 3.5: vote_count

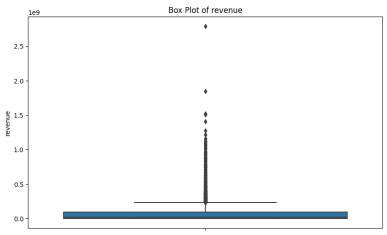

Figure 3.6: revenue

Figure 3.7: Box Plots

3.2 Histograms

Histograms were used to display the distribution of different variables as Figure3.14 shown. The histograms showed significant insights into the distribution of various attributes in the movie dataset. The vote_count, budget, revenue, and popularity attributes all exhibited highly right-skewed distributions, indicating that most movies had relatively low values in these categories, while a few movies had exceptionally high values, acting as outliers. In contrast, the vote_average attribute showed a roughly normal distribution centered around a mean value of approximately 6 to 7, with fewer outliers on both ends. The runtime attribute displayed a slightly right-skewed distribution, with most movies having runtimes between 80 and 120 minutes, but some outliers had significantly longer runtimes. These distribution patterns suggested that while high budget, high revenue, and high popularity were common phenomena in the film industry, they were not the norm. These distributions were crucial for effective data preprocessing and feature engineering in predictive modeling of movie box office performance.

3.3 Correlation Heatmap

We created a correlation heatmap as Figure3.15 to show the correlations between different movie indicators. The correlation matrix revealed significant relationships between various attributes in the movie dataset. Higher budget movies tended to generate higher revenue and receive more

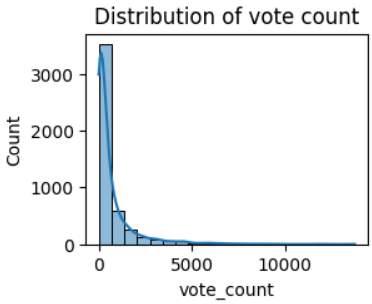

Figure 3.8: vote_count

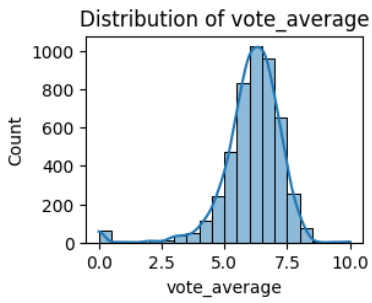

Figure 3.9: vote_average

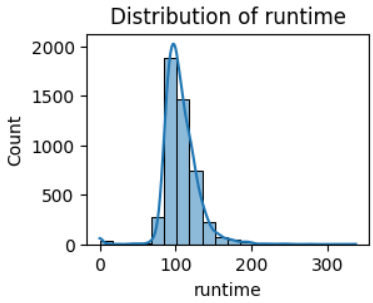

Figure 3.10: runtime

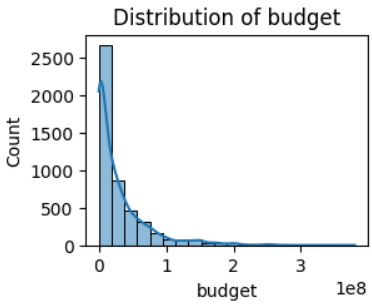

Figure 3.11: budget

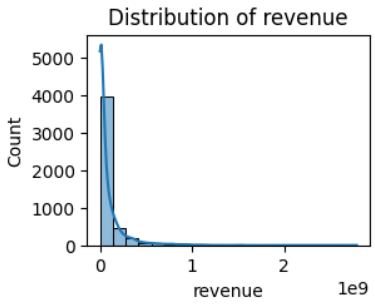

Figure 3.12: revenue

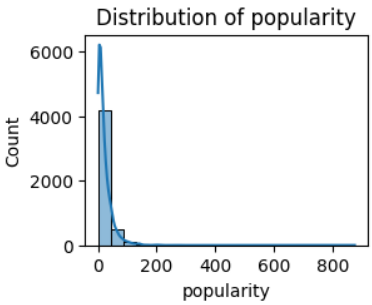

Figure 3.13: popularity

Figure 3.14: Distribution Plots

votes, as indicated by strong positive correlations between budget and revenue (0.73) and budget and vote count (0.59). Popularity was strongly associated with vote count (0.78) and revenue (0.64), suggesting that more popular movies garnered more votes and higher revenue. Runtime showed a moderate correlation with vote average (0.38), indicating that longer movies might receive slightly higher ratings. The correlations between vote average and other attributes such as budget (0.093), popularity (0.27), vote count (0.31), and revenue (0.2) were weaker, indicating that average ratings were less influenced by these factors.

