DSTI Applied MSc in Data Science & Artificial Intelligence

Python Machine Learning Labs – Goodreads dataset: Prediction of book ratings

By Gabriela Copetti

Table of contents

Introduction

The Goodreads website allows users to give books a rating of 1 to 5 stars.

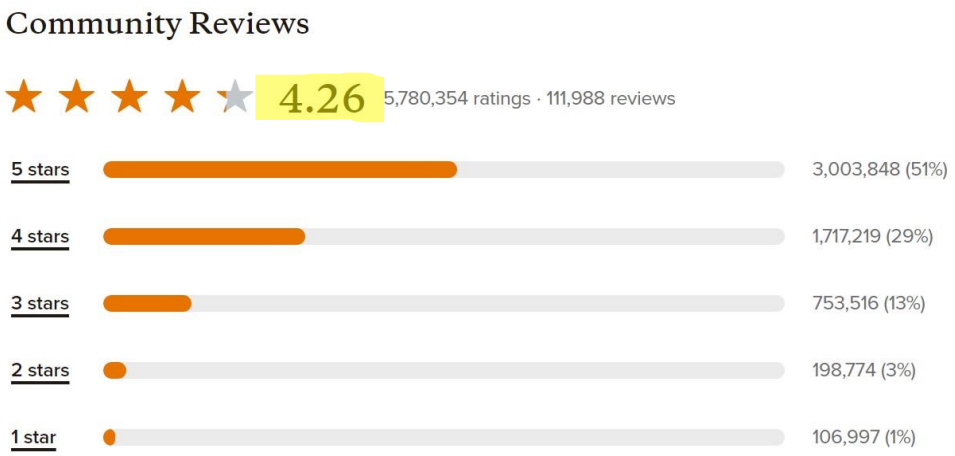


Figure 1 - Image taken from Goodreads website, showing the rating system. Average rating is highlighted.

The aim of this project is to apply Machine Learning techniques, including data cleaning and analysis, feature selection, model training, and evaluation, to **predict book average ratings** at the Goodreads website using the dataset provided in the course. All computations were performed using Python 3.9 and Jupyter Notebook 6.5.

The dataset

The data obtained from Goodreads website is available in the csv file “books.csv” and contains the following attributes, using the comma as column delimiter:

- **bookID**: a unique identification number.
- **title**: the name of the book.
- **authors**: includes the names of the authors, as well as other people involved in the making of the book such as illustrators and editors.
- **average_rating**: an average of the ratings given to the book by the Goodreads users.
- **isbn**: the International Standard Book Number is a 10-digit unique commercial book identifier, used until 2007.
- **isbn13**: 13-digit ISBN, assigned after 2007.
- **language_code**: the language in which the book was written.
- **num_pages**: the number of pages.
- **ratings_count**: how many times the book was rated by users.
- **text_reviews_count**: how many written reviews has the book received.
- **publication_date**: date the book was published.
- **publisher:** name of the publisher.

The dataset had a total of 11127 book entries.

The original csv file "books.csv" was edited manually to ensure that there were no commas within cells. Issues were found in lines 3350, 4704, 5879, 8981 of the column “authors”. For example, "Sam Bass Warner, Jr" had to be modified to "Sam Bass Warner Jr" to prevent an error when reading the dataset. The corrected table was saved as "books_edited.csv".

Data analysis

In the Jupyter notebook “Goodreads_Dataset_Data_Analysis”, exploratory data analysis was used to summarize main characteristics and to check for patterns within the data. Data cleaning was performed, as well as preliminary feature selection and engineering.

Data cleaning included searching for missing values, duplicates and making sure that the column titles were clean. For example “ num_pages” has a space that prevented this column to be properly processed and had to be transformed into “num_pages”.

Features were split into numerical and categorical.

Numerical variables

Numerical included the target variable – average rating – as well as ISBNs, ratings count, number of pages and text reviews count. Since ISBNs are identification numbers that should not have any impact on the ratings of its book, “isbn” and “isbn13” columns were immediately dropped.

Ratings count: The number of ratings a book would receive had a very skewed distribution. While the median ratings count was of 745, there were book with several million votes. The books with most ratings were:

#1 Twilight by Stephenie Meyer

#2 The Hobbit by J. R. R. Tolkien

#3 The Catcher in the Rye by J. D. Salinger

These are all very popular books and would make sense that they would have millions of ratings.

Some books had ratings count equal to zero. It was assumed that a book had to have at least 1 rating for the entry to be valid. Otherwise, some books would have an average rating of zero. Therefore, books with less than 1 rating were dropped from the dataset.

