COMP30027 Project 2 Report

Anonymous

1. Introduction

In today's film industry, accurately predicting a movie's market performance is crucial for film producers, distributors, and potential audiences. IMDB (Internet Movie Database) provides a comprehensive movie rating system to evaluate the quality of films across multiple dimensions. Therefore, the ability to predict movie ratings is of great significance and value for formulating market strategies and predicting the commercial success of movies.

The objective of the project is to construct and analyze several machine learning models that use movie-related features extracted from IMDB (such as gross, genres, director, actors, and Facebook likes, etc.) to predict movie ratings. Each model has unique characteristics that could ultimately lead to different prediction outcomes. We aim to combine the theoretical knowledge from courses with practical data analysis methods to construct and solve a real-world prediction problem through this task.

This report will also describe the data processing techniques, feature selection processes, specific evaluation methods, and the comparison among the diverse machine learning models. Through comparative experiments and in-depth analysis, we will demonstrate the performance of these movie rating prediction models and explore the underlying reasons.

2. Methodology

2.1 Preprocessing

This study used a dataset of movie evaluations based on publicly accessible IMDB, with some data and advanced text preprocessed. There are 25 movie features with each movie assigned one of the five rating levels: 0, 1, 2, 3, and 4 in the training dataset. After the overall inspection with the correlation heat map (Figure 1), we further applied some preprocessing techniques to improve the models' training performance and prediction accuracy. Here are several primary adjustments:

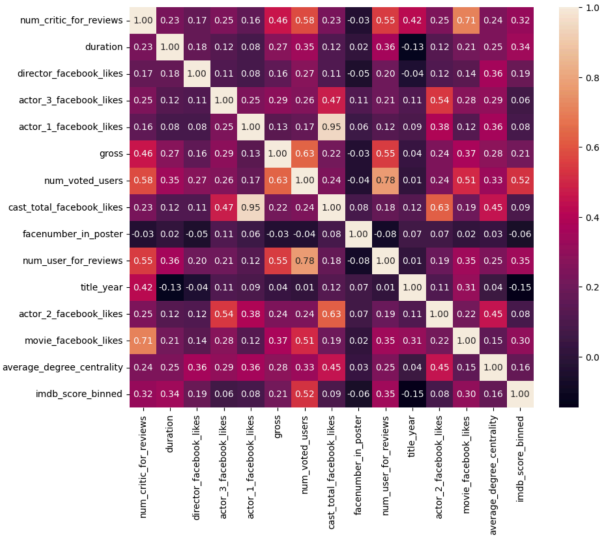

Figure 1 - Numerical Features' Correlation matrix heatmap in training dataset

1. Remove all plaintext: As these texts have been vectorized with advanced processing methods. We have decided to remove the original plaintext as they are not well-suited for model utilization.

2. Change feature type: Some features may have a minor impact on the prediction performance, while certain features such as language may have a dominant role. To mitigate these performance impacts, we converted it to a binary boolean.

3. Separate features and labels from the training set: As the standard test set does not include labels, we need to adjust the training set by separating its features and labels to facilitate subsequent cross-validation and training.

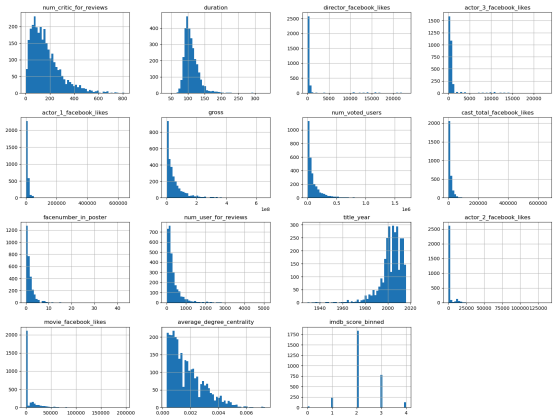

Figure 2 - Feature distribution histograms

4. Data transformation: From Figure 2, most numerical features, such as Facebook likes, exhibit a clear positive skewed distribution. To reduce the impacts of skewness, logarithmic transformation was first applied to these features.

