2025

Best Seller Identification with Logistic Regression

PDAN8412 PART 2

MAXIMILIAN WALSH – ST10203070

27 OCTOBER 2025

Table of Contents

Introduction

This analysis aimed to develop a logistic regression classification model from scratch using a simple neural network capable of predicting whether a book will become a bestseller, using real-world bibliometric and engagement data. The chosen dataset, *Best Books Ever*, was sourced from Kaggle and represents a large-scale collection of over 40,000 books, each described through multiple quantitative and categorical attributes such as user ratings, rating counts, liked percentages, genre information, and publisher details (Mostafapoor, 2025).

This task was approached from the perspective of a data analyst in a large book publishing company, tasked with identifying the key factors that differentiate top-performing books from average ones. Spark was leveraged for scalable data handling, while scikit-learn and TensorFlow were used for feature engineering and modelling.

A structured methodology was followed in the analysis, encompassing data ingestion, cleaning, feature engineering, model training, and performance evaluation. The logistic regression approach was chosen to meet the project requirements; however, it is considered suitable as this is a binary classification task with the output being either bestseller or non-bestseller. Two logistic regression models were trained: a neural logistic regression (from scratch) implemented as a one-layer network with a sigmoid activation, and a Scikit-learn logistic regression used as a benchmark. Logistic regression offers a balance of interpretability, computational efficiency, and robustness (OpenAI, 2025). All of which contribute to the understanding of which features most strongly influence bestseller outcomes.

Ultimately, the results of this project demonstrate that a neural logistic regression model can effectively replicate traditional logistic behaviour and achieve high accuracy, close to 88%, while the scikit-learn model further optimises the approach to near perfect performance at an accuracy of roughly 98%.

Executive Summary

This project develops two logistic regression models, a neural network logistic regression (from scratch) and a scikit-learn baseline, to predict whether a book will become a bestseller using the *Best Books Ever* dataset from Kaggle, which contains over 40,000 records with bibliometric and engagement features. Two handle numeric scaling, categorical encoding, and TF-IDF text vectorisation, a unified preprocessing pipeline was used to convert missed data into over 5,600 model ready features. The dataset was moderately imbalanced, with 80% non-bestsellers and 20% bestsellers; however, this was handled through class weighting. The

neural logistic regression achieved a final model accuracy of 88% and a ROC-AUC of 0.95, which validates the from scratch approach, whilst the scikit-learn logistic regression reached 98% accuracy and an ROC-AUC of 0.999, confirming benchmark level performance. Bestseller key predictors included reader engagement metrics like rating counts, liked percentage, and votes, along with publisher prominence. The combination of models provide a transparent and interpretable framework for predicting book success. Both models offer strengths, the neural model satisfies the technical requirements for manual logistic regression and the scikit learn model shows the scalability and precision expected in applied analytics.

Analysis Plan

Developing a structured plan ensures that data cleaning, feature preparation, model training, and evaluation are all handled systematically. The following is the analysis plan that outlines how the Best Books Ever dataset will be transformed into a predictive model capable of classifying books as bestsellers or non-bestsellers using logistic regression from scratch:

Exploratory Data Analysis (EDA) Plan

The goal of the EDA is to understand the dataset's structure and key statistical relationships between reader engagement and other book features.

Planned steps:

- Check for missing or null entries in fields like 'rating', 'numRatings', 'likedPercent', and 'bbeScore'.
- Examine the distribution of numeric fields.
- Verify class balance between 'bestseller' as 1 and 0.
- Visualise distributions through histograms and boxplots, and correlations among numeric features.
- Generate summary statistics for categorical variables like 'language', 'publisher', and 'genre' to understand diversity in the dataset.

Feature Preparation Plan

Given that the dataset consists of both numeric and categorical variables, preprocessing will standardise inputs and encode qualitative information for use in logistic regression (Muller & Guido, 2016).

Steps:

- Cast columns 'rating', 'numRatings', 'likedPercent', 'bbeScore', and 'price' to numeric types and scale using StandardScaler.
- Skewed features like 'numRatings' may have log transformations applied to normalise their distribution.
- Encode 'language', 'publisher', and 'genres' using OneHotEncoder (handle_unknown="ignore").