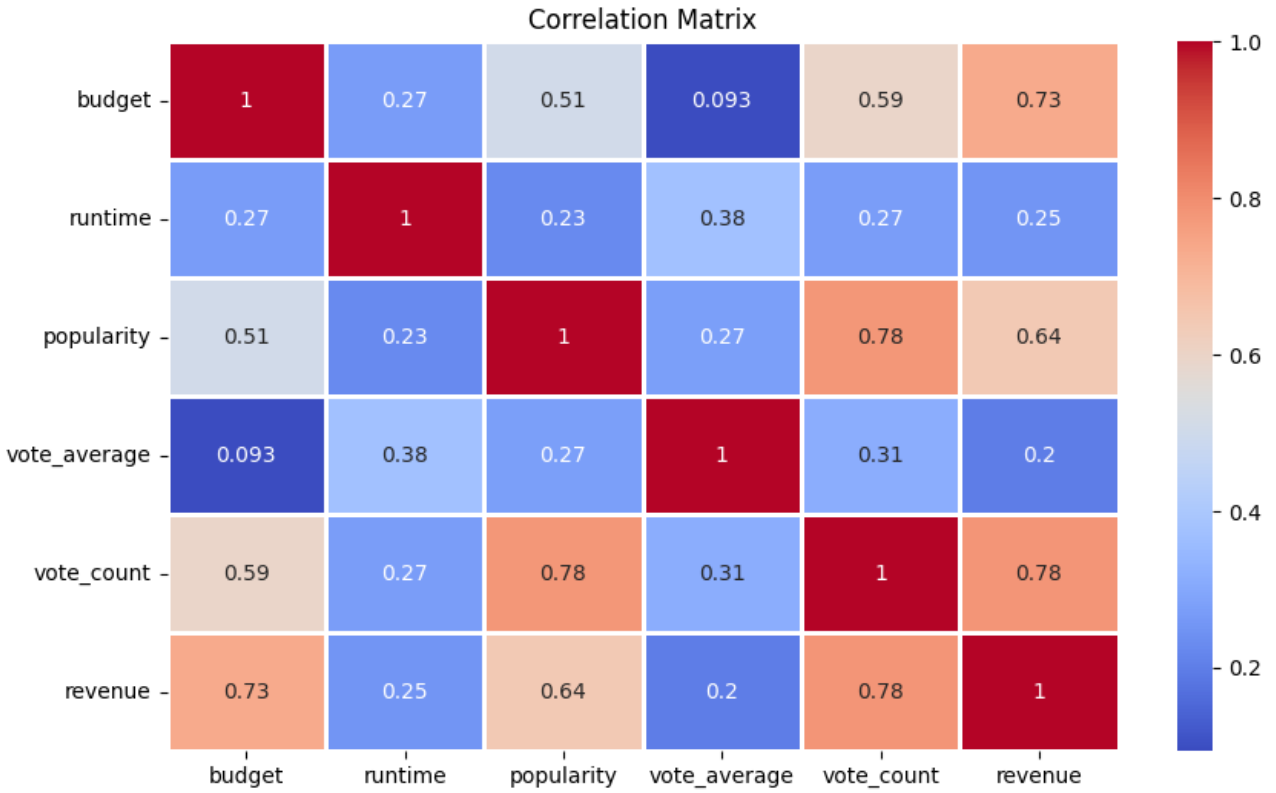

Figure 3.15: correlation matrix

3.4 Scatter Plots with Regression Lines

Scatter plots with regression lines were drawn to show the linear relationships between variables.The Figure3.21 collectively revealed insightful relationships between various movie attributes (budget, vote count, popularity, revenue, and runtime) and the average vote (rating). Analyzing these plots, we observed a slight positive correlation between budget and vote average, indicating that movies with higher budgets tended to have slightly higher ratings, although the correlation was not very strong. The relationship between vote count and vote average showed a moderate positive correlation, suggesting that movies receiving more votes generally had higher average ratings. This was indicative of a broad audience appeal often correlating with better ratings. Popularity also showed a positive correlation with vote average, with more popular movies tending to have higher ratings. However, as popularity increased significantly, there was also an increase in variance, suggesting that not all highly popular movies received high ratings. The revenue and vote average relationship displayed a slight positive correlation, implying that higher revenue movies tended to have better ratings, though the correlation remained weak. Similarly, runtime exhibited a weak positive correlation with vote average, showing that longer movies might receive slightly higher ratings, but the wide distribution indicated that runtime alone was not a strong predictor of average ratings.

These insights highlighted that while budget, vote count, popularity, revenue, and runtime had some influence on movie ratings, their individual predictive power varied. Vote count and popularity emerged as more reliable predictors of movie ratings, whereas budget, revenue, and runtime played a comparatively smaller role. Finally, we still decided to reserve these five attributes as our key features.Because certain correlated features may become important when combined with other features, the interactions between multiple features can enhance the predictive power of the model.