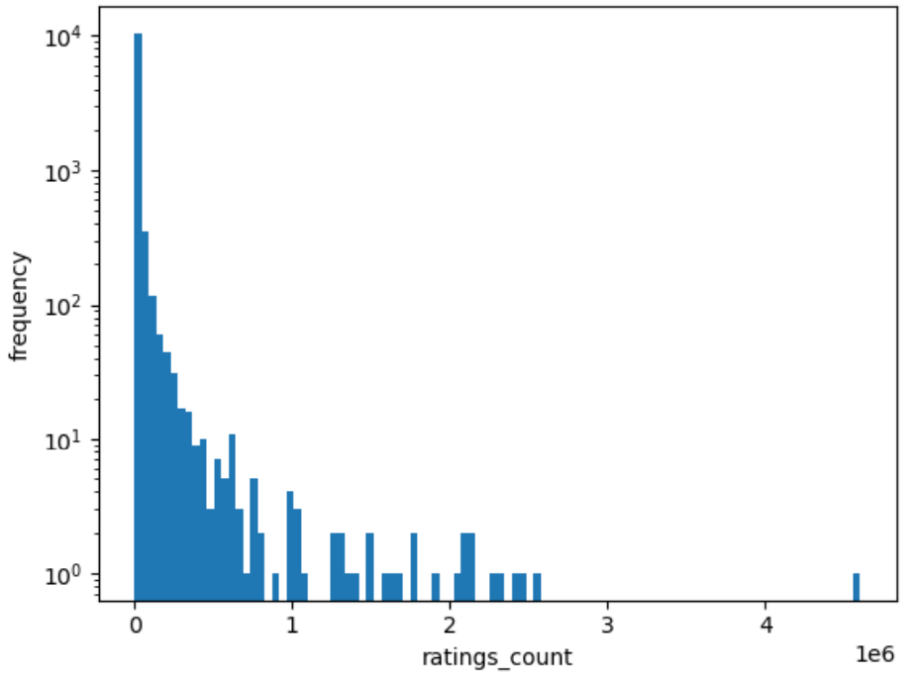


Figure 2 - Distribution of rating counts

Number of pages: Some entries in the dataset indicated books with 0 pages. It was concluded that all books with a very low number of pages (<20) were in fact audio books. These 81 entries were discarded – less than 1% of the size of the dataset.

The distribution of number of pages was also very skewed. While the median was 304 pages, some books had thousands of pages. Books with more than 1700 pages were box sets, author’s complete works or a study Bible. These were kept in the dataset at this point.

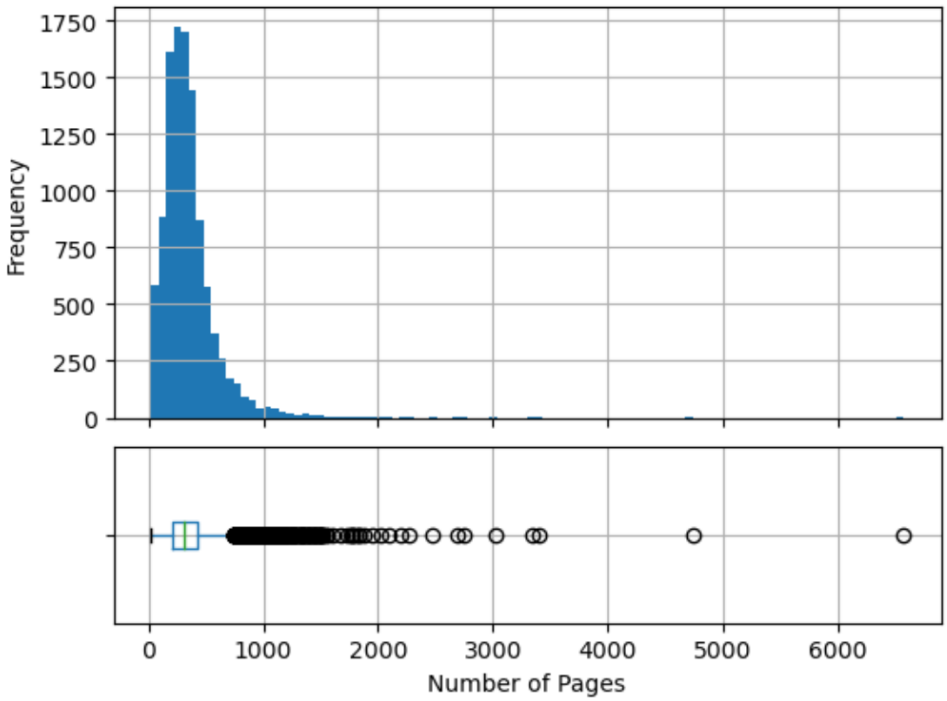


Figure 3- Distribution of number of pages

Text reviews count: More than half the books have under 100 text reviews. However, some books have dozens of thousands of text reviews. The number of text reviews was strongly correlated with the number of reviews (Pearson's correlation coefficient of 0.87 – see Fig 6). To avoid training the machine learning model with correlated features (other than with the target variable), it was chosen to discard text reviews counts at feature selection.

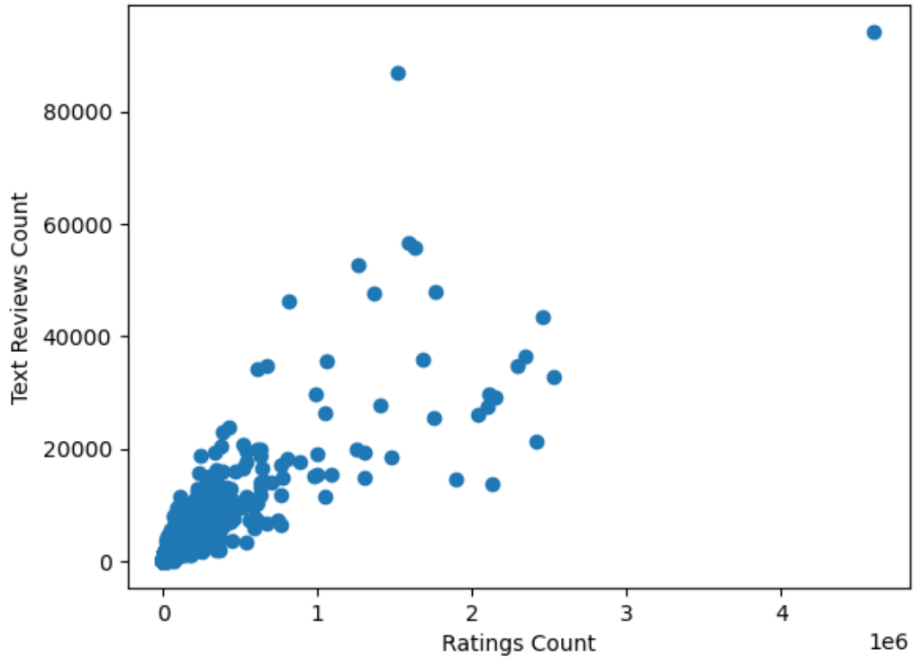


Figure 4 - Text reviews count vs ratings count.

