COMP30027 Report
Project 2: IMDB Movie Rating Prediction

Anonymous

1. Introduction

In an era where data-driven decisions are vital, the capacity to reliably predict outcomes from complicated information has become crucial across numerous industries.

This research project includes a comprehensive investigation aiming at constructing predictive models to estimate the IMDB ratings of movies based on a wide collection of features including social media presence, critical evaluations, and internal metadata such as genre and directorial influences. By applying a range of machine learning techniques—from simple models like Decision Trees to advanced ensemble approaches such as Random Forest and Stacking—we attempt to find the most influential indicators of cinematic success and boost the precision of entertainment analytics.

The goal is to accurately predict the class label of movies, consisting of five classifications from zero to four binned IMDB scores. To achieve the best model, several data preprocessing methods will be implemented along with data plotting to further delve into the model performances.

2. Methodology

2.1 Data Preprocessing

Prior to training the model, the dataset goes through several steps of data preprocessing to optimise model performance.

2.1.1 Label Encoding

Several categorical features are label encoded to ease the model training process and to accommodate models that require numerical data. We implemented a combination of using the feature vectors provided and performing our own label encoding.

2.1.2 Standardizing

Performing standardisation is essential to ensure that numerical values have an appropriate scale where attributes with larger values affect the model more than others. We chose this normalisation method over min-max scaling because it handles outliers better.

2.1.3 Data Sampling

In addressing an imbalance in the class distribution in the training set, several techniques were implemented to adjust the distribution. Firstly random undersampling is performed where training instances were randomly removed from the dataset to match the minority class. Another technique is random oversampling, where instances were randomly selected from the minority class and duplicated to match the majority class. The last technique applied is the Synthetic Minority Over-sampling Technique (SMOTE), where new instances are synthetically generated from the minority class to reach a balanced dataset.

2.1.4 Feature Selection

Feature selection is the last step of the data preprocessing phase. This step is essential to reduce dimensionality and improve model performance by reducing the complexity. Feature selection is applied using the random forest classifier model. For this step, we pre-determined the threshold value of 0.1. This tuning was done by testing different threshold values and examining the average accuracies of the models. We selected the random forest classifier as the feature selection model due to its robustness to overfitting, and its ability to handle non-linear relationships.

2.2 Model Selection

2.2.1 K Nearest Neighbors

KNN is a non-parametric, lazy learning algorithm that classifies data points based on the 'K' closest training examples in the feature space. KNN was used due to its effectiveness in scenarios with irregular decision boundaries. In domains where similar cases lead to similar outcomes, KNN can perform exceptionally well without any assumptions

about the underlying data distribution. Its straightforward implementation and the intuitive nature of its mechanism—classifying samples based on proximity—make it suitable for our initial explorations into the dataset.

2.2.2 Naive Bayes

The choice of Naive Bayes was driven by its efficiency and performance in dealing with large datasets and its effectiveness in applications with a high dimensionality-to-sample size ratio, making it a good base learner. Its probabilistic approach provides a good basis for handling uncertainty and variability in data, making it apt for preliminary data analysis, especially with nominal data.

2.2.3 Decision Tree

Decision trees create models that predict the value of a target variable by learning simple decision rules inferred from the data features. They are easy to interpret and can handle both numerical and categorical data. Decision trees were selected due to their transparency in decision-making, ease of visualisation, and ability to handle non-linear relationships. The model's capability to explicitly show which fields are most important for prediction is invaluable for understanding feature relationships and data structure.

2.2.4 Random Forest

Random Forest is an ensemble learning method that builds and merges multiple decision trees to achieve a more accurate and stable prediction. It combines the simplicity of decision trees with flexibility, reducing the risk of overfitting. The robustness of Random Forest in handling overfitting, while maintaining the ability to scale to large datasets, makes it ideal for this project. Its ensemble approach ensures that the variances of individual decision trees are averaged out, leading to improved accuracy and reliability.