5. Feature standardization: In order to address the issue of scale inconsistency between different features, StandardScaler was further applied to standardize all numerical features in the dataset.

2.2 Model Selection

Based on the above preprocessing procedures and considering the problem's nature as well as data characteristics, the following models were selected for comparison in this study:

1. Decision Tree Classifier (Week 4, Lecture 2): Initial exploration at the basic level, easier to understand and implement, as well as its ability to process classified data.

2. Random Forest Classifier (Week 7, Lecture 2): Can Improve classification accuracy and control overfitting by constructing multiple decision trees and taking the average of their predictions.

3. Logistic Regression (Week 6, Lecture 2): Used to evaluate the weights of different features that would affect movie ratings.

4. Multilayer Perceptron (Week 10, Lecture 1): A neural network model used to handle possible nonlinear relationships and complex patterns.

5. Stacking Classifier (Week 7, Lecture 2): An integrated learning strategy used by combining predictions from multiple base models to make predictions through a final meta-model. This method aims to combine the strengths of various models to attain a better prediction performance than a single model.

2.3 Training Method

Due to the absence of labels in the test set, this study employed Cross Validation (CV) as the main model training and validation strategy, to ensure the models' robustness and reliability. This statistical and analytic method is used to assess the model's generalization capacity to an independent dataset. It reduces the randomness of the models' performance and improves the accuracy of evaluations. Meanwhile, we applied 5-fold cv to each model, which means the dataset is evenly divided into 5 subsets, each part taking turns to serve as the test set, while the remaining 4 subsets were merged as the training set. This approach allows the model to be trained and validated on different subsets of data iteratively, ensuring consistency and reliability in evaluation outcomes.

2.4 Evaluation Methods

Finally, the performance of different models is evaluated based on accuracy, recall, and F1 score, while paying attention to models' consistency across various folds to assess their stability. Moreover, for stacked models, we pay special attention to their performance improvement compared to individual models to evaluate the effectiveness of stacking strategies. After selecting the model with the best and most stable performance as the final model, we will further analyze its feature importance to explain the primary factors that influence the IMDB score prediction most.

3. Results

Model	Mean score	Std Dev
Decision tree	0.609527	0.018135
Random forest	0.722706	0.010892
Logistic regression	0.703064	0.008545
Stack classifier	0.702731	0.008476
Multilayer perceptron	0.718375	0.010505

Table 1 - Mean accuracy scores and standard deviation for each model

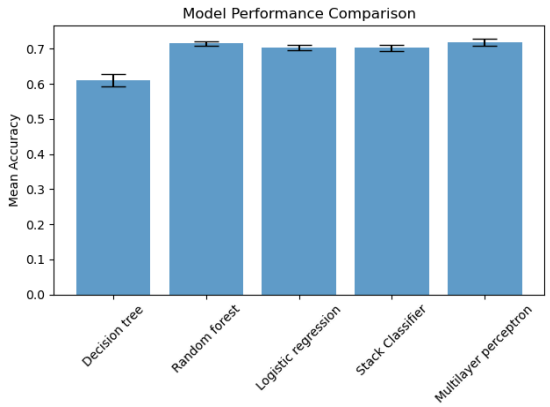

Figure 3 - Model performance comparison with error estimation

The table and figure presented above show the performance of five different machine learning models we have constructed including decision tree, random forest, logistic regression, stacking classifiers, and multilayer perceptron. All of them are applied with the 5-fold cross-validation method. As shown in Table 1, the average accuracy of each model is over 0.60, indicating a relatively stable performance. Additionally, from Figure 3, we can see that except for the decision tree, the performance and error margins of the other four models are relatively similar.

The decision tree showed the lowest average accuracy (about 0.61), which may be attributed to its simplistic nature, making it susceptible to data noise and overfitting. Conversely, random forests and multi-layer perceptrons presented better with both slightly above 0.71, indicating that ensemble methods can provide more robust predictive performance when confronted with complex datasets.