Average rating: This is the **target** variable that the model with try to predict. The distribution of average ratings is bell shaped. Removing entries with rating counts < 10, it was possible to reduce the kurtosis and the skewness of the distribution bringing it closer to normality. 552 entries were removed in total, which accounts to about 5% of the dataset. The distribution of ratings is very narrow, with around 95% of ratings in between 3.35 and 4.52 (mean +- two standard deviations). This shows that users of Goodreads tend to grade well the books that they interact with, usually do not giving less than 3 stars.

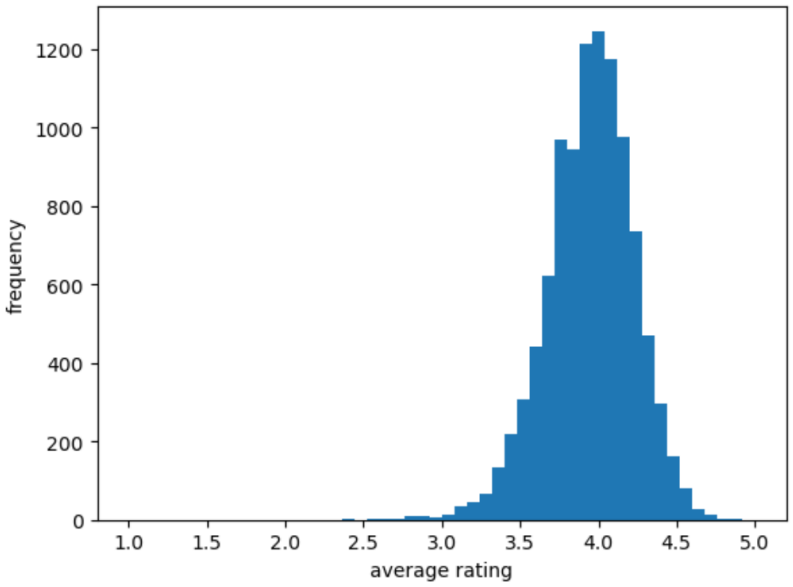


Figure 5 - Distribution of average rating

Correlation heatmap showed that average rating is not strongly correlated to any of other numerical features. The highest correlation coefficient was between average rating and number of pages (0.2).

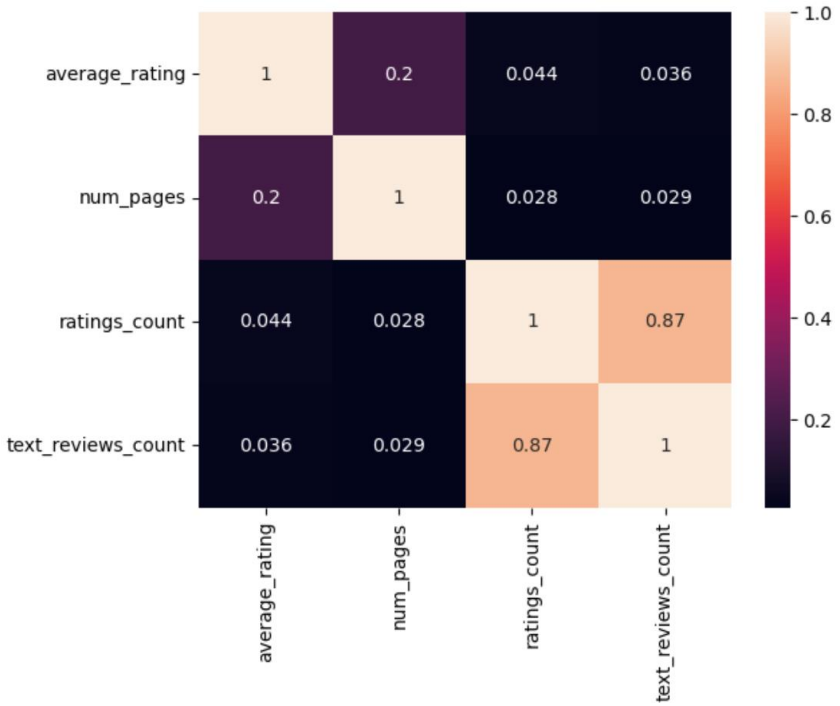


Figure 6 - Correlation heatmap

It is interesting to note that as the number of pages or the ratings count increased, the distribution of average rating became narrower.

