COMP30027 Report
Book Rating Prediction

Anonymous
Word Count: ~ 1900 words (excl. Tables and Headings)

1. Introduction

1.1 Problem Statement

The machine learning model developed within this project is intended to predict book ratings based on several features relating to the book such as book name, author, publisher, book description, etc.

1.2 Objective

The main objective of this project is to explore effective features, implement and compare different machine learning models, and conduct error analysis to ultimately produce a model that solves the problem to a satisfactory degree.

2. Data Description

2.1 Data Source

All data used in this project was sourced from Goodreads[1], a platform that allows users to search its database of books, rate books and write reviews.

2.2 Data Overview

The data consists of 10 features (book name, authors, publish year, publish month, publish day, publisher, language, number of pages, book description). There are 5766 observations in the provided testing data (which do not include the target variable), and 23063 observations in the provided training data. The target variable is the book's rating out of 5 stars on Goodreads.

2.3 Exploratory Data Analysis

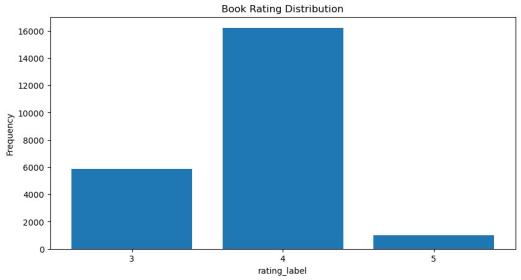

Figure 2.3 - Class Distribution for Book Rating

The training dataset appears to be quite unbalanced / skewed as a large portion of the instances have a 4 star rating, a much smaller amount have a 3 star rating, and an even smaller portion of the instances have a 5 star rating as seen in the figure above (Figure 2.3).

There are a significant number of categories present within the categorical features 'name', 'authors', 'publisher', and 'description', which is unsurprising considering the textual nature of these features. Therefore, feature extraction and selection seems appropriate for this dataset. The 'language' feature seems to consist mostly of missing values, the category 'eng' (i.e. the book is written in English) and occasionally some other languages. All other features are numeric.

[1] https://www.goodreads.com/

3. Methodology

3.1 Pre-processing

One hot encoding was applied to the categorical features 'publisher' and 'language' to allow machine learning models to better utilise them. This simultaneously dealt with any missing values present within these features by providing an encoding for the missing values.

3.2 Feature Engineering / Selection

As noted in the exploratory data analysis (Section 2.3), there are several categorical textual features present within the data, with which most machine learning models would struggle to utilise if used in their original form. Therefore, doc2vec and count vectorizer bag-of-word features were extracted from the 'name', 'authors', and 'description' features.

The final features were then selected based on which combination of features provided the best performance across the different classifiers after the classifiers were trained on all instances. This resulted in the count vectorizer feature extractions being used for 'name' and 'authors', and the doc2vec feature extractions being used for 'description'. F-test feature selection was used to select the 5000 best features, as this appeared to be an amount which didn't noticeably impact model performance across different classifiers for this feature set.

3.3 Model Selection

The machine learning models chosen were Naïve Bayes, Logistic Regression, and a Support Vector Machine. These models were selected due to their promising performance metrics in the feature selection phase, and also due to their varying strengths and weaknesses in an attempt to increase the chance of a satisfactory model being produced. These classifiers also seemed to be appropriate considering the discrete and binary nature of the features (besides the description doc2vec vector).

3.4 Training Process

Each model was trained and evaluated using stratified 5-fold cross validation and averaging of evaluation metrics to form a fair and accurate evaluation of each model over different training and validation splits. Stratified 5-fold cross validation was used to train and evaluate the models on splits that consistently and closely match the class distribution of the whole training dataset. This was performed under the assumption that the training dataset is a reasonable representation of the true distribution of book ratings on Goodreads, and also to produce consistent results across different training-validation splits.

For the Logistic Regression and Support Vector Machine models, regularisation was applied to discourage overfitting while also still attempting to create a satisfactory fit.

Grid search cross validation was utilised to try to optimise the hyperparameters (including regularisation amounts) for each model where applicable.

3.5 Hyperparameters

Naïve Bayes: Laplace smoothing: 1,

Bernoulli Distribution

SVM: *C*: 0.5,

Kernel: Linear,

One-vs-Rest

Logistic Regression: *C*: 0.1,

Solver: Newton-CG,

Penalty: L2

3.6 Evaluation Metrics

The models were all evaluated based on their accuracy, precision, recall, and f1-scores. Confusion matrices were also used to evaluate and perform error analysis on the models. All models were compared against each other and a 0R baseline model using these metrics.