- Numeric and categorical transformations will be combined using a ColumnTransformer within the scikit-learn pipeline to ensure consistent preprocessing during training and inference (OpenAI, 2025).

Model Training Plan

The task is a binary classification predicting whether a book is a bestseller.

Steps:

- Split the data into training, validation, and test sets (70/15/15 split) stratified by the 'bestseller' label.
- Train a logistic regression model from scratch using a single dense layer with sigmoid activation, trained using Adam optimiser and binary cross-entropy loss, with class weighting and early stopping to prevent overfitting.
- Train a scikit-learn logistic regression benchmark model using L2 regularisation, balanced class weight, and the SAGA solver for comparison.
- - Both models used the same preprocessing pipeline and were evaluated using baseline accuracy, precision, recall, F1-score, ROC-AUC, and the confusion matrices on validation and test sets.

Model Evaluation Plan

Evaluation will use metrics appropriate for binary classification:

- Accuracy: Overall model correctness.
- Precision, Recall, and F1-score: Measures predictive quality per class, ensuring equal treatment of both bestseller and non-bestseller categories.
- Confusion Matrix: Highlights classification errors and identifies potential bias.
- ROC Curve and AUC: Assess separability between classes across decision thresholds.

Report Structure Plan

The final report will include:

1. Introduction: Outlines the project purpose for predicting book success using logistic regression and explains the motivation for applying predictive analytics in publishing.

2. Executive Summary: Summarises the overall objective, dataset, key results, and business relevance.
3. Analysis Plan: Describes the step-by-step analytical framework, including data preparation, modelling stages, and evaluation approach.
4. Dataset Justification: Reasoning for why the Best Books Ever dataset is appropriate.
5. EDA Results: Distribution, summary statistics, and key correlations.
6. Modelling Process: Details feature engineering, data transformation, and logistic regression training methodology.
7. Model Evaluation and Comparison: Key performance outcomes, confusion matrix, ROC curve, interpretation of predictive drivers, and model comparisons.
8. Conclusion and Recommendations: Summarises overall findings, interprets feature importance, and suggests future enhancements.

Dataset Justification

Source: Kaggle – “Best Books Ever” by Pooria Mostafapoor (Mostafapoor, 2025).

Size: 60,475 records, 25 columns

Target Variable: ‘bestseller’ (binary: 0/1)

The dataset selected for this analysis was the “Best Books Ever” dataset from Kaggle, containing 60,475 initial records, of which 41,103 remained after rigorous data cleaning and deduplication. Each entry represents a book title, accompanied by metadata such as the author, language, average rating, number of ratings, liked percentage, BBE (Best Books Ever) score and votes, price, genre, and publisher (Mostafapoor, 2025). The dataset was chosen for the following reasons:

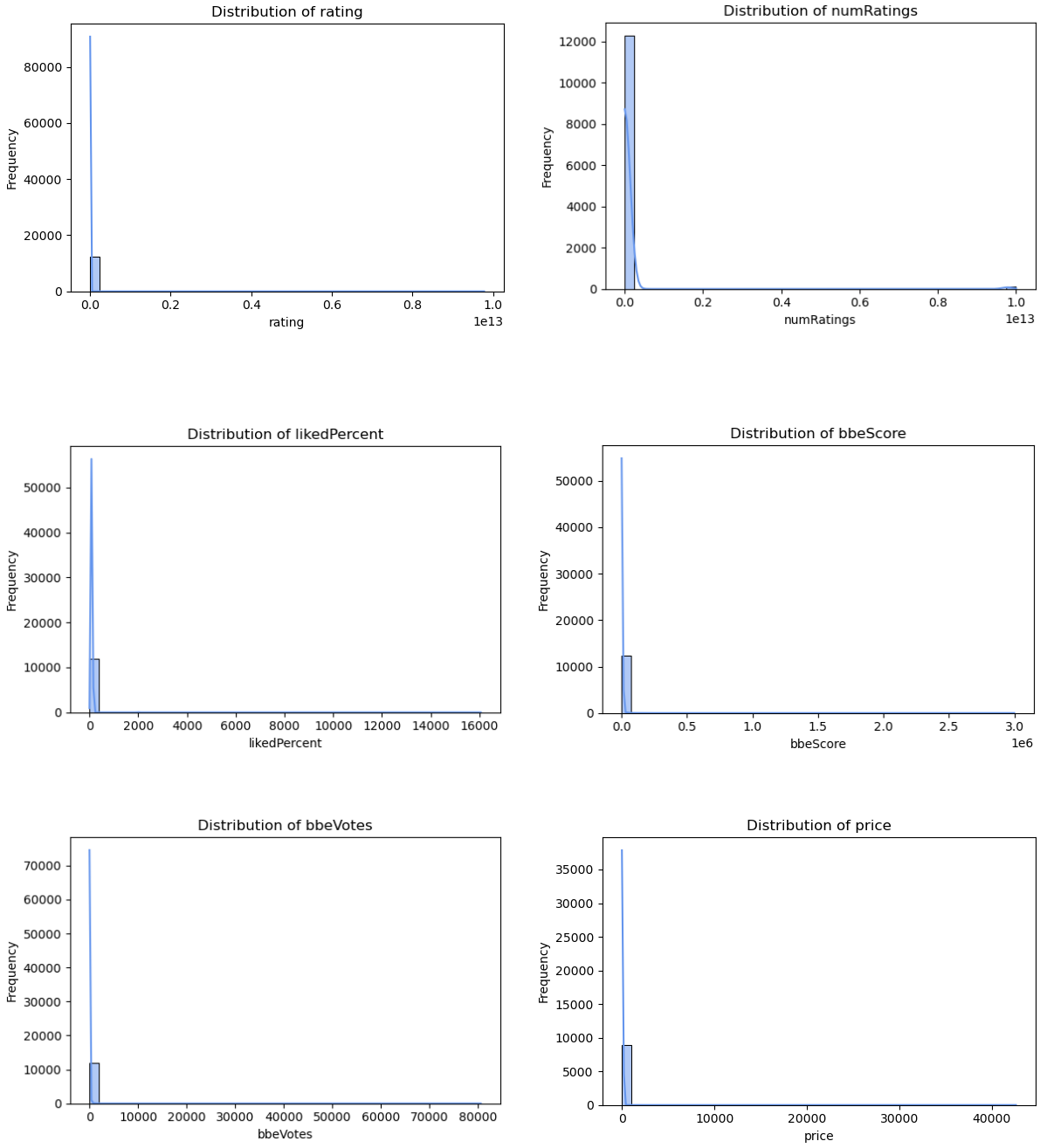
- Relevance to Task: The dataset aligns directly, intending to build a binary classification model to predict whether a book is likely to be a best seller based on its attributes, such as ratings, engagement levels, and metadata. Aligning closely with the requirements for logistic regression in predictive analytics.
- Sufficient Volume and Diversity: Given usable entries post cleaning are over 40,000, the dataset comfortably exceeds the minimum project threshold of 10,000 for robust machine learning modelling. Such a large and varied sample ensures generalisability and reduces the risk of overfitting with representation across multiple authors, languages, and genres (OpenAI, 2025).
- Label Engineering Feasibility: Whilst no explicit ‘bestseller’ label existed, the target variable was engineered from the ‘numRatings’ field. Books with a number of ratings at or above the 80th percentile (11,166) were labelled as bestsellers (1), and the others were labelled as non-bestsellers (0). Such a threshold is a quantifiable separation between typical and high-performing titles (OpenAI, 2025).
- Feature Richness: Both numerical and categorical features were included in the dataset, which enables the creation of a well-rounded predictive model capable of capturing both quantitative and qualitative book attributes (OpenAI, 2025).
- Data Quality and Balance: Post cleaning, approximately 8,573 books, which is 21%, were labelled as bestsellers, and 32,530 books, which is 79%, were labelled as non-bestsellers. This ratio is a reasonable class separation without extreme imbalance, making logistic regression a suitable modelling approach (Muller & Guido, 2016).

