Movie Ratings Prediction:

An Analysis of Data Processing and Model Performance

Made by:
John Sebastian

Contents

Introduction

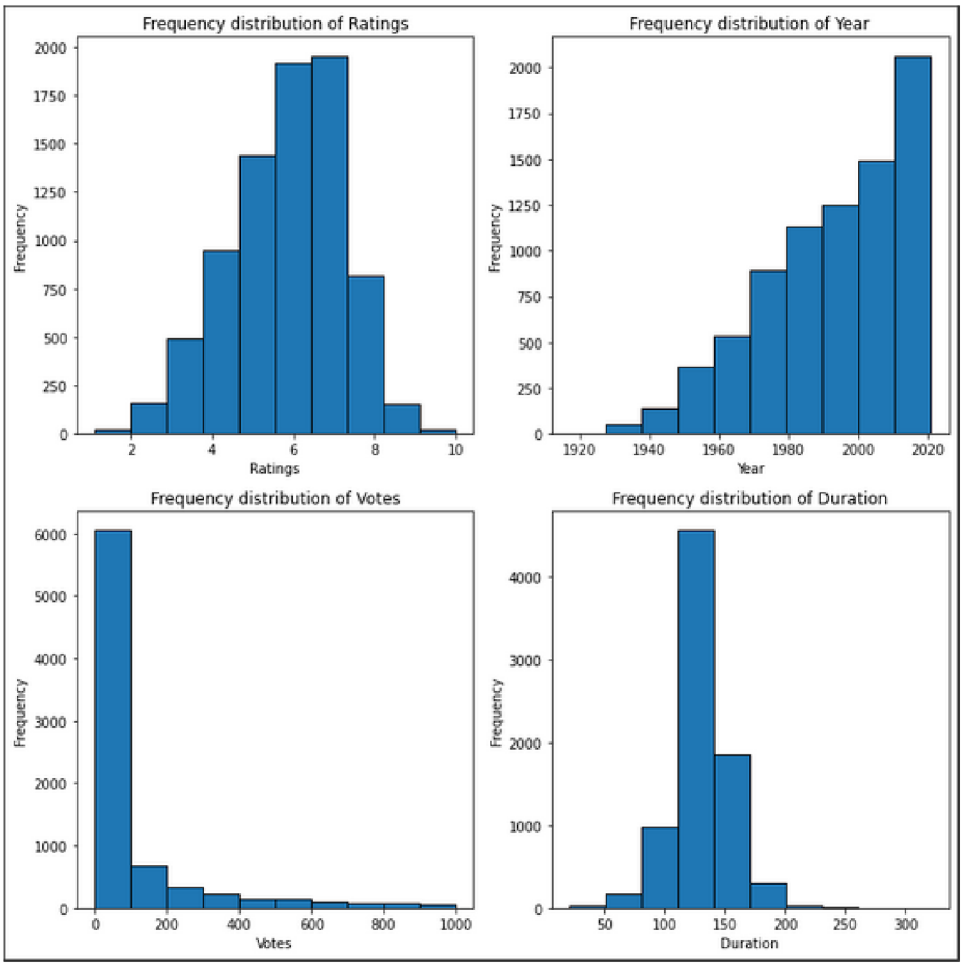
The purpose of this report is to present the results of a machine learning assignment aimed at predicting movie ratings. The project consisted of several phases including exploratory data analysis, pre-processing, model selection, hyperparameter tuning, and evaluation. During the exploratory analysis, the distribution and relationship between the features and target variable (Rating) were analyzed to gain a better understanding of the data. The findings from the exploratory analysis were used to inform the pre-processing steps, which involved scaling and encoding the data. The model selection process involved comparing the performance of several machine, namely Linear Regression, Decision Tress, Random Forests and Gradient Boosting learning algorithms and selecting the best performing model. Out of the four models Gradien Boosting gave the highest r2 score of 79.6%. The hyperparameter tuning process aimed at finding the optimal parameters for the model to improve its performance.