4 Results

Below are the averaged classification reports and confusion matrices for 0R and all 4 models after stratified 5-fold cross validation was performed.

4.1 0R

	precision	*recall*	*F1-score*	*support*
3.0	0.00	0.00	0.00	1172.8
4.0	0.70	1.00	0.83	3241.6
5.0	0.00	0.00	0.00	198.2
Training Accuracy			0.70	4612.6
Validation Accuracy			0.70	4612.6
Macro Average	0.23	0.33	0.28	4612.6
Weighted Average	0.49	0.70	0.58	4612.6

Table 4.1.1- Classification Report (5 fold average)

	3.0	*4.0*	*5.0*	*Predicted*
3.0	**0**	**1172.8**	**0**	
4.0	**0**	**3241.6**	**0**	
5.0	**0**	**198.2**	**0**	
Actual				

Table 4.1.2- Confusion Matrix (5 fold average)

4.2 Bernoulli Naïve Bayes

	precision	*recall*	*F1-score*	*support*
3.0	0.67	0.38	0.49	1172.8
4.0	0.79	0.94	0.86	3241.6
5.0	0.92	0.38	0.54	198.2
Training Accuracy			0.81	4612.6
Validation Accuracy			0.78	4612.6
Macro Average	0.79	0.57	0.63	4612.6
Weighted Average	0.77	0.78	0.75	4612.6

Table 4.2.1- Classification Report (5 fold average)

	3.0	*4.0*	*5.0*	*Predicted*
3.0	**450.0**	**772.0**	**0.8**	
4.0	**184.0**	**3051.2**	**6.4**	
5.0	**36.2**	**86.2**	**75.8**	
Actual				

Table 4.2.2- Confusion Matrix (5 fold average)

4.3 Linear Kernel SVM

	precision	*recall*	*F1-score*	*support*
3.0	0.65	0.20	0.31	1172.8
4.0	0.74	0.96	0.83	3241.6
5.0	0.73	0.10	0.18	198.2
Training Accuracy			0.77	4612.6
Validation Accuracy			0.73	4612.6
Macro Average	0.71	0.42	0.44	4612.6
Weighted Average	0.71	0.73	0.67	4612.6

Table 4.3.1- Classification Report (5 fold average)

	3.0	*4.0*	*5.0*	*Predicted*
3.0	**238.2**	**934.0**	**0.6**	
4.0	**126.6**	**3108.2**	**6.8**	
5.0	**4.2**	**173.8**	**20.2**	
Actual				

Table 4.3.2- Confusion Matrix (5 fold average)

4.4 Logistic Regression

	precision	*recall*	*F1-score*	*support*
3.0	0.67	0.19	0.30	1172.8
4.0	0.74	0.97	0.83	3241.6
5.0	0.71	0.07	0.13	198.2
Training Accuracy			0.76	4612.6
Validation Accuracy			0.73	4612.6
Macro Average	0.70	0.41	0.42	4612.6
Weighted Average	0.72	0.73	0.67	4612.6

Table 4.4.1- Classification Report (5 fold average)

	3.0	*4.0*	*5.0*	*Predicted*
3.0	**226.8**	**945.4**	**0**	
4.0	**109.2**	**3127.4**	**5.0**	
5.0	**2.2**	**182.2**	**13.8**	
Actual				

Table 4.4.2- Confusion Matrix (5 fold average)

4.5 Feature Extraction Testing

Below are the classification reports for a subset of classifiers used for feature selection (Bernoulli Naïve Bayes and Linear Kernel SVM). These classifiers were trained on different combinations of feature extractions (with F-test feature selection also being applied to each combination to reduce the feature set while still retaining model performance). All classifiers used were evaluated using a 20% holdout set.

