During the project, we encountered some data consistency errors.
When extracting the file by panda, we noticed that there were errors and 4 lines were missing.

```
Skipping line 3350: expected 12 fields, saw 13
Skipping line 4704: expected 12 fields, saw 13
Skipping line 5879: expected 12 fields, saw 13
Skipping line 8981: expected 12 fields, saw 13
```

After looking at the lines one by one, and comparing the information on the Goodreads website, we concluded that there were human hand errors in the data. There are excess commas that have been removed. This had the effect of shifting the columns and creating new ones. For example:

12224,Streetcar Suburbs: The Process of Growth in Boston 1870-1900,Sam Bass Warner, Jr./Sam B. Warner,3.58,0674842111,9780674842113,en-US,236,61,6,4/20/2004,Harvard University Press
Becomes : 12224,Streetcar Suburbs: The Process of Growth in Boston 1870-1900,Sam Bass Warner-Jr./Sam B. Warner,3.58,0674842111,9780674842113,en-US,236,61,6,4/20/2004,Harvard University Press

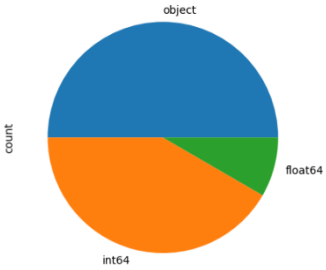
We changed the format of the dates to be able to work on them and new errors appeared which were modified in the new file.

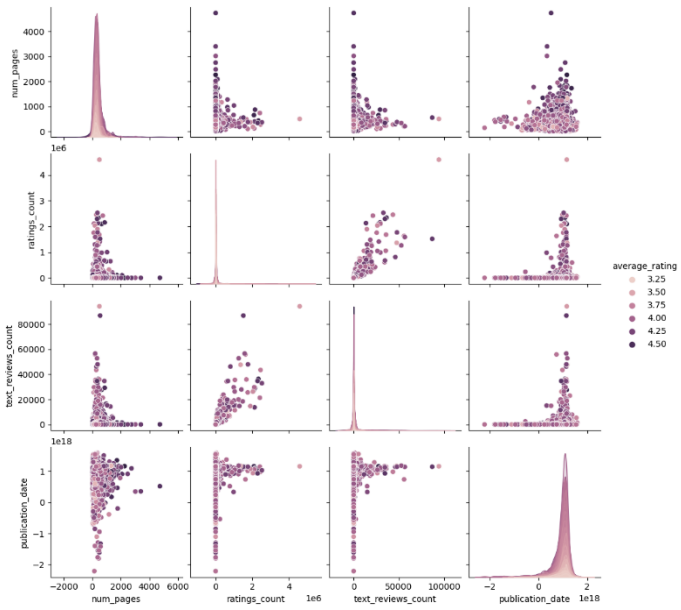
```
ValueError: day is out of range for month, at position 8177. You might want to try:
    - passing `format` if your strings have a consistent format;
    - passing `format='ISO8601'` if your strings are all ISO8601 but not necessarily in exactly the same format;
    - passing `format='mixed'`, and the format will be inferred for each element individually. You might want to use `dayfirst` alongside this.
```

31373,In Pursuit of the Proper Sinner (Inspector Lynley #10),Elizabeth George,4.10,0553575104,9780553575101,eng,718,10608,295,11/31/2000,Bantam Books. H
Here we see that the date is incorrect, November 31 does not exist and we have modified it to November 30 as shown by the Goodreads website.

Thanks to this manual control we manage to reload the file with all the initial lines.

We transformed dates from object to numeric but data repartition remains like this, with half hard usable objects:



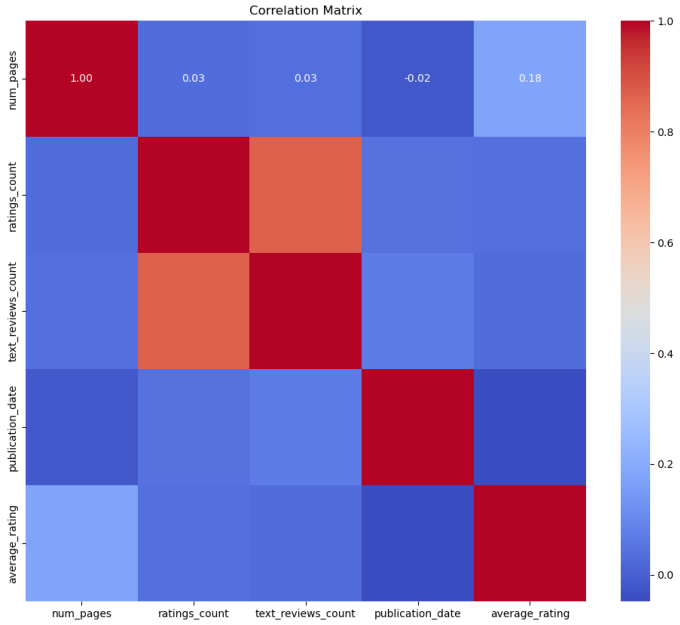


We can see that text_reviews_count/rating_count are highly correlated because they grow together. Ratings_count/num_pages are decorrelated because we can see some L on the graph that is the sign that the variables evolve independently. Num_pages/rating_count and num_pages/text_reviews_count are weakly correlated because they do not do a perfect L.