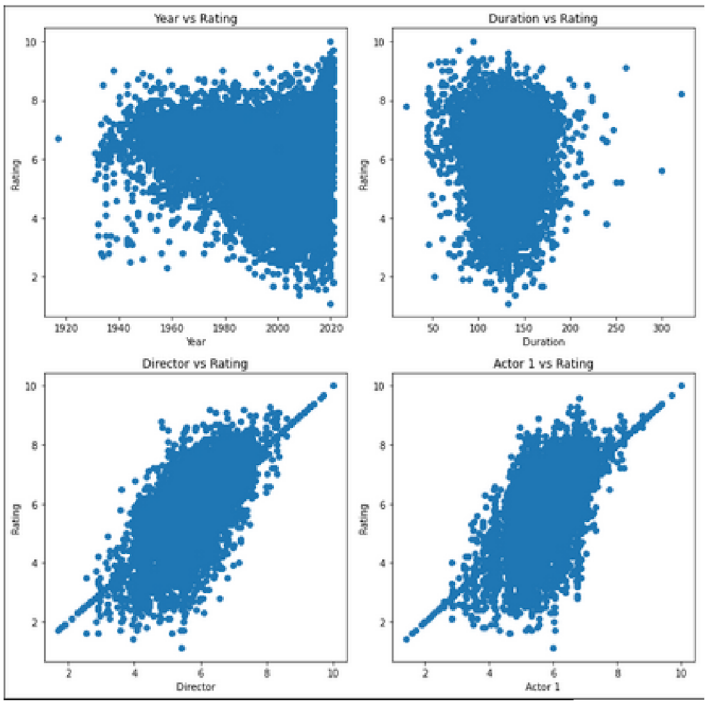
Links:

- Driver folder: https://drive.google.com/drive/folders/1gGoYI8ciAU-28w0TgVP7UB6RlMw8Hvs_?usp=sharing
- Notebook link: https://colab.research.google.com/drive/1FPyaTJa83cphzZpQ9DKIwTG7BF2qzxoc?usp=sharing