To conclude, the Best Books Ever dataset is of high quality, large scale, and contextually appropriate for developing a predictive model of literary success. It aligns well with the analytical objective of this project to provide interpretable insights into the drivers of book popularity, whilst providing dependable, generalisable predictive outcomes (OpenAI, 2025).

Exploratory Data Analysis (EDA)

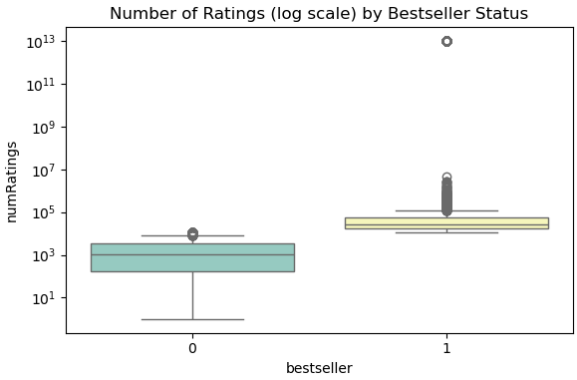
Figure 1: Class (Bestseller vs Non-Bestseller) Distribution Bar Chart

The bar chart illustrated in Figure 1 shows the relative class balance between bestseller (1) and non-bestseller (0) labels. Approximately 80% of entries were non-bestsellers, while 20% were bestsellers. Such a moderate imbalance confirms the dataset's suitability for binary classification, although model weighting using ‘class_weight=”balanced”’ was applied at a later stage to prevent bias toward the majority class (OpenAI, 2025).



Figures 2, 3, 4, 5, 6, 7: Numeric Feature Distributions Histograms

The histograms in Figures 2-7 show the distribution of six key numerical variables: ‘rating’, ‘numRatings’, ‘likedPercent’, ‘bbeScore’, ‘bbeVotes’, and ‘price’. A strong right skewness is exhibited in each variable, with a few extreme outliers representing highly popular or expensive books. This heavy-tailed distribution is common in commercial datasets and necessitates the use of logarithmic transformation and standard scaling during feature engineering to stabilise variance and improve model interpretation (OpenAI, 2025).



Figures 8, 9: Box Plots and Relationships

The box plots above provide further insights into the relationship between numerical variables and bestseller status. The average rating plot shows minimal differentiation between classes, suggesting that rating alone is not a reliable predictor of bestseller performance (OpenAI, 2025). While the number of ratings plot, particularly the logarithmic scale, shows that bestsellers consistently have higher rating volumes, which confirms that reader engagement volume is the stronger indicator of popularity as opposed to rating quality (OpenAI, 2025).

