Machine Learning for Book Rating Prediction (Individual, 1938 words)

1. Introduction

Book rating prediction is a task in which book predictions are obtained based on book features such as book description.

So far, there has been a surge in applications of machine learning in book rating, especially when deep learning has demonstrated state-of-the-art performance in language modelling tasks in the recent years, especially transformer-based models such as BERT [1], GPT-4 [2].

The dataset is web-scraped via Goodreads, and 2 separate training and test sets are provided. They consist of 10 features in total, in which the ***rating_label*** is the response variable.

This project aims at evaluating book rating classification through 2 approaches: traditional machine learning methods via LGBM-Classifier[3], and deep learning method via transfer learning with BERT (Bidirectional Encoder Representations from Transformers)

2. Methodology

2.1 Dataset partitioning

The original training set (23063 instances) is further split into 2 different subsets with a stratified shuffle split: train (80%) and validation (20%) subsets. Partitioning was conducted before any data wrangling methods to avoid information leak.

		# Instances
Training set	Train subset	18451
	Validation subset	4612
Test set		5766

Table 1 - Overall Summary of Data Partitioning

2.2 Data Pre-processing

2.2.1 Feature selection

As *PublishYear, PublishMonth*, and *PublishDay* are categorical features, a chi-square test is implemented at 5% significance level**,** which makes *PublishMonth* be omitted.

	Feature	chi_score	p_value
2.	PublishDay	108.571	2.65×10^{-24}
3.	PublishMonth	0.919	0.631

Table 2 – Chi-squared tests on categorical features

Similarly, *pagesNumber* and *PublishYear*, 2 continuous variables, are evaluated against ANOVA-test; however, there is insufficient evidence to reject them at 5% significance level.

	Feature	F-statistic	p_value
1.	PublishYear	108.571	1.81×10^{-50}
2.	pagesNumber	84.555	2.58×10^{-37}

Table 3 – ANOVA tests on continuous features

Language is excluded because of many missing instances. The training set only contains 5861/23063 non-null instances for language, while there are only 1526/5766 non-null instances in test-set. Moreover, there are certain languages in test-set, which are unobserved in trainset and would drastically reduce classifiers' performance if data imputation were applied.

Publisher is a highly predictive categorical variable for the response variable. For instance, Vintage, Wiley and Bantam are top recorded publishers for both training and test sets; however, there are no books rated as 5.0 in Vintage Publisher, and a relatively similar trend is also observed in the other 2 publishers.

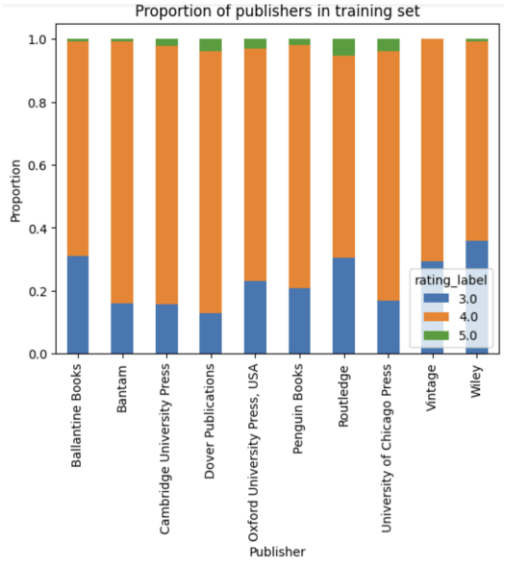


Figure 1 – Examples of proportions of publishers in training-set

2.2.2 Missing value and Scaling

pagesNumber is standardized by centering and scaling to boost model's evaluation performance.

Publisher, which is a categorical variable, is handled by one-hot encoding.

Text data features are pre-processed in section 2.2.3

2.2.3 Text Data Pre-processing

Most of predictive features are in text form, which requires an efficient text representation.