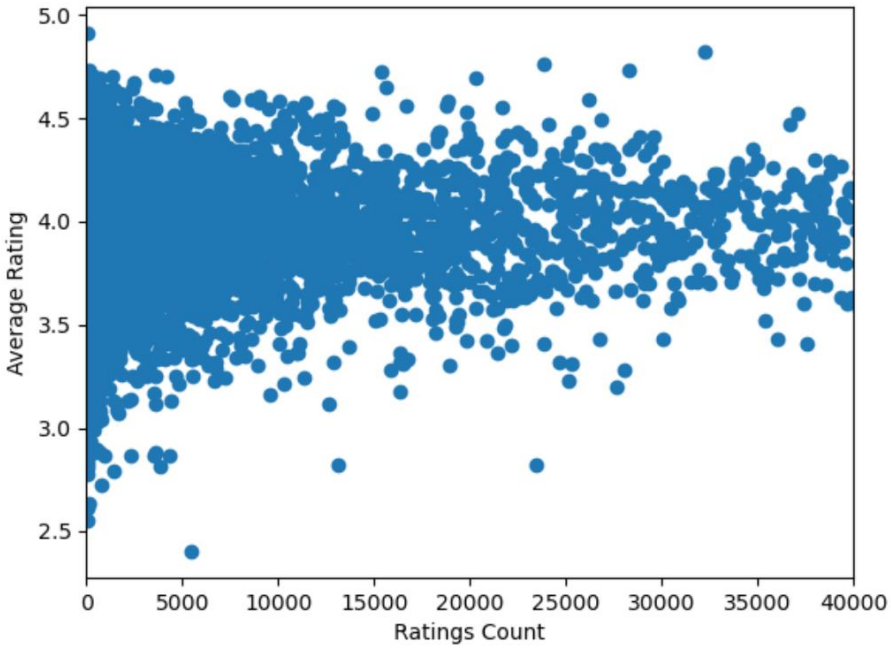


Figure 7 - Average rating vs ratings count.

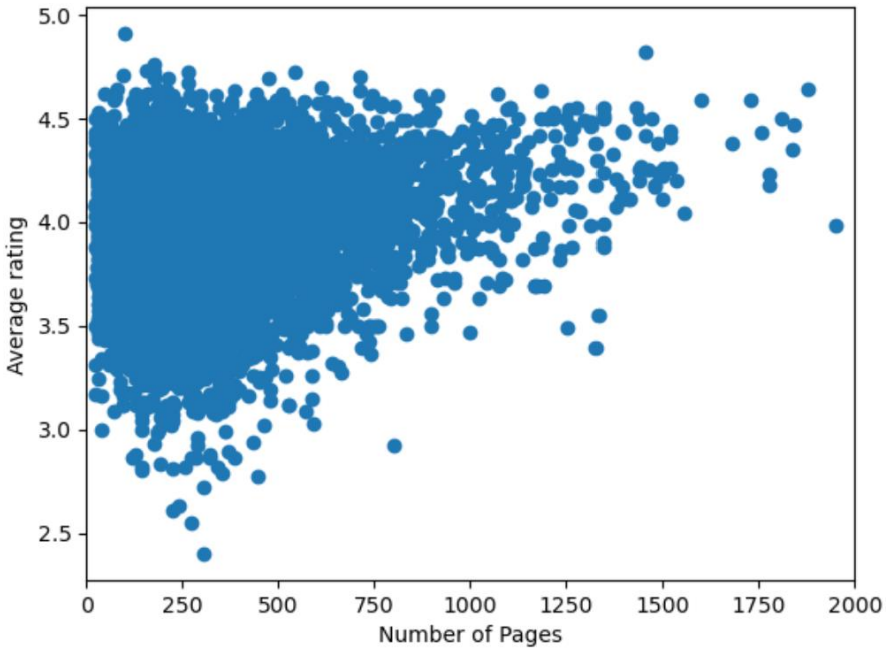


Figure 8 - Average rating vs number of pages.

Categorial variables

Categorical variables included title, language, publication date, authors and publisher.

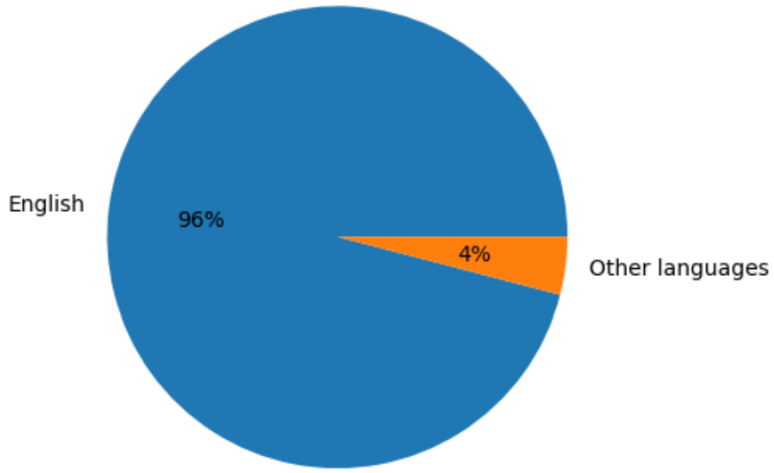
Title: After the cleaning of the dataset based on the numerical variables, the titles with the highest ratings were:

1 Existential Mediation by Simon Cleveland (4.91)
#2 The Complete Calvin and Hobbes by Bill Watterson (4.82)
#3 Harry Potter Boxed Set by J.K. Rowling (4.78)
The lowest rated titles were:
#1 Citizen Girl by Emma McLaughlin (2.40)
#2 The Trouble with the Pears: An intimate portrait of Erzsebet Bathory by Gia Al Babel (2.55)
#3 A Matter of Trust by Penny Jordan (2.61)

The title is longtext data that was assumed not be a relevant feature for the machine learning model. It was discarded from the dataset in feature selection.

Language: It was noted that language codes were not consistent. For example, there were several codes for English (eng, en-US, en-GB, en-CA and enm), which were grouped into a single category with language_code "eng".

96% of books in the dataset were in English, while only 4% were written in other languages. Second and third most frequent language were Spanish and French, with 174 and 98 books, respectively. Given that descriptive statistics of both books in English and other languages were very similar and that the dataset is extremely unbalanced in terms of language, it was decided to not keep language code as a relevant feature for the machine learning model.



language_code	count	mean	std	min	25%	50%	75%	max
eng	9808.0	3.942781	0.275585	2.40	3.7800	3.96	4.13	4.91
other	414.0	3.997367	0.270597	3.06	3.8125	3.99	4.19	4.59

Figure 9 – Percentage and stats of books in the dataset written in English or other languages.

