IMDB Movie Rating Classification Report

Joseph Keelagher

1 Introduction

Given a derivation of the IMDB 5000 Movie Dataset[1] with 25 features per movie and a label based on ratings, the objective is to build a classification model to predict binned movie rating on a test dataset. Various pre-processed features, including FastText and Doc2Vec, were provided. The process included data analysis, pre-processing, and exploring machine learning techniques to construct an accurate model. This report documents the processes undertaken to analyse and evaluate the models and all findings about the problem.

2 Methodology

The process by which this task was completed can be categorised into 3 distinct sections. Firstly, there's data pre-processing where the dataset was transformed into a form on which learning can occur optimally. Following the data pre-processing stage is the model exploration stage in which four machine learning algorithms with different approaches were implemented and compared on effectiveness for this task. Finally, with the goal of squeezing as much as performance out of the models as possible prior to evaluation, they each underwent hyper-parameter tuning to determine the best combination of hyper-parameters which allow high performance and generalisability. Detailed documentation of the three aforementioned steps are in the following sections.

2.1 Data Pre-Processing

Textual features and instances with missing values are first removed from the data in order to accommodate models which do not support them. Discretisation is done on country since high rate films are mainly produced in USA, UK or France (See Table 1). PCA is then done on the other pre-processed features as they have significant dimensionality "*making it difficult for models to generalize well*"(Bellman, 1961). This is followed by comparing correlation with the class label, the Doc2Vec version of 'genre' stands out as informative among the rest and thus included in the data (See Figure 1), irrelevant feautures are removed so the models can avoid "*overfitting and computational inefficiencies, a phenomenon known as the curse of dimensionality*"(Verleysen and François, 2005). To prepare the data for the model exploration stage, the features were standardised to have a mean of 0 and standard deviation of 1.

2.2 Model Exploration

The first model considered was logistic regression. Logistic regression makes for a good starting point because of its simplicity and reputation as "*a benchmark for more complex classification*"(Hastie et al., 2009). The next model for comparison, random forest, conducts feature selection itself and as an ensemble of sub-models can capture more complex relationships in the data than the baseline. Gradient boosting was employed next in an attempt to triumph over class imbalance in the data since it can easily have its loss function configured to have it consider class weights. Finally, a stacking classifier was constructed, combining all three models' predictions and uses random forest as a final estimator. As an ensemble classifier of multiple approaches it benefits from the complementary strengths of each previous model.

2.3 Model Tuning

Lastly hyper-parameter tuning was conducted on all four models. Grid search was used along with stratified k-folds to ensure fair comparison between different parameter combinations. Grid search iteratively runs each model comparing the performance of stratified k-folds cross validation on all combinations of parameters be-

[1] https://www.kaggle.com/datasets/carolzhangdc/imdb-5000-movie-dataset

ing tuned and outputs the one which exhibits optimal performance. Logistic regression was tuned in regularisation strength, regularisation type 'penalty' and max iterations. Gradient boosting was tuned on number estimators, max depth of the estimators and learning rate. Random forest was also tuned on number of estimators but also its branching criterion, the same approach was taken on stacking classifier's final random forest estimator.

3 Results

When presenting the results, logistic regression will be referred to as LR, random forest as RF and gradient boosting as GB. Furthermore, the following results for the models were measured after hyper-parameter tuning and thus are the 'final' versions.

3.1 Data Analysis

In Table 1, the training dataset is presented as instances grouped by countries and highlights the predominance of USA, UK and France originating instances over all other countries when looking at highly rated films.

Country	Instances
USA	632
UK	116
France	37
All others	121

Table 1: Instances with $rating > 3$, grouped by *country*

Table 2, depicts the imbalanced class distribution in the training dataset. Considering the total count of 3004 instances, values of 2 for imdb_score_binned make up more than half the total amount of training instances and including values of 3, the two make up for almost 90% of the whole training dataset.

imdb_score_binned	Instances
2	1839
3	777
1	235
4	129
0	25

Table 2: *imdb_score_binned* instance counts, in descending order

Depicted in Figure 1 is the correlation of each feature with the label 'imdb_score_binned'. Importantly, the only relevant features kept from the pre-processed data were the doc2vec version of 'genres' and ordinal encoded version of 'country' made during data pre-processing.

