Introduction

Predictive analytics is a very broad set of practices aimed at analysing the data available to a company and making predictions on that data. It uses predictive models, supported by algorithms that learn from the data. Its areas of application are as diverse as the data available: in this case, we will focus on literature.

In this study, we aim to define and train a predictive model for book ratings. In particular, because book ratings are continuous values, the problem we are solving is called a regression model. For this purpose we will use different models, compare them and choose the most optimal one.

This work will be divided into three parts:

- *Data Exploration*
- *Features Engineering*
- *Data Modeling*

Data exploration

For this project we worked with data from the Goodreads website.

During the exploratory analysis, we first conducted a thorough investigation of our database. This involved observing the format of the dataset, seeing if there were any missing or incorrectly filled in dat and choosing the appropriate processing method.

Then we proceded to the preliminary analysis (i.e. to assess the balance of the data) to determine the useful variables according to the information contained in these variables, and to eliminate those that a not very useful for our analysis.

☐ **Preview Dataset **

	bookID	title	authors
0	1	Harry Potter and the Half-Blood Prince (Harry Potter #6)	J.K. Rowlin
1	2	Harry Potter and the Order of the Phoenix (Harry Potter #5)	J.K. Rowlin
2	4	Harry Potter and the Chamber of Secrets (Harry Potter #2)	J.K. Rowlin
3	5	Harry Potter and the Prisoner of Azkaban (Harry Potter #3)	J.K. Rowlin
4	8	Harry Potter Boxed Set Books 1-5 (Harry Potter #1-5)	J.K. Rowlin
5	9	Unauthorized Harry Potter Book Seven News: "Half-Blood Prince" Analysis and Specu	W. Frederic
6	10	Harry Potter Collection (Harry Potter #1-6)	J.K. Rowlin
7	12	The Ultimate Hitchhiker's Guide: Five Complete Novels and One Story (Hitchhiker's G	Douglas Ad
8	13	The Ultimate Hitchhiker's Guide to the Galaxy (Hitchhiker's Guide to the Galaxy #1-5)	Douglas Ad
9	14	The Hitchhiker's Guide to the Galaxy (Hitchhiker's Guide to the Galaxy #1)	Douglas Ad

The dataset has 11123 rows (thus as many book entries) and 12 columns including the target variable (average_rating).

Quantitative variables :

- *Average_rating* : The average rating of the book received in total
- *num_pages*: The number of pages the book contains
- *ratings_count*: The total number of ratings the book received

- *text_reviews_count*: The total number of written text reviews the book received

Qualitative variables:

- *title*: The name under which the book was published.
- *authors*: The names of the authors of the book. Multiple authors are delimited by “/”.
- *language_code*: Indicates the primary language of the book. For instance, “eng” is standard for English.
- *publication_date*: The date the book was published.
- *publisher*: The name of the book publisher.

We also have some other variables for books identification: (*bookID*, *isbn*, *isbn13*).

