Best Seller Classification from Book Descriptions Using Logistic Regression: Binary Classification Project Report

# Introduction

This report documents the implementation and training of a machine learning (ML) model for best seller status classification based on book descriptions. A dataset by [1] sourced from Kaggle titled “Goodreads Book Datasets With User Rating 2M” was used for this project with. The best model pipeline achieved an F1-Score of 74.22% on the testing set showing the effectiveness of the model in differentiating between best sellers from and non-best seller books based on their descriptions on noisy data. The primary goal of this project was to leverage Logistic Regression (LR) for binary classification tasks. LR has been shown to perform well in various text classification scenarios as highlighted in a study by [2], [3].

Recent studies have shown the efficacy of LR in text classification tasks. For example, [3] carried out a comprehensive analysis for text classification between KNN, LR and Random Forest discussing their technical contributions and text classification performance. Additionally, [4] emphasized the advantages of deep learning (DL) and ML models such as Logistic Regression in handling large-scale text classification problems. The current project extend based on the recent literature by creating a LR model using Python for best seller classification based on the book description given. The project uses libraries such as Pandas and Spark for data examination. Term Frequency-Inverse Document Frequency (TF-IDF) vectorization was utilized and for evaluation metrics such as accuracy, confusion matrix and classification report was used to report the best performing model pipeline. The project also involved hyperparameter tuning to optimize model performance.

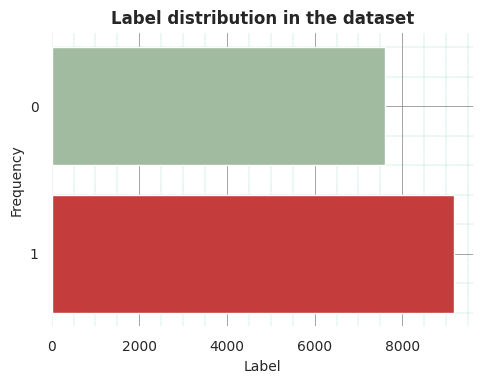
This report is structured into different sections; Section II outlines the methods used, Section III lists and outline results and then finally, Section IV will conclude the report.

# Methods

This section of the report will outlines the methods that was used for data analysis, data cleaning, feature selection, classifier training, classifier evaluation and classifier inference.

## Dataset, Data Cleaning and Preprocessing

In this project the dataset titled “ Goodreads Book Datasets With User Rating 2M” by [1] was used to train and evaluate the LR classifier. The dataset was obtained from Kaggle at this URL “<https://www.kaggle.com/datasets/bahramjannesarr/goodreads-book-datasets-10m>” with a lot of csv files. The downloaded file had a file name book4000k-5000k.csv with initially 19 columns and 280256 rows [1]. Out of 19 columns in the dataset two (2) columns were selected which are “Description” and “Rating”. The dataset included book descriptions as text along with their corresponding ratings, ranging from 0 to 5 and contained additional columns with some missing values. All column values that had null values were dropped leaving the dataset having 184554 paired description of books with their respective ratings. Descriptions with less than 230 words were removed from the dataset leaving the dataset size as 16791. A threshold of 2.5 was applied to the ratings, such that all book descriptions with a rating greater than or equal to 2.5 were labelled as “best seller” (1) and the rest were categorised as “non-best seller” (0) in a new dataframe. Fig. 1 shows label distribution after applying doing this preprocessing step.



1. Unbalanced label distribution.

Fig. 1 shows the frequency of samples with (0) representing the book with the description of non-best sellers and (1) description of books for the best sellers. There was huge class misbalance between the 2 classes. To solve this class misbalance, a technique known as down sampling was used to balance classes based on the class with less rows of data as suggested in recent research by [5], [6]. The balancing technique is done so that the model will not be biased towards the class with the majority samples [5]. Fig. 2 illustrated the frequency of samples after balancing the dataset.

A green and red pie chart

AI-generated content may be incorrect.

