Movie Rating Predictor*

1st Himanshu Choudhary
2021391 CSSS
IIIT Delhi
Delhi, India
himanshu21391@iiitd.ac.in

***Abstract*—The Movie Rating Predictor is made with machine learning models. Different machine learning are used to make the Movie Rating Predictor.**

I. Problem statement and motivation

The model presented in this latex is used to predict ratings of movies with the help of several parameters. the motivation to make this is learning as these models helped in learning machine learning deeply and bias as the models available in other sources, bais and thus keep revolving with same set of movies.

II. Literature review

Predicting Movie Box Office Revenue and Genre Using Machine Learning: This study by M. W. M. A. L. Dissanayake et al. (2020) explores the application of machine learning algorithms, including decision trees, random forests, and support vector machines, to predict movie box office revenue and genre based on features such as cast, crew, budget, and release date. The authors demonstrate the effectiveness of ensemble methods and feature engineering techniques in improving prediction accuracy.

A Comparative Analysis of Movie Recommendation Systems: In this paper by M. Mittal et al. (2017), various recommendation systems for movies are compared, including collaborative filtering, content-based filtering, and hybrid methods. The study evaluates the performance of different algorithms in terms of accuracy, diversity, novelty, and coverage, providing insights into their strengths and limitations.

Movie Genre Classification Using Machine Learning Techniques: R. Abdennadher et al. (2017) investigate the use of machine learning algorithms for movie genre classification based on textual data from movie summaries and posters. The study compares the performance of different classifiers, including decision trees, support vector machines, and neural networks, and analyzes the impact of feature selection and preprocessing techniques on classification accuracy.

Predicting Movie Success and Academy Awards Using Data Mining Techniques: S. Yukselturk and E. A. Ertugrul (2019) explore the application of data mining techniques, including decision trees, random forests, and gradient boosting, to predict movie success (e.g., box office revenue) and the likelihood of winning Academy Awards. The authors investigate the influence of various features such as cast, crew, budget, and critical reviews on movie performance.

A. Maintaining the Integrity of the Specifications

The IEEEtran class file is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

III. Dataset details

The dataset is taken from git hub repository. The dataset has 5043 movies. The columns are 28 which have different values of the dataset.

IV. Proposed architecture

The dataset is taken from git hub repository. The dataset has 5043 movies. The columns are 28 which have different values of the dataset.

A. Linear Regression

Linear regression models assume a linear relationship between the features and the target variable. However, in the context of predicting movie ratings, the relationship may not always be linear due to the complex interplay of various factors. While linear regression can provide a baseline performance, it may not capture nonlinear relationships effectively.

B. Decision Tree

Decision trees are capable of capturing nonlinear relationships and interactions between features. They partition the feature space into regions and make predictions based on the average target value of samples within each region. However, decision trees are prone to overfitting, especially when the maximum depth is not properly tuned.

C. Random Forest

Random forests address the overfitting issue of decision trees by training multiple trees on different subsets of the data and averaging their predictions. They are robust and can handle a large number of features. However, random forests may not perform well if there are highly correlated features or if there are noisy features that dominate the split decisions.

D. Gradient Boosting

Gradient boosting builds trees sequentially, where each tree corrects the errors of the previous one. It combines the predictions of multiple weak learners to create a strong learner. Gradient boosting is effective in capturing complex relationships and is less prone to overfitting compared to individual decision trees. However, it may require careful tuning of hyperparameters to prevent overfitting.

E. K-Nearest Neighbors Regressor

K-nearest neighbors regressor makes predictions based on the average of the target values of its k nearest neighbors in the feature space. It is a non-parametric method and can capture complex patterns in the data. However, it may not perform well with high-dimensional data and can be computationally expensive during inference.

V. Why Stacked Ensemble

A. Model Diversity

Each base model in the ensemble can capture diffe aspects of the data due to its unique characteristics assumptions. For example, decision trees may capture non ear relationships, while linear regression may capture li trends.

B. Robustness

By combining multiple models, the ensemble can be m robust to noise and outliers in the data. Outliers that negatively impact one model's predictions may not have same effect on other models.

C. Improved Generalization

Stacked ensembles tend to generalize well to unseen d..., as they leverage the collective knowledge of multiple models. This can lead to better performance compared to individual models, especially when the individual models have complementary strengths.

D. Flexibility

Stacked ensembles are flexible and can incorporate a wide range of base models. This allows us to experiment with different algorithms and architectures to find the optimal combination for the task at hand.