We are not able to see on this graph that a couple of columns are correlated with average_rating.

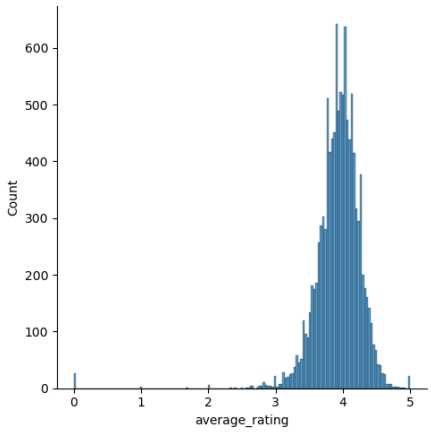
Obviously the number of ratings given is lower for books that were published before the creation of the website. The prediction will probably be less reliable for old publication dates.

ABDOUL BA
EMNA BEN CHAABANE
LAURA LACOMBE
SERGE NYAMSIN



The most interesting line/column is average_rating. The best correlation is with num_pages, then ratings_count, text_reviews_count. Publication_date is decorrelated from average_count but is correlated with text_reviews_count. Using both publication_date and text_reviews_count may help an algorithm to better predict average_rating.

We can see on this graph what the correlation with average_rating looks like. It has a good repartition and is concentrated around 4.

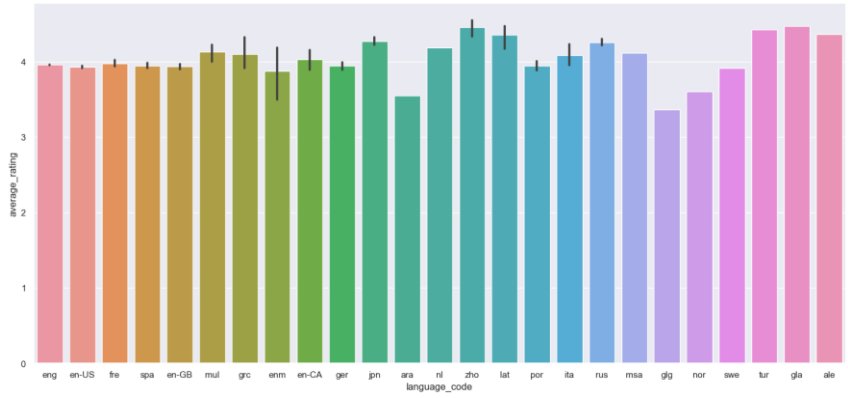


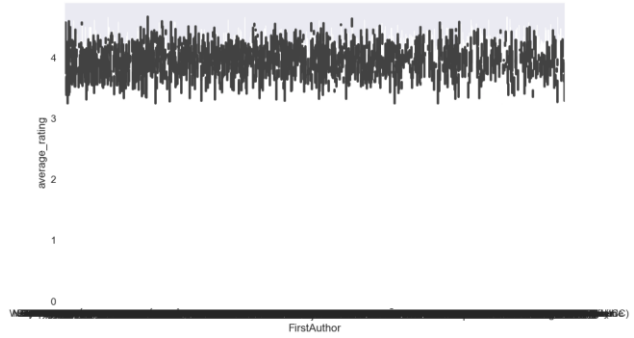
Feature selection:

We don’t use randomly generated columns (bookID, isbn, isbn13) because they can't help us to predict average_rating due to their ramdomness.

The title can't be handled in a machine learning algorithm too so we have to drop it.

We drop the publisher column because it is very hard to process: there are many duplicates publisher with different names like "Avery" and "Avery Publishing Group". But it should be a very interesting input because there is someone from the publisher who has read the book and who has judged its quality to sell it or not. The sale by a publishing house means that they have bet on this book to sell well. Some publisher may want quality or just quantity. We can therefore say that a publisher should normally be well rated by critics or not.





We use the next features:

- num_pages: because we saw in the correlation matrix that it is directly correlated with average_rating

- ratings_count: this criterion allows you to know the enthusiasm surrounding a book. If the book was divisive, this does not predict its rating but in the other case, it is a good indicator. the correlation matrix shows it is moderately correlated with average_rating

- text_reviews_count: same reason as ratings_count but the reader felt the need to comment. This is usually a sign that he either loved or hated the book.

