Automatic book genre classification

Vera Milovanović
University of Tübingen
vera.milovanovic@student.uni-tuebingen.de

Yeong Hwangbo
University of Tübingen
yeong.hwangbo@student.uni-tuebingen.de

Filip Radović
University of Tübingen
filip.radovic@student.uni-tuebingen.de

Abstract

We use the collection of books to see whether we could classify them into the predefined set of genres using only the book summaries. For this purpose, we use complement naive Bayes, logistic regression, random forest and XGBoost, and compare their performances. The project repository on GitHub can be found on https://github.com/radovic/automatic-book-genre-classification .

1 Introduction

In recent times, especially after the start of Covid-19 pandemic, the proportion of books sold online has drastically increased. Often, people's search criteria are genres, which could also help online bookstores to identify one's preferences. Therefore, a book recommendation system that uses genre classification would be useful for customers as well as for online bookstores to increase their profit.

In this project, we perform the analysis on CMU Book Summary Dataset [1] that contains plot summaries for 16,559 books extracted from Wikipedia, along with aligned metadata from Freebase, including publication dates, authors, book titles, and book genres. We classify the collection of books into the predefined set of genres using the book summaries. For this purpose, we apply TF-IDF vectorization of summaries and additional data pre-processing. We conduct label filtering to reduce the set of genres to the most commonly known. Then we implement four models for multi-label classification and compare their performances using different types of metrics.

2 Experiments

2.1 Data pre-processing

For the first dataset used in this project, out of 277 labels, i.e. genres from the original dataset, some are dropped as they represent a tag rather than genre, for example, *morality play* and *sea story*. Others are grouped around the broader genres, for example, *cyberpunk*, *military science fiction*, *time travel* are grouped as *science fiction*. In this way, we largely denoise our dataset, as there are lots of infrequent genres that mostly do not represent commonly known genres. Figure 1 shows the final 15 genres used for the classification and their frequency.

In order to possibly improve the performances of the classification we drop genres from the previously mentioned dataset with the lowest F1 scores obtained after classification. The genres with the F1 score below the 0.4 threshold are dropped, leaving 9 labels for the second dataset.

We transform summaries from the filtered datasets into the vectors using TF-IDF vectorization [2]. TF-IDF is an algorithm that assigns weights to terms depending on how relevant they are to the corpus on which it is performed. Inspired by Zipf's law [3], before the vectorization apply some additional steps. First, we tokenize the corpus and remove all tokens that are either stop words or contain non-alphabetic characters as we want to have only informative features. The remaining words are lemmatized, i.e. reduced to their dictionary form, in order to avoid having different weights for different inflected forms of the same word.

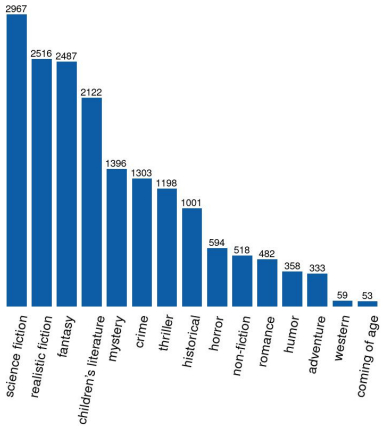


Figure 1: Number of books per genre

We also exclude words with small frequencies in the datasets because we want to reduce the vocabulary and avoid potential overfitting caused by the connection of the terms which are scarcely represented in the training set to specific genres. It is done by iterative search of the cut-off value on the interval $[0, 0.001]$. It represents the minimal portion of texts that should contain each word of the corpus. By comparing the accuracy of the complement Bayes model we come to the optimal cut-off value of 0.125%.

2.2 Classification

The pipeline for the classification process is the same for both datasets. In order to prevent selection bias and subsequently underestimating the generalization error, hyperparameter tuning is conducted independently from the fitting and evaluation procedures in each trial [4]. For this purpose, k-fold nested cross-validation is used, where the inner 2-fold loop serves for hyperparameter tuning and the outer 5-fold loop for generalization error estimation. Additionally, as the first step in the model pipeline, TF-IDF vectorization is applied to test and train folds of the outer loop independently, so data leakage into the test sets is prevented. Random search cross-validation is used for the hyperparameter tuning for all models except for complement Naive Bayes for which hyperparameter tuning is not conducted. For logistic regression inverse of regularization strength is tuned, and for random forest and XGBoost it is the maximum depth of the trees. We tune only one hyperparameter with three search iterations to reduce the number of fittings, which are especially time-consuming with XGBoost. During the process of hyperparameter tuning and generalization error estimation, we keep track of the three evaluation metrics: F1 score(micro), accuracy, and Hamming loss. The best model found during the hyperparameter search is fitted on the train sets using F1 micro score.

