COMP30027 - Assignment 2 Report

Anonymous

1 Introduction

Since ancient times, books have always played a significant role in human development. Nowadays, with the advancement of technology, books are abundant around us. However, there is a emerging issue of books of low quality.

In this study, we will use data collected from Goodreads (a platform that allows user to search its database of books, rate books and write reviews) to predict the rating of books based on various books features.

2 Dataset

Our dataset contains 23063 books obtained from Goodreads. In our data, each book contains:

- Book features: name, authors, publish year, publish month, publish day, publisher,language, page numbers, and description.
- Text features: produced by various text encoding methods for name, authors, and description. Each text feature is provided as a single file with rows corresponding to the file of book features.
- Class label: the rating of a book rating label (3 possible levels, 3, 4 or 5)

2.1 Train-Test Split

We use 80% of our instances for training and the other 20% for testing.

2.2 Exploratory Data Analysis

Looking at the data distribution in (Figure 1), the distribution of classes in both training dataset and testing dataset is similar. Also, this data is unbalanced since the number of *ratinglabel 4.0* is significant larger the other 2 labels. Therefore, we have to use method to deal with the inherent unbalance of this dataset.

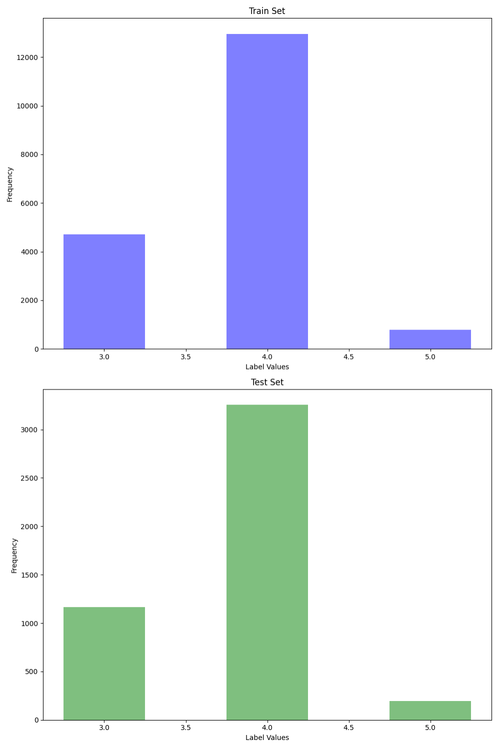

Figure 1: Distribution of classes in Training Dataset and Testing Dataset

3 Methodology

3.1 Data Preprocessing

3.1.1 Missing Data

Here is the proportion of missing data in features with missing values.

Features	Proportion
Publisher	0.006417
Language	0.745870

Table 1: Proportion of missing data in a feature

Since the proportion of missing data in features *Publisher* is less than 1%, we just convert the missing data into a new label *NA*.

Having a look at the distribution of classes for each value of feature *Language*:

Class	Full	Missing
4.0	70.277067	67.672364
3.0	25.426007	27.467736
5.0	4.296926	4.859900

Class	eng	fre	spa
4.0	78.128440	73.376623	79.194631
3.0	19.357798	22.077922	18.791946
5.0	2.513761	4.545455	2.013423

Table 2: Label distribution in *Language*

In addition to having 75% of missing data, distribution of the class for each value in *Language* is close to the distribution in the whole dataset. Hence, we remove this feature from our model.

3.1.2 Text Preprocessing

- *Authors*:
  We use one-hot encoding to transform this categorical features to numerical. This creates features representing authors appearance in each instance. Using χ^2 for feature selection, we choose the best 10 features from newly created features.
  The result features are: *Anonymous, Carole Mortimer, Charles Fillmore, Creflo A. Dollar, M. Fethullah Gülen, Paramahansa Yogananda, Paul Metcalf, Peter Egan, Quino, W.E. Vine.*

- *Publisher*:
  We convert this categorical feature to numerical using the same method as using on *Authors.*
  The result features are: *Banner of Truth, Hal Leonard Publishing Corporation, Harrison House, ICS Publications, 'Light, Inc.', Museum of Modern Art, Self-Realization Fellowship Publishers, TAN Books, TFH Publications, VIZ Media*

- *Name*:
  We used the provided pretrained *CountVectorizer* for this features. Using χ^2 for feature selection, we choose the best 10 features from newly created features.
  The result features are: *bible, holy, mafalda, nausicaä, niv, novel, pokemon, st, vine, yotsuba*

- *Description*:
  We convert this categorical feature to numerical using the same method as using on *Name.*
  The result features are: *bible, concordance, god, herriot, liszt, nahuatl, objectivist, rinpoche, web, yogananda*

3.2 Models

3.2.1 Baseline

First of all, we use a simple Zero-R model, which make predictions solely based on the most frequent class in the training data, as the baseline model.

3.2.2 Logistic Regression

The second model is Logistic Regression. Logistic Regression is a suitable algorithm for multiclass classification. The coefficients from the models can also give us the impact of each features on the model predictions. Regularization techniques such as Ridge or Lasso can also be applied to mitigate overfitting.