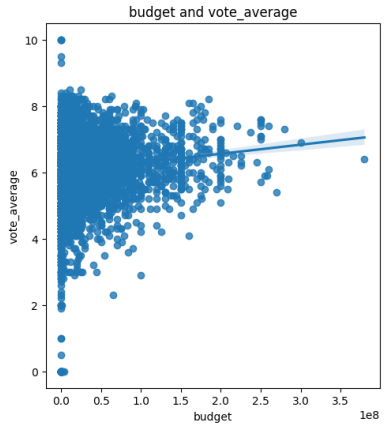

Figure 3.16: budget

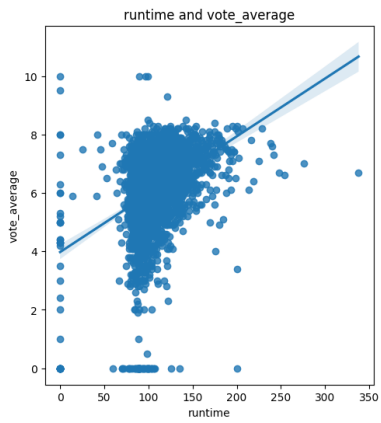

Figure 3.17: runtime

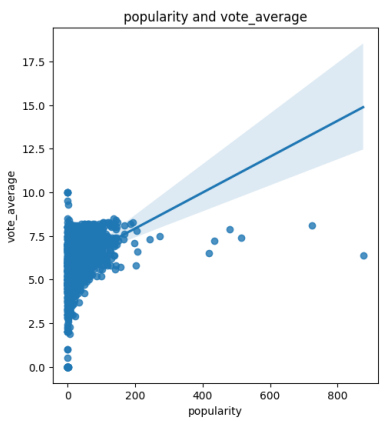

Figure 3.18: popularity

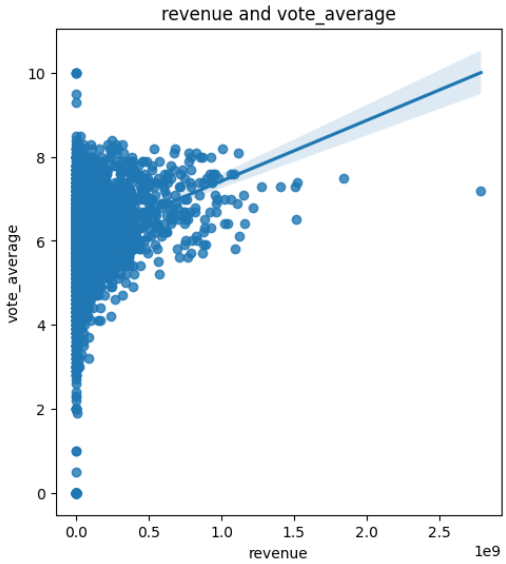

Figure 3.19: revenue

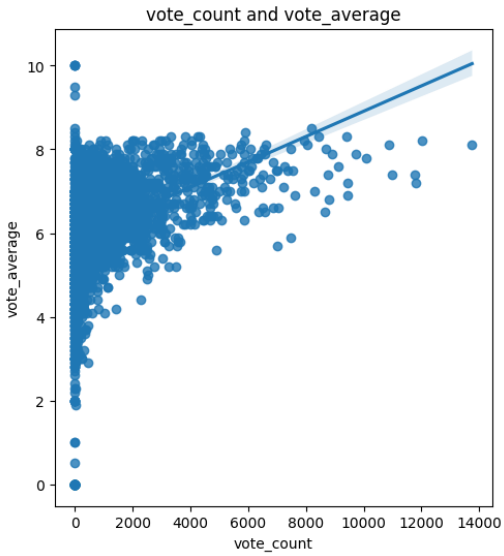

Figure 3.20: vote_count

Figure 3.21: Scatter Plots

4 Predictive Modeling

4.1 Data Set Split

After data preprocessing, we split the dataset into training and test sets to prevent data leakage and maintain data processing consistency. This ensured that the model's evaluation results on the test set were closer to real-world performance. Preprocessing steps, such as filling missing values and encoding features, were applied to the entire dataset for consistency. We used the train_test_split function to split the dataset in an 80/20 ratio and ensured consistent results with random_state=42. Further standardization and scaling steps guaranteed the independence of the training and test sets, preventing data leakage.

4.2 Modeling Methods Considered

The models tested include:

• linear regression, with and without regularization

• SVMs for regression

• MLPs with different architectures

• random forests for regression

4.3 Model Processing

Each model was tested with various parameters. The Linear Regression and random forest models were built using default settings, followed by cross-validation, and then hyperparameter tuning.In SVMs,we selected hyperparameters randomly,followed by cross-validation, and then hyperparameter tuning.In MLPs,we also used default settings, followed by cross-validation.then we modified number of hidden layers and made hyperparameter tuning seperately.

4.4 Validation Protocol

During model validation, k-Fold cross-validation was chosen as the final validation method due to the limited sample size[Bis06]. Using five-fold cross-validation to evaluate linear regression, support vector machine regression, multi-layer perceptron, and random forest models helped improve the stability and reliability of the evaluations. By dividing the training set into five subsets, each time using four subsets for training and the other subset for validation, this method reduced the uncertainty caused by a single data split and made full use of the data, especially when the sample