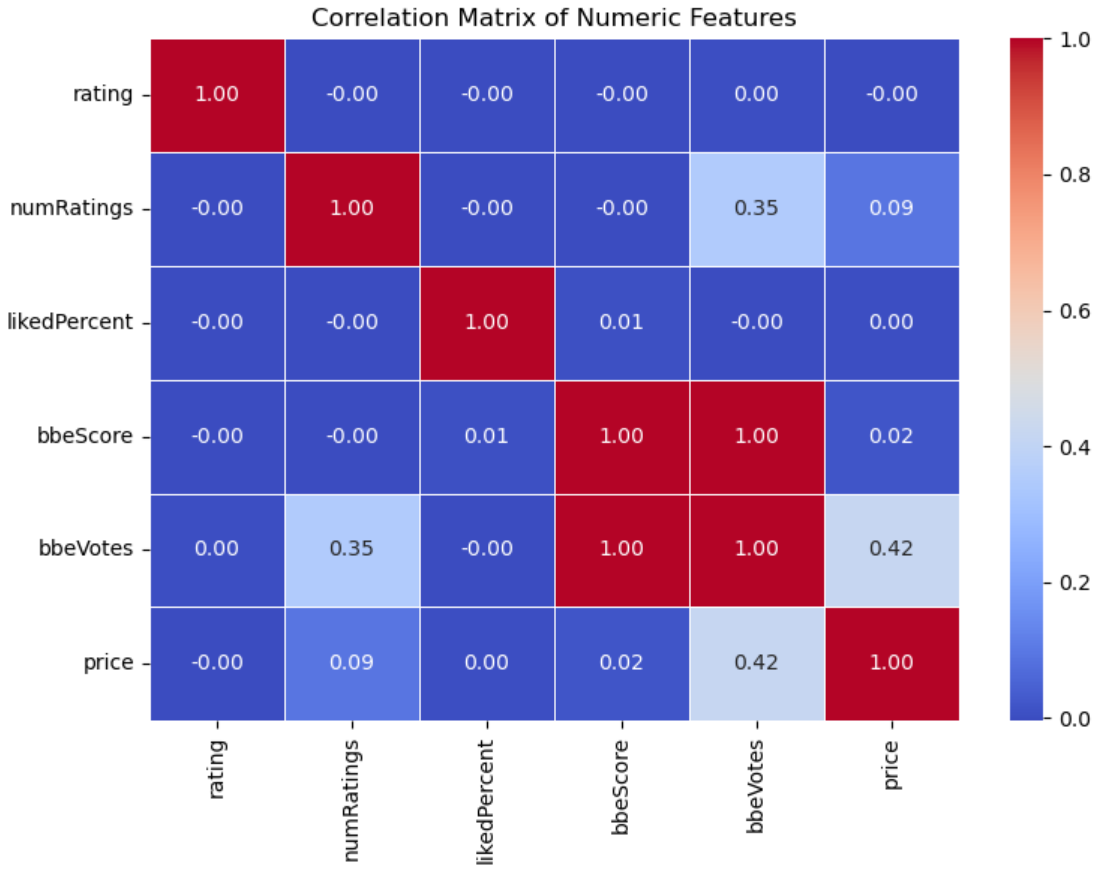
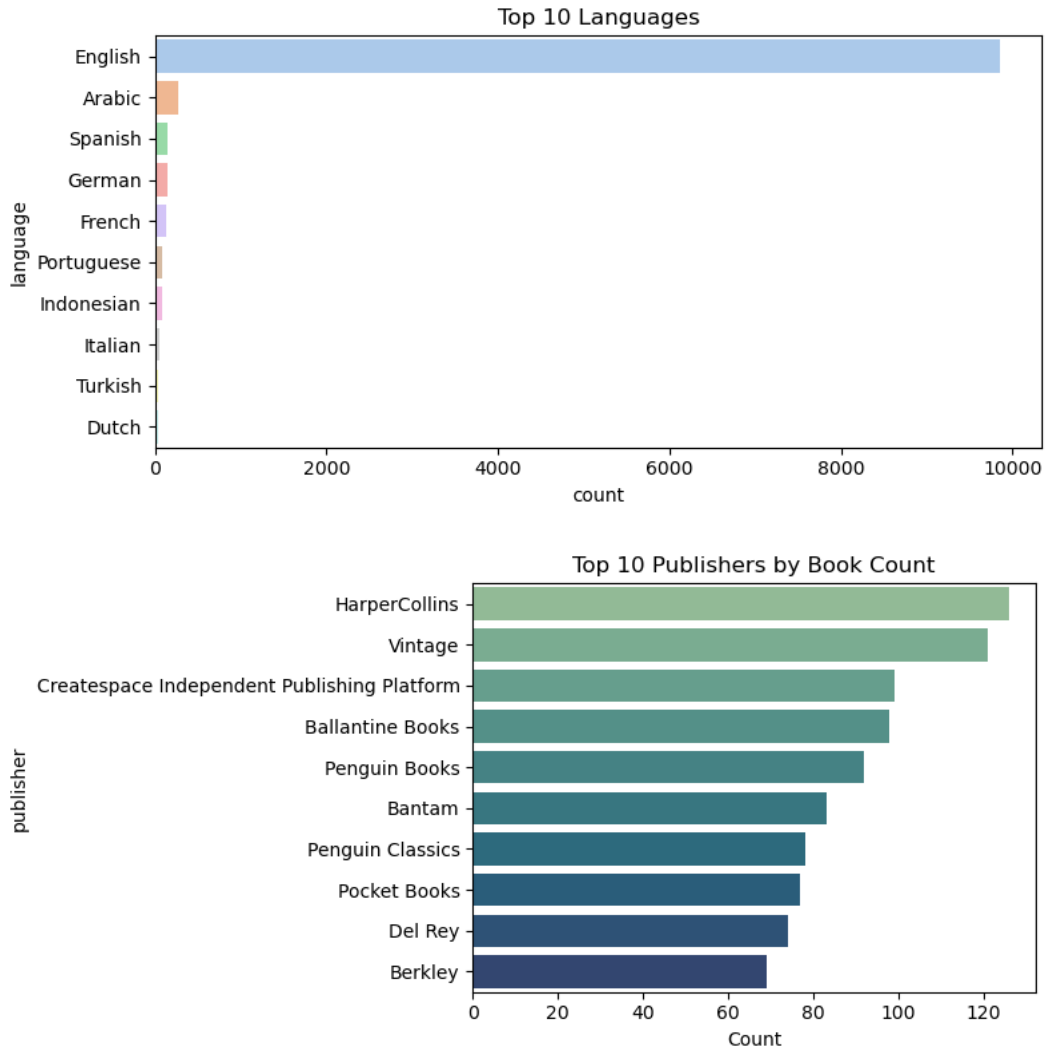