4.5.1 Bag-of-words 'Name', 'Authors', 'Description'

	precision	*recall*	*F1-score*	*support*
3.0	0.46	0.32	0.37	1200
4.0	0.74	0.86	0.79	3208
5.0	0.52	0.12	0.20	205
Training Accuracy			0.81	4613
Validation Accuracy			0.68	4613
Macro Average	0.57	0.43	0.45	4613
Weighted Average	0.65	0.68	0.66	4613

Table 4.6.1.1- BNB Classification Report

	precision	*recall*	*F1-score*	*support*
3.0	0.51	0.16	0.24	1200
4.0	0.72	0.94	0.81	3208
5.0	0.58	0.10	0.17	205
Training Accuracy			0.84	4613
Validation Accuracy			0.70	4613
Macro Average	0.60	0.40	0.41	4613
Weighted Average	0.66	0.70	0.64	4613

Table 4.6.1.2- SVM Classification Report

4.5.2 Bag-of-words 'Name', 'Authors', and Doc2Vec 'Description'

	precision	*recall*	*F1-score*	*support*
3.0	0.60	0.14	0.23	1200
4.0	0.72	0.97	0.82	3208
5.0	0.78	0.03	0.07	205
Training Accuracy			0.84	4613
Validation Accuracy			0.71	4613
Macro Average	0.70	0.38	0.37	4613
Weighted Average	0.69	0.71	0.63	4613

Table 4.6.2.1- BNB Classification Report

	precision	*recall*	*F1-score*	*support*
3.0	0.62	0.10	0.18	1200
4.0	0.71	0.98	0.82	3208
5.0	1.00	0.05	0.09	205
Training Accuracy			0.76	4613
Validation Accuracy			0.71	4613
Macro Average	0.78	0.38	0.36	4613
Weighted Average	0.70	0.71	0.62	4613

Table 4.6.2.2- SVM Classification Report

4.5.3 Bag-of-words 'Authors', and Doc2Vec 'Name', 'Description'

	precision	*recall*	*F1-score*	*support*
3.0	0.54	0.18	0.27	1200
4.0	0.72	0.94	0.82	3208
5.0	0.58	0.03	0.06	205
Training Accuracy			0.82	4613
Validation Accuracy			0.70	4613
Macro Average	0.61	0.38	0.38	4613
Weighted Average	0.67	0.70	0.64	4613

Table 4.6.3.1- BNB Classification Report

	precision	*recall*	*F1-score*	*support*
3.0	0.59	0.07	0.12	1200
4.0	0.71	0.98	0.82	3208
5.0	1.00	0.04	0.08	205
Training Accuracy			0.73	4613
Validation Accuracy			0.70	4613
Macro Average	0.77	0.36	0.34	4613
Weighted Average	0.69	0.70	0.61	4613

Table 4.6.3.2- SVM Classification Report

4.5.4 Doc2Vec 'Name', 'Authors', 'Description'

	precision	*recall*	*F1-score*	*support*
3.0	0.47	0.26	0.33	1200
4.0	0.72	0.86	0.79	3208
5.0	0.20	0.14	0.16	205
Training Accuracy			0.72	4613
Validation Accuracy			0.67	4613
Macro Average	0.46	0.42	0.43	4613
Weighted Average	0.63	0.67	0.64	4613

Table 4.6.4.1- BNB Classification Report

	precision	*recall*	*F1-score*	*support*
3.0	0.56	0.11	0.19	1200
4.0	0.71	0.97	0.82	3208
5.0	0.43	0.03	0.05	205
Training Accuracy			0.72	4613
Validation Accuracy			0.70	4613
Macro Average	0.57	0.37	0.35	4613
Weighted Average	0.66	0.70	0.62	4613

Table 4.6.4.2- SVM Classification Report

5 Results Discussion and Critical Analysis

5.1 Feature Selection / Engineering

Feature selection was performed to reduce the dimensionality of the data as much as possible without heavily impacting model performance by losing important features. In relation to the feature extractions that ended up being utilised, intuitively, it makes sense that word counts would be a more effective representation of short textual features like book names and authors, while a doc2vec representation would be more effective for capturing contextual similarities between the longer descriptions of books, thus better enabling models to utilise those similarities in their predictions.

The varying performance present between the classifiers (Tables 4.6.1.x - 4.6.4.x) when using different combinations of feature extractions can be explained by the fact that different models perform differently with different data representations, and by the fact that certain feature extractions capture information about certain types of text more effectively than others. Additionally, the varying performance would seem to indicate that book name, authors and book description have a measurable relationship with book rating that requires suitable feature extraction to be properly utilised by the models.

5.2 Model Performance

Bernoulli Naive Bayes assumes that the input features are binary (i.e. 0 or 1), and therefore works well for text classification problems such as this where there are features which represent the presence or absence of words. Technically bag-of-words are word counts, however, these can still function as binary attributes as it's extremely likely that important words only appear once in book names. Obviously, author names would only appear once per instance as well and therefore their word counts can also be treated as binary. The suitability of the data representation for Bernoulli Naïve Bayes could explain why the Naïve Bayes model performed slightly better than the other models (Tables 4.2.1 - 4.4.1).