3.2.3 Boosting

The third model is Boosting with Decision Trees as base classifiers. Boosting is suitable for this dataset since it creates new samples based on the current dataset, which can reduce the sampling bias of the current dataset. We can also change the base classifiers of the boosting algorithm to deal with overfitting or increase the strength of our model. In this study, our boosting algorithms is AdaBoost.

4 Model Training and Evaluation

4.1 Training Process

4.1.1 Baseline

First of all, we use a simple Zero-R model, which make predictions solely based on the most frequent class in the training data, as the baseline model.

4.1.2 Logistic Regression

The second model is Logistic Regression. We look for the changes in the performance when we use different algorithms for optimisation, different penalty strength or without any penalty.

Hyperparameters Tuning

We use cross-validation on the training dataset to find the best hyperparameters.

- **No Penalty**
  This model we use Limited-memory Broyden–Fletcher–Goldfarb–Shanno Algorithm

(L-BFGS) as the algorithms for optimisation and no penalty is added.

- **With Penalty**
  This model we include the Ridge regularization as the penalty. Using regularization algorithms could reduce overfitting of a model.
  We also change the optimisation algorithms and the strength penalty (regularisation strength).
  Optimisation Algorithms: Newton (*'newton-cg'*), L-BFGS (*'lbfgs'*), Large Linear Classification (*'liblinear'*).
  Inverse of Regularisation Strength: 100, 10, 1.0, 0.1, 0.01

4.1.3 Boosting

The third model is Boosting with Decision Trees as base classifiers. In this study, our boosting algorithms is AdaBoost. We look for changes in the performance when we change the number of decision trees or the maximum depth of decision trees used in the ensemble.

Hyperparameters Tuning

We use cross-validation on the training dataset to find the best hyperparameters.

- **Number of Base Classifier Decision Trees**
  We find the best number of Decision Trees to deal with overfitting and increase the strength of the model. More Decision Trees could increase the strength of the model, but it could also lead to overfitting.
  Number of Trees: 10, 50, 100, 500, 1000, 1500

- **Maximum depth of Base Classifier Decision Trees**
  We find the best maximum depth of Decision Trees to increase the strength of the model. As the depth of the base Decision Trees increase, it may increase the strength of the model.
  Range of Maximum Depth: [1, 10]

4.2 Evaluation

We use accuracy as evaluation metrics. We also use confusion matrix for a detailed breakdown of the model predictions and the actual classes of the instances.

4.3 Results

Logistic Regression with Regularisation Parameters

The accuracy of Logistic Regression models with different hyperparameters.

C	solver	Mean	Std
100.00	newton-cg	0.7107	0.0028
100.00	lbfgs	0.7023	0.0010
100.00	liblinear	0.7032	0.0013
10.00	newton-cg	0.7107	0.0028
10.00	lbfgs	0.7022	0.0010
10.00	liblinear	0.7032	0.0012
1.00	newton-cg	0.7090	0.0025
1.00	lbfgs	0.7024	0.0012
1.00	liblinear	0.7032	0.0013
0.10	newton-cg	0.7076	0.0024
0.10	lbfgs	0.7025	0.0012
0.10	liblinear	0.7028	0.0012
0.01	newton-cg	0.7033	0.0013
0.01	lbfgs	0.7023	0.0009
0.01	liblinear	0.7023	0.0009

Table 3: Accuracy of selected models

The best Logistic Regression Model here is the Logistic Regression with hyperparameters 'C': 100, 'solver': 'newton-cg'.

AdaBoost with Regularisation Parameters

- **Number of Base Classifier Decision Trees**
  The best number of trees here is 100 trees.

n_trees	Mean	Median	Std
10	70.373984	70.352304	0.164160
50	70.915989	70.921409	0.291073
100	70.925023	70.975610	0.265041
500	70.861789	70.867209	0.294085
1000	70.834688	70.840108	0.306077
1500	70.816621	70.785908	0.293112

Table 4: Accuracy of selected models

- **Maximum depth of Base Classifier Decision Trees**
  Surprisingly, the best maximum depth is 1, which is the default AdaBoost.

The accuracy of the selected models.

Confusion matrix for each model

depth	Mean	Median	Std
1	70.919603	70.921409	0.286370
2	70.719061	70.731707	0.290090
3	70.289070	70.271003	0.348484
4	69.181572	69.214092	0.466377
5	67.185185	67.208672	0.782528
6	63.123758	63.170732	0.729796
7	60.863595	60.731707	1.164823
8	61.121951	61.029810	0.854485
9	63.026197	63.143631	0.755196
10	63.971093	63.875339	1.014862

Table 5: Accuracy of selected models

Model	Accuracy
Zero-R	70.54%
No Penalty Logistic Regression	70.50%
With Penalty Logistic Regression	71.17%
Default AdaBoost	71.17%
100-Tree AdaBoost	71.13%