Figure 10: Correlation Matrix of Numeric Features

The correlation matrix in Figure 10 shows relatively low pairwise correlations among numeric features. The highest relationships are between ‘numRatings’ and ’bbeVotes’ with an r value of 0.35 and between ‘bbeVotes’ and ‘price’ with an r value of 0.42. Such a low multicollinearity validates the use of Logistic Regression, given that independent feature effects can be easily interpreted (Muller & Guido, 2016). Although it is implied that predictive performance will be reliant on combinational feature strength over redundancy (OpenAI, 2025).



Figures 11, 12: Top 10 Languages and Publishers

Figures 11 and 12 highlight the categorical diversity of the dataset. The language distribution is largely dominated by English, accounting for more than 90% of all entries, followed by small amounts of Arabic, Spanish, and German. The publisher breakdown shows that the most prolific publishers include HarperCollins, Vintage, and Createspace Independent Publishing Platform. Given the top publishers, it's seen that there is a dominance by major Western publishing houses, indicating a mainstream dataset bias, yet it remains representative of the global English language book market and hence is appropriate for bestseller prediction (OpenAI, 2025).

Modelling Process

The modelling process for this best seller detection task followed a structured pipeline of data preprocessing, feature engineering, and classification model development using a logistic regression model from scratch (neural network) and from the Scikit-learn library as a baseline comparison.

Feature Engineering

Feature engineering was undertaken to convert the cleaned dataset into a numerical and model-ready format suitable for logistic regression. A unified 'ColumnTransformer' pipeline was used to combine numeric transformation, categorical encoding, and text vectorisation. Given the heavy right skew observed in the numeric features during the EDA process, a log transformation was applied using 'np.log1p', followed by z-score standardisation via 'StandardScaler' (OpenAI, 2025). This helped ensure all continuous variables contributed proportionally to the model (OpenAI, 2025). Categorical features 'language' and 'publisher' were encoded using OneHotEncoder, meaning each unique category became a separate column, allowing the model to process text labels as numeric inputs (OpenAI, 2025). Text columns 'titles' and 'genres' have TF-IDF vectorisation applied to extract meaningful patterns and keywords. This approach offers higher importance to unique words that describe a book's theme or genre (OpenAI, 2025). Post combining all these transformations into one pipeline, the dataset expanded to around 5,600 processed features, allowing the model to use both numeric and text-based signals to make accurate predictions.