3.2 Model Evaluation

Table 3 highlights the LR models strong performance specifically in recall 0.92 of instances with label 2 and the highest of all individual models' f1-scores for under-represented label 4. Unfortunately, the LR model fails to classify label 1 correctly almost entirely with a 0.01 f1-score and label 3 is not much better with a 0.48 f1-score. These results serve as a baseline for the following models.

Table 4 suggests that RF improved significantly with its knowledge of label 1 instances with a jump to 0.70 in precision, still present is the problem of low recall with a 0.07. RF performed slightly better than the baseline with label 2 and made improvement in recall of labels 3 and 4 helping it secure a 0.57 and 0.66 respectively for f1-score. Overall the accuracy was an improvement on the LR baseline.

Table 5 indicates that GB, had the best performance of all models in relation to f1-score of label 1 suggesting GB's power when it comes to dealing with data imbalance. Though GB had the best score of all, it is still a poor result overall comparing it to the f1-scores of lables 2,3 and 4 which were again consistently strong with 0.81, 0.60 and 0.68 respectively. The overall accuracy was comparable with the baseline.

Finally, Table 6 exhibits the results of the best model of them all by accuracy, the stacking classifier. The f1-score for label 1 dips slightly from the 0.20 which GB boasted with a 0.16 for itself, though higher than any other individual model. Stacking produced the highest overall f1-score for label 4, label 3 and label 2 as well as the overall highest accuracy of all models.

Figure 2, Figure 3, Figure 4, Figure 5 all depict confusion matrices constructed on the results of cross validation. The persistent problem throughout all confusion matrices is the two coloured squares for True label 1 and true label 3 where they were predicted as label 2. This is a symptom of each models difficulty with predicting relatively under-represented classes.

