Python Machine Learning Labs

Develop an end-to-end Machine Learning Pipeline

Book Rating Prediction Model

**Group 5**

Marius Akre

Vaishnavi Jagtap

Patrick Githendu

Stefanny Peraza

**Instructor**

Hanna Abi Akl

Abstract

#TODO

Keywords: Add keywords here.

Book Rating Prediction Model

The following analysis aims to predict book ratings using different machine learning methods. A model of this type could help predict whether a book would be well or poorly evaluated by the reading community, based on certain characteristics, and in such a way the reader could focus on those that would have better ratings. Although it is not the objective of this analysis, the available data could also be used to make recommendations not only based on the estimated score, but rather based on the characteristics that are most important to the reader. However, it should be noted that the characteristics or variables available for this study are quite limited and correspond only to the books that have been evaluated in the database of Goodreads[[1]](#footnote-2).

## Data

The database contains 12 columns and 11,127 registers. However, of these 12 columns there are four (*bookID, isbn, isbn13, title*), which correspond to book identifiers and are therefore discarded as predictors variables; the *authors* and *publishers* are also not taken as variables on this occasion. Finally, the *average\_rating* variable is the one to be predicted, therefore only 5 possible predictors are available: language\_code, num\_pages, ratings\_count, text\_reviews\_count and publication\_date.

## Data cleaning

During data loading, problems were detected for the correct identification of the columns due to the presence of "," in the names of the authors, and the use of the "," at the same time as separator character of the fields in the file. To solve this problem, the "," in the authors field was replaced by a " -"; this was applied specifically to 4 records that were the only ones showing this problem, thanks to that change it was possible to keep them.

## Exploratory analysis and data processing

In the initial analysis of the data, the structure is checked to ensure that it is in the appropriate format for the analysis. In the case of the predictors "num\_pages", "ratings\_count" and "text\_reviews\_count" it is verified that they are of integer type (*int64*), "language\_code" is a categorical variable and therefore is set to "*object*", and in the case of "publication\_date" incorrect dates were adjusted and date format applied. The idea with this last variable is to analyze whether the year of publication shows any relationship with the scores given by readers. In the case of the variable to be predicted "average\_rating", its type is verified to be *float64*, although it is worth mentioning that it was also categorized, turning it into a dichotomous variable to be able to apply other ML techniques.

Once these changes were made, the distributions of the variables were analyzed, and additional adjustments made. For example, in "language\_code" we observe more than 10 categories, which were regrouped to later analyze if the language in which the book is published has any relationship with the score. Books in English were grouped into a single category, books in French, Spanish, German and Japanese were left untouched, but the rest with lower frequency of appearance in the database were regrouped under a single category of others[[2]](#footnote-3).

Thus, in the different graphs it can be observed that the variables are biased and present extreme values, as well as different magnitudes, which makes it necessary to apply some transformations so that the algorithms are not negatively affected. The graphs of the numerical variables can be seen in annexes xxx.

The first modification is to delimit the variables "num\_pages" and "text\_reviews\_count", applying an upper bound equivalent to three standard deviations, as a method to deal with extreme values. There are many different methodologies that can be implemented, Standard Deviation Method is one of them xxbibliographyx. However, this is not enough, therefore, in order to have variables with slightly more normal distributions and that work better in the regression and classification methods, the *RobustScaler* method is applied, with the parameter *quantile\_range*=(25.0, 75.0) which seeks to reduce the effect of the tails of the distributions in the transformation, centering and scaling the variables based on the defined interquartile range. In the case of categorical variables, *OneHotEncoder* method implemented in the preprocessing class of the *sklearn* library is applied.

On the other hand, the correlation between the variables is also reviewed (annexes xxx), and it is observed that there are two of them highly correlated "ratings\_count" and "text\_reviews\_count". Even though in some methods it does not necessarily affect that the predictors are correlated, in others it can be very problematic, therefore it is decided to eliminate one of the variables ("text\_reviews\_count" is kept since it shows lower extreme values), discarding what seems to be redundant information. In this correlation analysis, it can also be observed that there is no relationship, at least linear, between the variable to be predicted and the numerical predictors, and this is a first sign that it will be difficult to obtain an adequate model to predict the "average\_rating". No other variable preselection is applied, given the small number of them available for this model.

## Methodology

For model calibration, the samples are initially separated into training and test samples, in a ratio of 80%/20% respectively. As mentioned above, the objective is to perform two exercises, one for the prediction of the "average\_rating" in its numerical form and the other for the variable categorized.