First, HTML tags are removed from *Author, Name, Description* features because the dataset is web-scraped to avoid any unnecessary information. Punctuation is then removed from these variables' instances, before being converted to individual token through basic text tokenization. Lowercasing, English Stop-words removal and Porter Stemmer are applied to reduce the dimensionality and ensure consistency, before re-joining into sentences as input for TF-IDF Vectorizer.

Also note that a number of descriptions is written in other languages, which may rise certain difficulties for stop-words removal.

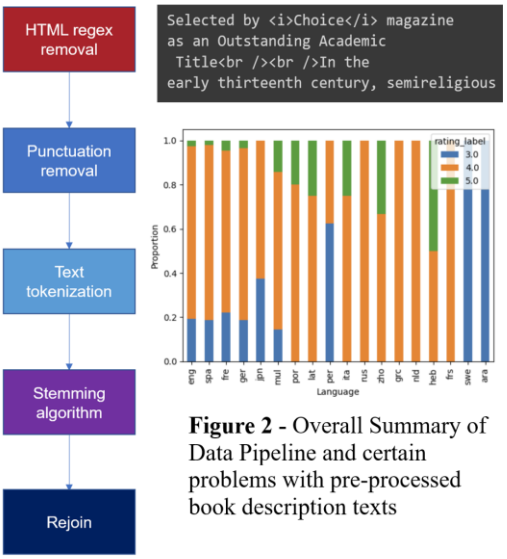


Figure 2 - Overall Summary of Data Pipeline and certain problems with pre-processed book description texts

Texts are then processed by TF-IDF Vectorizer instead of Count Vectorizer because not only does TF-IDF concentrate on the word frequency, but it can also assign higher weights for more important features in the vocabulary based on the word's frequency.

2.3 Model selection

2.3.1 Traditional machine learning approach (*LGBM-Classifier*)

LGBM-Classifier was selected for experiments because of its robust performance in competitions and research studies. Moreover, it also provides faster training time compared to other models such as XGBoost and Random-Forest.

An additional reason for LGBM-Classifier is because it is a gradient boosting framework, which is highly powerful for classifying hard-to-classify cases. This is useful as the training set is highly imbalanced, which could cause difficulties for other approaches (**Figure 3**)

Figure 3 – Book rating labels distribution

2.3.2 Motivations for a novel approach

Upon closer insight into book descriptions, they contain valuable information for book rating classification. While books with high rating generally contain positive words such as "most comprehensive" and "detailed", those with low rating are typically associated with negative features such as "fat", "ambitionless" (**Figure 4**). Therefore, this project formulated a hypothesis to determine whether book descriptions are adequate for building strong classifiers.

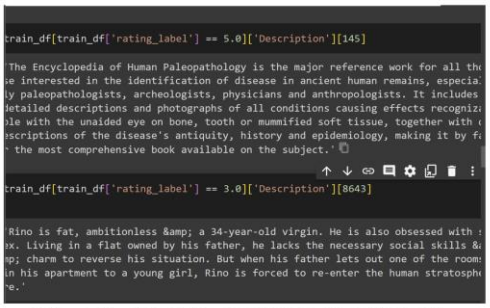


Figure 4 – Examples occurrence of positive and negative words in different book types

In addition, text representation for count-based methods solely depends on corpus distribution, while large language models generally "learn" their own word embeddings, grouping similar words closer together

2.3.3 Deep Learning Approach – BERT

BERT (Bidirectional Encoder Representations from Transformers) has gained significant popularity as it adopts Transformer architecture and achieves outstanding results.