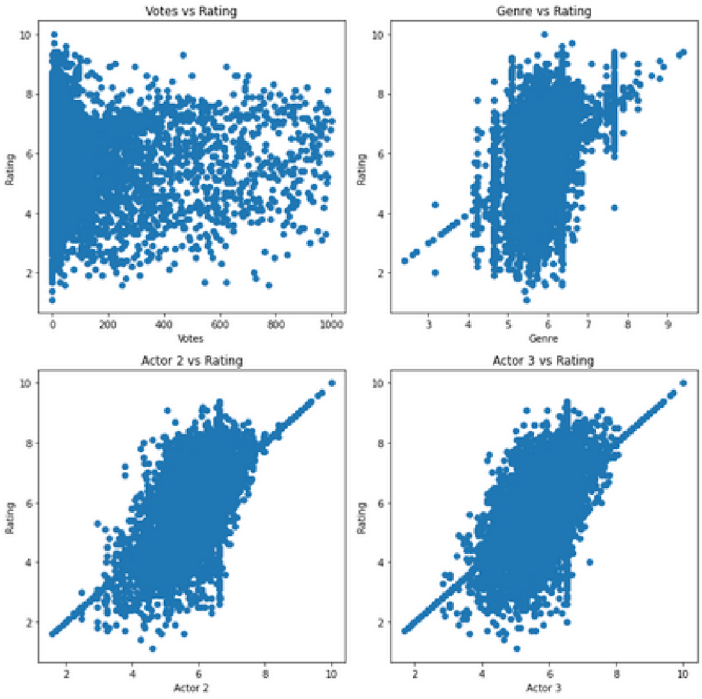
Exploratory Analysis

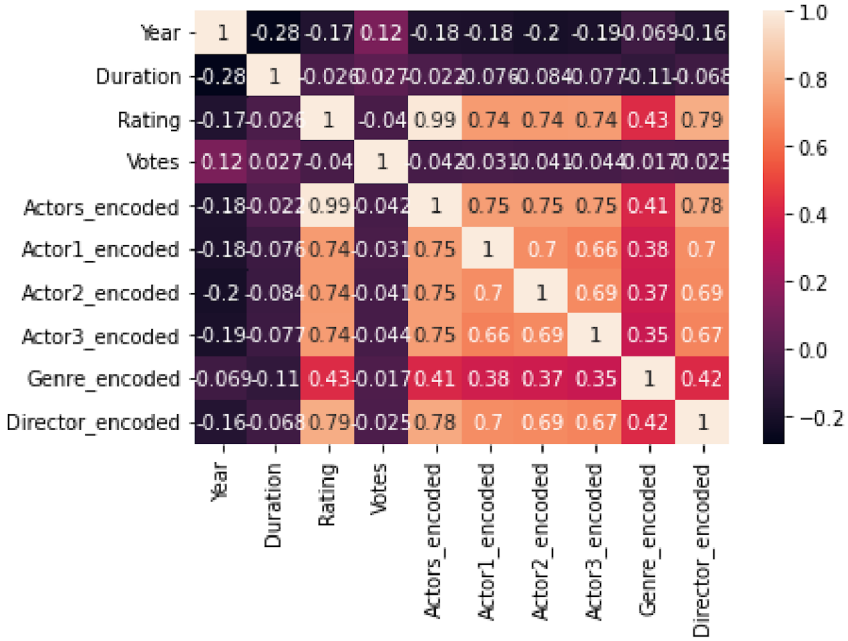
Exploratory analysis is an important step in any machine learning project as it helps to understand the data, identify patterns and relationships, and uncover any potential problems or issues with the data. In this project, an exploratory analysis was conducted on a movie dataset which included features such as genre, director, actors, year, duration, ratings, votes, etc. The distribution plots for each feature were created to understand the distribution of the features before scaling and encoding. The scatter plots were used to visualize the relationship between features and the target variable (rating). A correlation matrix heatmap was created to understand the correlation between features and the feature importance plot was created to understand the importance of each feature in the final model. The exploratory analysis helped to gain a deeper understanding of the data and set the foundation for the pre-processing steps, model selection, hyperparameter tuning, and evaluation results.





Scatter Plots





Correlation Heat Map

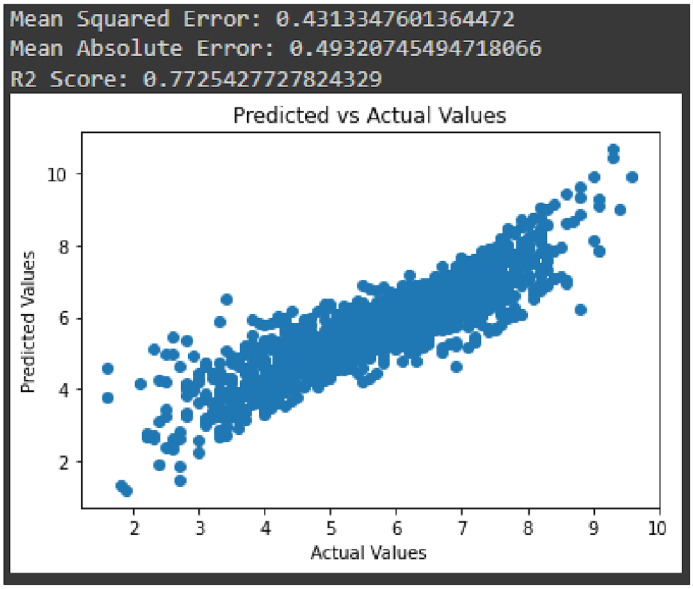
Data Pre-processing

In the pre-processing step, the data set was cleaned and transformed to prepare it for modeling. The year, duration, and votes variables were initially in the form of -xxxx, x min, and x,xxx, respectively, but were converted into numeric form. The empty values of the duration were filled with the mean value, the year with the median, and the votes with the median. To bring the numeric values within a common range, Min-Max scaling was applied to these variables. The directors, genres, and actors were initially thought to be one-hot encoded, but this method would have resulted in an overly large dataset. Instead, target encoding was used for these variables. The initial approach was to consider the actors as a single variable by combining actor 1, 2, and 3, but this resulted in overfitting and an accuracy of 96%. Hence, the actors were encoded separately. The genres were not separated as they had weak correlation with the target variable and did not affect the accuracy significantly.