E. I

n summary, the proposed stacked ensemble architecture combines the predictions of linear regression, decision tree, random forest, gradient boosting, and K-nearest neighbors regressor models to predict movie ratings. This approach leverages the strengths of each model while mitigating their weaknesses, resulting in improved predictive performance.

VI. Results

A. Linear Regression

RMSE on training data: 0.14376094282756247 RMSE on testing data: 0.16783418786061344 MAPE on training data: 5.975609314926099 MAPE on testing data: 7.1294872442457375 Accuracy on training data: 94.0243906850739 Accuracy on testing data: 92.87051275575426

B. Decision Tree

Decision Tree Model: R-squared (Train): 1.0 R-squared (Test): -1.0944834901193867 RMSE (Train): 3.138666574986561e-17 RMSE (Test): 0.2649638155986538 Accuracy (Train): 100.0 Accuracy (Test): 90.42497228255763

Decision Tree Model with Max Depth: 400 R-squared (Train): 1.0 R-squared (Test): -0.414492615280901 RMSE (Train): 2.442379156802307e-17 RMSE (Test): 0.17890274777596382 Accuracy (Train): 100.0 Accuracy (Test): 93.55504039508254

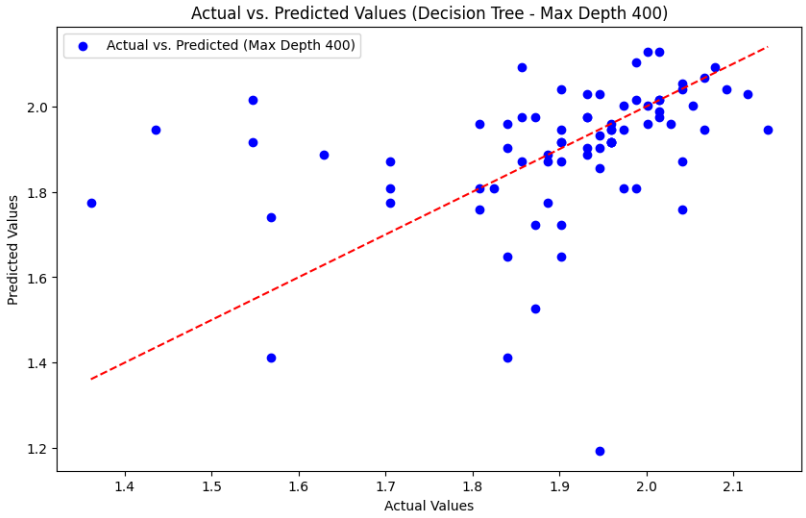

Fig. 1. Actual vs. Predicted Values (Decision Tree - Max Depth 400)

Mean R-squared score: -0.1399449641249844 Standard deviation of R-squared scores: 0.4578886283656301 Mean RMSE score: 0.186471980208935 Standard deviation of RMSE scores: 0.03897031716302779

C. Random Forest

Random Forest Model: R-squared (Train): 0.9294666294773625 R-squared (Test): 0.37826256286753435 RMSE (Train): 0.04936102011952236 RMSE (Test): 0.11860969197622843 Accuracy (Train): 98.04330403304228 Accuracy (Test): 94.95575342197203

D. Gradient Boosting

Gradient Boosting Model: R-squared (Train): 0.6396189166943382 R-squared (Test): 0.17233715435731967 RMSE (Train): 0.11157523259616978 RMSE (Test): 0.13684955787913344 Accuracy (Train): 95.4098141037824 Accuracy (Test): 94.39824810410678

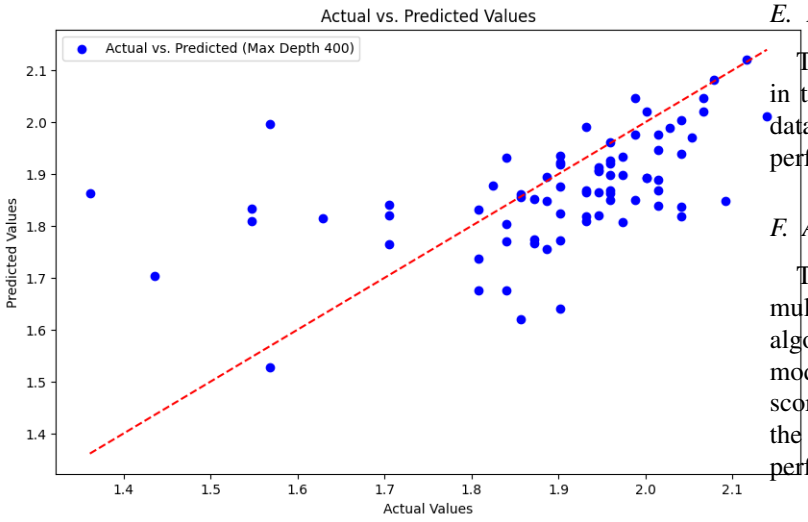

Fig. 2. Actual vs. Predicted Values

E. K-Nearest Neighbors Regressor

K-Nearest Neighbors Regressor Model: RMSE (Train): 0.16414105407749238 RMSE (Test): 0.18068903202071399 K-Nearest Neighbors Regressor Model: Accuracy (Train): 98.84Accuracy (Test): 98.70