1. Frequence distribution of samples after balancing book description based on class frequency.

Fig. 2 shows that there was equal distribution of class labels after down sampling the dataset with 7613 samples for each class. Two (2) subsets were created from the dataset and Fig. 3 shows the per class distribution of labels after this data split.

A diagram of a training and testing

AI-generated content may be incorrect.

1. Per set sample distribution.

The dataset was split into 2 subset for this task with 20% of the dataset reserved for classifier testing set and 80% being used for fitting the classifier set as illustrated in Fig. 3. Features (book descriptions) was further cleaned cleaning to ensure data quality and consistency. The book descriptions were normalized by converting them to lowercase, removing punctuation and stripping special characters, numbers and URLs using regular which normally add noise to textual data as asserted by [7]. Other preprocessing techniques to improve textual feature quality was added directly to vectorizers. These techniques include removal of stopwords and generating n-grams. According to [8], stopwords are words that carries less semantic meaning on their own for example words like “the”, “is”, “are”, etc.

## Exploratory Data Analysis (EDA)

In this project EDA was performed using both Pandas and Spark to understand the data. The labelled dataset with columns “Description”, “Rating” and “BestSeller” were analyzed. Table I shows the descriptive summary of the ratings in the dataset.

1. Descriptive statistics about rattings in the dataset

|  |  |
| --- | --- |
| Summary | Value |
| Count | 17278 |
| Mean | 3.4681277672359285 |
| Starndard Deviation | 52.13563911450714 |

The mean and standard deviation rating of the book descriptions were 3.47 and 52.13 respectively as illustrated in Table I.

1. Descriptive statistics about the length of each book description

|  |  |
| --- | --- |
| Summary | Value |
| Count | 18411 |
| Mean | 1358.7222855901364 |
| Starndard Deviation | 504.58481145117065 |
| Min | 856.8054649266676 |
| Max | 31515 |

Table II shows that the average size of the textual description was 630.1 having a 3 minimum number of words and 31515 maximum number of words. Table III illustrates the most common words in the boom description.

1. Most Common Words Overall (5 works)

|  |  |
| --- | --- |
| ****Word**** | ****Count**** |
| the | 235426 |
| of | 164685 |
| and | 155130 |
| in | 84786 |
| to | 81983 |

Table III displays the top five (5) common words in all descriptions of books. The most used words in the book descriptions were stopwords. These words carries little meaning on their own [8].

## Logistic Regression Training

A LR model was initialized evaluated with 3 different vectorizers namely, CountVectorizer, TfidfVectorizer and HashingVectorizer to compare, which model will perform best. The best pipeline for classification was chosen for the and then GridSearchCV was used to obtain best parameters for the chosen vectorizer and LR classifier.

## Classifier Inference

The best LR classifier that was identified by *GridSearchCV* was used to create a Bot that binary classifier for the best seller based on the textual description of a book. The whole project was implemented using Python programming. Table VII summarizes the technologies that was used for this task.

1. Summary of technologies that was used

|  |  |  |
| --- | --- | --- |
| ****Technology**** | ****Purpose**** | ****Reason for Use**** |
| Python 3.10 | Core language for implementing data processing and model training tasks. | Provides a broad ecosystem for machine learning and NLP. |
| Matplotlib & Seaborn | Created EDA plots and confusion matrix. | Enables clear visual interpretation of data trends and model performance. |
| Pandas | Reading and manipulating CSV files. | Offers efficient and user-friendly dataframe operations. |
|  | Performed distributed data analysis and preprocessing. | Handles large-scale datasets efficiently with parallel processing capabilities. |

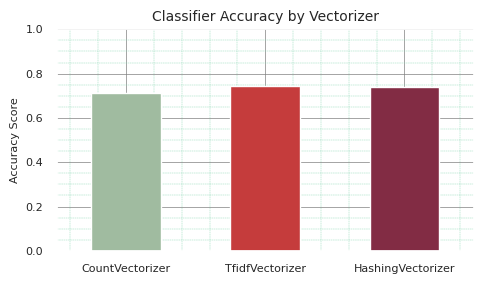
# Results

This section outline the results that was obtained for the best seller classification using LR model. Table V shows the classification performance for each vectorizer paired with LR classifier.