The performance of logistic regression slightly trails behind the above two, with an accuracy of approximately 0.70. Compared with the decision tree, this reflects the efficiency of linear models in handling feature relationships, especially after the appropriate data preprocessing and feature engineering.

Our stacking classifier is constructed based on the standard logistic regression mentioned before, along with a separate logistic regression stack synthesis for preprocessed title embedding with fitting. However, the accuracy of the stacking classifier is still around 0.70, which seems to not fully realize its higher accuracy characteristics.

Overall, the performance of all models is very close, possibly because the current feature set and model architecture are nearing optimality.

4. Discussion and Critical Analysis

4.1 Decision Tree

Considering the decision tree has the worst implementation performance among all models, we decided to assess the impact of data normalization on its performance.

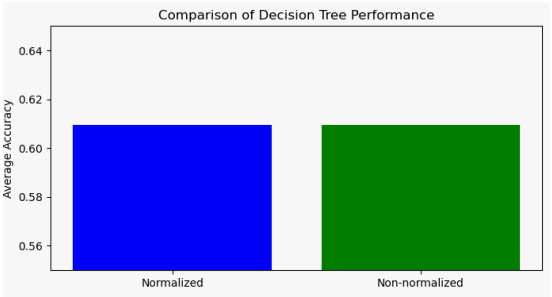

Figure 4 - Normalization impact on decision tree

From Figure 4, it can be observed that even though the data has undergone standardized normalization, the final performance of the decision tree was not affected. This is because the decision tree models mainly rely on category information and thresholds in data segmentation, rather than the specific values or their scales. Thus, normalizing the data has little impact on its performance and decision-making process.

In addition to this small experiment, we want to explore if overfitting of the data may cause poor performance to the decision tree. We used a learning curve (scikit-learn, 2024) to help check if this phenomenon has occurred.

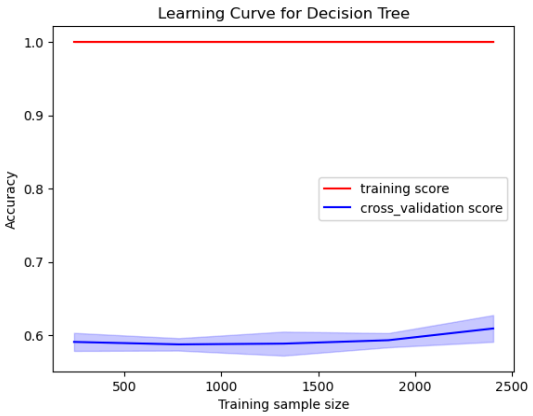

Figure 5 - The learning curve for decision tree

From Figure 5, we can observe that the training score of the decision tree is 1.0, indicating that it is fully fitting on the training data and captures all the patterns and noise inside. However, its cross-validation score is relatively low and shows a growth trend with an increased training sample size. This is a sign of overfitting, indicating that the model's generalization ability on unseen data is weak. We believe that adding more training data, if feasible, could enable the model to learn a broader distribution, thereby improving its generalization ability.

4.2 Random Forest

In the models we have established, the random forest has the best overall performance and a lower standard deviation. Therefore we intend to conduct deeper research and exploration of the content related to features based on it and draw the learning curve for comparison with the less effective decision tree model.

Random forest supports the feature importance function, which displays the significance and degree of determination of each feature for the final label prediction. As shown in Figure 6,

