COMP30027 Machine Learning – Project 2: Book Rating Prediction (Report)

Anonymous
Word Count: 1792 words

1. Introduction

How well are different machine learning methods able to predict the ratings of books based on their titles, authors, descriptions, and other features? Predicting book and movie ratings has been an active area of research in machine learning, including predictions before releasing through analysis of visuals and predicting user ratings based on rating history [1]. By applying machine learning and natural language processing (NPL) to textual data, this project aims to predict book ratings and determine the most effective model and its parameters in predicting book ratings.

The data on books were collected from Goodreads. In the training dataset, there were 23063 observations and 5766 observations in the testing dataset. The features included were:

Feature	Type
Name	Textual
Authors	Textual
Publish Year	Numerical
Publish Month	Numerical
Publish Day	Numerical
Publisher	Categorical
Language	Categorical
Pages Number	Numerical
Description	Textual
Rating Label	Numerical

Table 1- Table of features and their data types

With the target variable being the rating label. Through exploration of the training data, the mean publishes year is 2000 and the mean rating is 3.789. As shown in Figure 1, the data is heavily centred in the year 2000, with outliers near 1852 as the minimum publish year. Books published by Oxford University Press and Cambridge University Press make up a plurality of the datasets. There are also 149 missing values for Publisher and 17202 for Language. The testing data also encountered similar proportions of missing values.

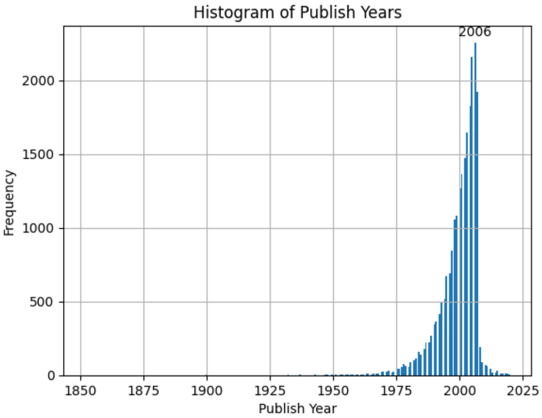

Figure 1- Histogram of frequencies of publishing years.

2. Methodology

2.1 Pre-processing

Firstly, categorical variables like Publisher and Language were encoded into numerical values. This meant that missing values of these features were also encoded as one label. Other missing values of textual and numerical types in the training dataset were dropped.

2.2 Feature Engineering & Selection

Using the NPL tool, Doc2vec, the textual features Name, and Description were converted into vectors of 100 features, whilst Authors was converted into a vector of 20 features. Resulting in a total of 227 features.

The Pointwise Mutual Information (PMI) score, which measures the association between two variables was calculated between all the features and its corresponding rating label. Using this PMI score, the top 20 features were selected to be trained and tested. This included: Publisher, Pages Number, Publish Year, Language, and a mixture of variables from the Doc2vec vectors of Name, Description, and Authors.

2.3 Model Selection

As for machine learning classifiers, Logistic Regression, Random Forest, and Gaussian Naïve Bayes were chosen.

2.3.1 Naïve Bayes

Naïve Bayes is a probabilistic model that assumes independence. And will act as a baseline for comparison. The main parameter is a smoothing parameter.

2.3.2 Logistic Regression

Logistic Regression is a model that calculates the probability of a binary or ordinal outcome from a set of input variables. The main parameters of this model include penalty, which specifies the type of regularisation; C, which controls the inverse of regularisation strength; and solver, which specifies the algorithm to be used.

2.3.3 Random Forest

Random Forest is an ensemble learning algorithm that consolidates multiple decision trees to make predictions. The main parameters include the number of estimators, which specify the number of decision trees in the forest; max depth, which controls the maximum depth of the decision trees; minimum samples split, which determines the minimum number of samples in a node for it to be split; and max features, which controls the number of features to be examined when considering the best split.

3. Results

3.1 Model Training

Firstly, the logistic regression model underwent hyperparameter tuning using grid search, which searches the entire grid of parameter combinations. The parameters were tuned to be, C=0.1, penalty=l1, and solver=liblinear. By setting this penalty with a type of regularisation also known as Lasso, it prevents overfitting by encouraging feature elimination of less important features. Then, trained against an 80-20 split on the training dataset.