Publication date: It was assumed that the year in the publication date was of greater relevance than day and month. Therefore, the year was extracted from the publication date column and placed in a new column "publication_year". The original column "publication_date" was discarded. Most books in the dataset were published in the early 2000s, with a big drop in counts for books published after 2006. The mean average rating varied very little for every year. Taking this into account, together with the fact that the dataset was very unbalanced in terms of publication year, this variable was not selected as a relevant feature and ended up being discarded.

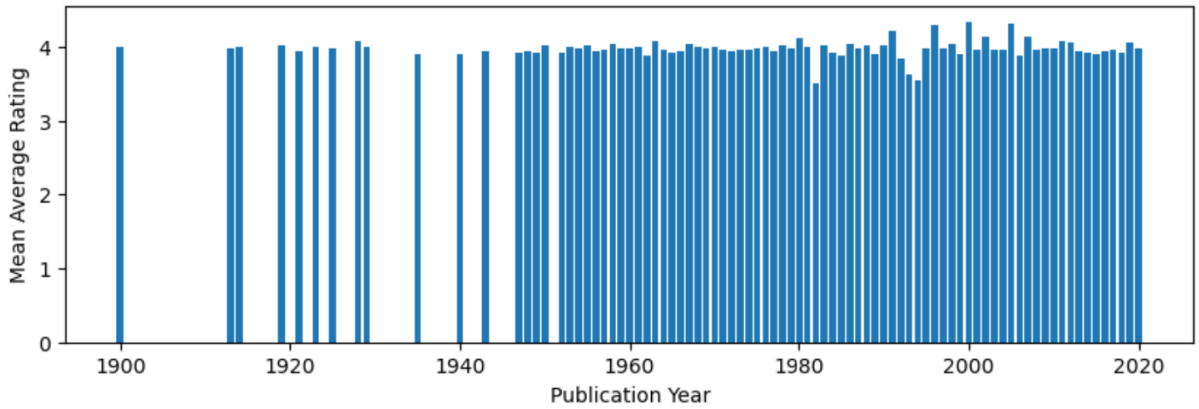


Figure 10 -Mean average rating vs publication year.

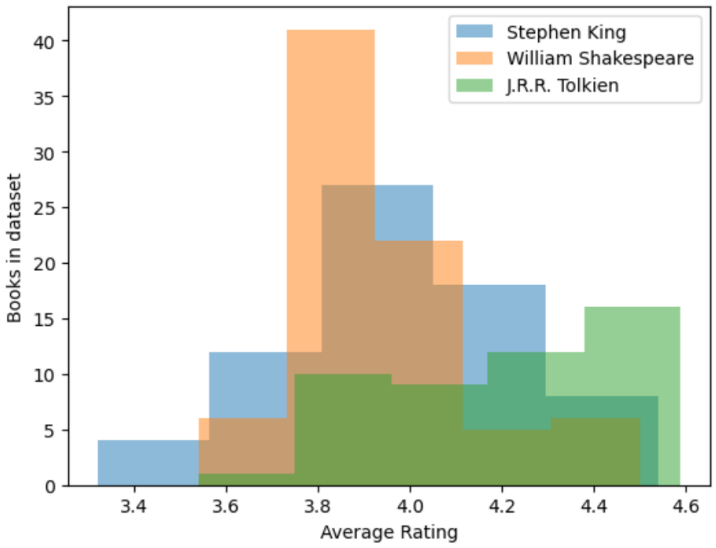
Authors: The "author" column could contain one or more names, separated by "/". However, examining the authors' names and their respective books, it was possible to infer in most entries the first name cited was the main and most relevant author. For example:

- The book **Harry Potter and the Half-Blood Prince** has **J.K. Rowling/Mary Grandpré** listed as authors. J.K. Rowling is the author and Mary Grandpré is the illustrator.
- The trilogy of **The Lord of the Rings** has **J.R.R. Tolkien/Alan Lee** listed as authors. J.R.R. Tolkien is the author and Alan Lee is the illustrator.
- The book **Anna Karenina** has **Leo Tolstoy/Richard Pevear/Larissa Volokhonsky** listed as authors. Leo Tolstoy is the author and Richard Pevear and Larissa Volokhonsky are translators.

The first name was extracted from "authors" and placed in a new column called "main_author". This allowed the grouping of books by the same main author. It was observed that different main authors had different mean average rating and standard deviation.

	average_rating		
	count	mean	std
main_author			
William Shakespeare	80	3.93	0.20
Stephen King	69	4.00	0.26
J.R.R. Tolkien	48	4.21	0.26
P.G. Wodehouse	41	4.16	0.16
James Patterson	40	3.89	0.25
Mercedes Lackey	40	4.00	0.14
Orson Scott Card	38	3.79	0.27
Rumiko Takahashi	38	4.19	0.11
Gabriel García Márquez	36	3.93	0.14
Margaret Weis	36	3.84	0.25

(a)



(b)

Figure 11 - (a) Count of books, mean average rating and its standard deviation for different main authors. (b) The average rating distribution for William Shakespeare, Stephen King and J. R. R. Tolkien.