We compare the generalization errors of the Complement Naive Bayes [5], logistic regression [6], random forest [7] and XGBoost [8]. We use the Complement Naive Bayes classifier instead of the other versions because it was shown that the Complement Naive Bayes preceded by TF-IDF with normalization performed well [5]. In the rest of this report, we will call this model transformed weight-normalized Complement Naive Bayes, or TWCNB for short.

Since we have a multi-label problem, and TWCNB, logistic regression and XGBoost do not support it, we transform the multi-label classification to the two-class problem using OneVsRestClassifier, i.e. fitting one classifier per class. Random forest inherently supports multi-label classification [9].

In order to compare the models' performances, we choose three metrics: Accuracy, F1 score(micro), and Hamming loss. Micro F1 score and Hamming loss are chosen because they weigh all samples evenly and all classes are equally important to us. Additionally, accuracy is used because it counts labels as incorrectly predicted if they do not entirely match the true set of labels. Contrary to accuracy, Hamming loss acknowledges partially correct classification which is important for multi-label classification.

3 Results

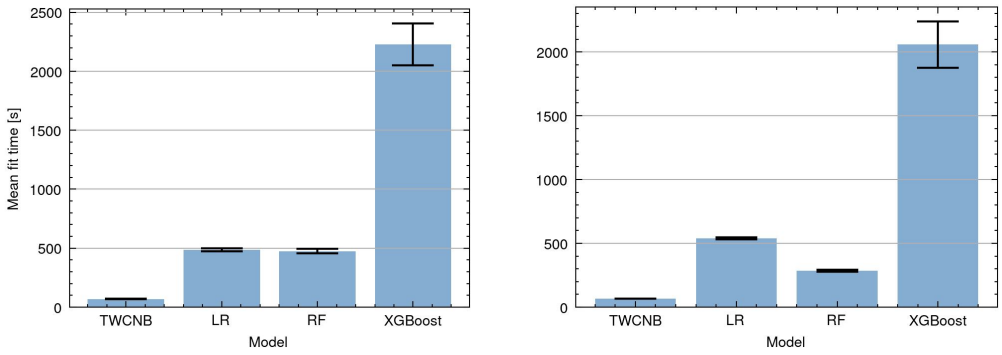
Table 1: Mean scores and their standard errors of different models for each dataset

Dataset	Model name	F1 score (micro)	Accuracy	Hamming loss
15 genres	TWCNB	0.5572 ± 0.0030	0.3158 ± 0.0033	0.0780 ± 0.0006
	LR	*0.5629* ± 0.0030	*0.3305* ± 0.0018	***0.0709*** ± 0.0006
	RF	0.2361 ± 0.0121	0.1192 ± 0.0048	0.1044 ± 0.0017
	XGBoost	0.5138 ± 0.0027	0.2783 ± 0.0020	0.0774 ± 0.0005
9 genres	TWCNB	***0.6049*** ± 0.0023	0.3671 ± 0.0058	0.1123 ± 0.0010
	LR	0.6025 ± 0.0021	***0.3760*** ± 0.0058	*0.1035* ± 0.0010
	RF	0.2577 ± 0.0095	0.1291 ± 0.0056	0.1642 ± 0.0029
	XGBoost	0.5602 ± 0.0030	0.3266 ± 0.0051	0.1144 ± 0.0011

The values written in cursive are the best scores obtained from the dataset, and the values written in bold are the best obtained values per metric on both datasets. The higher F1 score and accuracy and the lower Hamming loss the better classification is.

Overall, by comparing the results in Table 1, we note that logistic regression has the best performance on both datasets. We also note that TWCNB and logistic regression have similar performances, so TWCNB might be the preferred model because of its simplicity and training time (Figure 2). It is also worth mentioning that none of the scores has the standard error greater than 0.01.

Although we anticipated better results on the dataset with a smaller number of genres, we see only slight improvement in F1 score and accuracy of the models. In the case of Hamming loss, we also see the increase in the value. However, this is not as bad as it seems, since Hamming loss is greatly influenced by the imbalance of the classes. For example, if we take one of the minority classes, coming-of-age (0.5% of books), the Hamming loss for the classifier that does not assign anything to coming-of-age would be less than 0.005 which is almost perfect result. Therefore, it would drastically pull the overall Hamming loss down and make an illusion of good classification.



