MACHINE LEARNILNG LABS PROJECT

GROUP 14 : YEINKO TAGO Alex Stéphane

DEENA ZamZam

Project 1 : Book Rating Prediction Model

GitHub : Yeinkal/ML_G14_Average_Rating (github.com)

PROBLEM OBJECTIVE :

This project uses the GoodReads dataset to train a simple Machine Learning model as a book rating predictor that aims to answer the question: "What information determines a book's rating?" It includes a Jupyter Notebook of Anaconda with three sections: 1) exploratory analysis of the data, 2) feature engineering and selection, and 3) model training and evaluation. The project is deployed on GitHub as a complete data pipeline in Python with documentation, which can easily be rerun and reproduced on an updated or similar dataset.

1) DATA ANALYSIS AND FEATURE SELECTING

A - Checking the composition of DataFrame

Before really exploring our data frame we first imported all the packages necessary for carrying out the different operations on the data set. This is to avoid imports at each stage of our program.

After importing the packages we decided to do an overall check of our data set and we proceeded as follows:

- **Check the type of each dataframe attribute:**

We noticed that we have **5 Numerical** type attributes *(average_rating, isbn13, num_pages, ratings_count, text_reviews_count)* and **6 Non-Numeric** type attributes considered as **object** by python (*title, authors, isbn, language-code, publication_date, publisher*)

- Search missing values :

To know what potential transformation we will have to do empty lines laterin our case we do not have **MISSING VALUES.** This is partly due to the fact that we made a small manual transformation on **4 lines** (with the bookID: **12224, 16914, 22128, 34889**) of our DATAFRAME.

- Verification of uniques values by column :

This can be helpful to see how which attribute can be categorize or group by interval. At this stage we can notice that non-numeric variables like **title, authors, isbn, publisher** all have a high number of unique values which shows us that it will be very difficult to consider them as useful variables for training our model.

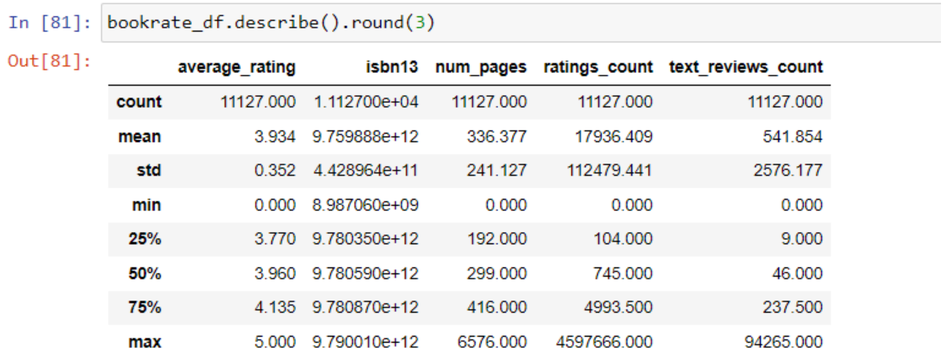
B - Analysis of our DataFrame

In this step, we would like to have a better understanding of the collected data and have better intuition of whether the collected data represents the problem we are trying to solve or not.

To achieve our objective we have analyzed the numerical attributes and the non-numerical attributes separately. To globally explore and see some information like outliers , BIAS or other useful information **.**And we obtained the following results:

- Analysis of numerical attributes

Analysis of numerical attributes

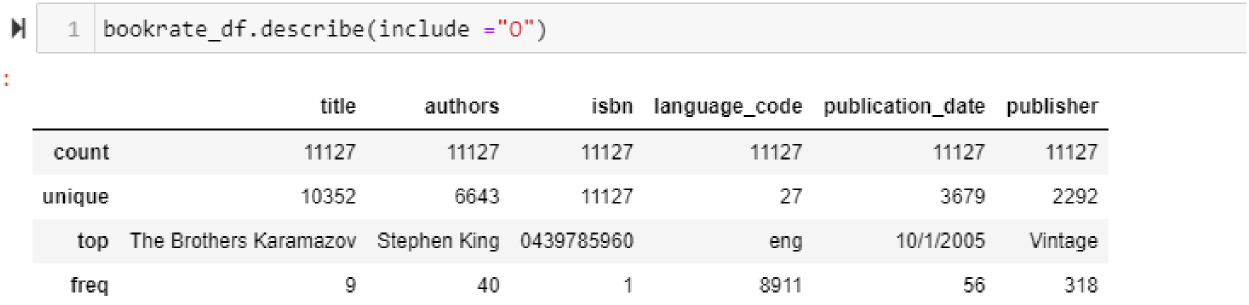


```
In [81]: bookrate_df.describe().round(3)
```