size was limited. Five-fold cross-validation also better observed the performance of the models on different datasets, reducing the risk of overfitting and underfitting. Specifically, linear regression models were sensitive to data, and five-fold cross-validation helped evaluate their stability and regularization parameter selection; SVM models required tuning of kernel functions and regularization parameters, and cross-validation provided stable parameter evaluation; MLP, being a complex neural network model, had its stability ensured across different architectures and parameters through five-fold cross-validation; random forest models, consisting of multiple decision trees, had their overall performance and the contribution of each tree evaluated through cross-validation. Thus, five-fold cross-validation provided these models with a stable, reliable, and effective evaluation mechanism, helping to select the best parameters and ensuring the generalization ability of the models.

4.5 Parameters Choices

4.5.1 Linear Regression

We used linear regression model with and without regularization[DB08].There were three methods:linear regression,Ridge regression and Lasso regression.In ridge regression and lasso regression, the alpha parameter controlled the strength of the regularization term. "Best Ridge Alpha: 100" indicated that ridge regression performed best with an alpha of 100, requiring stronger L2 regularization. "Best Lasso Alpha: 0.1" indicated that lasso regression performed best with an alpha of 0.1, requiring weaker L1 regularization. Different models required varying regularization strengths, and we determined the optimal alpha through cross-validation and grid search.

• Best Ridge Alpha: 100

• Best Lasso Alpha: 0.1

4.5.2 SVMs

In the Support Vector Machine (SVM) regression task,we used the random search to find the optimal hyperparameters[VGS97]. The kernel function (kernel) was set to 'rbf' (Radial Basis Function) to handle non-linear relationships; the -insensitive loss function threshold (epsilon) was 0.1, determining the model's tolerance for error; and the regularization parameter (C) was 1, controlling the penalty for error terms, balancing model complexity and error. These hyperparameter combinations optimized the model's generalization capability and performance.