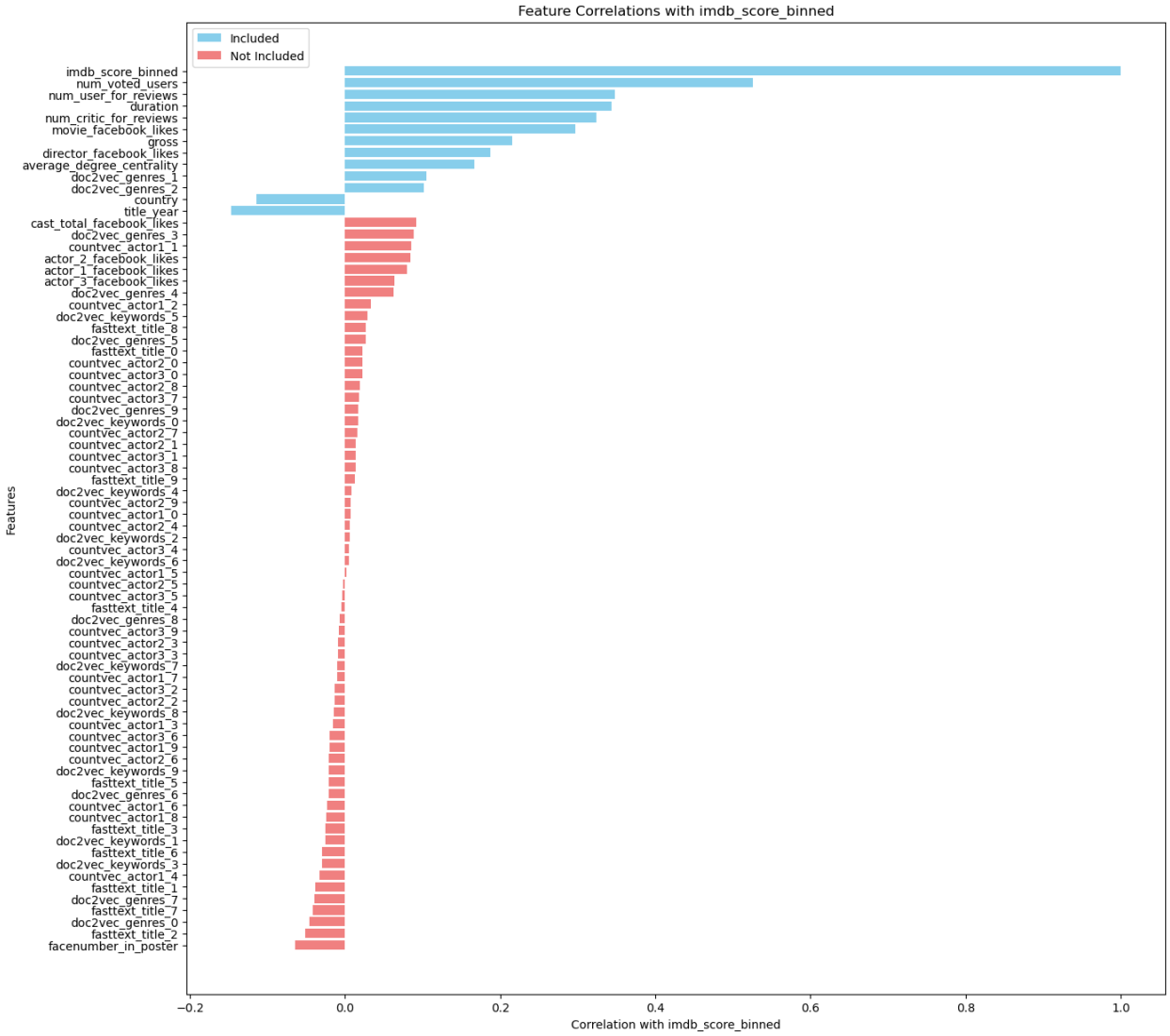

Figure 1: Results of correlation computation between all non-textual features (including given pre processed ones) and the label imdb_score_binned, labels in red had $-0.1 <$ correlation score < 0.1

4 Discussion

4.1 Dataset

The given datasets for this task were significantly impacted by class imbalance (See Table 1. "*Class imbalance poses significant challenges to machine learning models, often leading to biased predictions toward the majority class.*"(He and Garcia, 2009). Due to the fact the class distribution was so skewed towards label 2, models acquired biased tendencies in learning that led to predictions of the majority class when it had difficulties deciding the label for classes in the minority (See Figures 2, 3, 4, 5).

4.2 Model performance

The approach for choosing the 3 individual machine learning models was to pick models that have a reputation for performing well in practice and most importantly had varied strengths and weaknesses due to their unique approaches to learning.

Logistic regression (LR) was chosen first due to its reputation as a simple and powerful baseline model. LR performed strongly overall and is even somewhat comparable to the best final model, the stacking classifier (See Table 3). From Figure 2 and Table 3, it is evident that LR's biggest problem is that it simply could not capture much if any information about the relationships of under-represented classes. Due

label	precision	recall	f1-score	instances
0	0.00	0.00	0.00	24
1	0.11	0.00	0.01	235
2	0.70	0.92	0.80	1839
3	0.63	0.39	0.48	777
4	0.84	0.63	0.72	129
accuracy	-	-	0.70	3004
macro avg	0.46	0.39	0.40	3004
weighted avg	0.64	0.70	0.65	3004

Table 3: Classification report for logistic regression evaluated with stratified k-folds cross validation

label	precision	recall	f1-score	instances
0	0.00	0.00	0.00	24
1	0.70	0.07	0.12	235
2	0.73	0.93	0.82	1839
3	0.67	0.50	0.57	777
4	0.89	0.52	0.66	129
accuracy	-	-	0.72	3004
macro avg	0.60	0.40	0.43	3004
weighted avg	0.71	0.72	0.69	3004

Table 4: Classification report for random forest evaluated with stratified k-folds cross validation

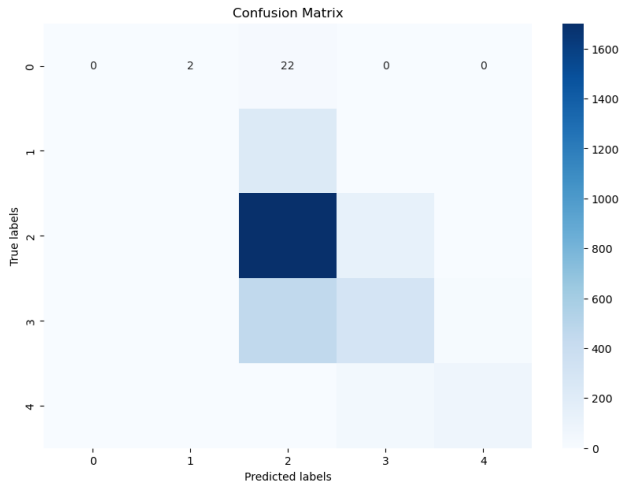

Figure 2: Confusion matrix based on cross validation results for logistic regression, predictions on the x-axis and ground truth on the y-axis