Out[81]:

	average_rating	isbn13	num_pages	ratings_count	text_reviews_count
count	11127.000	1.112700e+04	11127.000	11127.000	11127.000
mean	3.934	9.759888e+12	336.377	17936.409	541.854
std	0.352	4.428964e+11	241.127	112479.441	2576.177
min	0.000	8.987060e+09	0.000	0.000	0.000
25%	3.770	9.780350e+12	192.000	104.000	9.000
50%	3.960	9.780590e+12	299.000	745.000	46.000
75%	4.135	9.780870e+12	416.000	4993.500	237.500
max	5.000	9.790010e+12	6576.000	4597666.000	94265.000

- Analysis of non-Numerical attributes



```
bookrate_df.describe(include ="O")
```

:

	title	authors	isbn	language_code	publication_date	publisher
count	11127	11127	11127	11127	11127	11127
unique	10352	6643	11127	27	3679	2292
top	The Brothers Karamazov	Stephen King	0439785960	eng	10/1/2005	Vintage
freq	9	40	1	8911	56	318

After these global analyzes of numerical and non-numerical attributes we noted that the vast majority of books in our data set are well rated.

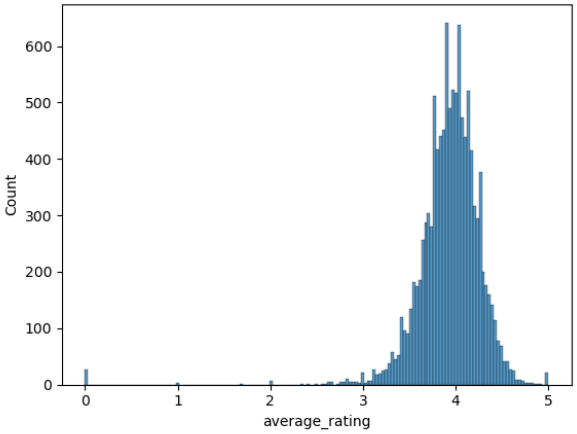
We also observed the data features and get an intuition about which feature could be useful, so we considered the five main feature which are the **number of pages**, **the publication date, rating count, text reviews count** and **language of the book**.

We therefore decided to analyze the average rating to find out which model we could use to predict them for other books. And after finding the relationships that can exist between average rating and the 5 attributes of our intuition

C - Analysis and exploration of Attributes

1) Average_rating analysis

We first decided to highlight the histogram of distribution of averages rating



By closely visualizing the distribution histogram of average_rating we see that the data is concentrated between 3 and less than 5 so whatever the machine learning model used it will be good for designing books that are well rated or which will have higher ratings to 3.

- Try to find a model which will allow us to say whether a book is well rated or poorly rated. based on the average scores. this because we have a fair distribution between the categories. "average rating less than 3.8 = AVERAGE and average rating greater than 3.8 = GOOD.
- Or simply find a Linear regression model.