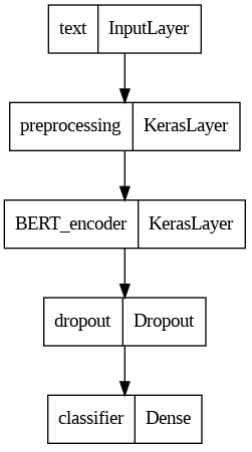
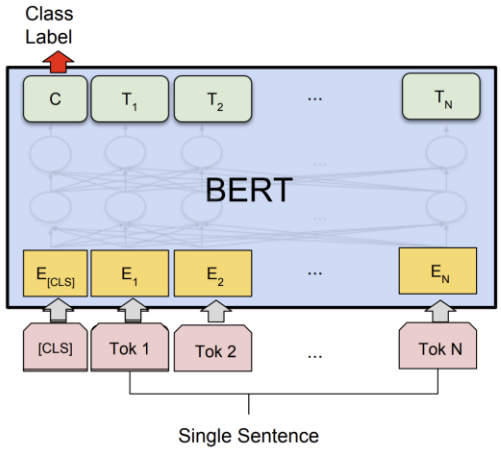


Figure 5 – BERT architectural summary

Texts, after going through Figure 2 pipeline, are then converted into input formats with a fixed input length (max_length=128) by Bert-Tokenizer before passing through embedding layers to learn embedding space, capturing semantic space. More importantly, a [CLS] token (classifier token – a learnable embedding) is added at the beginning of sequence, which captures pooled representation of the sequence



(b) Single Sentence Classification Tasks: SST-2, CoLA

Figure 6 – BERT architecture with CLS token

Having passed through the pre-processing layer, the text then goes through BERT encoder, which consists of multiple transformer-based encoders [4] for capturing the contextual relationship between words in sequence, hence understanding the meaning of the sentence.

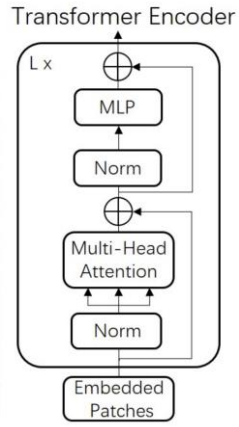


Figure 7 – BERT/Transformer Encoder architecture

The final state of hidden output from CLS token, representing the whole sequence, can then be fed into a customized MLP head/classifier for classification. A Dropout layer with 20% input units to drop – a **regularization method**, is also added to prevent overfitting and improve model generalization. A customized classifier layer with 3 units and 'SoftMax' activation is implemented to fit the model requirements to classify 3 levels of rating.

Model alternative:

Due to limitations in terms of computational resources, an alternative to BERT – Small-BERT [5], is implemented with competitive results and faster training time.

3. Model Training and Initial Evaluation

3.1 Evaluation metrics

2 architectures both use overall accuracy as the main evaluation metrics; however, other metrics such as precision, F1 score, and recall are also considered.

3.2 Traditional Approach

L2 normalization for TF-IDF Vectorizer will be implemented as it reduces weights yet not to 0, retaining certain useful information while L1 normalization tends to drive some weights to 0.

Cross-validation is implemented to avoid overfitting and improve generalization.

Model configuration

The specified range of hyper-parameters is listed in the table below (best parameters obtained via cross-validation are bold):

Feature	Feature range
n_estimators	[100,200, **500**]
Learning_rate	[0.01, **0.05**, 0.1]
Max_depth	[-1,3,6,**8**]
Num_leaves	[20,**31**,50]
Min_child_samples	[**20**,50,80]

Table 4 - Hyper-parameter tuning

3.3 Deep Learning Approach

A Dropout layer with 20% input units to drop – a **regularization method**, is also added to prevent overfitting and improve model generalization

Model configuration

The new model employs L = 4 hidden Transformer blocks/Encoders, hidden size H = 512, and A = 8 attention heads.

The whole model employs a loss function of 'sparse_categorical_crossentropy' along with an AdamW optimizer (learning rate is 0.00003 as suggested, learning rate is between 0.00001 and 0.00005).

$$CE = -\sum_{i}^{C} t_i log(f(s)_i)$$

Figure 8 – Cross entropy loss function

3.4 Results on validation

Figure 9 shows while BERT is more powerful regarding precision and F1-score, LGBM-Classifier is slightly stronger than Small-BERT regarding accuracy, which signals a slightly favour towards LGBM-Classifier for better overall accuracy performance. Further analysis is in Section 4.