Publisher: In the "publisher" column, the name of one publisher can be found written in different formats. The column also contains names of different branches of the same parent publisher. For example:

Bloomsbury / Continuum	Bloomsbury Paperbacks
Bloomsbury Academic	Bloomsbury Publishing
Bloomsbury Arden Shakespeare	Bloomsbury Publishing PLC
Bloomsbury Childrens Books	Bloomsbury U.S.A. Children's Books
Bloomsbury Children's Books	Bloomsbury USA
Bloomsbury Methuen Drama	Bloomsbury UK

To see if the publisher affects the average rating of a book of the dataset, it was decided to group different spellings/branches under the same publisher's name. A new column "parent_publisher" was created and filled with the first part of each name in "publisher". For example, all publishers mentioned above will have "Bloomsbury" as parent publisher. For publishers names that begin with "The", the second word will be considered the first part of the name. For example, "The MIT Press" has parent publisher "MIT". This strategy allowed reducing of the number of publishers from 2018 down to 1191.

It is important to note that while this strategy is helpful to group more books by publishers it leads to mistakes in some cases. Taking for example the publishers that start with the word "Book":

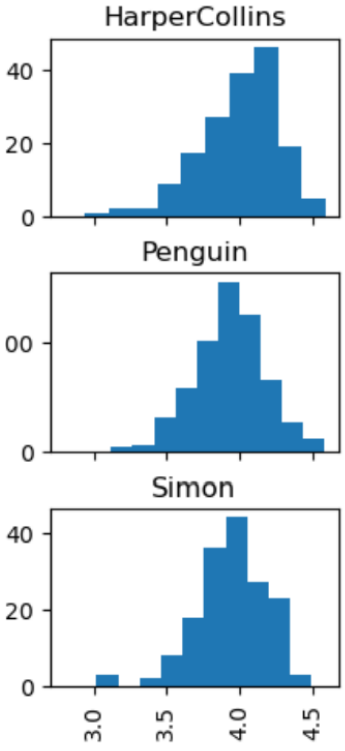
- Book of the Month Club
- Book Publishing Company
- Book Publishing Company (TN)

The last two represent the same publisher and will be named "Book" in the column "parent_publisher". However, "Book of the Month Club" is a different publisher that ended up receiving same name. Other issues include, for example, that Harper and HarperCollins remaining as different publishers or that Random and Vintage, branches of Penguin, remaining ungrouped.

Nevertheless, creating the column parent_publisher allowed a better analysis of the distribution of average rating for each publisher. Notice in Figure 11 that the average rating distribution, the mean and the standard deviation do not show much variation from one publisher to another.

	average_rating		
	count	mean	std
parent_publisher			
Ballantine	148	3.887500	0.284779
Bantam	194	3.907938	0.267702
Berkley	169	3.904083	0.201844
Harper	226	3.935221	0.247911
HarperCollins	167	3.994611	0.275962
Mariner	148	3.940608	0.297891
Oxford	143	3.910490	0.208250
Penguin	583	3.929160	0.239583
Pocket	142	3.913239	0.211065
Random	142	3.939366	0.270883
Simon	164	3.938415	0.247612
Vintage	405	3.907284	0.257573

(a)



(b)

Figure 12 - (a) Counts of books, the mean average rating and its standard deviation for different publishers. (b) The average rating distribution for publishers Harper Collins, Penguin and Simon & Schuster.

Feature Selection

The features pre-selected to be used in the machine learning model were: “num_pages”, “ratings_count”, “main_author” and “parent_publisher”, though data analysis indicated that the number of pages and the main author might be the ones with the most relevance. The processed dataframe was saved in a new .csv file called “books_processed.csv”.

The main author and the parent publisher still needed further encoding to be used in machine learning, but this was left to be done at the modelling section.

bookID	average_rating	num_pages	ratings_count	main_author	parent_publisher
1	4.57	652	2095690	J.K. Rowling	Scholastic
2	4.49	870	2153167	J.K. Rowling	Scholastic
4	4.42	352	6333	J.K. Rowling	Scholastic
5	4.56	435	2339585	J.K. Rowling	Scholastic
8	4.78	2690	41428	J.K. Rowling	Scholastic
...	...	...	...	...	...
45631	4.06	512	156	William T. Vollmann	Da
45633	4.08	635	783	William T. Vollmann	Penguin
45634	3.96	415	820	William T. Vollmann	Penguin
45639	3.72	434	769	William T. Vollmann	Ecco
45641	3.91	272	113	Mark Twain	Edimat

10222 rows × 5 columns

Figure 13 – Dataframe after processing

Model training and evaluation

Methodology

Since the model will try to predict a continuous variable, average rating, a regression algorithm is needed. In this project, four different regression algorithms from the scikit-learn library were used and compared: Linear Regression [1], Ridge [2], Decision Tree Regressor [3] and Random Forest Regressor [4]. The experiment performed consisted in applying these algorithms to different sets of selected features:

- Feature set 1: num_pages, ratings_count, main_author, parent_publisher
- Feature set 2: num_pages, ratings_count, main_author
- Feature set 3: num_pages, ratings_count
- Feature set 4: num_pages, main_author
- Feature set 5: ratings_count, main_author
- Feature set 6: main_author
- Feature set 7: main_author, parent_publisher

The objective was to find the algorithm that performed the best, while minimizing the number of variables used. The evaluation was made using the following metrics [5, 6]:

- MAE (Mean Absolute Error)
- MAPE (Mean Absolute Percentage Error)
- MSE (Mean Squared Error)
- RMSE (Root Mean Square Error) (the square root of MSE)
- R^2 score

These steps were performed in the Jupyter notebook “Goodreads_Dataset_Model_and_Evaluation”.

Columns “main_author” and “parent_publisher” were encoded using the *get_dummies* method from the pandas library [7]. This method converts categorical variable into dummy/indicator variables, with each variable being converted in as many 0/1 variables as there are different values.

