Classifying a book's genre using Classification Machine Learning Methods

November 8, 2023

1 Introduction

With the invention of mass printing and the advent of technology, books have found their ways to become one of the most important and effective forms of information transmission, allowing humans to record and exchange ideas and stories on a larger scale than ever before. However, categorising books remains a difficult and daunting task, since one must at least have a sufficient overview of what the book is about. Therefore, this project aims to efficiently classify a book based on its general description in order to reduce the workload of librarians and book enthusiasts. The project will be organised as follows. First, the problem is formulated as a machine learning problem, and then two different methods used for classification will be discussed. Finally, the results will be presented, along with final conclusions and some further possible improvements.

2 Problem Formulation

In this project, we are trying to predict whether a book is fictional or non-fictional based solely on its description - which makes this a binary supervised classification problem. The dataset was retrieved from Kaggle [2], collected from Goodreads from the list *Books That Everyone Should Read At Least Once* [3]. The list was compiled according to a community voting system, with the number of voters mounting up to over 100,000. The genres for each book is determined by the website's crowd-sourcing system from users' shelves and does not consider the author's input [4]. Even though this might not be the most reliable categorising system, it is definitely a good place to start. Each data point corresponds to a book, consisting of seven properties featured on the Goodreads website: the title of the book, its author, description, a list of its genres, average rating, number of ratings, and an URL leading to its respective Goodreads page. For the purpose of this project, only the book's title, author, and list of genres are relevant, and the description will be transformed to a TF-IDF matrix, which will serve as the features for the model.

3 Method

3.1 Data Preprocessing and Feature Selection

As mentioned above, we will only consider the description of the book to predict its genres, as a book's description is one of the best fitted overview of the book. The other features, which are average rating, number of ratings and URL, do not have much of an impact on the book's genres. Hence, the former label contains valuable information on a book's content, while the latter could be removed from consideration.

There are 10,000 data points (i.e., books), and 617 distinct genres. After removing duplicates, books that do not have genres and books whose description is not in English, we are left with 7745 data points. Besides the selected features, we will expand the dataset with another label corresponding to fiction/non-fiction. One particular aspect to keep in consideration is that if a book has been labelled as "Fiction", it can not be labelled as "Non-fiction", and vice versa, since they are mutually exclusive. A book can be labelled as true ($= 1$) or false ($= 0$), meaning it does classify as fiction or not.

We will also need to perform further data preprocessing with the books' descriptions. This includes tokenization (splitting the initial text into individual words), lowercasing (convert all words to lowercase), stopword removal (remove words such as "a", "the", "is" that may not add much meaning to the whole text), and lemmatization (reduce words to their root forms for text normalisation).

After the necessary data preprocessing, we will transform the description into a TF-IDF (stands for Term Frequency-Inverse Document Frequency) features matrix using *TfidfVectorizer* from sklearn. TF-IDF is made up of two components, the first one being Term Frequency - representing the relative frequency of a word in a document, and the second one being Inverse Document Frequency - referring to the amount of information the word provides, or its relevance. Together, TF-IDF determines the importance of a particular word in the document [5]. I decided to set the minimum Document Frequency (DF) as 0.015, and the maximum DF as 0.8, so as to end up with a reasonable 1036 features (i.e., words) to work with.

Data splitting

The data set is split into three sets, training, validation and test of sizes 60%, 20%, and 20% respectively. The data set is sufficiently large, so splitting it this way would guarantee a proportional training set (4956 data points) to validation and testing. First, the data set is split into training and test with a ratio of 1:8:2, and then the training set is further split into training and validation with a ratio of 1:7.5:2.5.

3.2 Method selection

Random Forest

Since this problem is a Natural Language Processing problem, we will be using the random forest classifier method in combination with TF-IDF. Random Forest is an ensemble learning method that combines various decision tree classifiers, using majority vote in the case of classification to improve the predictive accuracy and reduce overfitting. Each decision tree spans a hypothesis map that takes in features and map them to the binary labels, so a forest consists of multiple trees could approximately represent a non-linear map [1]. Additionally, random forest is also suitable for handling high-dimensional data, which in this case is TF-IDF vectors, without the need for dimensionality reduction techniques that may result in the loss of important information. The weight class is also set to balanced, as the number of fictional books doubles nonfictional ones.

Gini impurity is chosen as the loss function in accordance with this method, since it is readily available in the library chosen. While it might be less sensitive to outliers compared to Information gain, Gini impurity is more computationally efficient for classification tasks. Both the model and the loss function are included in the library *sklearn.ensemble.RandomForestClassifier*.

Logistic Regression

Since logistic regression is a classic binary classification method, it will be implemented as the second method. Its simple and easily interpretable nature, along with its high efficiency allows me to easily use it. The method could be understood as a linear map, with the input features make up the hypothesis space. They will be mapped to the probability of the binary outcomes, modelling the relationship between the features and the events. The weight class is, again, set to balanced just like random forest for the above mentioned reason.

The logistic loss is chosen as the loss function for this method, as it is readily available in the library. As a natural choice for logistic regression, it measures the distance between the predicted and true labels, and aligns with the fundamental assumption of the method that its output follows a logistic function, generating values between 0 and 1.

4 Results

Accuracy scores are used to determine the preferred model since this is a classification problem. Random forest returns an average training accuracy of 93.67% and average validation accuracy of 80.78%, while logistic regression returns a 92.3% training accuracy and 89.44% validation accuracy. Logistic regression is chosen as the final method since it has a better validation accuracy. The model is tested with the remaining test set, and has an accuracy score of 89.92%. Providing that this is a NLP problem, I believe this is a rather positive result.

5 Conclusion

The project applies two classification methods that are different in nature, random forest and logistic regression to classify books as fiction or non-fiction based on their general descriptions. While the former utilizes multiple decision trees to capture the complex relationships in the textual data sets, the latter tries a simple linear map to show the relationship between the input features and the binary outcome. Even though both methods are not widely used for textual data, they both show great accuracy scores (above 85%), making room for more chances to implement them for textual data in the future. Improvements could be made in various places. The description could be further preprocessed by stemming, or the number of trees in the forest could also be tuned in to a more suitable fit. Moreover, it would also be interesting to compare different word embedding techniques such as Word2Vec to see whether it could perform better than TF-IDF. Overall, for a first machine learning project, it has provided me with exciting experience and lessons to further refine my knowledge regarding natural language processing.

6 References

[1] A. Jung,“Machine Learning: The Basics,” Springer, Singapore, 2022

[2] https://www.kaggle.com/datasets/ishikajohari/best-books-10k-multi-genre-data/discussion/409535

[3] https://www.goodreads.com/list/show/264.Books_That_Everyone_Should_Read_At_Least_Once

[4] https://help.goodreads.com/s/article/How-can-I-set-my-book-s-genres#:~:text=Goodreads%20determines%20a%20book’s%20genre,the%20book%20in%20our%20algorithm

[5] *Tf-idf*. Wikipedia. https://en.wikipedia.org/wiki/Tf%E2%80%93idf

[6] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html