For Random Forest, parameters were tuned using randomised search, which randomly samples hyperparameter values to explore a wider hyperparameter space. Resulting in, the number of estimators=300, minimum sample split=10, max features=square root, and max depth=none. Then, trained similarly to the logistic regression model.

Similarly, the Gaussian Naïve Bayes parameters were tuned using grid search. Resulting in a smoothing factor equalling 1e-6, then trained similarly to the other models.

3.2 Evaluation Metrics

The performance metrics used in the evaluation are accuracy, precision, recall and F1-score.

3.2 Results

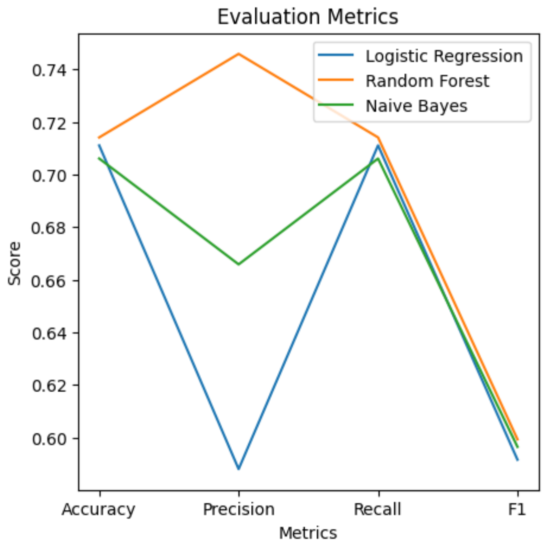

Figure 2 – Line chart of metrics for all models

The logistic regression model has an accuracy of 71.10% and the random forest model has 71.41%. Both have higher accuracy than the baseline accuracy, 70.60%. However, the logistic regression model has a precision of 58.80% which is lower than the baseline (66.58%). Whilst the random forest model has a precision higher than the baseline (74.59%).

Using Naïve Bayes as the benchmark, it can be observed that random forest outperforms both Naïve Bayes and logistic regression models.

4. Discussion and Critical Analysis

4.1 Model Interpretation

All models were trained under the same 20 feature lists and underwent hyperparameter tuning using either randomised search or grid search. Considering accuracy, efficiency, and performance on Kaggle, the random forest was the best model.

20 was chosen as the ideal number of features to be trained with due to improved model performance with lower and more important features. Thus, reducing noise and unnecessary complexity of the model. Through this, the models are also able to be better generalised to

unseen data as overfitting is reduced. By using a smaller feature dataset, the execution times of the models were also improved drastically, which was a main concern of this project.

Naïve Bayes was chosen as the baseline model as it is simple, commonly used and can provide competitive results. Since the rating label has an ordinal scale (3-5), logistic regression was chosen as it would be a well-suited model for this classification task. Moreover, random forest was chosen to test if a non-linear decision tree model could provide better predictions for the data. Its robustness to numerical and categorical features made it suitable for this dataset.

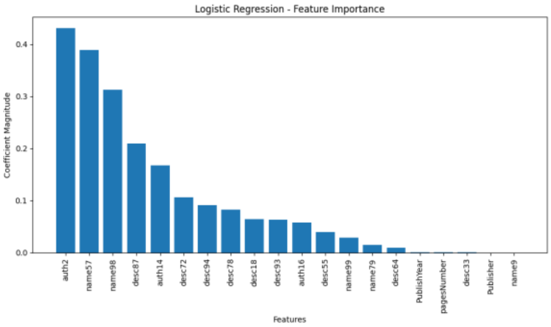

Figure 3 – Bar chart of feature coefficients in the logistic regression model

According to the feature coefficients in Figure 2, the vector representations of authors and names were most important to the logistic regression model.