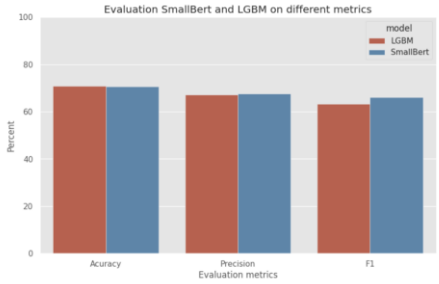


Figure 9 – Evaluation metrics on 2 models

Moreover, **Figure 10** also points out that Small-BERT, thanks to pre-trained learning, has a very high accuracy at 70% in the initial epoch; however, the model seems to become overfit as of the third epoch based on validation loss and accuracy.

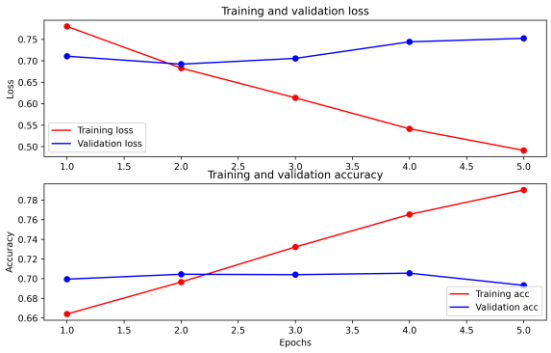


Figure 10 – History plot for Small-BERT

4. Results Discussion

4.1 Model interpretation and analysis

Figure 11 shows LGBM is more powerful in detecting books labelling with 4.0 (3107 compared to 2935), whereas Small-BERT is more useful for finding those labelling with 3.0 and 5.0 (288 vs 144, 25 vs 12 correct predictions).

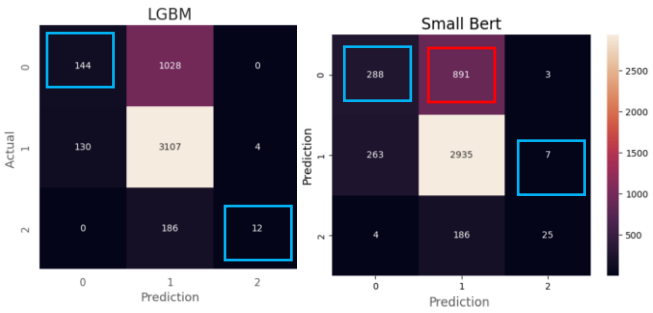


Figure 11 – Error analysis of 2 models

However, one common problem shared by both classifiers is despite outstanding performance for 4.0 labels regarding accuracy score, they are making too many predictions for 4.0, which makes them behave relatively similar to 0R-baseline (always outputting label with highest distribution). However, the problem is less severe for Small-BERT when it is making more correct predictions and less incorrect labelling for books of type 3.0 and 5.0 (**Figure 11**).

Therefore, a solution is demanded to raise the recall for the other 2 labels, while at the same time, retaining relatively high accuracy of both classifiers on 4.0 labels.

4.2 Addressing errors and Error Analysis

4.2.1 Class weights

Note that the dataset is highly imbalanced towards books labelling 4.0; therefore, a naïve approach is to adjust class weights inversely proportional to class frequencies to impact their predictions:

$$\text{class\_weight}(label) = \frac{n_{samples}}{n_{classes} * \text{count}(label)}$$

Figure 12 – Class weights formula

Figure 13 shows that by making adjustment to class weights, both models can predict/detect books with class 3.0 and 5.0 more accurately. The number of correct predictions increase significantly from 288 and 144 to 732 and 628 for Small-BERT and LGBM, respectively. Moreover, when both models are predicting 4.0 for books, the number of incorrect predictions is also significantly smaller, especially when the figures for LGBM decreases by half from 1028 to 523. (**Figure 11+13**)