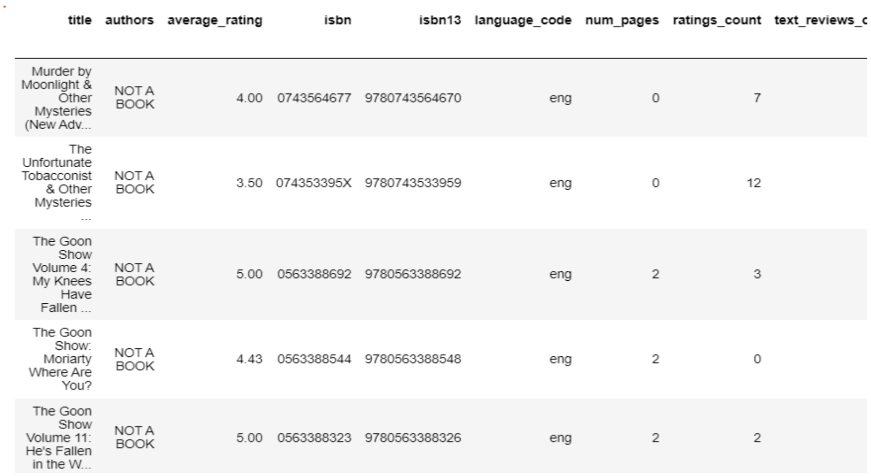
We decided to study the max (average_rating=5) and min (average_rating=0) value of the scores to have more information on this distribution of notes contained between 3 and 5

OBSERVATION

After studying our extremum we notice that:

- We have **"26 books with average rating = 0"** and **"22 books with average rating = 5"**
- **Books with a rating of 0** almost all have no text review and rating counts. this can be explained by the fact that these different books have never been read by any reader or, less likely for us, the information in these books has not been found.
- **Books with a rating of 5** almost all have **rating counts** less than or equal to 5, perhaps because the books have not been read by many readers or for other reasons that we don't know. we also noticed that the **texts reviews** were all less than or equal to zero which can be normal because not all readers who vote are obliged to leave a comment.
- After studying the rating of 5 we noticed that they exists the book who have author's Name **NOT A BOOK** and we decide to observe more about this kind of authors

Finding Autors who have the name NOT A BOOK



title	authors	average_rating	isbn	isbn13	language_code	num_pages	ratings_count	text_reviews_c
Murder by Moonlight & Other Mysteries (New Adv...	NOT A BOOK	4.00	0743564677	9780743564670	eng	0	7	
The Unfortunate Tobacconist & Other Mysteries ...	NOT A BOOK	3.50	074353395X	9780743533959	eng	0	12	
The Goon Show Volume 4: My Knees Have Fallen ...	NOT A BOOK	5.00	0563388692	9780563388692	eng	2	3	
The Goon Show: Moriarty Where Are You?	NOT A BOOK	4.43	0563388544	9780563388548	eng	2	0	
The Goon Show Volume 11: He's Fallen in the W...	NOT A BOOK	5.00	0563388323	9780563388326	eng	2	2	

We realized that in the data set there were 5 books whose author name had the words “**NOT A BOOK**”

DECISION

We decide finally after all these observations **to remove all the book who have Average_rating equal to 5 and 0 of our Dataframe**. Because for us these are outliers which can disturb our Model and given their number (48) we think they could be forget. **We also decided to remove all the 5 books who have "NOT A BOOK"as author names because they are not considered as a books.**

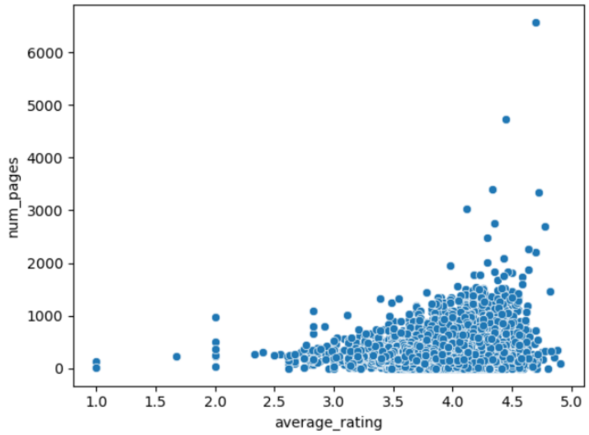
2) Relation between Average_rating and Numerical values

After the global view of average rating .We can therefore find out if the average ratings have correlations with other attributes.

- *Average_rating and Num_pages*

First of all After several tries without success we were forced to make a modification on the columns to prevent the spaces between the columns from creating errors of understanding in python especially at the plotting level.

After this to look for the relationship between average rating and other values we decided to use the scatter plot on all our comparison data.