4.2 Error Analysis

4.2.1 Naïve Bayes

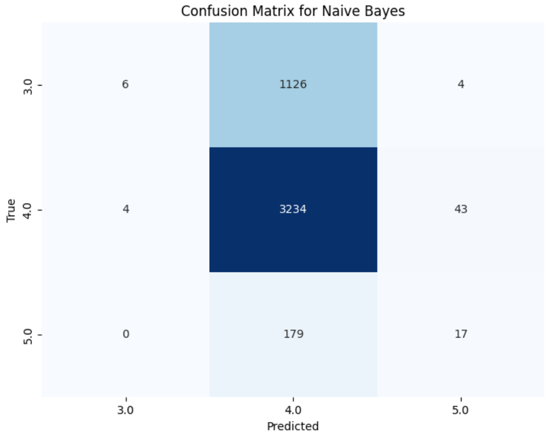

Figure 4 – Confusion matrix of the Naïve Bayes model

The Naïve Bayes model struggled to accurately differentiate books with a true rating label of 3 from 4. This may be due to the model assuming a Gaussian distribution of the features, which may be untrue. The Naïve Bayes model also assumes independence of features. Therefore, due to a combination of these reasons, the model may not be able to capture the patterns in the dataset as accurately as possible.

To counter this, it may be worthwhile to explore ensemble methods like bagging or boosting to improve the model's accuracy.

4.2.2 Logistic Regression

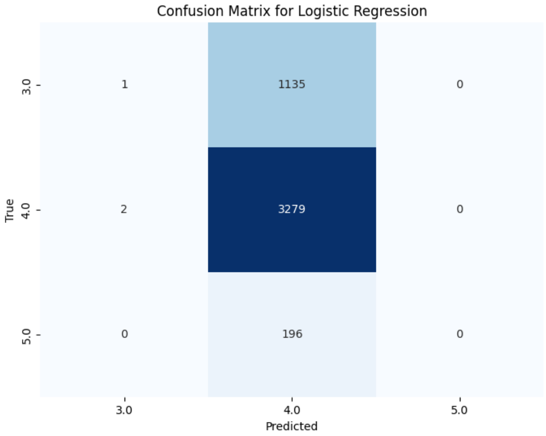

Figure 5 – Confusion matrix of logistic regression model

The logistic regression model made almost all of its predictions to be 4 although the true ratings of a plurality of the data were 3 or 5. Thus, it is more likely to produce errors when the true rating is 3 or 5. This may be because the model is predicting probabilities of the classes that are not distinctly different, as the rating scale (3-5) is neighbouring each other. Thus, the model is unable to effectively differentiate between the different ratings, which leads to many of the predictions falling into the middle value (4). This can be combatted by increasing the decision threshold for middle values so that other values are prioritised and the probability of predicting middle values can be reduced.

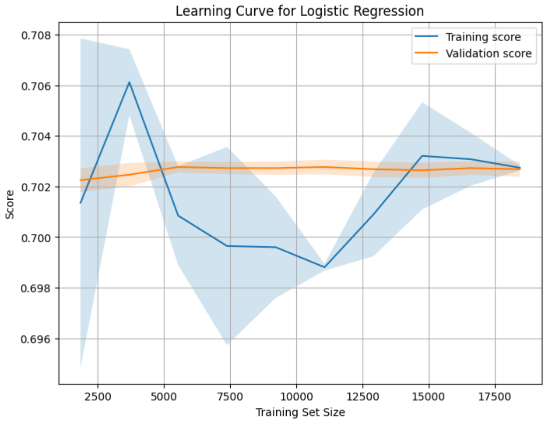

Figure 6 – Learning curve of logistic regression model

As shown in Figure 5, the logistic regression model has learned the underlying patterns of the data well as the training and validation score converge near the end of the dataset. This means that the model may not benefit from more data and other means such as new features or a change to the parameters are needed.