Additionally, it could simply be that the classes are difficult for models such as SVM and Logistic Regression to separate due to a lack of distinctive meaningful relationships between features and the class labels.

Overall, all the models performed fairly well in terms of predicting 4.0 book ratings but did quite poorly in terms of predicting 3.0 and 5.0 book ratings.

5.3 Error Analysis

5.3.1 Error Identification

Predictions were obtained through stratified 5-fold cross validation of the training data as explained in Section 3. The results of these predictions (Section 4) will be used to perform error analysis.

Upon further inspection of the confusion matrices produced during cross validation (Tables 4.2.2 – 4.4.2), it appears that all models tested tend to misclassify 3.0 book ratings as 4.0 book ratings very often. A similar misclassification also tends to occur where the models misclassify 5.0 book ratings as 4.0 book ratings. Only Bernoulli Naïve Bayes (Table 4.2.2) seems to correctly predict 5.0 book ratings nearly as much as it incorrectly classifies them as 4.0. However, Bernoulli Naïve Bayes also appears to incorrectly classify 5.0 book ratings as 3.0 book ratings much more often than the other classifiers (Tables 4.3.2 and 4.4.2), which tend to mostly misclassify 5.0 as 4.0.

5.3.2 Error Explanation

It is immediately clear from the class label distribution (Figure 2.3) that the tendency for the models to misclassify instances as 4.0 book ratings can be partially explained by the large presence of 4.0 book ratings in the training set. This could cause the models to become much better at classifying 4.0 book rating instances than 3.0 or 5.0. Another explanation as to why the models struggle to classify 3.0 and 5.0 book ratings is likely that the features obtained in the training data do not correlate strongly enough with the rating of a book, therefore causing the models to have insufficient information to reliably distinguish between the majority class (4.0) and other less prevalent classes.

Regarding the Bernoulli Naïve Bayes classifier's tendency to misclassify 5.0 book ratings as 3.0 more often than other book ratings, this is likely due to the nature of Naïve Bayes utilising prior probabilities for the classes. The 3.0 book rating instances, while appearing much less in the training data than 4.0, appear much more frequently in the data than 5.0 book ratings (Figure 2.3). Therefore, it isn't unusual that the Bernoulli Naïve Bayes tends to make such errors since the prior probability of an instance having a 3.0 rating is much larger than the prior probability of it having a 5.0 rating. Similar logic also applies to the reasoning for why Bernoulli Naïve Bayes tends to misclassify both book ratings as 4.0.

The imbalance of the dataset likely isn't affecting the performance of the SVM and Logistic Regression models as heavily as it is the Naïve Bayes model. This would be due to how these models rely less on the class distribution and moreso on how easy it is to separate the different classes, which is largely determined by how well different feature values correlate with the different classes. Due to the lack of strong correlations between the distinct class labels and feature values, several instances with 3.0 and 5.0 ratings must be falling within the margin for instances with 4.0 book ratings when compared with the rest. This misclassification is evident in the confusion matrices (Tables 4.3.2 and 4.4.2).

5.3.3 Error Resolution

The most appropriate steps that could be taken to try and resolve these errors include obtaining more samples that have 3.0 and 5.0 book ratings to give the models more information to train on in relation to those classes (i.e. forming a more balanced training dataset). Alternatively, although not as ideal as obtaining new data, the training data could be modified to have a more balanced class distribution. This likely would have a more noticeable impact on the performance of the Naïve Bayes classifier which relies directly on the prior probabilities of classes to make predictions.

Engineering new features or obtaining new features relating to the book which may better predict a book's rating or are more usable by the selected models would certainly reduce misclassification and would likely have the greatest impact across most models in general.

6 Conclusion

6.1 Summary of Findings

In conclusion, moderately accurate models were produced for predicting book labels. However, the main reason the models may have appeared to perform well (accuracy-wise) during cross validation is largely due to the major presence of 4.0 book ratings in the training data. The models unsurprisingly are good at predicting 4.0 book ratings but perform quite poorly when it comes to predicting 3.0 and 5.0 book ratings. The errors present in these models are likely mostly explained by the imbalance of class distribution and the lack of meaningful relationship expressed in the training data between features and class labels.

6.2 Future Work

Gathering features that better predict book rating, gathering more balanced data, utilising data that is better scaled, and usage of textual feature extractions that better capture the relationship between the original features and book ratings are things which could all be done in future to improve the performance of these models.