As both classifiers make more predictions for 3.0 and 5.0 labels, they are also prone to

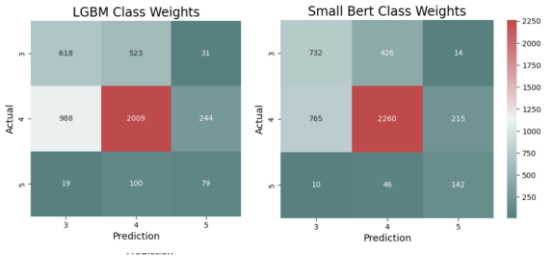


Figure 13 – Evaluation metrics on class weights version

making incorrect predictions about these 2 labels whereas the ground truth is 4.0. The number of correct predictions for 4.0 labels declines significantly, especially for LGBM when it decreases from 3107 to 2009. Despite certain improvements regarding 2 minor classes, the number of correct predictions is not sufficient to make up the loss in accurate predictions in the major class, decreasing overall accuracy for both models (Section 4.3).

4.2.2 SMOTE (LGBM) and Focal Loss (Small-BERT) evaluation

SMOTE (Synthetic Minority Oversampling Technique) [6] generates synthetic data based on interpolation and highly useful for imbalanced datasets. The resampled data is designed to be relatively balanced as shown in Figure 1

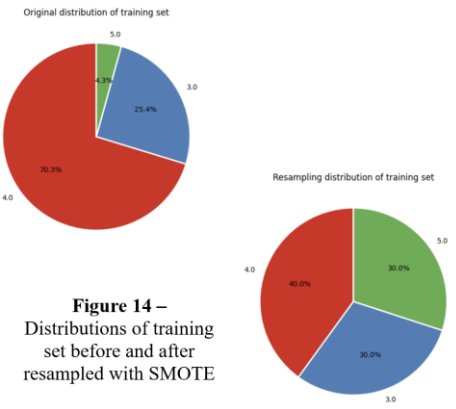


Figure 14 – Distributions of training set before and after resampled with SMOTE

Focal Loss [7] is a loss function to handle imbalanced dataset problems and is widely used in Object Detection. Focal Loss essentially down-weights instances with high accuracy and concentrates more on hard-to-classify instances. Recommended values of gamma and alpha (2.0 and 0.25, respectively) are implemented in this experiment.

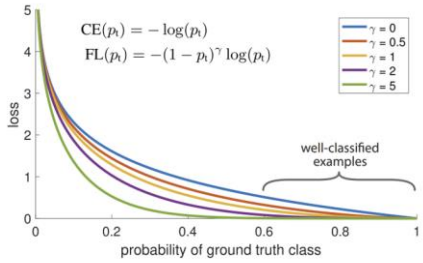


Figure 15 – Focal loss function

Similar to class weights modification (4.2.1), both classifiers can make more correct predictions for minor classes and more incorrect inference for major class. Small-BERT trained with Focal Loss has a significantly lower recall for 3.0 labels than the normal version (6.06% vs 24.43%), yet with higher precision (75.53% vs 51.89%). Although LGBM classifiers employing SMOTE shows a worse overall accuracy than the normal version, Small-BERT trained on focal loss shows the **most outstanding performance** in accuracy and weighted-precision for validation.