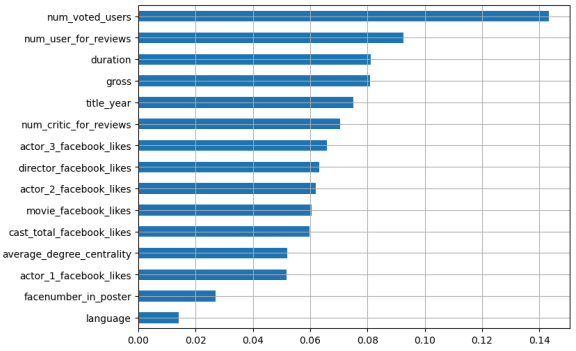

Figure 6 - Feature importances in random forest

Random Forest identifies the 'num_voted_users' feature as the most crucial reason, accounting for more than 14%, which is far ahead of other features. This can also be explained by common sense, as the popularity of a movie usually reflects its final score to some extent. Popular movies tend to have more interaction and discussion among people, while poor movies are relatively quiet. This also precisely proves that the second- ranked num_user_for_reviews, as it reflects audience engagements. On the contrary, language ranks last not only because English dominates in thousands of our dataset, but also because language itself does not directly affect the quality of the movie.

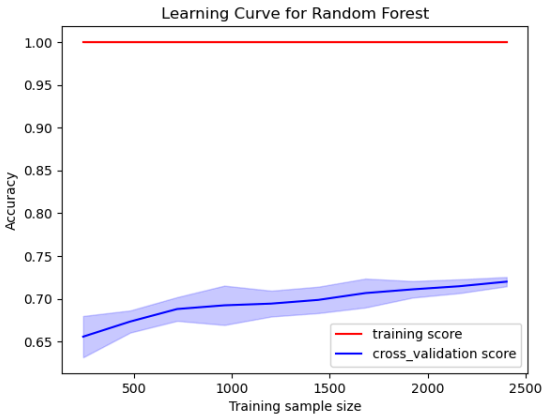

Figure 7 - The learning curve for decision tree

Overall, the learning curve (scikit-learn, 2024) of Random Forest is basically consistent with the decision tree, which exhibits overfitting on the training set. As the number of training samples increases, the cross-validation score gradually improves, indicating better model performance with more data.

4.3 Logistic Regression

Given the mediocre performance of our logistic regression model, we would like to examine its specific performance indicators closely. Hence, we have decided to use classification reports and confusion matrices for the analysis.

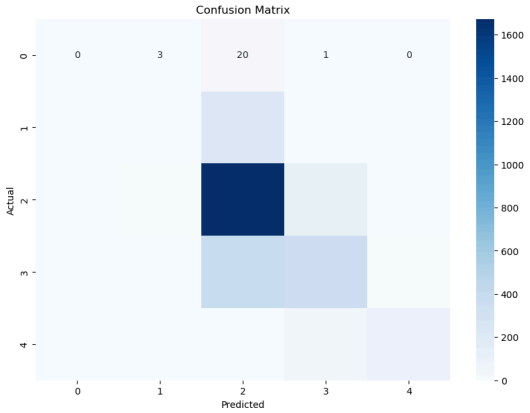

Figure 8 - Confusion matrix for logistic regression

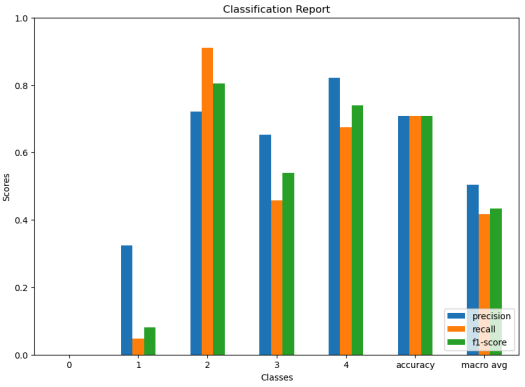

Figure 9 - Classification report for logistic regression

Through Figures 8 and 9, we can see that our logistic regression performs unevenly across different classes:

Class 0: The samples' classification performance in class 0 is very poor, with the model rarely correctly classifying these samples. This may be due to the insufficient samples of label 0 in the original dataset, resulting in inadequate learning. This indirectly leads to the problem of fewer false positives but more missed ones.

Class 1: This class is similar to class 0 in that it has a small number of samples, which results in low accuracy and recall in the model's recognition of class 1. The overall classification performance is poor, and most of them have not been correctly classified.

