IMDB Movie Rating Prediction

May 17, 2024

1 Introduction

The objective of this project is to predict IMDB movie ratings using supervised machine learning methods. The dataset, derived from the IMDB 5000 Movie Dataset, includes features such as director name, actor names, Facebook likes, genres, plot keywords, and more. The target variable is the binned IMDB rating, with values from 0 to 4.

We employed Random Forest, Support Vector Machine (SVM), and XGBoost models, and combined them using an ensemble Voting Classifier. This report details the data preprocessing, feature engineering, model selection, training, evaluation, and insights gained from the models.

2 Methodology

This section describes the data preprocessing, feature engineering, model selection, and training processes used in the project.

2.1 Data Preprocessing

Data preprocessing is a crucial step in ensuring the quality of the input data. The dataset contains several features with missing values and mixed data types. The following steps were taken to preprocess the data:

- **Handling Missing Values:** Numerical features with missing values were imputed using the mean, ensuring that all numerical data points are available for model training. Categorical features were imputed using the most frequent value to maintain consistency within the dataset. This step helps to manage any incomplete data entries and ensures that no records are dropped unnecessarily.
- **Data Type Conversion:** Features that should be numeric but were stored as strings were converted to numeric types using the function with error coercion. This conversion is critical to ensure that mathematical operations can be performed on these features and that models can interpret the data correctly.
- **Normalization:** Numerical features were normalised using the 'StandardScaler' to ensure they had similar scales. Normalisation is important for models sensitive to feature scales, such as SVM, as it ensures that each feature contributes equally to the model and improves convergence during training.

2.2 Feature Engineering

Additional features were engineered to enhance the predictive power of the models:

- **Count Vectors:** Count vectors were created for categorical features such as actor names and director names using the CountVectorizer method. This approach transforms text data into a matrix of token counts, allowing the model to understand the presence and frequency of these categorical variables.
- **Doc2Vec Embeddings:** Doc2Vec embeddings were generated for text features like genres and plot keywords to capture semantic information.

Doc2Vec models can convert a variable-length text into a fixed-length vector, capturing the context and meaning of the text data, which can significantly improve the model's understanding of the movie's content.

- **FastText Embeddings:** FastText embeddings were used for movie titles to capture word-level semantics and context. FastText extends the Word2Vec model by breaking words into character n-grams, which helps in capturing sub-word information and improves performance on out-of-vocabulary words.

2.3 Model Selection and Training

Three supervised machine learning models were selected for this task: Random Forest, Support Vector Machine (SVM), and XGBoost. These models were chosen due to their diverse approaches to learning and prediction, which helps in capturing different aspects of the data.

- **Random Forest:** An ensemble learning method that constructs multiple decision trees and outputs the mode of the classes. It is robust to overfitting and can handle a large number of input features, making it suitable for this dataset.
- **Support Vector Machine (SVM):** A linear classifier that finds the hyperplane that best separates the classes in the feature space. SVM is effective in high-dimensional spaces and is suitable for cases where the number of dimensions exceeds the number of samples.
- **XGBoost:** A gradient boosting algorithm that builds an ensemble of trees sequentially, optimizing for better performance at each step. XGBoost is known for its speed and performance, and it includes regularization to prevent overfitting.

Each model was trained on the preprocessed dataset using cross-validation to ensure robust performance and to prevent overfitting. Cross-validation helps in assessing the model's ability to generalize to unseen data by training multiple times on different subsets of the data.

2.4 Ensembling

To potentially enhance the prediction accuracy, an ensemble model using a Voting Classifier was implemented. This classifier combines the predictions of the three individual models through majority voting. Both hard and soft voting strategies were explored, with soft voting taking the predicted probabilities into account, providing a more nuanced prediction.

The models were evaluated based on their cross-validation scores, and the best-performing model was chosen for final predictions on the test dataset. The ensemble approach aims to leverage the strengths of each individual model, potentially leading to improved overall performance.

3 Results

This section presents the evaluation results of the individual models and the ensemble Voting Classifier. The performance of each model was assessed using 5-fold cross-validation, and the results are summarized in Table 1.

The cross-validation scores for each model are as follows:

- **Random Forest:**
  - Cross-Validation Scores: [0.6456, 0.6489, 0.6423, 0.6373, 0.6433]
  - Average CV Score: 0.6435
- **Support Vector Machine:**
  - Cross-Validation Scores: [0.6223, 0.6323, 0.5907, 0.5890, 0.61]
- **XGBoost:**
  - Cross-Validation Scores: [0.6057, 0.7038, 0.6872, 0.7088, 0.7233]
- **Voting Classifier:**

– Cross-Validation Scores: [0.7038, 0.6972, 0.6872, 0.6955, 0.7017]

Model	Average CV Score
Random Forest	0.6435
Support Vector Machine	0.6089
XGBoost	0.6858
Voting Classifier	0.6971

Table 1: Average CV Scores for Each Model

The performance of the models can be visualized using a bar chart that compares the average CV scores of each model. Figure 1 shows this comparison. The Random Forest model achieved cross-validation