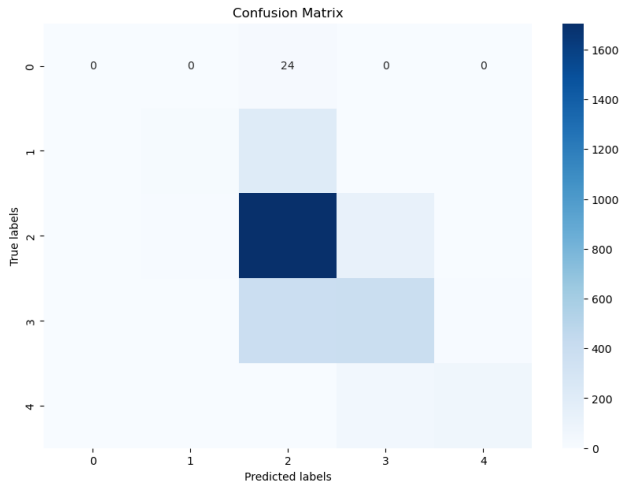

Figure 3: Confusion matrix based on cross validation results for random forest, predictions on the x-axis and ground truth on the y-axis

to LR's process of fitting linear decision boundaries and computing probabilities, it stood to reason that the model was too simple to capture the complexities of the relationships in the data.

To counter LR's complexity problem, random forest (RF) was then employed. RF is an ensemble model of decision trees trained on random subsets of data which can perform feature selection for themselves. RF's process makes it much more suited to capturing more complex relationships in the data than LR. Ultimately, RF did perform better overall than LR (See Table 4) and improved upon predicting

label	precision	recall	f1-score	instances
0	0.00	0.00	0.00	24
1	0.43	0.13	0.20	235
2	0.75	0.89	0.81	1839
3	0.66	0.55	0.60	777
4	0.86	0.57	0.68	129
accuracy	-	-	0.70	3004
macro avg	0.54	0.43	0.46	3004
weighted avg	0.70	0.72	0.70	3004

Table 5: Classification report for gradient boosting evaluated with stratified k-folds cross validation

label	precision	recall	f1-score	instances
0	0.00	0.00	0.00	24
1	0.50	0.10	0.16	235
2	0.75	0.91	0.82	1839
3	0.68	0.55	0.60	777
4	0.85	0.64	0.73	129
accuracy	-	-	0.73	3004
macro avg	0.55	0.44	0.46	3004
weighted avg	0.71	0.73	0.70	3004

Table 6: Classification report for stacking classifier evaluated with stratified k-folds cross validation

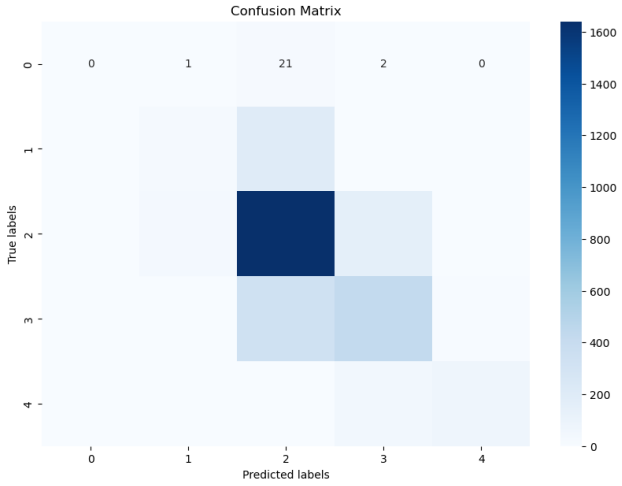

Figure 4: Confusion matrix based on cross validation results for gradient boosting, predictions on the x-axis and ground truth on the y-axis