Table 6: Accuracy of selected models

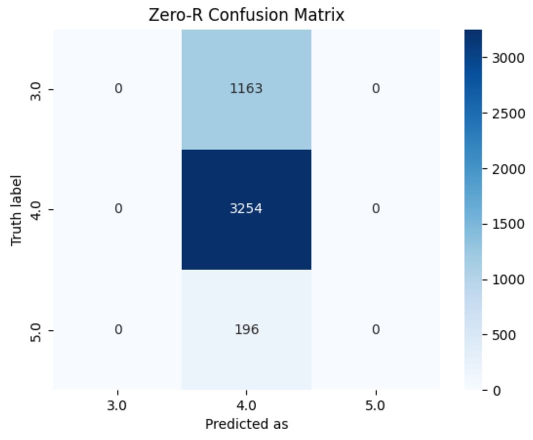

Figure 2:

5 Discussion and Critical Analysis

First, we can see that all of our model predictions contain a lot of rating_labels 4.0. However, the accuracy is still high, which is around 70%. This behavior is expected since the distribution of classes in our dataset is not balanced. We can solve this problem by getting more data or use any random sampling methods such as bagging or decision tree.

There is an increase in accuracy when we use regularization techniques on Logistic Regression

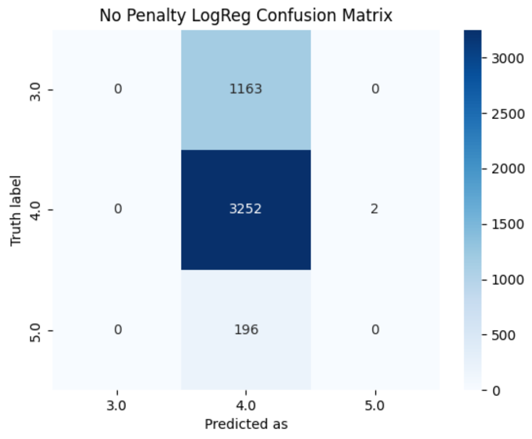

Figure 3:

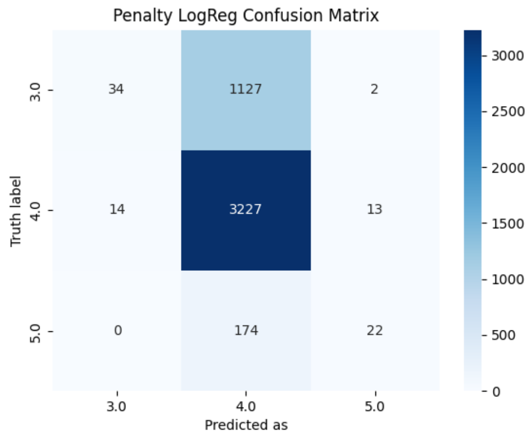

Figure 4:

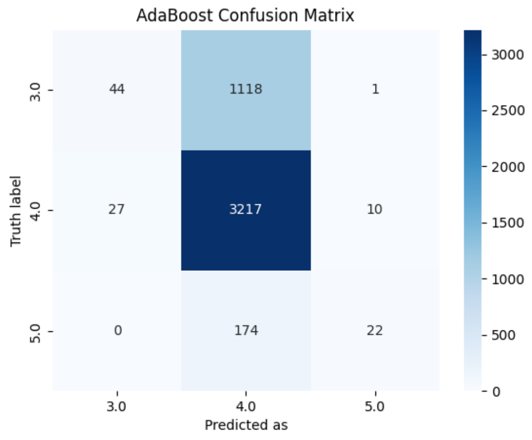

Figure 5:

models. From confusion matrix Figure 3 and Figure 4, we can see that accuracy of predicting the rating labels 4.0 decrease. However, there

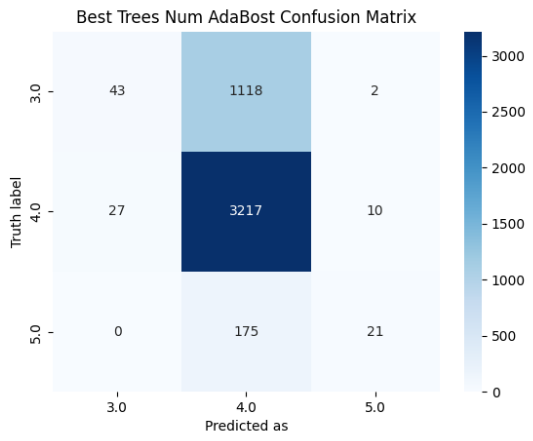

Figure 6:

is a slight increase in the accuracy of predicting others labels. This is because regularization techniques have mitigated overfitting of the model.

There is a surprise result is that when we are finding the best maximum base Decision Tree depth for AdaBoost, the best maximum depth is 1, which means the best base model is the Decision Stump. And our best number of Decision Trees model perform slightly worse than the original model. These may be caused by the unbalanced data in the whole dataset and in both training and testing. We can solving this by using cross-validation to train and choose the best model or using a different random train-test split to reduce overfitting.

6 Conclusions

This dataset is unbalanced. Despite using Boosting and Logistic with Regularisation technique, our models still overfit. We can improve the performance of the model by getting more data or use random sampling technique on the data to train the model.

References