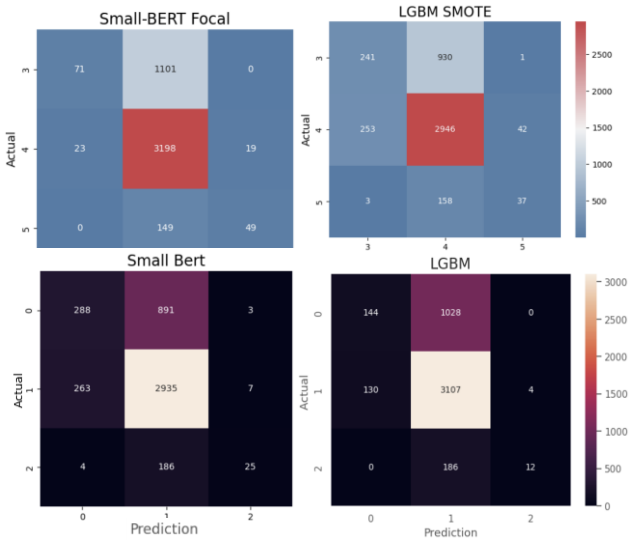


Figure 16 –Evaluation metrics on SMOTE LGBM and Focal Loss Small-BERT

Small-BERT customized with Focal Loss also adopts high performance on initial epochs and starts to overfit from third epoch when validation loss and accuracy do not improve.

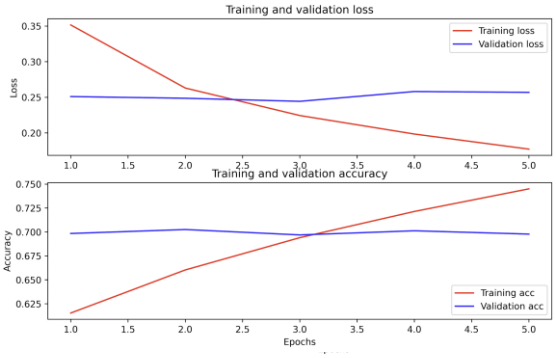


Figure 17 – History plot of focal loss

4.3 Overall results

The overall results on different evaluation metrics can be found here (best models highlighted as **bold**):

Models		Accuracy (%)	Precision (%) Weighted	F1-Score (%) Weighted
LGBM	Normal	70.77	67.12	63.30
	Class weights	58.69	64.28	60.55
	SMOTE	69.92	65.64	65.41
Small-BERT	Normal	70.58	67.61	66.10
	Class weights	67.98	72.13	**69.23**
	Focal loss	**71.97**	**72.83**	62.91

Table 5 – Summary of metrics on validation set

Evaluation on test set gives **71.522%** accuracy for LGBM-Classifier while Small-BERT records an accuracy of **70.62%.** This is consistent with our observations on the validation set.

5. Conclusion

Overall, LGBM-Classifier offers competitive results compared to deep learning models. Moreover, although modified versions of LGBM offer better recall, precision performance on minor classes, LGBM with tuned hyperparameters achieves highest accuracy – main evaluation metrics.

Despite worse overall accuracy in the normal version, Small-BERT achieves higher accuracy than LGBM models thanks to customization to the loss function (Focal Loss). Thus, this customized model achieves outstanding/best performance on overall accuracy and weighted-precision.

All 3 variants of Small-BERT models adopt

transfer learning, which makes them have a relatively high performance in initial epochs and require a smaller number of epochs than ordinary neural networks.

Book description contains highly valuable for book classification because if extracted appropriately, a strong model solely depending on this feature could achieve competitive results compared to another one utilizing all features.

Further improvements are restricted due to computational limitations. Possible improvement is training the dataset with Additionally, BERT multilingual models is potential because there is a large number of documents written in other languages besides English.

6. References

This project adopts certain codes, information from Chat-GPT; however, the core ideas are original.

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2018)

[2] OpenAI: GPT-4 Technical Report (2023)

[3] Guolin Ke , Qi Meng , Thomas Finley , Taifeng Wang , Wei Chen , Weidong Ma , Qiwei Ye , Tie-Yan Liu: LightGBM: A Highly Efficient Gradient Boosting Decision Tree

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin: Attention Is All You Need (2017)

[5] Iulia Turc, Ming-Wei Chang, Kenton Lee, Kristina Toutanova: Well-Read Students Learn Better: On the Importance of Pre-training Compact Models (2019)

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer: SMOTE: Synthetic Minority Over-sampling Technique (2011)

[7] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár: Focal Loss for Dense Object Detection (2017)