The data was split into train and testing, with 25% of the data being reserved for the testing. The splits and algorithms were performed several times to test the consistency of results, i.e., to determine if similar error metrics were obtained after different data splits.

Some tests also included removing some extreme values in number of pages (>1700) and ratings count (>100,000), but this did not provide better results. Therefore, such values were kept.

Results

Algorithm comparison: The best results overall were obtained using the **Ridge Regressor**, with the lowest error metrics and highest R^2 score. Residuals were randomly scattered, which indicates that the model is adequate. Error metrics were fairly consistent between trials, and computation took only a few seconds.

	Linear Regression	Ridge	Decision Tree	Random Forest
MAE	3.918623e+03	0.169724	0.214875	0.180845
MAPE	1.038046e+03	0.043920	0.055390	0.046852
MSE	4.952234e+09	0.050591	0.078364	0.056631
RMSE	7.037211e+04	0.224925	0.279936	0.237973
R^2	-6.616741e+10	0.324045	-0.047030	0.243345

Figure 14 - Error metrics and R^2 score for feature set 2.

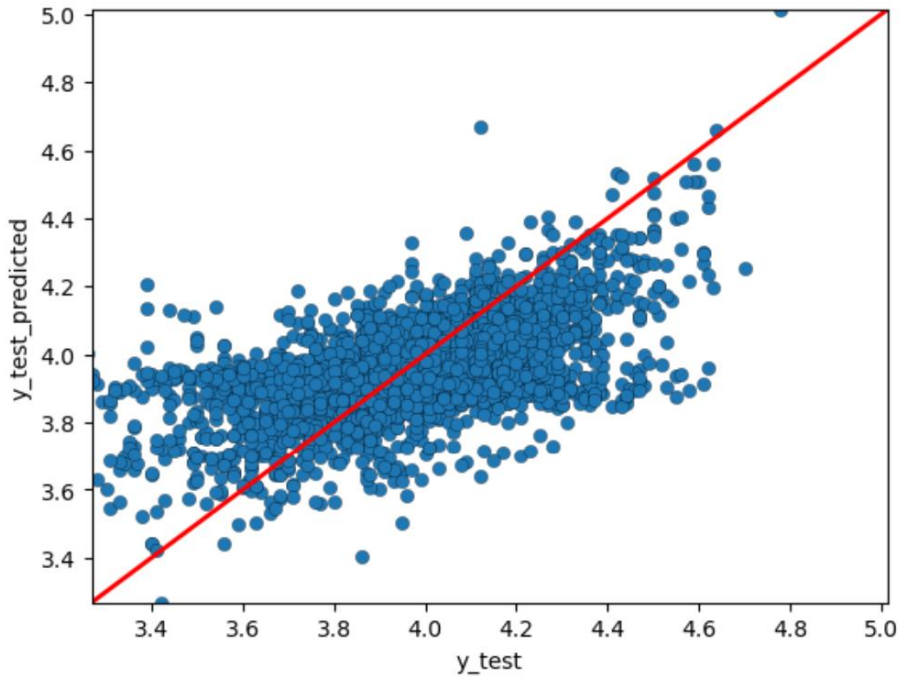


Figure 15 - Predicted average rating vs true average rating, using Ridge regressor, for feature set 2. The red line marks when predicted value is equal to true value.

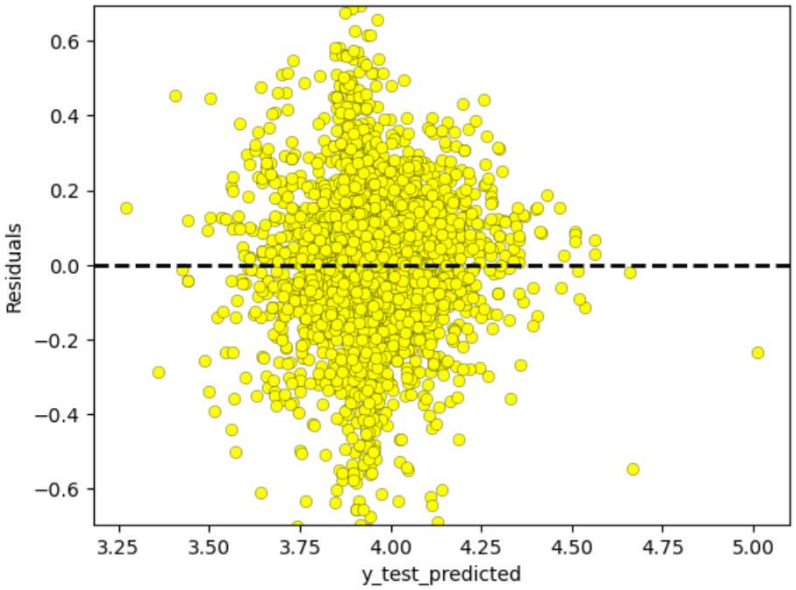


Figure 16- Residual plot obtained using the Ridge regressor and feature set 2.

Still, in the best trials, R^2 score was of only 0.32-0.34, which means that only about a third of the variation in the predicted ratings could be accounted for by the test values. This result is not unexpected, since there were no clear trends between average rating and the dataset features during data analysis. Error values are small because the distribution of ratings is quite narrow, with around 95% of books having a rating between 3.35 and 4.52 (mean +- two standard deviations).

Ordinary least squares **Linear Regression** was not appropriate for this set of data as the algorithm would often diverge. Using feature sets 1 and 2 better results were obtained, but slightly different train/test splits could lead to very different error metrics and sometimes divergence.

