Data ScienceTech Institute

PYTHON MACHINE LEARNING PROJECT:

Book's Rating Prediction

Lavanya BASAVARAJU – Josselin CACHET
Mounir M'BARKI – Abdoulkader GASSAMA

Table Of Contents:

1) Project Perimeter

We chose to work on Project 1: Book Rating Prediction Model.

The objective of the project is to process the dataset extracted from a book referencing website so that we can predict the average rating given by users.

Every processing and analyses were made on Python Notebook, we co-worked on a GitHub platform to make progress together and deliver the final version of our notebook.

Our input dataset was a file supplied by the professor untitled "books.csv" to which we brought some small modifications to remove potential trim error in csv conversion.

2) Dataset Cleaning and Analysis

DATASET CLEANING

By trying to import the data in Python, we noticed there were 3 lines wrongly processed by CSV that we cleaned manually.

Fortunately, the data was mostly clean as there were no null values :

```
RangeIndex: 11127 entries, 0 to 11126
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   bookID             11127 non-null  int64
 1   title              11127 non-null  object
 2   authors            11127 non-null  object
 3   average_rating     11127 non-null  float64
 4   isbn               11127 non-null  object
 5   isbn13             11127 non-null  int64
 6   language_code      11127 non-null  object
 7   num_pages          11127 non-null  int64
 8   ratings_count      11127 non-null  int64
 9   text_reviews_count 11127 non-null  int64
 10  publication_date   11127 non-null  object
 11  publisher          11127 non-null  object
```

Looking at the dataset, we defined these fields as potentially useful to predict the rating :

- bookID (kept to identify the book but not for model training)
- title (kept to identify the book but not for model training)

- authors, publisher, language_code, num_pages and publication_date to make potential link between several books / clustering
- average_rating main numerical parameter to evaluate the book that we will try to predict
- ratings_count will help us to understand the pertinence of a rating
- text_reviews_count may be useful to see the commitment of readers for the book

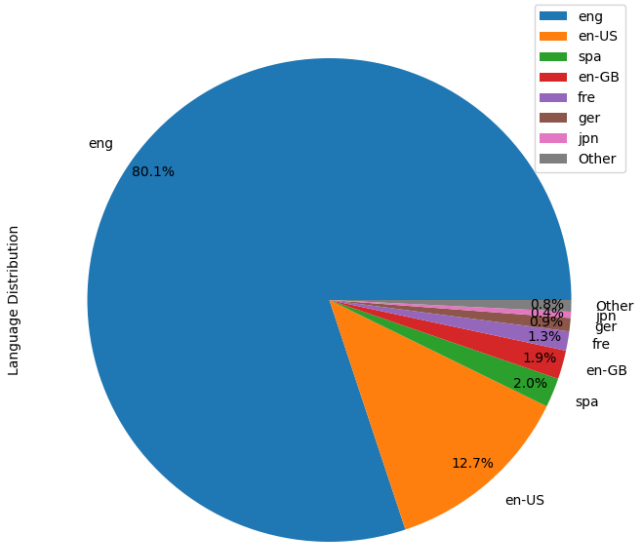
We removed isbn and isbn13 as other identifier were not useful in our study.

For future processing we transformed the publication_date column to be of datetime type instead of strings which permitted to identify 2 wrong dates (date going to 31 in a 30 days month), those 2 lines were removed.

ANALYSIS OF THE DATASET

We plotted several graph to understand the content and distribution of our dataset and make sure there were no additional wrong data.

For example the list of language in the language_code column seemed coherent and the repartition shows that most of the books are in English which is what we expected:

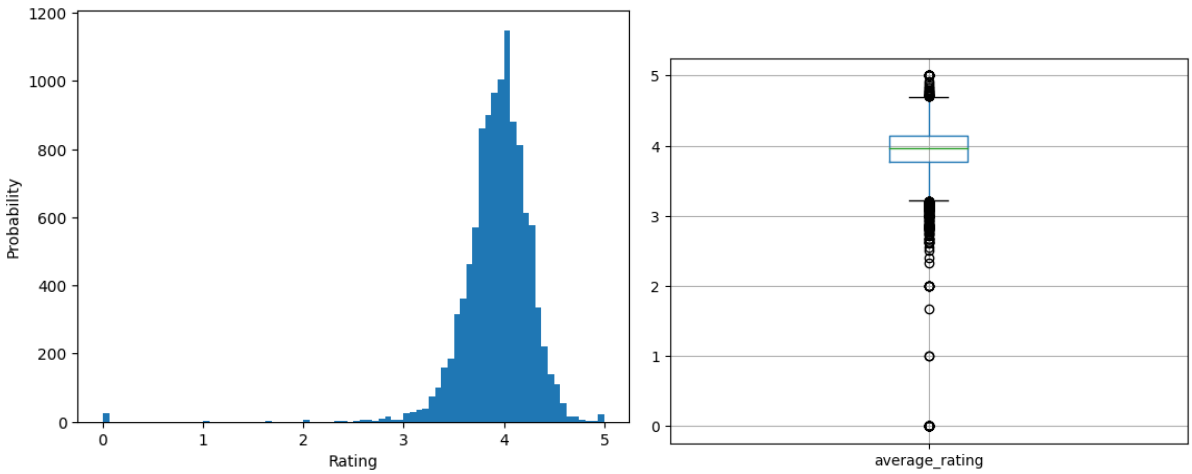


On the numerical values the describe function permits us to verify there is no aberrant values :

	average_rating	num_pages	ratings_count	text_reviews_count
count	11125.000000	11125.000000	1.112500e+04	11125.000000
mean	3.933613	336.315326	1.793868e+04	541.925213
std	0.352473	241.104641	1.124894e+05	2576.402036
min	0.000000	0.000000	0.000000e+00	0.000000
25%	3.770000	192.000000	1.040000e+02	9.000000
50%	3.960000	299.000000	7.450000e+02	46.000000
75%	4.140000	416.000000	4.991000e+03	237.000000
max	5.000000	6576.000000	4.597666e+06	94265.000000

We still noticed that there were 76 books with 0 pages but kept the data as it could be audio books.

The average rating seemed to be normal and following a normal distribution around the mean value 3.93.



Some visualizations permitted us to predict potential groups for our model training, like for example the possibility to regroup the books by publisher as some publisher owns a lot of books by looking at the top 10 publishers :

```
Vintage              318
Penguin Books        261
Penguin Classics     184
Mariner Books        150
Ballantine Books     144
Harper Perennial     112
HarperCollins        112
Pocket Books         111
Bantam               110
VIZ Media LLC         88
```

Finally, by looking at the correlation we wanted to have a first idea of the most important fields and the possible relations between them, however, the results were not that clear and the only obvious correlation was between rating counts and text review count that we expected.

	average_rating	num_pages	ratings_count	text_reviews_count
average_rating	1.000000	0.150763	0.038209	0.033740
num_pages	0.150763	1.000000	0.034387	0.037043
ratings_count	0.038209	0.034387	1.000000	0.865978
text_reviews_count	0.033740	0.037043	0.865978	1.000000

3) Machine Learning Model

Linear Regression on numerical columns

We firstly tried to directly applied a linear regression model on numerical columns it is to say on the number of pages, the rating count and the review count.

However the precision of this model was very low (< 2%) so we tried to exploit more of the original dataset.

Model Testing on Factorized data

To quickly encode our alphabetical data we created a function to factorize such columns :

```
def fac_df(df):
    #factorizes every column in the dataframe, starting with 0.
    df = df.apply(lambda x: pd.factorize(x)[0]+1)
    return df
```

We applied it on Author, publisher, and language_code and tested several model : Linear Regression, Random Forest Regressor and Tree Regressor.

We defined a function to test these models over several iterations and get the average precision of each model which are given below :

	Precision
Linear Regression	0.03
Decision Tree Regressor	-0.67
Random Forest Regressor	0.15

As this was still not enough, we then tried to transform the alphabetical columns into numerical column by processing them in a way that could correlate them to the final average rating.

Model Testing on processed data

For this part we decided that we needed to give more sense to our numerical values if we wanted the model to be more accurate.

In this way we transformed several columns as follow :

- Authors: we computed the average rating given to each author and ranked them over this value to replace "authors" by their rank "author_rank"
- Publisher: Same than for authors
- Language_code : Same than for authors
- Publication_date: we transformed it into a numerical timestamp value so that the valur would be proportional to how old the publication date is

We then reproduced the same model fitting and testing that in the previous part and these are the precision results we obtained:

	Precision
Linear Regression	0.636125
Decision Tree Regressor	0.745390
Random Forest Regressor	0.876133
Logistic Regression	0.017978

4) Analysis of results and conclusion

As there were variations in precision, we tested each model several time to get the mean Precision.

Finally, Random Forest Regressor model gave us the best model with pre-processed data based on a ranking system for alphabetical fields, for an average precision of 86%. Hence, it was the model we chose to deploy for predicting average rating.

A final function "predict()" has been implemented to test the model, to use it just run the function and enter the different parameters asked.

User must keep in mind that for new author / publisher the function is trained to consider it as the lowest rank among all authors and publishers present on our dataset.