- publication_date: users may have been more or less critical about the books depending on the date and we can imagine that certain books which appeared at the launch of the website may have received fewer ratings than certain books published later. This results in ratings that are less reliable or may have been given by more knowledgeable and therefore more critical readers. It is therefore an interesting criterion. Other ancient authors like Jules Vernes were able to receive notes which arrived on the site in a more constant manner with results on the note which are to be studied. The correlation matrix shows it is not correlated with average_rating and this is normal. But it is correlated with text_reviews_count. Using both publication_date and text_reviews_count may help an algorithm to better predict average_rating.

- authors: They are important because some authors will be preferred and some people will give the book good ratings because the book was written by their favourite author while they would have given a lower rating to someone else. So some books of an author can be "bad" without having a big loss of average_rating. The last figure shows huge differences between authors (FirstAuthor is the authors column without translators) in their received average_rating.

- language_code: The language of the book is very important because it reflects the reader's culture and part of their personality. There may be differences in the way of criticizing or promoting the work of an author

depending on the culture. We can see very clearly in the previous figure that Chinese and Scottish people seem to give good marks to books on average and Galician people give bad marks.

We have chosen to differentiate between English languages because they reflect this cultural change. We see in the graph that the differences in ratings exist between the English even if they are rather small.

Motivations and analysis of the prediction models:

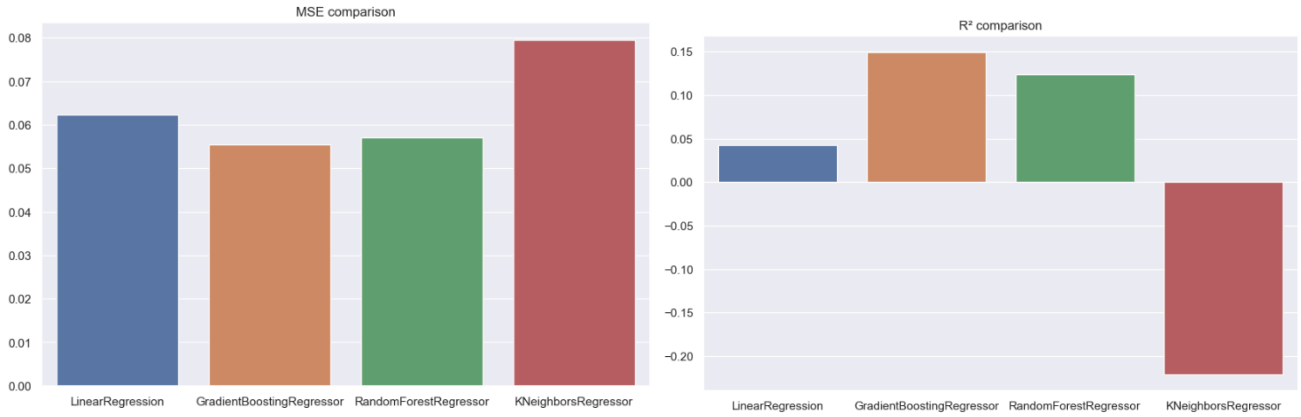
We use some models that are compatible with a variable like average_rating that is linear and not a category.

- LinearRegression: the classic one that we use as a base comparison. It uses a dependent variable, and one or more explanatory variables to form a linear equation estimating the values of the dependent variable. This is our scheme so it could be efficient.

- GradientBoostingRegressor: it is one of the most popular machine learning algorithms and this why we have to try it.

- RandomForestRegressor: it be useful with our data because we have intercorrelated columns and using tree decisions with these informations should have better results than other algorithms.

- KNeighborsRegressor: we have a lot of training data and using nearest neighbours of our known training points could be efficient to predict the test data. This algorithm can use the hard correlation between ratings_count and text_reviews_count but it will suffer from the lack of correlation with average_rating of it.



For the first test we used linear regression to predict the average rating based on quantitative variables such as page count and ratings. The mean square error was equal to 0.06440275632143355.

For the second test we used Gradient Boosting Regressor to predict the average rating built on the other prediction models. The mean square error was equal to 0.05922533682884795.

For the third model we used Random Forest Regressor which is a meta estimator that fits a number of decision tree regressors on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The mean square error was equal to 0.060023240299205236.

For the last one, we used KNeighborsRegressor with a mean square error of 0.08408471248246843. But it had a very bad R^2 that is negative.

Based on the mean square error, the GradientBoostingRegressor is the best model for our prediction model for books ranking. But in all models we tested, the R^2 is bad because it should be near 1.