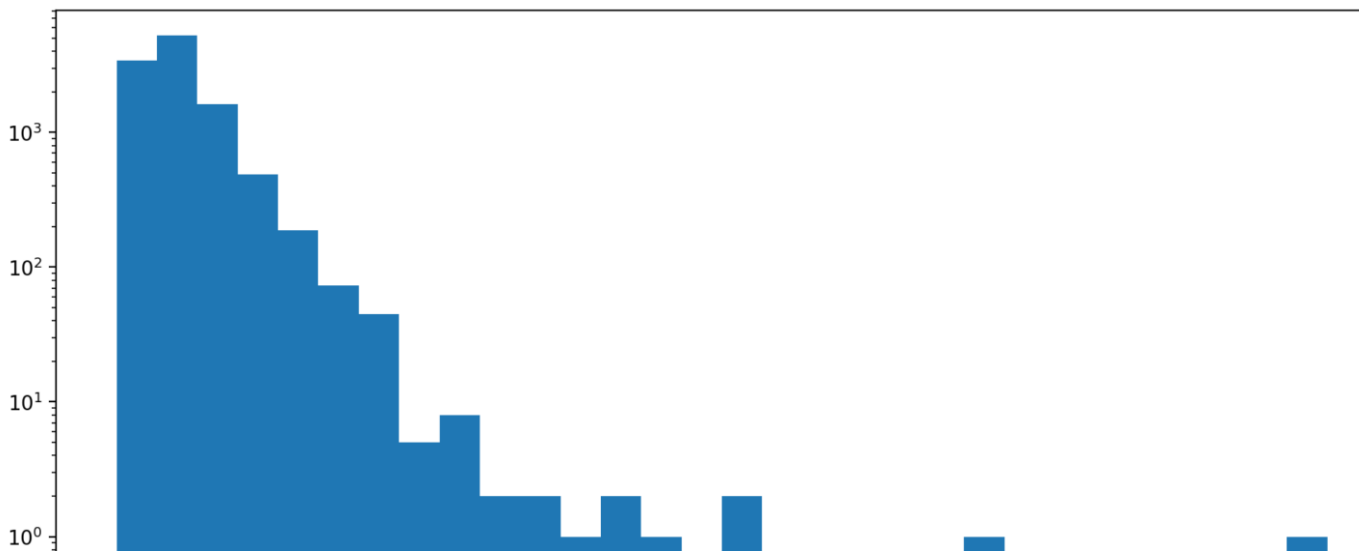
Clean up the dataset

Please note that there are no NA values.

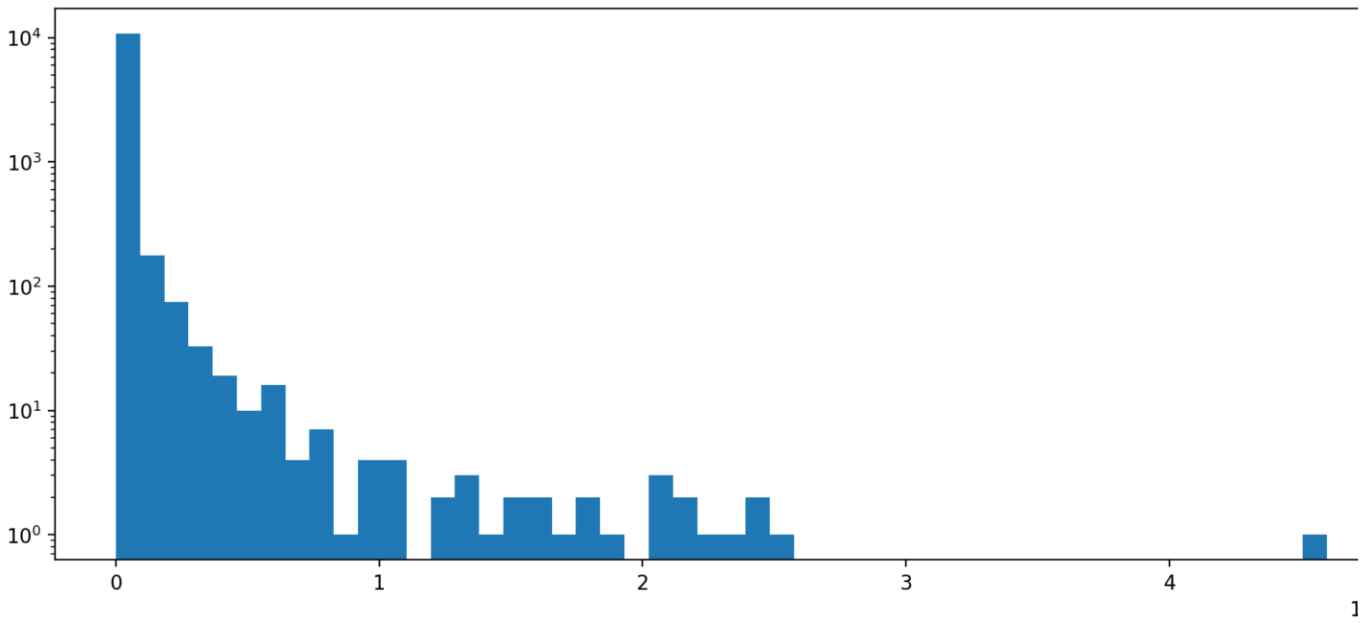
We first plotted the distributions of the main numerical features:

- *number of pages*
- *number of ratings*
- *number of reviews*
- *average rating*

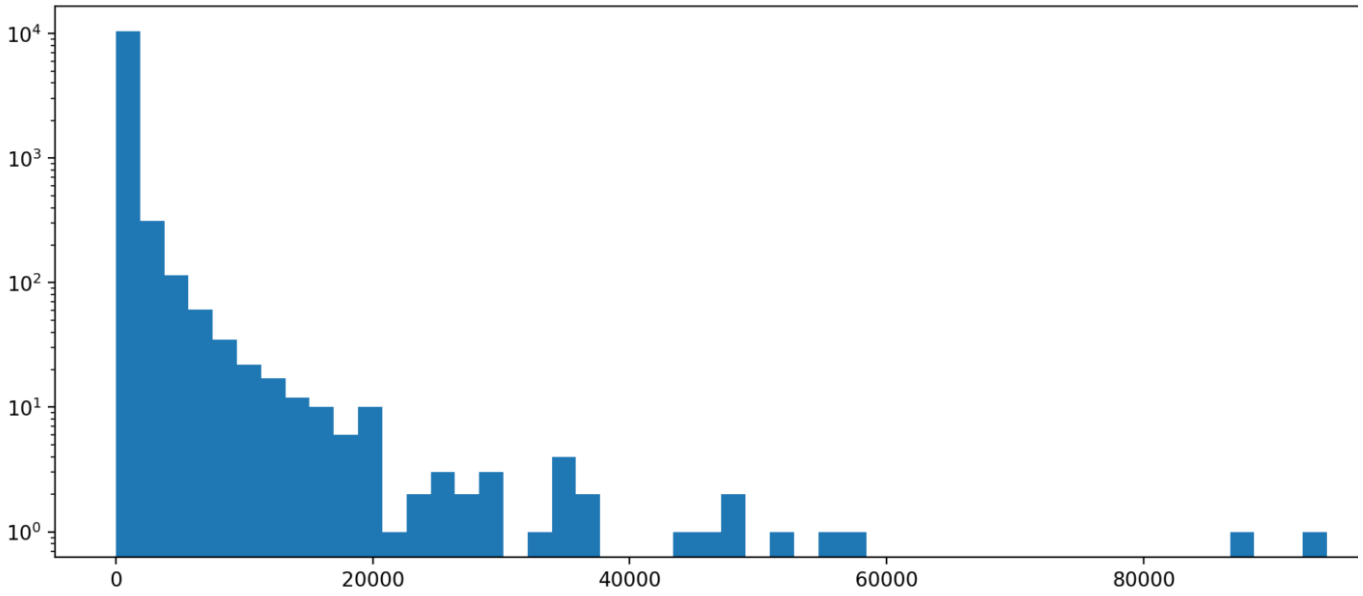
☐ **Show plots**



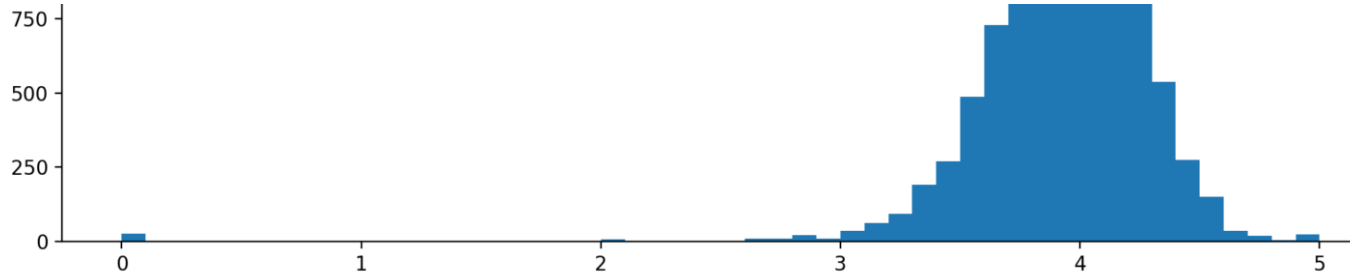
The distribution of number of pages



The distribution of number of ratings



The distribution of number of reviews



The distribution of average rating

We notice immediately that the target feature `average_rating` is skewed towards an average of 4. It resembles a normal distribution centered around 4. This data is inherently imbalanced (almost all ratin between 3 and 5, almost no ratings between 0 and 3).