OBSERVATIONS

- We can see a big problem here because we notice a lot of books that have **0 pages** but have high rating averages. This creates a big misunderstanding at our level and we decide to see all the information related to these books. To decide later whether they are useful in our algorithm or not.
- We see some books which have a huge number of pages more than 2000 pages

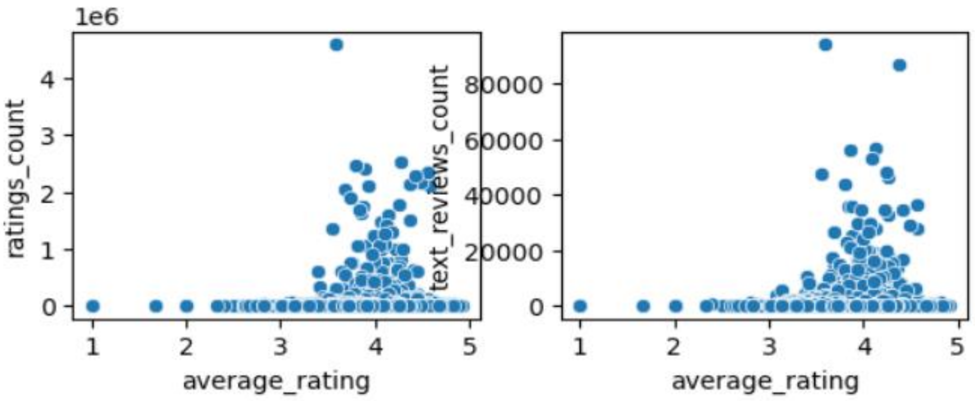
DECISIONS

At the end of the exploration of the Numbers of pages bases on the average rating we decide to :

- Affect to the books who have the numbers of pages less than 30 the number 30 because we have suppose that a book with a relevant number of pages must have a minimum of 30 pages. firstly we think about replace all of them by the mean of number of pages. But for us it seemed illogical to replace a number of pages lower than 30 by a number of pages of more than 330
- Use most of the Books who have more than 2000 pages in our modelling just because they are a symbolic books for us even if they can be a outliers. we have just remove the 2 big one with respectively **"6576"** and **"4736"**

- Average_rating with rating_counts and text review counts

We decided to study the relationship between average rating and rating count and average rating and text review count because we consider that these two attributes have an already existing relationship of proportionality.



OBSERVATION

After the different plotting we can see that we have

- We have a lot of books who have **"0"** as rating count may be because the book was not interresting for the readers . But All of them have the Average rating more or egal than 2 which is illogical.
- We have a lot of books who have **"0"** as text_reviews_count may be because reader didn't have the time to make their opinions about the book. this can be explained by many reason.
- We also see some outliers which can have a bad effect to our model

DECISION

After analysing the rating count we decide :

- To remove the 0 rating count which have the average rating because we didn't undersand the problem who can make this kind of non sense we also decide it because the major part of them **(50 out of 51)** also have "0"as text review count.
- To not remove the 5 values text review more than **"50.000"** or rating count more than **"1.000.000"** even if they can be a outliers values.

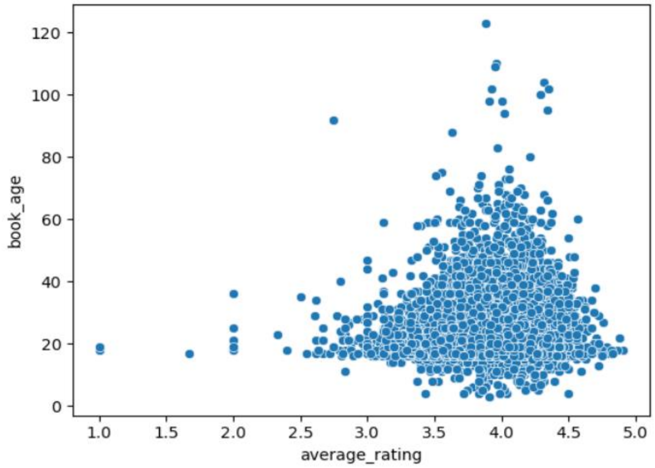
3) Average_rating and Non-Numerical attribute

After having studied the relationship with the numerical attributes we do the same thing. with non-numeric attributes

- Average_rating and publicacion Date

- Firstly we transform the format of the Date to make it more usable and flexible.
- Secondly We decided to transform the **publication_date** into **book_Age** (which is the age of the book in relation to the year 2023) in order to have an integer variable that could be used in our model because we think that this could be a factor that could influence the average rating of a book.