1. Logistic Regression and vectorizers perfomance

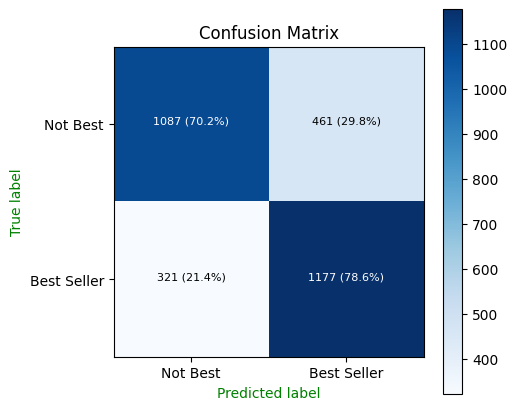
|  |  |
| --- | --- |
| vectorizer | score |
| CountVectorizer | 0.712738 |
| TfidfVectorizer | 0.743598 |
| HashingVectorizer | 0.740643 |

Fig. 4 shows the visual representation of these vectorizers performance inform of a bar chart.



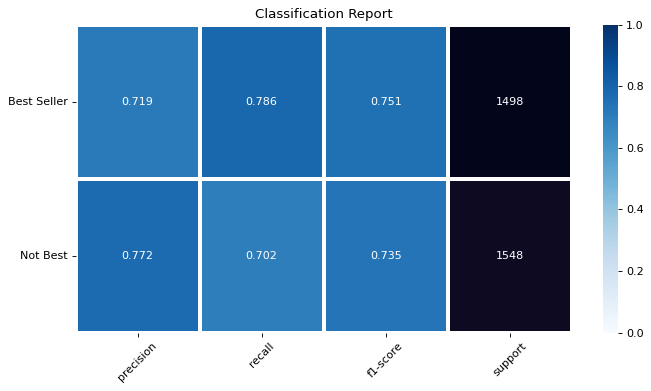
1. Performance of vectorizers for book description classification with LR.

Fig. 4 shows that *TfidfVectorizer* achieved a better score of 74.36% compared to other vectorizers used in combination with LR classifier as shown in Table V. The best pipeline that was selected incorporated *TfidfVectorizer* with 1-3 grams with LR model in predicting best sellers based on book description using LR classifier model. This combination attained a validation F1-score of 0.7422 evaluated on the testing set. Fig. 5 displays the confusion matrix for the best classifier evaluated on the testing dataset.



1. Confusion matrix showing model performance for Best Seller and Not Best Seller classifications

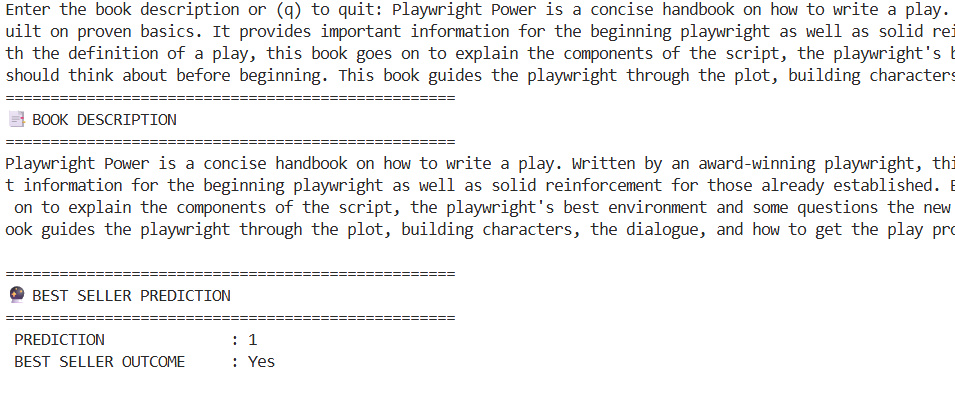
Fig. 5 illustrates that the LR classifier correctly classified 70.2% of “Not Best Seller” books and 78.6% of “Best Seller” books, indicating strong overall predictive performance with slightly better accuracy on best sellers. Fig. 6 displays the classification report for the LR model assessed based testing dataset.