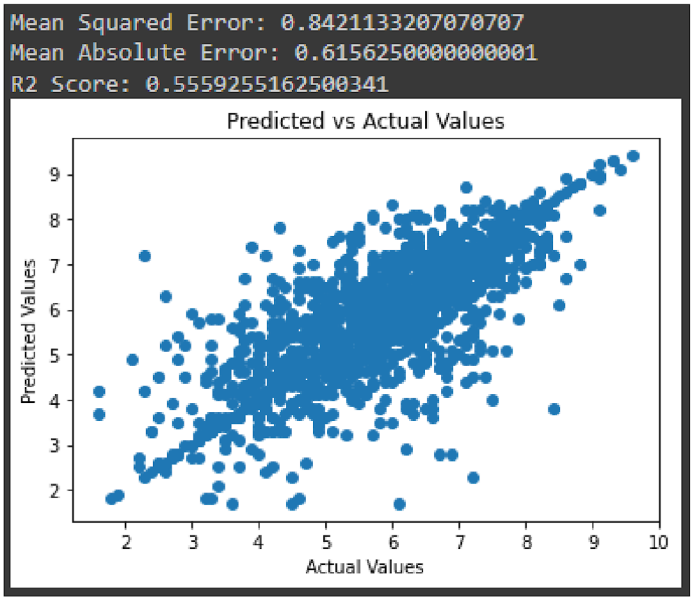
In order to split the dataset, stratified sampling was used as it ensures that each class in the target variable is represented proportionally in both the training and testing sets. This is particularly important for our dataset as it ensures that the distribution of the ratings in both sets is representative of the overall distribution in the dataset, reducing the chances of overfitting or underfitting the model. The train columns were chosen based on the correlation heatmap and included the encoded variables for Genre, Director, and Actors 1, 2 and 3. The code used StratifiedShuffleSplit from sklearn.model_selection with a test size of 0.2 and a random state of 2. This method split the data set into 20 strata based on the frequency distribution of the target variable, which was Rating. The resulting training set consisted of the encoded variables and the corresponding Ratings, while the testing set consisted of the encoded variables and their respective Ratings.

Model Selection

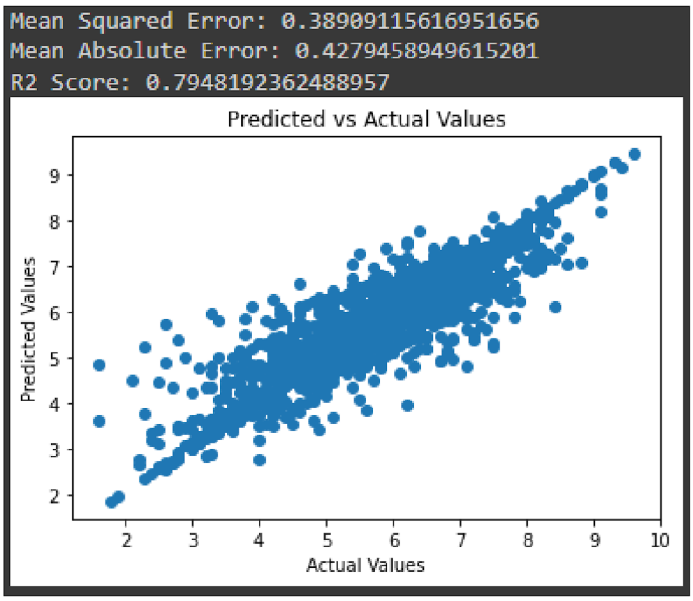
In this assignment, we evaluated several machine learning models to predict the movie ratings. The models used were linear regression, decision tree regression, random forest regression, and gradient boosting regression. We used the mean squared error metric to compare the performance of the models. After evaluating the results, gradient boosting regression was found to be the best performing model with the lowest mean squared error compared to the other models. Hence, this model was selected for further tuning and evaluation.