Class 2: The samples in this class have the best classification performance, with the majority of them being correctly classified and almost no false positives or omissions.

Class 3: The model demonstrates high accuracy in identifying class 3, but slightly lower recall. The overall classification performance is average, which may be due to a cliff-like decrease in the number of training samples compared to class 2.

Class 4: The amount of class 4 samples is less than that of class 3, but the model performs well in identifying class 4 with high accuracy and recall. This may be because some prominent features of class 4's were well-learned during the training process.

The unsatisfiable logistic regression outcome is likely due to the uneven class distribution. Also, the dataset may not have sufficient samples for the model to fully learn, with some labels, such as class 0, having very little data. Therefore, better performance may be obtained by certain classes, which lowers the overall average accuracy.

4.4 Stack Classifier

Since the results of the stack classifier did not meet our high expectations, we attempted to break down the stacking steps to identify the issues. Throughout the process, we first built a logistic regression model using title embedding processed with fast-text, and double checked its average cross-validation score of around 0.60.

While this score initially seemed acceptable, we discovered something interesting when using this model to predict the training set of title embedding, as illustrated in Figure 10.

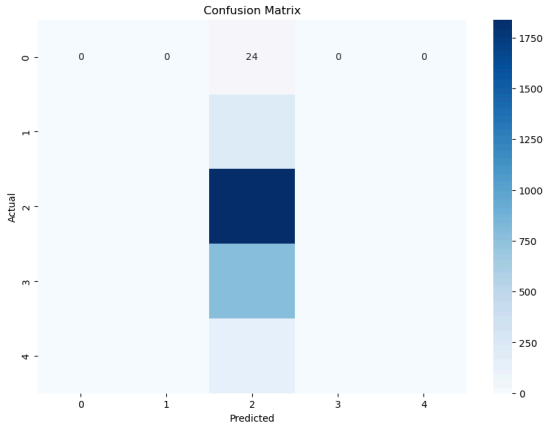

Figure 10 - Confusion matrix for title embedding specific logistic regression model

From the confusion matrix, the prediction outcome on all classes returns class 2, with no predictive ability for other classes at all. From the previous evaluation, we learned that there is far more training data related to class 2 than the others, which can lead to bias with the major vote, as most tend to favor a particular class, resulting in the final result also leaning towards that class.

To try to solve this problem, we first merged preprocessing features, including genres and plot keywords, into title embeddings to form a mega-training dataset. Then, using parameter tuning, we adjusted the hyperparameters of the logistic regression as a whole to see if the problem was alleviated. Figure 11 is a validation curve (scikit-learn, 2024) after tuning, indicating when C is 10, the model starts to be stable.

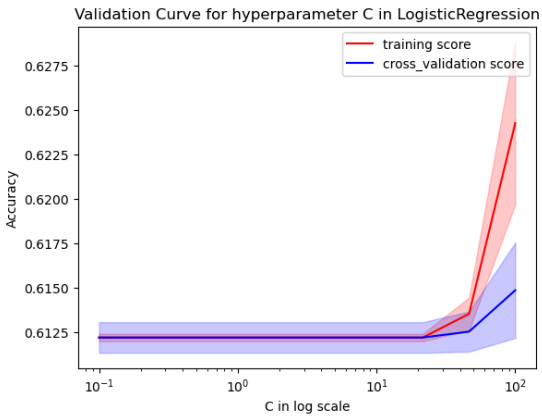

Figure 11 - Validation curve after tuning parameters in stack logistic regression

4.5 Multilayer Perceptron

We learned the Multilayer Perceptron (MLP) model relatively late, but its performance is quite impressive.