We have observed that most books have less than 1000 pages. Under 1000 pages the distribution is ove balanced.

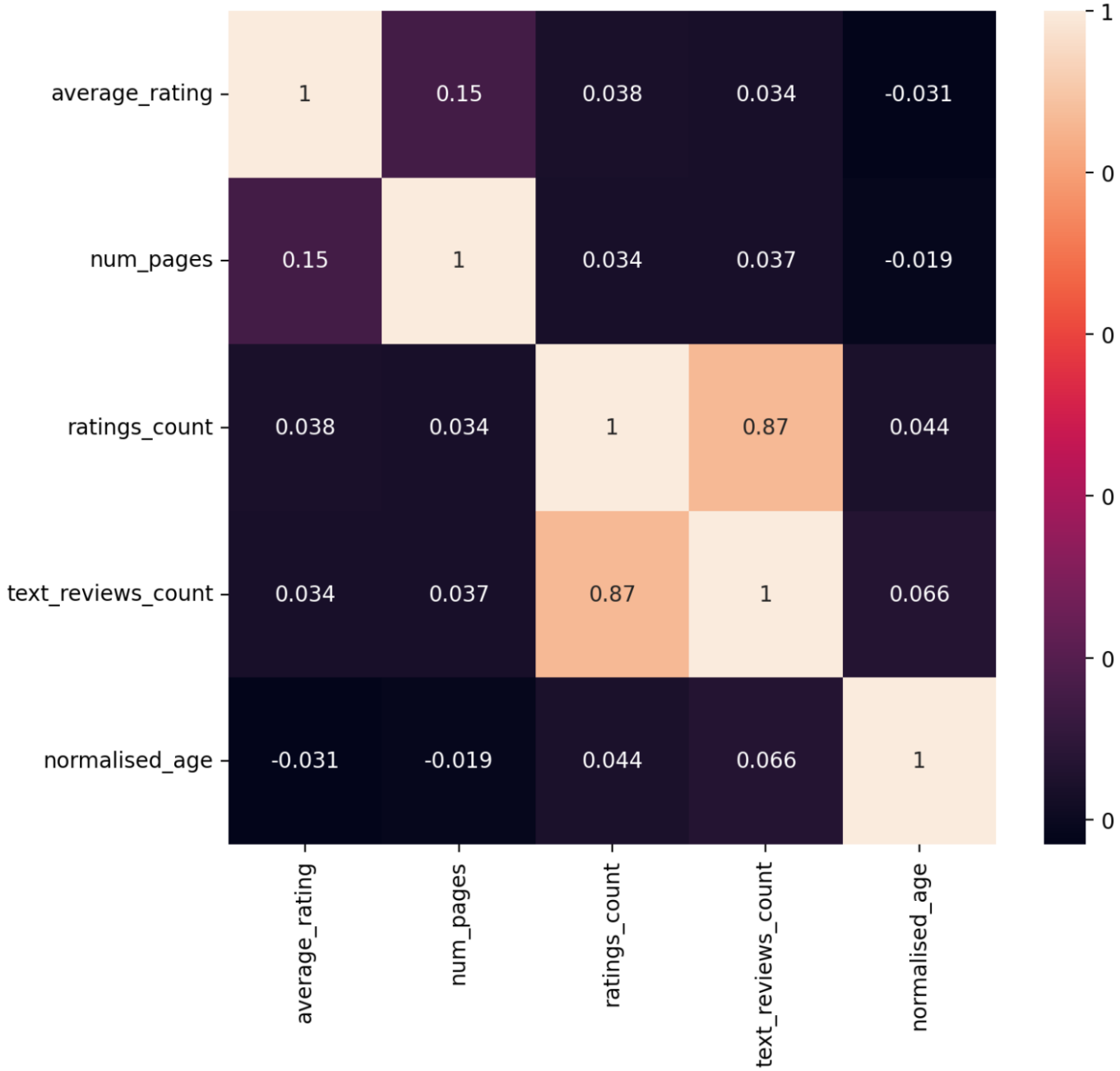
The distribution of number of ratings and reviews are skewed towards 0. This means that generally speaking, most books have fewer numbers of reviews and ratings while a few books have a lot. Such distribution resembles a power law distribution.