Random Forest Regressor did not outperform Ridge and requires much more computing time. It was not used with feature sets 6 and 7, since the computation surpassed 1 hour.

The **Decision Tree Regressor** did not perform well. Calculations using most feature sets led to a R^2 score near zero (or even negative), which means that the model performance was not better than simply assuming the mean as a prediction [8]. Interestingly, predictions improved when the model was provided with only the author.

Feature comparison: The numeric features, number of pages and rating counts, are not sufficient to make any prediction, as seen in feature set 3.

	Linear Regression	Ridge	Decision Tree	Random Forest
MAE	0.203790	0.203790	0.296763	0.219562
MAPE	0.052326	0.052326	0.075721	0.056245
MSE	0.067940	0.067940	0.143253	0.079086
RMSE	0.260654	0.260654	0.378488	0.281222
R^2	0.029752	0.029752	-1.045782	-0.129409

Figure 17 - Error metrics and R^2 score for feature set 3.

The most relevant feature seems to be the **author** of the book. The Ridge Regressor provided good predictions using only main author (set 6), though different splits between train/test data would results in big changes in error metrics.

	Linear Regression	Ridge	Decision Tree
MAE	3.782684e+06	0.170842	0.193679
MAPE	9.549386e+05	0.044299	0.050307
MSE	1.550681e+15	0.052015	0.069266
RMSE	3.937869e+07	0.228069	0.263184
R^2	-2.043380e+16	0.314579	0.087261

Figure 18 - Error metrics and R^2 score for feature set 6.

Predictions with the Ridge Regressor were more consistent when author was combined with numerical features. Combining author with the number of pages (set 4), resulted in higher R^2 score than combining author with ratings count (set 5). It is reasonable that the number of pages has more impact in the ratings than the rating counts the correlation coefficient is higher.

	Linear Regression	Ridge	Decision Tree	Random Forest
MAE	3.782684e+06	0.170842	0.193378	0.180417
MAPE	9.549386e+05	0.044299	0.050230	0.046990
MSE	1.550681e+15	0.052015	0.069121	0.057641
RMSE	3.937869e+07	0.228069	0.262908	0.240086
R^2	-2.043380e+16	0.314579	0.089177	0.240441

Figure 19 - Error metrics and R^2 score for feature set 4.

	Linear Regression	Ridge	Decision Tree	Random Forest
MAE	8.245143e+03	0.178957	0.210432	0.187814
MAPE	2.137505e+03	0.046366	0.054492	0.048799
MSE	4.935260e+09	0.055855	0.077922	0.061276
RMSE	7.025141e+04	0.236336	0.279145	0.247540
R^2	-6.490347e+10	0.265455	-0.024751	0.194160

Figure 20- Error metrics and R^2 score for feature set 5.

By its turn, the publisher, present in sets 1 and set 7, seems to have little impact and increases greatly the number of variables. This result was expected, since, as previously seen, there was little variation in the average rating distribution of each publisher. Therefore, the publisher can be discarded as a feature.

	Linear Regression	Ridge	Decision Tree	Random Forest
MAE	0.177713	0.167360	0.216352	0.180215
MAPE	0.045860	0.043256	0.055818	0.046629
MSE	0.057201	0.049096	0.081404	0.056032
RMSE	0.239168	0.221576	0.285315	0.236712
R^2	0.212692	0.324255	-0.120433	0.228783

Figure 21 - Error metrics and R^2 score for feature set 1.

	Linear Regression	Ridge	Decision Tree
MAE	2.026682e+12	0.178316	0.193212
MAPE	5.180602e+11	0.045995	0.049957
MSE	2.645497e+25	0.055470	0.069307
RMSE	5.143440e+12	0.235520	0.263262
R^2	-3.488017e+26	0.268645	0.086206

Figure 22 - Error metrics and R^2 score for feature set 7.

Conclusion

The Ridge Regressor was found to be the best algorithm for predicting the average rating of books using the Goodreads dataset. Using the main author as a feature was essential for improving the model. Best and consistent results, with minimal number of features, were obtained using main author, number of pages and ratings count. These predictions had a R^2 score of around 30%. The low R^2 is probably due to the lack of a strong correlation between the ratings and the other variables in the dataset. Low R^2 scores are sometimes expected [8], especially in problems involving predicting human behaviour, such as rating a book.

References

[1] Linear Models – Ordinary Least Squares. In: scikit-learn. https://scikit-learn.org/stable/modules/linear_model.html#ordinary-least-squares. Accessed 25 Aug 2023

[2] Linear Models – Ridge Regression and Classification. In: scikit-learn. https://scikit-learn.org/stable/modules/linear_model.html#ridge-regression-and-classification. Accessed 25 Aug 2023.

[3] Decision Trees. In: scikit-learn. https://scikit-learn.org/stable/modules/tree.html. Accessed 25 Aug 2023.

[4] Ensemble Methods – Forests of Randomized Trees. In: https://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees. Accessed 25 Aug 2023.

[5] Metrics and scoring: Quantifying the quality of predictions. In: https://scikit-learn.org/stable/modules/model_evaluation.html#regression-metrics. Accessed 25 Aug 2023.

[6] Pascual C (2023) Tutorial: Understanding linear regression and regression error metrics. In: Dataquest. https://www.dataquest.io/blog/understanding-regression-error-metrics. Accessed 25 Aug 2023.

[7] Pandas – get_dummies. In: https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html. Accessed 25 Aug 2023.

[8] 1. Frost J (2023) How to interpret R-squared in regression analysis. In: Statistics By Jim. https://statisticsbyjim.com/regression/interpret-r-squared-regression. Accessed 25 Aug 2023.