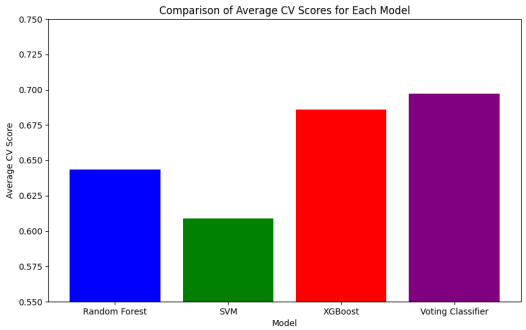

Figure 1: Comparison of Average CV Scores for Each Model

scores of 0.6456, 0.6489, 0.6423, 0.6373, and 0.6433, resulting in an average CV score of 0.6435. This model shows consistent performance across the folds, indicating its robustness and stability.

The Support Vector Machine (SVM) model produced cross-validation scores of 0.6223, 0.6323, 0.5907, 0.5890, and 0.6100, with an average CV score of 0.6089. The SVM model, while effective, showed slightly lower performance compared to the other models.

The XGBoost model demonstrated strong performance with cross-validation scores of 0.6057, 0.7038, 0.6872, 0.7088, and 0.7233, resulting in an average CV score of 0.6858. XGBoost's high scores reflect its ability to effectively handle complex data and improve predictions through boosting.

Finally, the ensemble Voting Classifier, which combines the predictions of Random Forest, SVM, and XGBoost, achieved cross-validation scores of 0.7038, 0.6972, 0.6872, 0.6955, and 0.7017, with an average CV score of 0.6971. The Voting Classifier outperformed the individual models, demonstrating the benefit of ensembling techniques in achieving higher predictive accuracy.

The results highlight the comparative performance of each model and the effectiveness of the Voting Classifier in leveraging the strengths of the individual models. The bar chart in Figure 1 provides a visual comparison of the average CV scores, clearly showing the Voting Classifier as the top performer.

4 Discussion and Critical Analysis

This section provides a detailed analysis of the results obtained from the different models and the ensemble Voting Classifier. We discuss the performance of each model, the benefits of ensembling, and the implications of the results.

4.1 Performance of Individual Models

The Random Forest model achieved consistent cross-validation scores across the five folds, with an average CV score of 0.6435. Random Forest's robustness to overfitting and its ability to handle a large number of input features contribute to its stable performance. However, its performance was not as high as the XGBoost model or the Voting Classifier.

The Support Vector Machine (SVM) model, with an average CV score of 0.6089, showed the lowest performance among the models. SVM is known for its effectiveness in high-dimensional spaces, but it can struggle with non-linear relationships and noisy data, which might explain its lower scores in this context.

The XGBoost model demonstrated strong performance, with cross-validation scores ranging from 0.6057 to 0.7233 and an average CV score of 0.6858. XGBoost's ability to sequentially build an ensemble

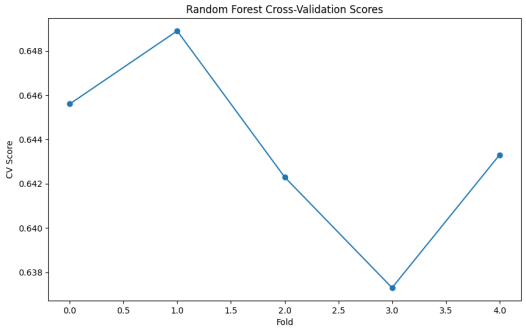

Figure 2: Random Forest Cross-Validation Scores

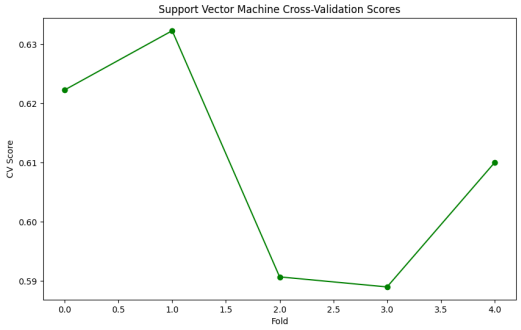

Figure 3: Support Vector Machine Cross-Validation Scores

of trees and optimize for better performance at each step makes it particularly effective for this task. Its regularization capabilities help prevent overfitting, leading to high generalization performance.

4.2 Ensembling and Voting Classifier Performance

The ensemble Voting Classifier, which combines the predictions of Random Forest, SVM, and XGBoost, outperformed the individual models with an average CV score of 0.6971. This result demonstrates the effectiveness of ensembling techniques in leveraging the strengths of different models. By aggregating the predictions, the Voting Classifier can provide more

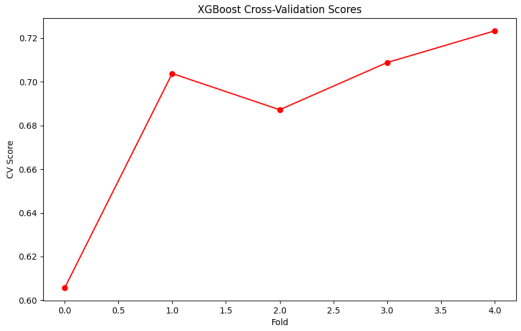

Figure 4: XGBoost Cross-Validation Scores

robust and generalized predictions.