2.2.5 Stacking Classifier

For our stacking classifier, we selected random forest and KNN as our base models and logistic regression as the meta-learner. Stacking involves stacking the output of individual classification models and using a second-level model to predict the output from the first-level model predictions. This approach leverages the strengths of individual learning algorithms. Stacking was chosen to potentially boost the predictive performance beyond what could be achieved by any single model. By combining various models, stacking harnesses their individual strengths while compensating for their weaknesses, making it a powerful method for complex predictive problems.

2.3 Model Evaluation

Cross-validation was applied to evaluate the performance of different models. This was a critical step to prevent overfitting and provide an unbiased estimation. This method was chosen over using a validation set to maximise the amount of training data. While cross-validation can be costly with many iterations, we had minimal time constraints and a reasonably sized dataset.

3. Results

We tested the performance of all our models using the accuracy evaluation metric. After performing data pre-processing and model selection, we computed the accuracy of each model with both cross-validation and the test set. Kaggle results were determined the accuracy of the models on the test set.

Model	Accuracy	
	Cross Validation	Test Set
Naive Bayes	28%	27%
3-NN	62%	60%
Decision Tree	60%	59%
Random Forest	72%	68%
Stacking	70%	70%

Table 1 - Comparison of different model performances by accuracy with cross-validation and on the test set.

We found that the accuracies between cross-validation and the test set were very similar, with the two best-performing models (Random Forest and Stacking) being ensemble models.

When performing feature selection, we selected the value for the threshold by plotting the model performances with different threshold values.

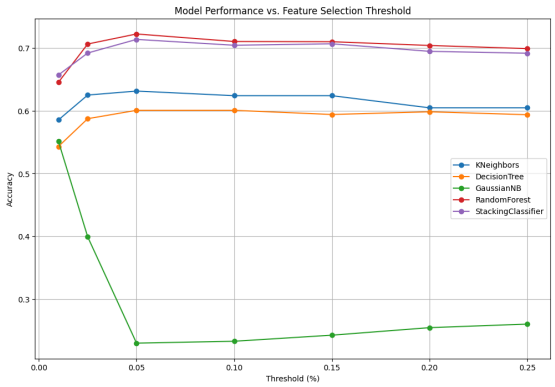

Figure 1 - Feature Selection Threshold Value vs. Model Accuracy

As seen in Figure 1, a threshold value of 0.05 and 0.1 resulted in the highest accuracy score among most models. Thus, we selected 0.1 as the threshold value for the feature selection. This resulted in the following features being selected:

actor_1_facebook_likes	actor_2_facebook_likes
actor_3_facebook_likes	
average_degree_centrality	cast_total_facebook_likes
director_facebook_likes	duration
genre_feat_4	genre_feat_5
genre_feat_9	genre_feat_12
genre_feat_15	genre_feat_27
genre_feat_33	genre_feat_41
genre_feat_42	genre_feat_54
genre_feat_78	genre_feat_86
genre_feat_88	genre_feat_92
gross	language
movie_facebook_likes	num_voted_users
num_user_for_reviews	num_critic_for_reviews
plot_feat_20	plot_feat_37
plot_feat_60	title_year

Model	Accuracy	
	No Feature Selection	Feature Selection
Naive Bayes	29%	28%
3-NN	59%	62%
Decision Tree	57%	60%
Random Forest	66%	72%
Stacking	67%	70%

Table 2 - Comparison of cross-validation accuracy with and without feature selection (threshold = 0.1).

Additionally, we compared the performance of different models using cross-validation with and without feature selection. As seen from Table 2, all models except for the Naive Bayes have an increase in accuracy with feature selection.

4. Discussion and Critical Analysis

Looking at the model performance comparison, it is evident that ensemble models perform significantly better than single algorithm models. This aligns with the theoretical explanation that combination classifiers create a stronger algorithm as it takes the strengths of different algorithms.

4.1 Error Analysis

However, we found that the model struggled to predict a variation of class labels, where most of the models seemed to have bias as it systematically creates more predictions of the '2' or '3' IMDB score class.