We will have to take this into consideration when creating the training and testing sets: these sets shoul both include books with a wide range of average ratings.

Feature Engineering

In this section we focused on leveraging techniques, in order to create new variables that were not verbatim in the original data. This helped us simplify and accelerate data transformations, while improving the model precision. The feature engineering was achieved in three steps.

We first proceeded to study the relationship existing between the selected variables of our dataset and our targeted variable. This was done with the use of a correlation matrix. Of all the chosen variables, we could see that only the number of pages had a better correlation to our targeted variable, average rating



In this way, we could get a first overview of which variables explain the average rating obtained by a boo and more likely help us predict potential average book ratings; nb_ratings, nb_reviews and nb_pages.

1- Defining training variable features

This was achieved through the creation of a function, the *normalise_age* function, that converts a date t number of seconds since a reference time. The function then normalises this number of seconds betwe 0 and 1 values, where 0 corresponds to the oldest book in our dataset and 1 the most recently publishe book. This allowed us carry out an age distribution evaluation. We noticed was that the distribution wa imbalanced, skewing towards more recent books.

2- Evaluating Other Variables

Then we proceeded to analysing the different languages in which were written the books, the book title publishers and authors.

By so doing, it could be observed that most books were written in English, and only *5.24%* of all the boo were written in foreign languages. We then created a binary feature for whether the book was written in english (0), or in a foreign language (1)

There are a total of 6639 unique authors, 2290 unique publishers and 10348 unique titles. This informat would need to be transformed into numerical values if we wanted to use it as features. Thousands of names cannot be converted into quantifiable measures in a realistic manner. Thus, we did not find it interesting to use these variables.

3- Handling Outliers

Outliers are values that are unusual in your dataset and might affect statistical analysis by challenging their presumptions. To avoid this, it is important to handle them prior to any analysis.

The first step was detecting them using a boxplot. Then, we went on to filter the outliers by setting up thresholds (books with more than 2e3 number of pages for instance), after which we observe the numb outliers we have above that scale and delete them. We repeated the same procedure for the ratings cou the text reviews count and the normalised age. Of course, by setting up rules based on what was initiall observed from the boxplot.

At the end we kept 96.27% of data.

Data Modeling

Based on the feature engineering, we decided to keep the following predictor variables:

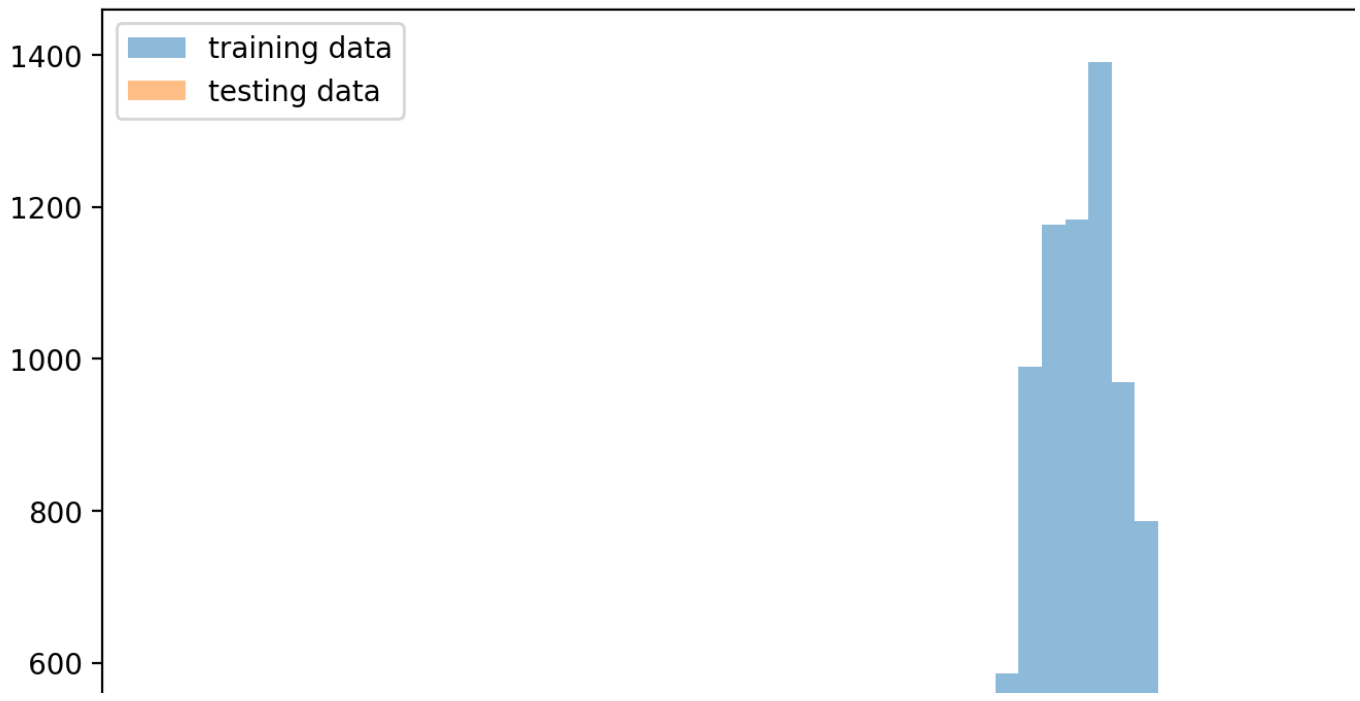
- *num_pages*
- *ratings_count*
- *text_reviews_count*
- *normalized_age*
- *language*

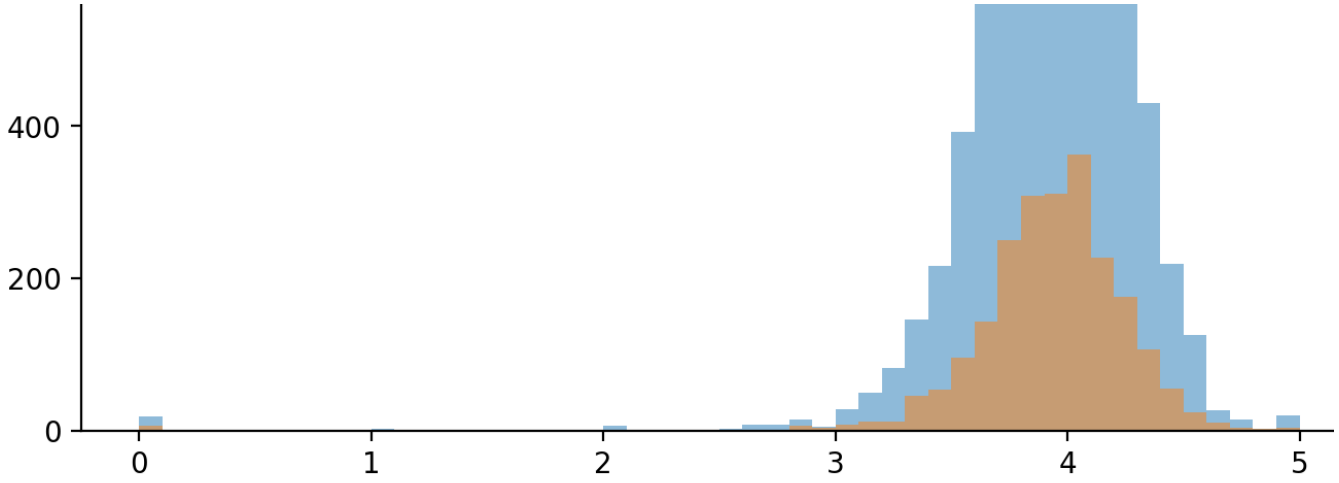
The variable to be predicted is *average_rating*.