1. LR classification report on the testing dataset.

Fig. 6 shows that the model achieved an F1-score of 0.751 for Best Seller and 0.735 for Not Best Seller showing a balanced performance across both classes with slightly better recall for best sellers.

Fig. 7 sample screenshot illustrating LR classifier output during model inference.



1. Sample screenshot illustrating Logistic Regression model inference.

# Conclusion

This project successfully demonstrated the application of ML for description book classification using LR. The classifier effectively distinguished between best seller and non-best seller books based on their descriptions with an F1-score of 74.22% on the testing dataset. The use of TF-IDF vectorization enhanced feature representation enabling the classifier to capture key linguistic patterns within textual data. This study aligns with previous findings by [2] and [3] who highlighted LR’s strong performance in text classification tasks compared to other algorithms. Moreover, [4] emphasized the effectiveness of ML models in processing large scale textual datasets, which was evident in this project’s use of Python, Pandas, and Spark for scalable data handling.

Data imbalance was mitigated through down sampling techniques, following approaches suggested by [5] and [6], ensuring a balanced dataset for fair model training. Overall, the implemented pipeline showed robust, interpretable and computationally efficient confirming that LR remains a reliable baseline for binary text classification problems such as best seller prediction.

# References

1. B. Jannesar and S. Ghaderi, “Goodreads Book Datasets With User Rating 2M.” Accessed: Oct. 09, 2025. [Online]. Available: https://www.kaggle.com/datasets/bahramjannesarr/goodreads-book-datasets-10m
2. Q. Li, S. Zhao, S. Zhao, and J. Wen, “Logistic Regression Matching Pursuit algorithm for text classification,” *Knowl Based Syst*, vol. 277, p. 110761, Oct. 2023, doi: 10.1016/j.knosys.2023.110761.
3. K. Shah, H. Patel, D. Sanghvi, and M. Shah, “A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification,” *Augmented Human Research*, vol. 5, no. 1, p. 12, Dec. 2020, doi: 10.1007/s41133-020-00032-0.
4. A. Varghese, G. Agyeman-Badu, and M. Cawley, “Deep learning in automated text classification: a case study using toxicological abstracts,” *Environ Syst Decis*, vol. 40, no. 4, pp. 465–479, Dec. 2020, doi: 10.1007/s10669-020-09763-2.
5. S. Ngwenya and T. C. Garidzira, “Enhanced Fake News Detection with Domain-Specific Word Embeddings: A TorchText-Based Method for News Semantics Representation.,” *The Indonesian Journal of Computer Science*, vol. 14, no. 4, Jul. 2025, doi: 10.33022/ijcs.v14i4.4831.
6. C. Padurariu and M. E. Breaban, “Dealing with Data Imbalance in Text Classification,” *Procedia Comput Sci*, vol. 159, pp. 736–745, 2019, doi: 10.1016/j.procs.2019.09.229.
7. W. T. Vambe and T. C. Garidzira, “A Comparative Analysis of Convolutional Neural Networks, Bi-Directional Gated Recurrent Units, and Bi-Directional Long Short-Term Memory for Sequential Data Processing Using Human Emotions Dataset,” *Proceedings of 2024 4th International Multidisciplinary Information Technology and Engineering Conference, IMITEC 2024*, pp. 531–538, 2024, doi: 10.1109/IMITEC60221.2024.10851188.
8. S. Sarica and J. Luo, “Correction: Stopwords in technical language processing,” *PLoS One*, vol. 19, no. 12, p. e0315195, Dec. 2024, doi: 10.1371/journal.pone.0315195.