4.2.3 Random Forest

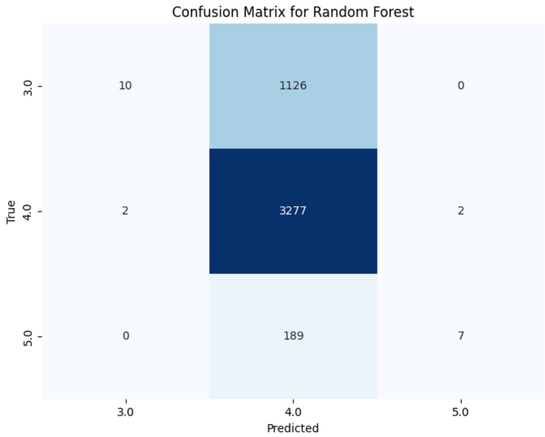

Figure 7 – Confusion matrix of random forest model

As shown in Figure 6, the random forest model also fails to accurately predict book ratings where its true value is 3 or 5. Therefore, being more likely to produce errors when the true rating is 3 or 5. This may be due to the imbalanced data where there are limited instances of data where the true rating equals 5. Thus, the model may be biased towards ratings of 4, which is the majority class, leading to the model struggling to predict the minority classes accurately.

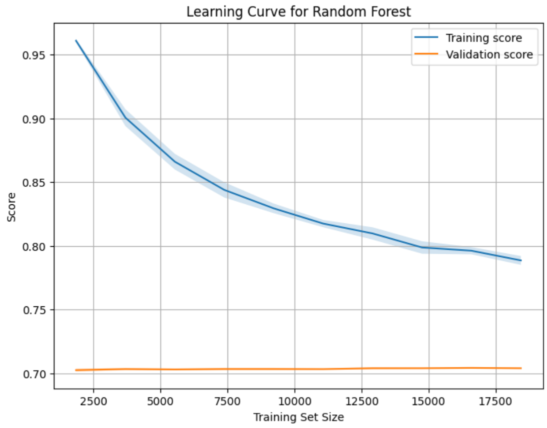

Figure 8 – Learning curve of random forest model

By examining the learning curve of the random forest model (Figure 6), the significant gap between the training and validation score indicates a potential overfitting problem and high variance. There may also be a high bias problem as seen by the training and validation score's reluctance to improve. However, the learning curve variability shows that the model has high levels of stability.

To tackle these errors, hyperparameters must be tuned differently for the random forest model to achieve optimal performance. So that the number of trees is reduced, the depth of trees is limited, and the minimum sample split is increased to find the most ideal combination of parameters.

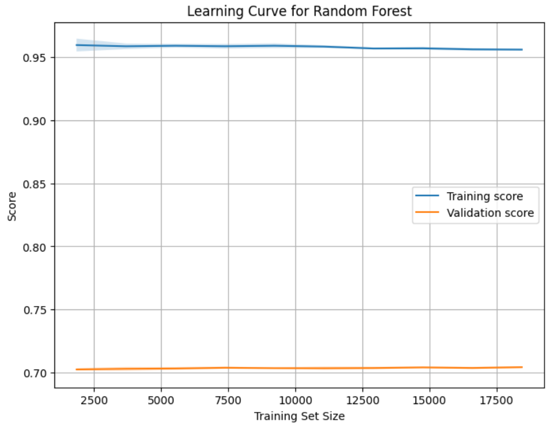

Figure 9 – Learning curve of adjusted random forest model

By eliciting these changes, a similar performing random forest model is produced. However, both the validation and training score of its learning curve is stagnant. This suggests that the model had reached a plateau in its learning process. This may be resolved by increasing the complexity of the model, by increasing the number of trees and other hyperparameters. Regardless of the random forest model's ability to learn the underlying patterns of the dataset, it is the best-performing model of the three.

5. Conclusion

Overall, by training three different models on the book dataset with feature selection and hyperparameter tuning, approximately 70% of the ratings were accurately predicted through the use of random forest and logistic regression. When compared to the Naïve Bayes model, the two models performed better than the baseline, predicting the book ratings with higher accuracy, with the random forest being the best-performing model. Among all features, features within the Doc2vec vectors of authors, names and descriptions had the highest correlation and predictive power for book ratings.

Adding new features, such as user preferences, social media posts related to the books, and book genres could help improve the rating predictions. There is also a notable opportunity to explore sentiment analysis on book reviews, as well as analysis of publish date times to capture trends and seasonality in books.

6. References

[1] Y. Yang, R. Ma and M. H. Cho. “Predicting Movie Ratings with Multimodal Data,” 2019 https://cs229.stanford.edu/proj2019aut/data/assignment_308832_raw/26260680.pdf