Hyperparameters of SVMs	**kernel**	**epsilon**	**C**
randomly generated	linear	0.01	100
optimal	rbf	0.1	1

Table 4.1: Hyperparameters of SVMs

4.5.3 MLPs

We compared different hidden layer architectures to show the best hyperparameter settings for each model[LeC+98]. These hyperparameters were determined through the hyperparameter tuning process, such as grid search, aiming to find the most suitable model configuration for specific tasks. Each configuration's activation function and regularization parameter varied, reflecting different methods to optimize model performance in various situations.

Best Parameters	**Activation**	**Alpha**	**Hidden Layer Sizes**	**Max Iterations**
One Hidden Layer	relu	0.01	(100,)	500
Two Hidden Layers	relu	0.001	(150, 100)	500
Three Hidden Layers	tanh	0.01	(150, 100, 50)	500

Table 4.2: Hyperparameters of the Best Models of MLPs

4.5.4 Random Forests

The hyper-tuned random forest model had a max depth of 10 with 300 estimators, and used a minimum of 4 samples per leaf and a minimum of 2 samples to split a node[Bre01]. These parameters ensured that each tree was relatively deep and used a different subset of features to predict the target. The best performing random forest used the following parameters:

• Max depth: 10

• Estimators: 300

• Min samples per leaf: 4

• Min samples to split: 2

5 Conclusions

5.1 Results

We used MAE, MSE, RMSE, R^2,and Explained Variance as evaluation metrics because they provided different perspectives on model error and explanatory power. In most cases, we found that the metrics after cross-validation did not differ significantly, indicating that the model had good generalization capability. However, the metrics improved significantly after hyperparameter tuning, suggesting that we identified the most suitable set of hyperparameters for the dataset. It allowed the model to better capture the patterns in the data, thereby enhancing overall performance.

Model	MAE	MSE	RMSE	R^2	Explained Variance
linear regression	0.7030	1.1089	1.0530	0.2116	0.2116
linear regression CV	0.7050	1.1153	1.0527	0.2094	0.2111
Ridge regression Default	0.7030	1.1089	1.0530	0.2116	0.2116
Ridge regression Default CV	0.7050	1.1153	1.0527	0.2094	0.2111
Ridge regression CV+Optimized	0.7050	1.1153	0.2094	0.2111	1.0527
Lasso regression Default	0.7050	1.1136	1.0553	0.2083	0.2083
Lasso regression Default CV	0.7063	1.1171	1.0537	0.2077	0.2092
Lasso regression CV+Optimized	0.7050	1.1149	1.0525	0.2096	0.2113
SVM Random	0.6933	1.1291	1.0626	0.1973	0.2085
SVM Random CV	0.6960	1.1368	1.0623	0.1953	0.2074
SVM CV+Optimized	0.6487	1.0146	1.0033	0.2823	0.2907
MLP Default	0.6412	0.9999	1.0000	0.3368	0.3388
MLP Default CV	0.6583	0.9914	0.9925	0.2972	0.2985
One Hidden Layer CV+Optimized	0.6554	0.9882	0.9908	0.2996	0.3011
Two Hidden Layers CV+Optimized	0.6503	0.9788	0.9861	0.3063	0.3086
Three Hidden Layers CV+Optimized	0.6500	0.9786	0.9864	0.3054	0.3090
Random Forest Default(Overfitting)	0.2186	0.0935	0.3057	0.9336	0.9336
Random Forest Default (Test)	0.5757	0.6382	0.7989	0.5768	0.5769
Random Forest Default CV	0.5921	0.6774	0.8225	0.5105	0.5115
Random Forest CV+Optimized	0.5782	0.6352	0.7965	0.5415	0.5422

Table 5.1: Comparison of key metrics among Different Models

It was noteworthy that the Default Random Forest model without 5-fold cross-validation exhibited overfitting on the training set. It was strongly validated by comparing the metrics obtained from the training set and the test set.Because the former metrics significantly outperformed the latter.

5.2 Final Model Estimation and Performance

The final model was run using Random Forest with feature selection applied on the dataset and and adjusting to the optimal hyperparameters. These results indicated that the model performed better on the test set compared to the cross-validation results. This suggested that the model had good generalization capability to new, unseen data. The feature selection and model optimization effectively enabled the model to capture data patterns and apply them to the new dataset.