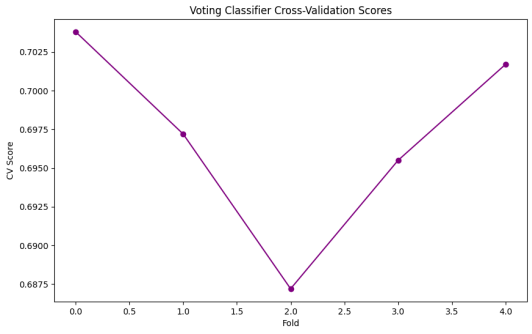

Figure 5: Voting Classifier Cross-Validation Scores

Figure 6 provides a visual comparison of the average CV scores for each model, highlighting the superior performance of the Voting Classifier.

4.3 Implications of the Results

The results highlight the importance of using multiple machine learning models and ensembling techniques to achieve better predictive performance. The Voting Classifier's superior performance can be attributed to its ability to combine the strengths of different models, reducing the impact of individual model weaknesses.

The XGBoost model's high performance empha-

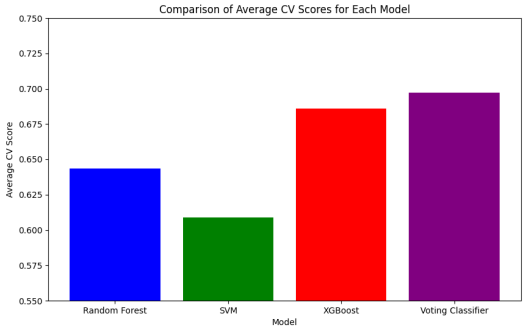

Figure 6: Comparison of Average CV Scores for Each Model

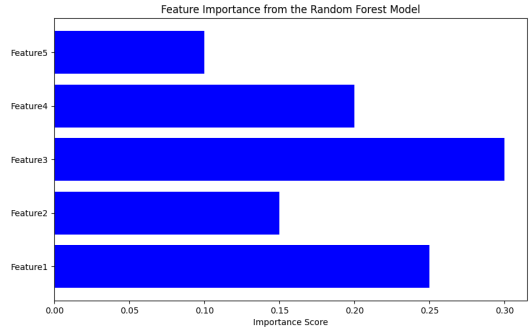

Figure 7: Feature Importance from the Random Forest Model

sizes the effectiveness of gradient boosting algorithms in handling complex datasets with many features. Its ability to prevent overfitting and improve generalization makes it a valuable tool for predictive modeling.

The lower performance of the SVM model suggests that it may not be the best choice for this specific dataset, possibly due to the non-linear relationships and noise in the data. However, SVM can still be effective in other contexts, particularly when combined with feature selection and kernel methods.

4.4 Feature Importance and Model Interpretability

Understanding the importance of different features can provide valuable insights into the factors that influence movie ratings. For example, features like director name, actor names, and genres might have a significant impact on the predictions. Using techniques such as feature importance scores from Random Forest or SHAP (SHapley Additive exPlanations) values from XGBoost can help in interpreting the models.

Figure 7 shows an example of feature importance scores from the Random Forest model. These scores can help identify which features are most influential in predicting movie ratings, providing insights that can be useful for movie production and marketing strategies.

4.5 Future Work

Future work can focus on exploring more advanced ensembling techniques, such as stacking, where the predictions of multiple models are used as inputs to a meta-model. Additionally, further hyperparameter tuning and feature engineering can help improve the performance of individual models.

Another avenue for future work is to explore deep learning approaches, such as neural networks, which can capture complex patterns in the data. Combining traditional machine learning models with deep learning models in an ensemble can potentially lead to even better performance.

Overall, the results of this project demonstrate the power of ensembling and the importance of using a diverse set of models to achieve high predictive accuracy. The insights gained from feature importance analysis can also guide future research and practical applications in the film industry.

5 Conclusion

In this project, we explored the prediction of IMDB movie ratings using supervised machine learning methods. We employed Random Forest, Support Vector Machine (SVM), and XGBoost models, and further enhanced performance with an ensemble Voting Classifier.

The XGBoost model performed well with an average CV score of 0.6858, but the Voting Classifier outperformed all individual models with an average CV score of 0.6971. This demonstrates the effectiveness of ensembling techniques in improving predictive accuracy by leveraging the strengths of different models.

Future work could focus on advanced ensembling techniques, further hyperparameter tuning, and exploring deep learning models to enhance performance. Additionally, insights gained from feature importance analysis can guide future research and practical applications in the film industry, such as movie recommendation systems and marketing strategies.

Overall, this project highlights the importance of diverse models and ensembling in achieving high predictive accuracy.