In the case of both regression and classification methods, the aim is to create an initial baseline scenario to compare the performance of the methods and thus choose the one that shows the best results in order to subsequently adjust the hyperparameters and try to improve the final model. This comparison is made from a cross validation[[3]](#footnote-4) with the *cross\_val\_score* method implemented in the *sklearn* library, using 5 splits and shuffle the data before splitting into batches (for the cross validation in the classification case, the *RepeatedStratifiedKFold* method is used, which in addition to splits, allows repeating the process, in this case 5 repetitions were used with 5 splits each). In regression, the coefficient of determination of the prediction is used to compare between methods, and in the classification case, the area under the ROC curve.

Thus, in order to predict the "average\_rating" (continuous variable), the following linear regression algorithms were initially tested: Ordinary least squares Linear Regression (LinR), Linear least squares with l2 regularization (RGE) y Linear Model trained with L1 prior as regularizer (aka the Lasso-- LSO), with different Alpha values in both cases, the Linear Support Vector Regression (LSVR), random forest regressor (RFReg) and even transformed data with a *PolynomialFeatures* method was used for applying a LinR on these new variables (polyLR). The documentation on these methods is extensive so no details will be provided in this report (add some bibliography).

The results of the summarized runs are presented below, where it is observed that although the *RFReg* shows better results these do not even reach 10%. The regression models show practically no predictive capacity when using the available variables.

A graph with numbers and lines

Description automatically generated with medium confidence

**Figure 1** Regression algorithm comparison

Given the above, we proceeded with the exercise of trying to predict the "average\_rating" but dichotomized, where 4.25 was chosen as the cut-off point, so that those books with a score equal to or higher than this value were classified as high rating and the rest as low rating. The new objective will be to establish whether, based on the available characteristics, it is possible to estimate whether a book will be ranked with a high score or not. For this exercise, the following methods were applied: Logistic Regression (LR with solver="lbfgs"), Linear Discriminant Analysis (LDA), the k-nearest neighbors vote (KNN), Random Forest Classifier (RF), Decision Tree Classifier (CART), Gaussian Naive Bayes (NB), and C-Support Vector Classification (SVM). It is important to point out that when categorizing the variable to be predicted, an unbalanced data sample was obtained since only 13.5% of the books have a score greater than or equal to 4.25. It was decided not to move the cutoff because it is of interest to evaluate whether a book is really going to be well rated by readers. To mitigate the effect of this imbalance, the parameter *class\_weight* = "balanced" was applied when possible.

A graph with a number of squares

Description automatically generated with medium confidence

**Figure 2** Classification algorithm comparison

## Results

TODO: Xxxx

For future analysis, it could be considered working with the authors column, for example, creating a variable that indicates whether the author has only one book, or more than one in the database. This suggestion is based on the fact that during the analysis of the variables it was observed that authors with only one book show ratings that are more normally distributed (although with some extreme values), while in the case of authors with more than one book in the database show ratings between 3.9 and 4 on average, that meaning high ratings and low variability (in some cases there are lower ratings and in a couple of authors the scores exceed 4.0, but in general terms the distribution is very uniform). That is, for authors with only one book, even when grouped around a mean of 3.9 there is more variability, while for authors with more books it is to be expected that in most cases the scores range from 3.9 to 4, so the characterization of the authors on one or more than one book could help to predict the average rating.

References

Last Name, A. B. (Year). Article Title. Journal Title, Pages #-#. URL. URL.

Annexes

Appendix 1

Number of records per language

|  |  |
| --- | --- |
| Language | Quantity |
| eng | 10,541 |
| fre | 144 |
| ger | 99 |
| jpn | 46 |
| other | 79 |
| spa | 218 |

Appendix 2

Descriptive statistics, numeric variables

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | average\_rating | num\_pages | ratings\_count | text\_reviews\_count | publicationYear |
| mean | 3.9 | 336 | 17,936 | 542 | 2000 |
| std | 0.4 | 241 | 112,479 | 2,576 | 8 |
| min | - | - | - | - | 1900 |
| 25% | 3.8 | 192 | 104 | 9 | 1998 |
| 50% | 4.0 | 299 | 745 | 46 | 2003 |
| 75% | 4.1 | 416 | 4,994 | 238 | 2005 |
| max | 5.0 | 6,576 | 4,597,666 | 94,265 | 2020 |

Appendix 3

Distribution of numerical variables

A group of graphs showing different sizes of numbers

Description automatically generated with medium confidence

Appendix 4

Box-plot of numerical variables

A screenshot of a graph

Description automatically generated

Appendix 5

Box-plot of average rating by language code grouped

A graph with blue and gray squares

Description automatically generated with medium confidence

Appendix 6

Correlation matrix

A graph of a number of rating

Description automatically generated with medium confidence

1. Goodreads: the world’s largest site for readers and book recommendations, launched in January 2007. [↑](#footnote-ref-2)
2. Appendix 2 for further details. [↑](#footnote-ref-3)
3. On training sample. [↑](#footnote-ref-4)