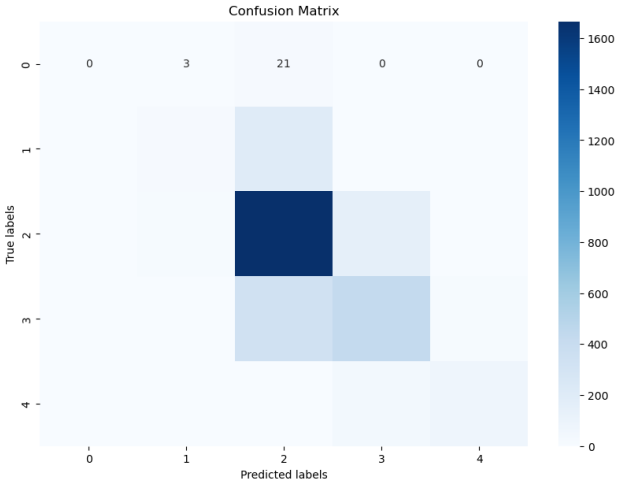

Figure 5: Confusion matrix based on cross validation results for stacking classifier, predictions on the x-axis and ground truth on the y-axis

under-represented classes but not significantly (See Figure 3). The results indicate that RF must have also over-fit to the majority classes in the data due to the class imbalance.

Gradient boosting (GB) was next to be considered. GB, like RF is an ensemble decision tree model which importantly builds trees iteratively which learn from this mistakes of the ones before it using a loss function which is easily optimised for class weights to directly combat imbalance in the data. This approach produced the best results when looking at under-represented classes (See Table 5), however, they weren't good (See Figure 4).

Lastly, a stacking classifier was implemented. The stacking classifier "*involves training a meta-model to combine the predictions of several base models, leveraging their individual strengths to improve overall performance*"(Zhou, 2012). Leveraging the strengths of each base model did in-fact produce the best accuracy of all models explored (See Table 6) by a small margin. Ultimately, stacking had also inherited the collective weakness of the group and failed to classify minority classes reliably or accurately.

4.3 Error Analysis and Evaluation

Confusion matrices and classification reports turned out to be pivotal in producing the final model. Confusion matrices helped to provide visualisations of what each model was deciding when encountering different types of classes. Classification reports gave a highly-detailed contextualisation of which classes were causing problems for the model and ultimately which types of relationships were failing to be captured by the models.

Confusion matrices had the biggest impact when revealing that class 2, the majority class, was being predicted heavily while the ground truth class differed (See Figures 2, 3, 4, 5).

Classification reports were analysed after getting a quick visual representation of the problem from confusion matrices. These reports provided more insight specifically which classes were being predicted or not predicted with precision and recall while providing an overall f1 score.

When comparing models, their performance was evaluated using stratified k-folds cross validation which "*ensures that each fold is representative of the overall class distribution, thus providing more reliable model performance estimates, especially in the presence of imbalanced datasets.*"(Kuhn and Johnson, 2013). Cross validation is crucial when comparing model performance because it counters over-fitting bias, maximises usage of the dataset and maintains a fair comparison between models as it splits consistently for each model.

5 Conclusion

After exploring 4 possible machine learning models and a variety of data pre-processing methods with the goal of predicted binned IMDB ratings on film instances from a dataset, one problem emerged consistently. The dataset's class imbalance proved to be a significant challenge which could not be overcome by the methods employed in this study. Logistic regression proved to be a strong baseline model, random forest improved upon performance by capturing complex relationships in the data and gradient boosting attempted to address the class imbalance problem through weighting its loss function. Ultimately, even a stacking classifier could not combined the strengths of all previous models and overcome the imbalance. Considering the likely context in which one might use this model, a low consequence and low-requirement for absolute precision environment, it is reasonable to conclude that the final stacking classifier has sufficient accuracy to be used in some capacity to get an estimate of binned imdb rating for a movie dataset. Using confusion matrices and classification reports as error analysers suggests that if a more powerful model is required, efforts should focus on robust methods to balance the class distribution without compromising the current strengths of the model.

References

Richard Bellman. 1961. *Adaptive Control Processes: A Guided Tour*. Princeton University Press.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning*. Springer.

Haibo He and Edwardo A Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.

Max Kuhn and Kjell Johnson. 2013. *Applied Predictive Modeling*. Springer, New York.

Michel Verleysen and Damien François. 2005. The curse of dimensionality in data mining and time series prediction. In *Computational Intelligence and Bioinspired Systems*, pages 758–770. Springer.

Zhi-Hua Zhou. 2012. *Ensemble Methods: Foundations and Algorithms*. Chapman and Hall/CRC.