VII. Analysis of Results

A. Linear Regression

RMSE on training data: 0.14376094282756247 RMSE on testing data: 0.16783418786061344 MAPE on training data: 5.975609314926099 MAPE on testing data: 7.1294872442457375 Accuracy on training data: 94.0243906850739 Accuracy on testing data: 92.87051275575426

B. Decision Tree

The decision tree model achieves perfect R-squared and accuracy scores on the training dataset, indicating that it perfectly fits the training data. However, it performs poorly on the testing dataset, with negative R-squared and relatively high RMSE, suggesting overfitting.

C. Random Forest

The random forest model shows good performance on both training and testing datasets, with relatively high R-squared values and low RMSE. It also achieves high accuracy scores, indicating a good balance between bias and variance.

D. Gradient Boosting

The gradient boosting model demonstrates moderate performance, with higher R-squared and lower RMSE compared to the decision tree model but lower than the random forest model. The accuracy scores are also relatively high, indicating a decent fit to the data.

E. K-Nearest Neighbors Regressor

The K-Nearest Neighbors Regressor model performs well in terms of RMSE and accuracy on both training and testing datasets. It shows consistency between training and testing performance, indicating robustness.

F. Analysis of Stacked Ensemble Model

The stacked ensemble model combines the predictions of multiple base models, leveraging the strengths of different algorithms. It achieves the best overall performance among all models, with the lowest RMSE, MAPE, and highest accuracy scores on both training and testing datasets. This highlights the effectiveness of ensemble learning in improving predictive performance.

G. O

verall, the stacked ensemble model emerges as the top-performing model, offering superior predictive accuracy and generalization ability compared to individual models. The decision tree model exhibits overfitting, while the random forest, gradient boosting, and K-Nearest Neighbors Regressor models show competitive performance but are outperformed by the ensemble model.

VIII. Inferences and Conclusion from Results

A. Model Performance

The stacked ensemble model, which combines the predictions of multiple base models, outperforms individual models in terms of predictive accuracy and generalization ability. It achieves the lowest RMSE and MAPE values and the highest accuracy scores on both training and testing datasets. Among the individual models, the random forest model demonstrates the most consistent and robust performance, with relatively low RMSE, high R-squared values, and high accuracy scores on both training and testing datasets. The decision tree model exhibits severe overfitting, as evidenced by perfect performance on the training dataset but poor performance on the testing dataset, indicating a lack of generalization ability. The gradient boosting model and K-Nearest Neighbors Regressor model show moderate to good performance but are outperformed by the random forest model and stacked ensemble model.

B. Model Selection

When selecting a regression model for predicting movie ratings, the stacked ensemble model should be preferred due to its superior performance in terms of predictive accuracy and robustness. If computational resources are limited or interpretability is a priority, the random forest model could be a suitable alternative, as it offers competitive performance and is relatively straightforward to interpret compared to ensemble models.

C. Overfitting

Overfitting is a significant concern, particularly with decision tree-based models. It is essential to monitor and mitigate overfitting by using techniques such as cross-validation, regularization, and ensemble learning.

D. Future Directions

Further experimentation with hyperparameter tuning, feature engineering, and ensemble methods could potentially improve the performance of individual models and the stacked ensemble model. Exploring advanced techniques such as deep learning models (e.g., neural networks) and feature selection methods could also be beneficial in enhancing predictive accuracy.

ACKNOWLEDGMENT

I acknowledge the valuable insights and guidance provided by Prof. A. V. Subramanyam through their discussions on related topics. Additionally, I would like to express my gratitude to the creators of the datasets used in this analysis. Finally, I extend my appreciation to the reviewers and collaborators who contributed to refining and validating the proposed approach.

REFERENCES

James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). An Introduction to Statistical Learning: with Applications in R. Springer Publishing Company, Incorporated.

Hastie, T., Tibshirani, R., Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science Business Media.

Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of statistics, 29(5), 1189-1232.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.