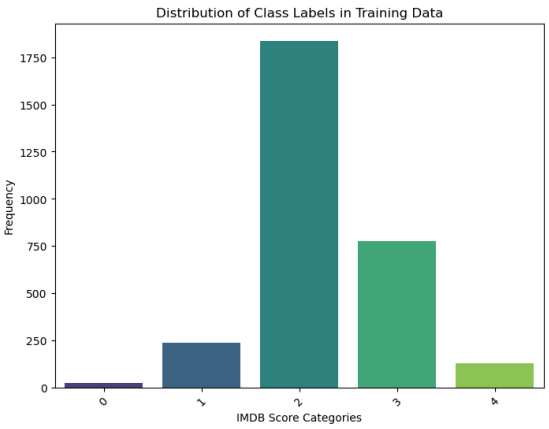

Figure 2 - Training Dataset Label Distribution

Delving further into this, we plotted the distribution of class labels in the training set, as seen on Figure 2, there is an imbalance in the training dataset where a significant portion of it is in the '2' class label, with class '3' following second. This creates sampling bias and thus may result in a biased behaviour in the models' performance. This may explain the high accuracies on the decision tree, random forest, and stacking classification as those models are less sensitive to class imbalance in the training set.

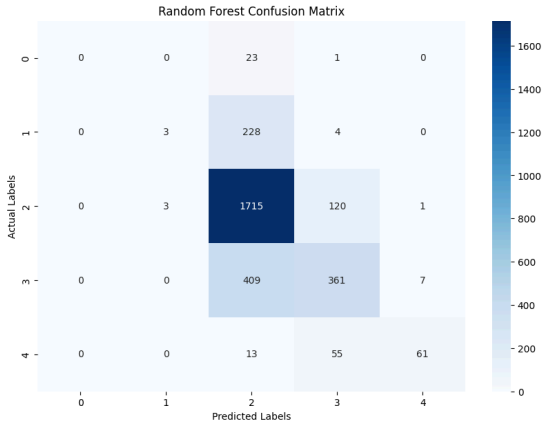

Figure 3 - Random Forest Confusion Matrix using Cross-Validation

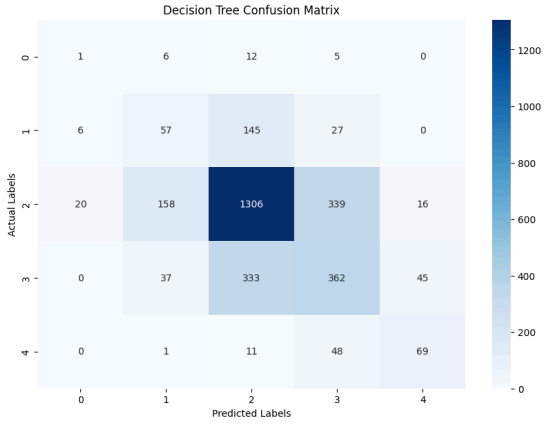

Figure 4 - Decision Tree Confusion Matrix using Cross-Validation

We can see that this sample bias is evident through the confusion matrix heatmap of each model. For example, in Figure 3, the random forest model produces a large number of incorrect predictions of class labels '2' and '3', which aligns with the sampling distribution bias. This is also evident in the decision tree model confusion matrix shown in Figure 4, where the sampling distribution is still somewhat preserved.