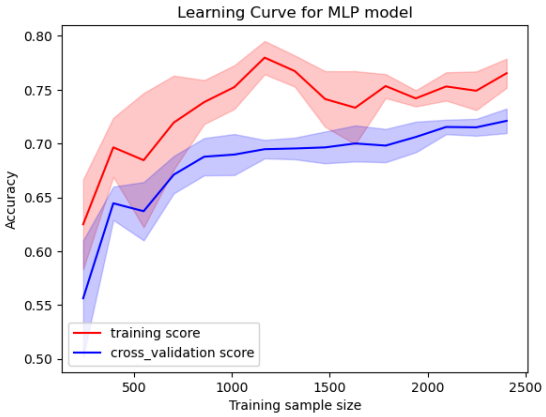

Figure 12 - Learning curve for MLP

From its learning curve (scikit-learn, 2024) in Figure 12, we can see that as the number of training samples increases, the performance of the model gradually improves and tends to stabilize when the number of samples is sufficient. Among them, the gap between the red and blue curves gradually narrows, indicating that the performance difference of the model on training and validation data gradually decreases, leading to a reduction in overfitting and further enhancing generalization ability. This also indirectly explains one of the reasons why the model performs well.

At the same time, we also plotted the loss curve (scikit-learn, 2024) for MLP model.

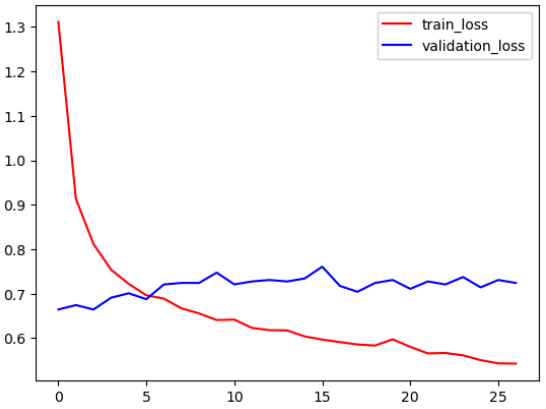

Figure 13 - Loss curve for MLP

From Figure 13, we can observe that the loss value of the model on the training set rapidly decreases in the initial epochs stage, and then tends to stabilize in the middle and later stages. There may be some fluctuations throughout the validation set, but overall it also tends to stabilize in the later stages.

Secondly, after the first intersection, although the difference between training loss and validation loss was widened again, as it gradually stabilized, the overall difference also gradually remained unchanged, indicating that the model performed more consistently on the training and validation sets, with less overfitting. This also suggests that the model has strong generalization ability and can perform well on unseen data.

5. Conclusions

In this project, we explored five distinct machine learning models to predict movie ratings based on IMDB data. We also conducted extensive data preprocessing to optimize the models and enhance their performance.

After the comprehensive evaluation, we found that decision tree classifiers performed the worst, logistic regression and stacked classifiers showed moderate performance, while random forests and multi-layer perceptrons performed the best. Nevertheless, their overall performance is still very close with no significant gap between them.

Considering that the current feature set and model architecture are approaching their limits, from the perspective of further improvement and optimization, we need to introduce more datasets and features in the future, while ensuring a more balanced distribution of data across different classes. Additionally, deep learning can be attempted, using more complex models or refining existing feature engineering processes to further enhance prediction accuracy.

Overall, this report emphasizes the importance of combining multiple machine learning techniques and strict preprocessing steps to develop robust movie rating prediction models, with each of their own evaluations. The insights gained from this project not only contribute to the field of predictive analysis in the film industry, but also provide a solid foundation for our future research and application in machine learning.

6. References

COMP30027 Lectures, Week 4, Lecture 2, 2024 Semester 1, Kris Ehinger, *Decision Trees*

COMP30027 Lectures, Week 6, Lecture 2, 2024 Semester 1, Kris Ehinger, *Logistic Regression*

COMP30027 Lectures, Week 7, Lecture 2, 2024 Semester 1, Ling Luo, *Classifier Combination*

COMP30027 Lectures, Week 10, Lecture 1, 2024 Semester 1, Kris Ehinger, *Neural networks I: Perceptron*

scikit-learn, 2024, *Validation curves: plotting scores to evaluate models*, 3.4, URL: https://scikit-learn.org/stable/modules/learning_curve.html#learning-curve