We therefore continued the program with book_age instead of Publication_date.



OBSERVATION

After the different plotting we can see that we have

- Most of the books are between 20 and 60

- We have books which are over 80 years old and which may constitute outliers

- We also noticed that there is a difference of **2 books** each time we count the **book_Age** in relation to the data that we visualize in the description. After several searches we noticed

Book Ages which had **“NaN”** values and we decided to study them and we found 2 book which doesn’t have the book age.

DECISION

The major decision at this level that we took was to assign to these missing book_age values the average age because we thought that a book with a good rating necessarily has a publication date.

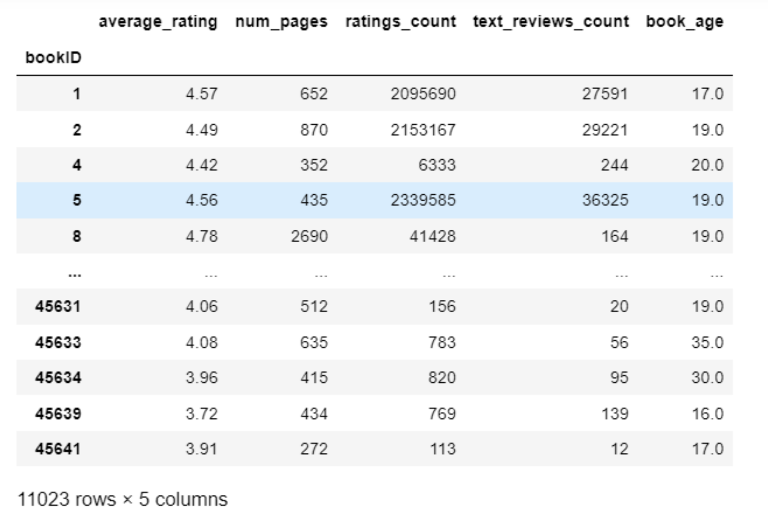
.

D) FEATURE SELECTIONS

In this part we carried out the selection of attributes that we used to train our model. We have already made all the necessary transformations during the data analysis phase to make our task easier at this stage.

At this step, we therefore removed from our dataframe all the data that was not necessary to train our model.

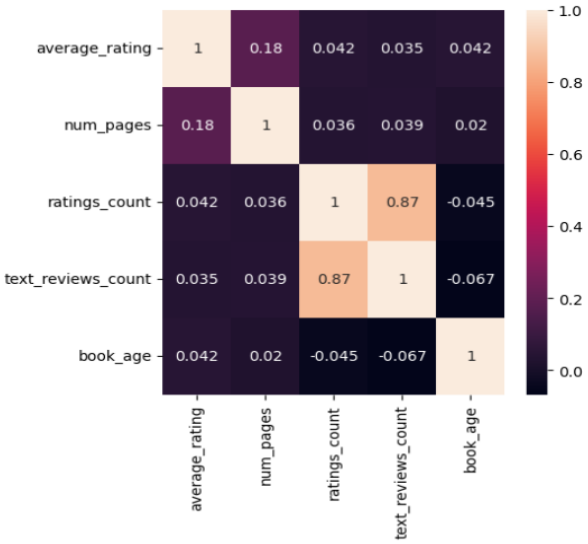
Et nous obtenons le nouveau dataset suivant composé uniquement de nos 5 attributs **average_rating**, **num_pages**, **ratings_count**, **text_review_count** et **book_age**.



	average_rating	num_pages	ratings_count	text_reviews_count	book_age
bookID					
1	4.57	652	2095690	27591	17.0
2	4.49	870	2153167	29221	19.0
4	4.42	352	6333	244	20.0
5	4.56	435	2339585	36325	19.0
8	4.78	2690	41428	164	19.0
...	...	...	...	...	...
45631	4.06	512	156	20	19.0
45633	4.08	635	783	56	35.0
45634	3.96	415	820	95	30.0
45639	3.72	434	769	139	16.0
45641	3.91	272	113	12	17.0

11023 rows × 5 columns

Before moving on to the modeling part we decided to visualize the different correlations that could exist between the values selected through a matrix from Correlation. And we obtained the following matrix.