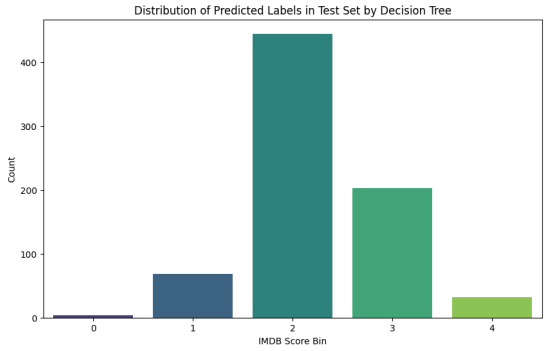

Figure 5 - Predicted Label Distribution on Test Set using Decision Tree

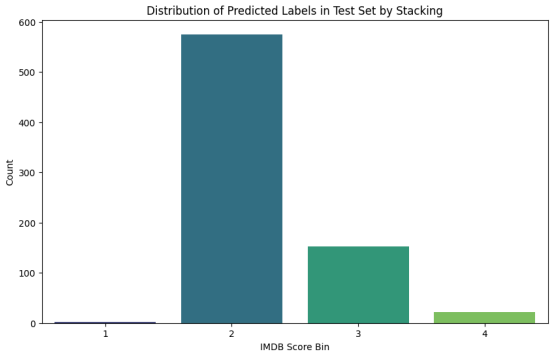

Figure 5 - Predicted Label Distribution on Test Set using Stacking Classifier

Furthermore, the sampling bias affects how the model performs on the test dataset. This can be seen by the predicted label distributions shown in Figures 5 and 6. From Figure 5, we see that the decision tree preserved the sampling distribution, while from Figure 6, it's shown that the stacking classifier did not predict a single '0' class label. This may be due to the tiny proportion of the class '0' label in the training dataset.

4.2 Dataset Comparison

We applied several methods to address the imbalanced class label distribution in the training set. Firstly, we performed random oversampling which significantly improved model performance as seen in Table 3. However, this may not be reflected in the accuracy of the test set, as it only resulted in 66% accuracy on the test set with the stacking classifier. This may be due to a lack of generalisation and overfitting as the training set initially has a large number of samples.

Model	Original Training Set	Random Oversampling	Random Undersampling	Synthetic Data Generator
Naive Bayes	28%	45%	51%	49%
3-NN	62%	86%	50%	85%
Decision Tree	60%	92%	40%	82%
Random Forest	72%	96%	55%	92%
Stacking	70%	96%	57%	91%

Table 3 - Comparative Performance of Machine Learning Models on Original, Undersampled, Oversampled, and SMOTE-Enhanced Datasets on

Validation Set.

Next, we applied random undersampling. As seen in Table 3, this significantly dropped the model performance, which could due to the significant loss of information from removing training instances. Lastly, we implemented the Synthetic Minority Over-sampling Technique (SMOTE). Despite being computationally expensive, this technique may be best for improved generalisation. As shown in Table 3, this technique increases the model performance significantly on the validation set. However, despite significant improvements in model performance in the validation set, these techniques did not improve model performance for the test set. The random forest classifier with SMOTE only produced 68% accuracy in the test set, equalling its performance when trained on the original training set in Table 1. This case also persists with the random oversampling technique, where the stacking classifier only produced 66% accuracy on the test set, lower than what it produced when trained with the original training set.

5. Conclusions

This research has systematically explored the impacts of various data preprocessing techniques to improve the performance of different models in predicting binned IMDB scores of a given movie through its features. One of the bigger challenges in this project is the highly imbalanced class distribution in the training dataset. Our findings reveal that although resampling techniques often enhance model performance on validation sets, they do not consistently translate to improved accuracy on test sets. Specifically, the Random Forest classifier exhibited equivalent accuracy when trained on both the original dataset and the SMOTE-enhanced dataset, highlighting the complexity of balancing model training between overfitting and underrepresentation of minority classes.

Additionally, our research proved that ensemble models are more robust to outliers, noise, and oversampling as they consistently produce the highest accuracies compared to single-algorithm models by combining the strengths of multiple learning algorithms.

Furthermore, we also observed that feature selection is essential to improve model performance by reducing model dimensionality and complexity. With that being said, the selection of the threshold value is critical, as it determines how many features are retained.

From this research, we advise future investigations to focus on developing more sophisticated resampling algorithms that can intelligently augment data without skewing the underlying distribution. Furthermore, exploring hybrid approaches that combine multiple resampling techniques could offer a pathway to models robust against both underfitting and overfitting.

Acknowledging these findings, it becomes evident that tackling class imbalance remains a challenge in machine learning, necessitating a balanced approach that carefully considers the trade-offs between model accuracy and complexity. We hope this study contributes to a deeper understanding of the dynamics at play and sparks further research in this pivotal area of machine learning.

6. References

Basim, A., & Ehinger, K. (2024). *COMP30027_2024_SM1 Machine Learning*. [Lectures]. The University of Melbourne.