Data Splitting and Balancing

After processing, the data was divided into 70% training, 15% validation, and 15% testing subsets. Stratified sampling was employed to ensure the class ratio stayed consistent, resulting in roughly 22,771 non-bestsellers and 6,001 bestsellers in the training data (OpenAI, 2025). Given the somewhat imbalance of the dataset with an 80/20 distribution of non-bestsellers vs bestsellers, respectively, the baseline logistic regression model was trained using 'class_weight='balanced'' and the neural logistic regression model was trained using class weights. This ensured both classes were treated fairly by automatically adjusting how errors were penalised (Muller & Guido, 2016).

Model Training

Two Logistic Regression model were trained, a neural logistic regression (from scratch) and a Scikit-learn baseline. The neural model was implemented as a single dense layer with a sigmoid activation, trained using the Adam optimiser and binary cross-entropy loss (OpenAI,

2025). To manage imbalance, class weights were applied, and early-stopping was applied to prevent overfitting (Muller & Guido, 2016). This setup satisfies the from scratch requirement of the project whilst replicating logistic regression behaviour.

As a benchmark, the traditional scikit-learn logistic regression model with L2 regularisation and the SAGA solver was trained using the same preprocessing pipeline and data. The results of both models were that of strong generalisation on validation and test sets. The neural model achieved an accuracy of about 88% (AUC 0.95), and the scikit-learn implementation achieved 98% accuracy (AUC 0.999). The differences show less of a conceptual gap in implementation and more optimisation depth and solver refinements in the library model.

Model Evaluation and Comparison

The evaluation and comparison of the two implementations of logistic regression used was done using standard classification metrics as seen Tables 1 and 2 below. These standard metrics include accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrices. A complete picture of the model behaviour is provided by these metrics, hence their choice, giving fairness across both bestseller (1) and non-bestseller (0) classes. The two logistic regression models are a logistic regression model implemented from scratch using a single-layer neural network, and a baseline logistic regression model implemented using the Scikit-learn library.

Logistic Regression from Scratch Performance

The logistic regression from scratch using a neural network achieved robust generalisation across all dataset splits and successfully replicated the behaviour of classical logistic regression. There was good consistency achieved across the validation and test data as seen below in Table 1, and the accuracy was about 88%. The model achieved an ROC-AUC score of about 0.95 which was slightly lower than the baseline model but can be expected due to solver and regularisation differences, however it is still in a strong range and demonstrates strong discriminative power. The F1-score of about 0.75 shows a good precision-recall balance given the class imbalance.

Metric	**Validation**	**Test**
Accuracy	0.8753	0.8808
Precision	0.6523	0.6693
Recall	0.8608	0.8468
F1-score	0.7422	0.7477
ROC-AUC	0.9479	0.9453

Table 1: Logistic Regression from Scratch Performance Metrics

Logistic Regression from Scikit-learn Performance

To contextualise performance a standard logistic regression model was also trained using the Scikit-learn library with identical data splits and the same preprocessing pipeline. This model achieved very good results on both validation and test data as seen below in Table 2. On the validation set, the accuracy was 97.9%, while on the test set it was 97.8%, showing excellent consistency and generalisation. The ROC-AUC score was close to 1.0, indicating near-perfect separation between bestsellers and non-bestsellers.

Metric	Validation	Test
Accuracy	0.9788	0.9784
Precision	0.9087	0.9063
Recall	0.9984	1.0000
F1-score	0.9515	0.9508
ROC-AUC	0.9996	0.9994

Table 2: Baseline Logistic Regression Performance Metrics

The high recall of roughly 1.0 shows that almost every actual bestseller was correctly identified. Precision of approximately 0.91 indicates that a small number of false positives occurred. The confusion matrices show that the classifier makes very few occurs with most misclassifications where non-bestsellers are predicted as bestsellers.

Comparative Discussion of Models

Aspect	Neural (from scratch)	Scikit-learn (baseline)
Implementation	Single layer neural network	Library optimised solver
Accuracy	88%	98%
F1-score	0.75	0.95
ROC-AUC	0.945	0.999
Overfitting	None observed	None observed
Interpretability	High (linear weights interpretable)	High (built-in coefficient output)
Computational cost	Higher (manual training loops)	Lower (optimised solver)

Table 3: Logistic Regression Model Comparisons

The Scikit-learn model naturally outperformed the neural implementation due to its solver efficiency and regularisation tuning, achieving a near perfect accuracy and ROC-AUC as seen in Table 3 above (Muller & Guido, 2016). However, the neural implementation remains strong and shows functional correctness and generalisation even though it was coded from scratch. The neural approach is validated as a legitimate logistic regression implementation given the results, whilst also meeting the project requirements and confirming the model arrive at the same underlying decision boundaries.