1. Data Splitting

Prior to the modeling, we split our data into two subsets; a 20% subset for test and the remaining 80% f training. We then went on to compare the two subsets through a histogram, and we could observe a resemblance between both subsets and the original distribution.

This means that even though the target feature in imbalanced, both the training and testing sets have similar distributions





2. **Model Evaluation**

It is helpful to consider which method to use in the evaluation of our models before we train the model. The ones we used are the Root-Mean-Square Error (RMSE) and the coefficient of determination (R-squa or R^2), both being the most popular metrics for regression models.

Looking at the maximum error, or the prediction that performed the poorest, can also be helpful. With t is mind, we developed a function to evaluate the model to help compare several models. It accepts the actual and expected scores and performs the following actions:

- Compares the prediction of the 20 first books in our test dataframe with the actual values in a bar plot.
- Plots a scatterplot of the predicted ratings as a function of the true ratings.
- Computes, prints, and returns the following metrics: Max error, RMSE and R^2 score

3. **Modeling**

We carried out 7 different models , in order to compare them and evaluate which of the 7 is the most optimal, the 7 models being:

- Linear regression
- Random Forest
- Decision Tree
- Support Vector Regression
- Gradient Boosting
- Adaboost

- StackingCV

Here are the summarised results of each model and for each metric:

	model_name	RMSE	r2_score	max_error
0	Gradient Boosting	0.3342	0.2463	3.7721
1	Adaboost Decision Tree	0.3365	0.2358	2.9721
2	Random Forest	0.3388	0.2253	3.7193
3	Stacking model	0.3454	0.1951	3.8298
4	Decision Tree	0.3507	0.1701	4.1667
5	Linear Regression	0.3806	0.0222	4.0272
6	SVR	0.3889	-0.0206	4.1041
7	Adaboost Linear Regression	0.3966	-0.0613	4.0272

After carrying out the modeling made the following observations:

First, we note the correlation between the RMSE and the R^2 score. In general, the better the R^2 (lower), t better (higher) the RMSE. This suggests that at the very least, these measurements are consistent with c another. Also, we observe that while the R^2 score displays significantly more diverse values, the RMSE scores are often comparable to one another ($0.3 < RMSE < 0.4$).

Secondly, we notice that the Gradient Boosting is the model that performs the best in terms of R^2 score and RMSE. However, the Adaboost Decision Tree model performs the best if minimizing the maximum error is preferred. In other words, compared to models like Gradient Boosting and Random Forest that often perform better, its worst errors are less incorrect.

Conclusion

This project consisted of defining a model for predicting the ratings of a set of books. For this we used tl Bookreads dataset.

We first carried out exploratory analysis, with the aim of getting to know our database, its flaws and the set of variables that were in it. We proceeded to the treatment of missing data, descriptive statistics, visualization of correlation coefficients and the choice of variables for the model.

After that we went on with the feature engineering, in which we did an evaluation of the remaining variables and handled outliers. Finally, we carried out the modeling. At the end of the modeling, it coul be deduced that the Gradient Boost model performed best with respect to the R^2 and RMSE.

Limits:

- Removing outliers simplified the training but seemed to make the prediction more dependant on input data (hint at overfitting)
- r and least squares regression are NOT resistant to outliers.
- Other factors than the predictor variables that are not studied but have an influence on the respons variable may exist. In other words, the data we have doesn't fully explain the results. This was to be expected due to the low values in the correlation matrix.
- A high degree of correlation does not imply causality and effect. Maybe the features our models trained do not have real predictive power and the model is just learning from correlation. If this wa the case, then our model wouldn't be generalisable to make predictions on other data.

Suggestions:

- Employing SMOTE data augmentation due to unbalanced dataset in terms of average_rating.
- Enriching the data by merging fresh data (using ISBN as merging key).
- Using NLP to exploit the dropped characteristics such as book titles.