Best Model	MAE	MSE	RMSE	R^2	Explained Variance
Cross-Validation Results	0.5782	0.6352	0.7965	0.5415	0.5422
Test Set Results	0.5518	0.5787	0.7607	0.6162	0.6163

Table 5.2: Results of the Best Model Random Forest

The feature importance of this model was shown in Figure5.1. It was evident that the "vote_count" feature had the highest importance score of 0.6327, significantly higher than the other features. This indicated that "vote_count" played the most crucial role in the prediction model. Following that were "runtime" (0.1735) and "budget" (0.0931), which also had some impact on the model. In contrast, "popularity" (0.0673) and "revenue" (0.0334) had lower importance scores. Based on the results, it could be considered to retain features with higher importance scores while performing feature selection on those with lower scores to reduce the model's complexity in the future.

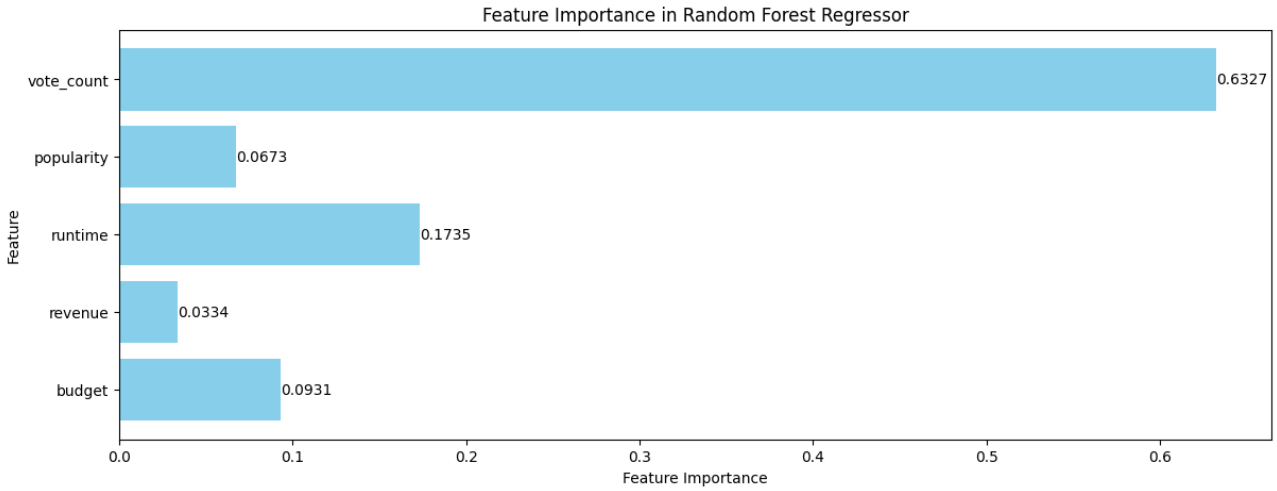

Figure 5.1: feather importance of Random Forest

Reference

Articles

[Bre01] Breiman, Leo (2001). "Random Forests". In: *Machine Learning* 45.1, pp. 5–32. DOI: 10.1023/A:1010933404324.

[LeC+98] LeCun, Yann et al. (1998). "Efficient BackProp". In: DOI: 10.1007/3-540-49430-8_2.

[VGS97] Vapnik, Vladimir N., Steven E. Golowich, and Alex J. Smola (1997). "Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing". In: 9, pp. 281–287.

[Yan19] Yang, Yichen (2019). "Predicting Movie Ratings with Multimodal Data". In: URL: https://api.semanticscholar.org/CorpusID:209517808.

Books

[Bis06] Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[DB08] Dobson, Annette J. and Adrian G. Barnett (2008). *An Introduction to Generalized Linear Models*. 3rd. CRC Press. ISBN: 978-1-58488-950-2.

Miscellaneous

[Gos24] Goshtasbi, A. (2024). *Arshiagosh/TMDB-5000-Movie-Dataset-Analysis-and-Modeling*. [Accessed 3 Jun. 2024]. URL: https://github.com/Arshiagosh/TMDB-5000-Movie-Dataset-Analysis-and-Modeling.

[Ng14] Ng, Andrew (2014). *Machine Learning*. Accessed: 2024-06-03. URL: https://www.coursera.org/learn/machine-learning.