(a) 9 genres dataset

(b) 15 genres dataset

Figure 2: Mean fit times and their standard deviations

In addition, since the standard errors of random forest in Table 1 are notably greater than those of the other models, we conjecture that this model is more sensitive to the hyperparameter change. Not surprisingly, since XGBoost is the most complex model among other models we test here, it takes the longest time to fit the data (Figure 2). Varying the value of XGBoost's hyperparameter maximum depth of trees does not influence the scores (Table 1). However, as we can see by the standard deviation of the mean training time in Figure 2, it significantly influences the training time. Therefore, it is better to keep the max depth at lower values, for example, 2.

4 Fairness

Conducting this kind of classification raises some questions about the fairness of the model. According to the [10], Wikipedia is characterized by highly uneven geographies of participation. This result is in compliance with our finding that non-western authors are in minority in the dataset used in this project, as only book articles in English were collected. This possibly makes the classification model biased toward recognizing books in a "western" writing style.

Moreover, the dataset contains disproportionally more male authors, which introduces a bias toward assigning certain genres to the book according to the gender of the writer. A study [11] showed that there was a significant difference in male versus female writing style, which confirms our claim.

Both of the previously mentioned fairness concerns should be taken into account if such kinds of models were used in a book recommendation system.

5 Limitations and future work

Our project has two main limitations. The first is the size and imbalance of the CMU dataset. Even before grouping and filtering the labels, more than 22% of the dataset is unlabeled and consequently useless for our project. The other problem is that the dataset is imbalanced. For example, the prevalence of the most frequent genre is 55 times greater than the most infrequent one. The second limitation is of technical nature, hyperparameter search was extremely time-consuming so we had to restrict the hyperparameter spaces accordingly. Hence, it might be that the random search used for hyperparameter tuning did not find the optimal hyperparameters for our models.

The dataset could be better exploited if other features were used, for example, the book author and the publishing date. However, this would immensely increase the dimensionality of the problem, as each author should be encoded. Also, summaries could be transformed differently, which would result in better performance. We propose Doc2Vec[12] or trying different n-gram models.

6 Conclusion

We conclude that the proposed models do not perform well in assigning genres to books. We assume that is because summaries do not contain features that are discriminative enough. However, it should be pointed out that the classifier deals with a big number of possible genres. On the other hand, the results are consistent, as the estimate of the standard error of the means of all metrics is negligible.

References

[1] David Bamman. *CMU Book Summary Dataset*. URL: https://www.cs.cmu.edu/~dbamman/booksummaries.html (visited on 01/27/2023).

[2] Juan Ramos et al. "Using tf-idf to determine word relevance in document queries". In: *Proceedings of the first instructional conference on machine learning*. Vol. 242. 1. Citeseer. 2003, pp. 29–48.

[3] Mark EJ Newman. "Power laws, Pareto distributions and Zipf's law". In: *Contemporary physics* 46.5 (2005), pp. 323–351.

[4] Gavin C Cawley and Nicola LC Talbot. "On over-fitting in model selection and subsequent selection bias in performance evaluation". In: *The Journal of Machine Learning Research* 11 (2010), pp. 2079–2107.

[5] Jason D Rennie et al. "Tackling the poor assumptions of naive bayes text classifiers". In: *Proceedings of the 20th international conference on machine learning (ICML-03)*. 2003, pp. 616–623.

[6] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[7] Tin Kam Ho. "Random decision forests". In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE. 1995, pp. 278–282.

[8] Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[9] Leo Breiman. "Random forests". In: *Machine learning* 45 (2001), pp. 5–32.

[10] Mark Graham, Ralph K Straumann, and Bernie Hogan. "Digital divisions of labor and informational magnetism: Mapping participation in Wikipedia". In: *Annals of the Association of American Geographers* 105.6 (2015), pp. 1158–1178.

[11] Shlomo Argamon et al. "Gender, genre, and writing style in formal written texts". In: *Text & talk* 23.3 (2003), pp. 321–346.

[12] Quoc Le and Tomas Mikolov. "Distributed representations of sentences and documents". In: *International conference on machine learning*. PMLR. 2014, pp. 1188–1196.