OBSERVATION

We can clearly see that there is a big correlation between Num_pages and Average_rating.

And for the other attributes we notice that the correlation is not significant but it exist.

3) MODEL TRAINING AND EVALUATION

A)Train the model : Motivation for choosing the machine learning model

Now the data is ready to be fed into the machine learning model. Choosing a suitable training model is a very critical step and it depends on all the previous steps. In general, there are seven criteria that you can select your model on. Briefly, they are the explain ability, in memory Vs out memory training, a number of features and samples, categorical vs numerical features, training time, prediction time, and normality of the data.

Given the fact that we want to determine decimal values we logically had a bias towards linear regression models because we judged that they are better in predicting decimal numerical values.

We also decided not to make a single choice but to take several models compared and based on the results choose the model with the best performance.

According to the Meterics already studies we have narrowed down our options to three Machine learning Models which are:

1- **Random Forest Regressor**

2- **Linear Regression Model**

3- **Decision Tree Regressor Model**

B) Setting Performance Baseline for Machine Learning Models:

A performance baseline provides a reference point against which you can compare the performance of the three models and determine to be able to decide which one had the best performance. Yet the three indicators are:

- **Mean Absolute Error (MAE)**
- **Mean Squared Error (MSE)**
- **Maximum Error (ME)** between the average rating of the test set and the predicted value.

C) Results

After having trained and tested our different models we can say based on the metrics used that the best model to predict our average rating is the Ramdom forest regressor. Because although the metrics show that it is practically as efficient as the linear regression model, it presents a slightly better MAE than the latter.

We know that MAE (mean absolute error) measures the average differences between predicted values and actual values. And the closer the MAE value is to 0, the better.

In our program we have **MAEs** of:

- **0.2134** for Ramdom Forest
- **0.303** for Decision Tree
- **0.2181** for Linear Regression

Based on the definition of a MAE we can say the results are quite satisfactory because the average of the errors for all of our models oscillates between 0.2134 and 0.303. Or even that the difference The average difference between the predicted values and the actual values is quite minimal. Our models, in particular the Ramdom forest, are therefore well adjusted.

FOR MSEs

- **0.0814** for Ramdom Forest
- **0.163** for Decision Tree
- **0.083** for Linear Regression

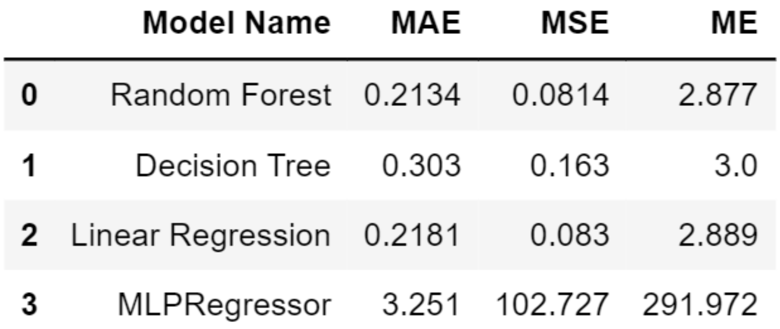
For MEs

- **2.877** for Ramdom Forest
- **3.0** for Decision Tree
- **2.889** for Linear Regression

We can therefore choose as models to use the Ramdom Forest Model and use it because the results of our metrics show that it is the most precise.

Linear Model Comparison Table

All the values of our metrics from the different Models studied are summarized in the comparative table of the models below:



	Model Name	MAE	MSE	ME
0	Random Forest	0.2134	0.0814	2.877
1	Decision Tree	0.303	0.163	3.0
2	Linear Regression	0.2181	0.083	2.889
3	MLPRegressor	3.251	102.727	291.972