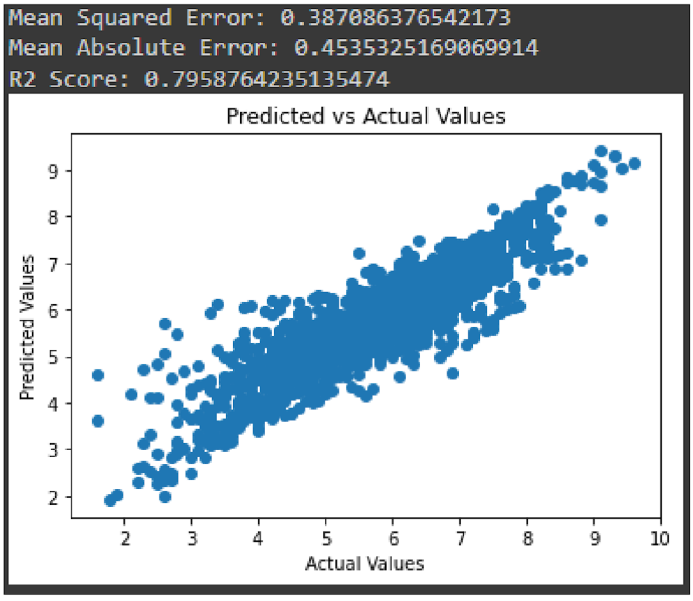
Linear Regression



Decision Tree



Random Forests



Gradient Boosting

Hyperparameter Tuning

In the process of building the model, the selection of appropriate hyperparameters is crucial to achieving the best performance. In order to fine-tune the hyperparameters of the Gradient Boosting Regressor model, we used Grid Search Cross-Validation (GridSearchCV). GridSearchCV is a powerful tool that allows us to exhaustively search for the best combination of hyperparameters that result in the highest performance of the model. We defined a grid of hyperparameters to search through, including the number of estimators, the maximum depth of each tree, the minimum number of samples required to split an internal node, and the learning rate. The param_grid used for the grid search consisted of the following combinations:

- n_estimators: [100, 200, 300, 400],
- max_depth: [3, 4, 5, 6],
- min_samples_split: [2, 4, 6, 8], and
- learning_rate: [0.1, 0.05, 0.01, 0.005].

The grid search would then train the model for each combination of hyperparameters and evaluate its performance using cross-validation to determine the best combination of hyperparameters. The combination of hyperparameters that gave the lowest mean squared error was chosen as the final set of hyperparameters for the model.

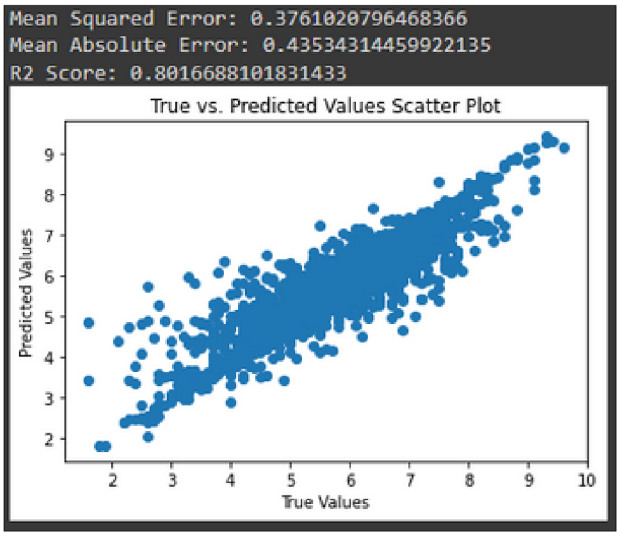
Evaluation Results

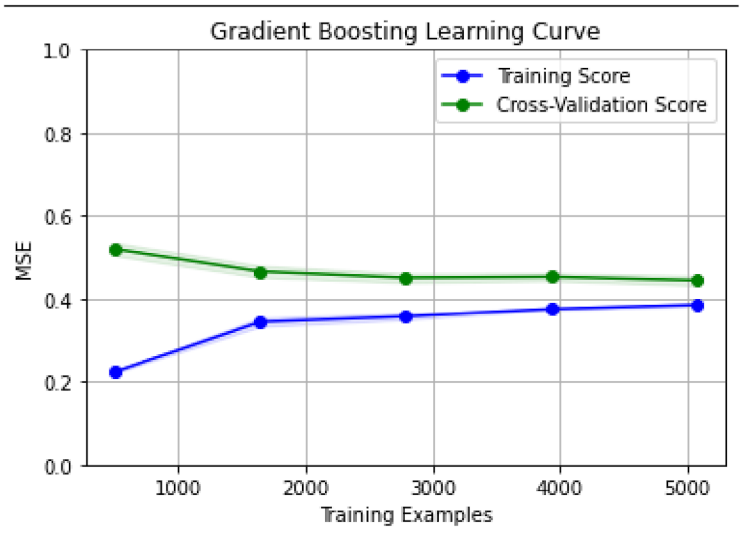
The gradient boosting model was selected due to its ability to handle both continuous and categorical variables, as well as its performance in reducing prediction errors. The learning curve showed that the model was able to generalize well and not overfit the training data, resulting in a good fit.