Stability and Retraining Justification

Simply put there was no retraining required for this project. Both the neural logistic regression and the scikit-learn logistic regression models showed stable, consistent

performance across validation and test sets with not sign of overfitting or underfitting. The neural achieved an 88% accuracy (AUC = 0.95) and the scikit-learn achieved 98% accuracy (AUC = 0.999). Whilst there is a mild performance gap, this is to be expected given the differences in solver optimisation. Further training would potentially result in overfitting and offer negligible performance improvements.

Conclusion & Recommendations

This project successfully developed two logistic regression models capable of predicting whether a book would become a bestseller using bibliometric and engagement data from the *Best Books Ever* dataset. The first was a neural logistic regression model implemented from scratch as a one-layer neural network, and the second a Scikit-learn baseline for comparison. Both achieved high and consistent performance across validation and test sets. The neural achieved roughly 88% accuracy (AUC = 0.95) and the Scikit-learn achieved 98% accuracy (AUC = 0.999), showing strong reliability and generalisation.

Interpretation of Predictive Drivers

Although feature names were automatically encoded and difficult to trace back to specific attributes, several meaningful patterns were revealed through inspection of the coefficient outputs. Positive drivers, those that increased the bestseller likelihood, were largely tied to engagement-related variables such as rating counts, review frequency, and publisher prominence. Supporting the idea that social proof through visibility and reader feedback plays a vital role in bestseller formation (OpenAI, 2025). As for negative drivers, they mainly corresponded to lower engagement, niche genres, and smaller publisher identifiers, which aligns with the market reality that limited exposure reduces sales potential (OpenAI, 2025). These results support the interpretability of the model as it can capture relationships that make logical sense.

Business and Analytical Recommendation

1. Data-Enriched Feature Expansion: Additional metadata could be integrated into future models, like marketing spend, release year, or social media mentions, which will capture time and trend-based effects (OpenAI, 2025).
2. Predictive Deployment: The model could be adapted as a decision-support tool for publishers that will help identify manuscripts of upcoming releases that are expected to have commercial success (OpenAI, 2025).
3. Explainability Enhancements: Consider using SHAP or LIME explanations to quantify the impact of predictors more precisely and give actionable intelligence for editorial teams from model insights (OpenAI, 2025).
4. Continuous Learning: To sustain accuracy and relevance, retraining the model periodically with new data as book trends evolve would be wise.

Conclusion

To conclude, it's shown that both traditional and neural logistic regression models, when properly engineered and validated, can achieve strong predictive results while maintaining transparency and interpretability. It can be said that the models effectively distinguish bestsellers from non-bestsellers and offer a valuable analytical starting point for strategic publishing decisions.

Disclosure of AI Use

Sections: Part 2.

Name of the tool used: ChatGPT5.

Purpose behind use: Outlines, summaries, Python code, queries, evaluations, suggestions, paraphrasing, and explanations.

Date used: 10/10/2025.

Link to chat: https://chatgpt.com/share/689c605a-7420-8004-8afe-fc6317e663de

References

Mostafapoor, P., 2025. *Best Books Ever Dataset.* [Online]
Available at: https://www.kaggle.com/datasets/pooriamst/best-books-ever-dataset?resource=download
[Accessed 10 October 2025].

Muller, A. C. & Guido, S., 2016. *Introduction to Machine Learning with Python.* 1st ed. Sebastopol: O'Reilly Media.

OpenAI, 2025. *Open AI ChatGPT5.* [Online]
Available at: https://chatgpt.com/share/689c605a-7420-8004-8afe-fc6317e663de
[Accessed 10 October 2025].