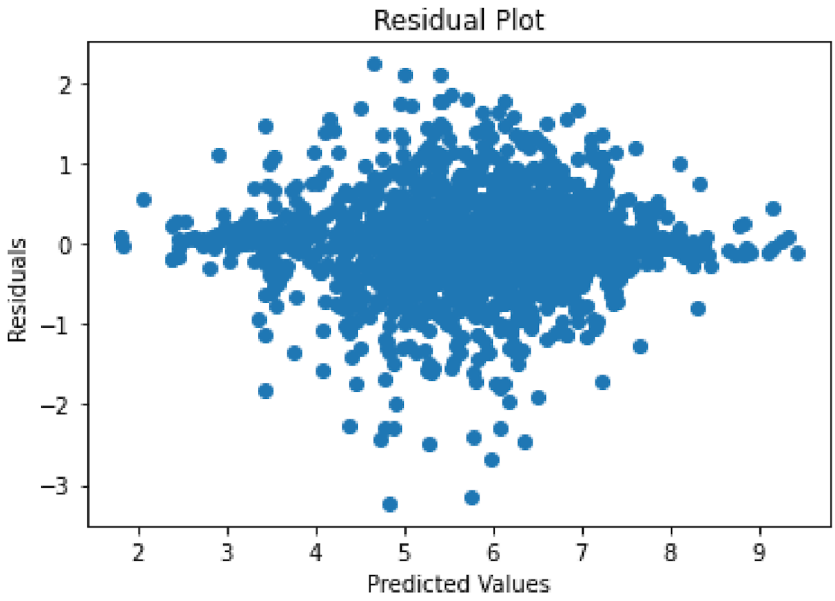
In terms of evaluation, the mean absolute error (MAE) of 0.4 was used to measure the difference between the predicted values and the actual values. This result indicates that the model's predictions are, on average, off by 0.4 units.

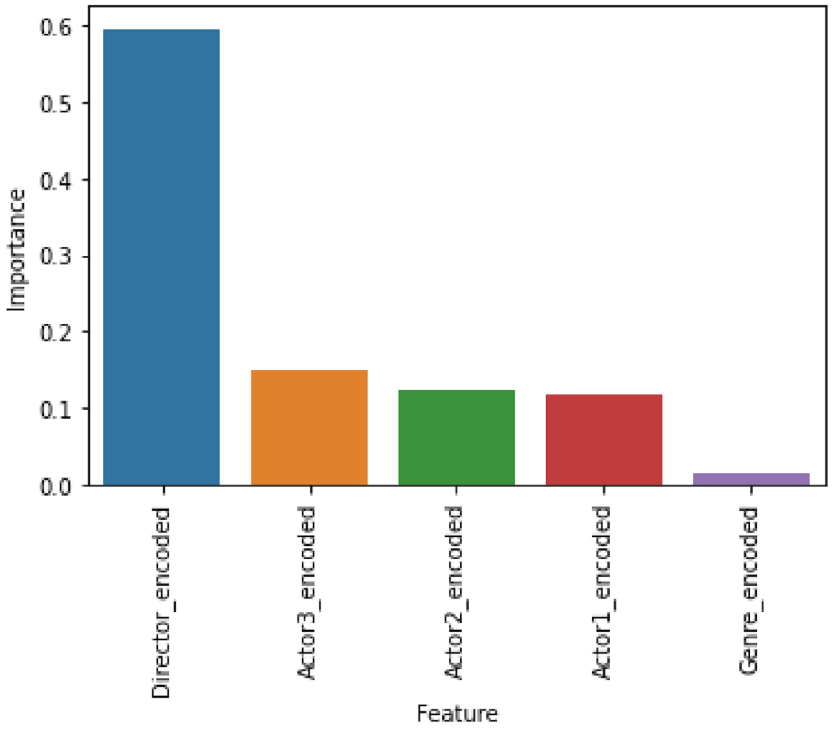
To further visualize the performance of the model, a scatter plot was created to show the relationship between the actual and predicted values. This plot revealed a linear relationship, indicating that the model made good predictions. The residual plot was also examined and showed evenly distributed residuals around zero, supporting the conclusion that the model was a good fit for the data. In terms of feature importance, the plot showed that the most significant features in the final model were the Director, followed by actors and genre.

In conclusion, the evaluation results suggest that the gradient boosting model was able to make accurate predictions and was a suitable fit for the data. The use of hyperparameter tuning and grid search CV helped us find the optimal parameters for the model, leading to improved performance.






Residual Plot
Residuals
Predicted Values
2
1
0
−1
−2
−3
2
3
4
5
6
7
8
9


Importance
Feature
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Director_encoded
Actor3_encoded
Actor2_encoded
Actor1_encoded
Genre_encoded