|  |
| --- |
| **Machine Learning with Python Labs – A22**  *Date*: March 26th, 2023  *Professor*: Hanna Abi Akl  *Project contributors*:  Diego Carriel Lopez : [diego.carriel@edu.dsti.institute](mailto:diego.carriel@edu.dsti.institute)  Romain Duhomme : [romain.duhomme@edu.dsti.institute](mailto:romain.duhomme@edu.dsti.institute)  Lan Le : [le.lan@edu.dsti.institute](mailto:le.lan@edu.dsti.institute)  Teresa Graffi : [teresa.graffi@edu.dsti.institute](mailto:teresa.graffi@edu.dsti.institute) |

**PROJECT REPORT**

**Contents**

[1. Overview of the targeted problem 1](#_Toc128036578)

[2. Exploratory Data Analysis and Data Preprocessing 2](#_Toc128036579)

[3. Feature engineering and Model selection 6](#_Toc128036580)

[4. Model training and evaluation 7](#_Toc128036581)

[5. Deployment 7](#_Toc128036582)

[6. Results and recommendations 7](#_Toc128036583)

\*\*\*\*\*

1. **Overview of the targeted problem**

The aim of this machine learning project is to build a model that can help Goodreads users get book recommendations based on authors, book types, and other relevant factors.

We believe that the problem at hand has an average complexity and will require a manageable number of features (independent variables) for the machine learning model.

*Questions to consider before starting the project?* (to be completed)…. NO INFORMATION ON THE COST OF THE BOOK, GENRE

*How we intend to address the problem?* (to be completed)…. GENRE FROM GOOGLE API

*Methodology*

The project is conducted with the following steps:

* **Exploratory data analysis** to understand the dataset and **data pre-processing** to clean and format the dataset to ensure it is in a suitable format for machine learning algorithms.
* **Feature engineering** by creating new features or extracting relevant features from the data to improve the model's predictive performance, and **Model selection** which will depend on the nature of the data and the required predictive accuracy for predicting book ratings.
* **Model training and evaluation**: Train the model on the training dataset, improve its performance and evaluate the performance of the model using the testing dataset.
* **Deployment:** Once the model is trained and evaluated, deploy it to a production environment.

1. **Exploratory Data Analysis and Data Preprocessing**

*About the dataset*

To achieve our objectives, we used a dataset containing approximately 11,000 rows of data scraped from the Goodreads API. This dataset includes customers' ratings of books, as well as other information such as author, book code, number of pages, count of text reviews, publication date, and publisher obtained from the Goodreads website[[1]](#footnote-1).

Here, questions regarding the volume and accuracy of the data arise. While we believe that the collected dataset may be sufficient to build an ML model, it is crucial to ensure that the dataset is representative of the target population and contains a diverse range of books and ratings. The first step in determining the sufficiency of a dataset is to experiment with exploratory data analysis and data preprocessing techniques to validate the dataset.

It's important to note that the dataset used for this project was last updated in May 2019. This may have a significant impact on the accuracy of the machine learning model, as it may not reflect the current distribution of the target variable or account for new patterns or trends that have emerged since the data was collected. To mitigate this risk, we have decided to limit the scope of the project by focusing only on the period before mid-2019 and analyzing long-term trends that can be detected from the available data.

*Observations about data*

In this stage, we will be using Excel and Python to conduct EDA and DP processes.

1. ***Excel***

In Excel, the first step was to analyze the original dataset, which was a CSV file separated by the delimiter *";"*. We noticed that there were some blank spaces before the title *"num\_pages"*, which we removed to ensure the file could be imported correctly in Excel.

The next steps involved correcting the data types. We deleted non-numerical values in the *"rating"* and *"num\_pages"* columns and transformed some numerical and non-text data in the *"language\_code"* and *"publisher"* columns into the appropriate data types.

We noticed that some book *“titles”* appeared multiple times across different rows in the dataset.

We also eliminated columns that were not relevant to the project (such as *“idbook”*).

We corrected the discrepancies between book titles and their corresponding data in some rows by matching them with the correct column.

We decided to address the issue of *“author”* rows containing more than 51 authors by selecting the first author, who is typically the principal writer, and separating them from the other authors.

1. ***Python***

In Python, we installed all necessary libraries and imported the dataset using Pandas.

HOW WE OBTAINED THE GENRE FROM THE GOOGLE API

Then, we checked the data types using the code *"data.info()"* to get an overview of the dataset's dimensions and description. We identified two types of variables: Categorical variables & numerical variables.

Update the picture to see the features from the joined dataset

Text, table

Description automatically generated with medium confidence

After that, we inspected the original dataset by checking the missing values and describing the data.

**Some observations:** We found that the dataset contains 11 123 records with 12 columns. There were no missing values, and the data types seemed coherent. We observed some redundancy in the “*titles*” (10 348 unique values) and “*authors*” (6 639 unique values), with Stephen King having the most works. There were 27 unique “*languages*”, with English books dominating the dataset (8 908 values). There were 2 290 unique “*publishers*”, with Vintage being the most common (318 works). The mean “*average\_rating*” was 3.93, with 25% at 3.77, the median at 3.96, and 75% at 4.14. Ratings were mostly high, and the spread was small (0.35). Skewness seemed low. The mean “*num\_pages*” was 336, with the median at 299, and it exhibited some skewness because of outliers (with the top value of 6 576). The mean “*ratings\_count*” was 1.79e+04, with the median at 7.45e+02, exhibiting strong skewness as the top value was > 1e+6. The mean “*text\_reviews\_count*” was 542, with the median at 47, exhibiting high skewness with a top value of 94 265.  
  
CHANGE THE PICTURE WITH DESCRIBE FOR NUMERICAL ROUNDED AND DESCRIBE FOR CATEGORICAL ROUNDED

Graphical user interface, application

Description automatically generated

* We renamed the columns correctly, converted everything to lowercase and removed blank spaces in the “num\_pages” column name.
* We converted every string to lowercase except for the ISBN since it is an identifier. We also corrected some data types, such as changing the "*publication\_date*" to MM-DD-YYYY format date (since it was assimilated as an object).
* We also inspected duplicates in the dataset and found 812 titles in duplicates. We decided to keep only the observation with the max “*average\_rating*” for each title.
* We decided to deal with the very low value of the number of pages by filling the "*num\_pages*" with the "*max\_pages*" value found in another row with the same book title, with the average pages of the author, with the average pages of the publisher, and with the number of pages found online. Additionally, we removed irrelevant data such as the value "not a book" from the "authors" column, and excluded any books with an average rating or rating count of 0.

Completing this step helped us gain a deeper understanding of the dataset and prepared it for the next steps of feature engineering and model selection.   
DATA VISUALIZATION part adding some of the plots for example the 2nd wordcloud, lineplot of average\_rating by genre… with related comments

1. **Feature engineering and Model selection**

The publication date can provide insight into the context in which the book was written, and books published in the winter season would have different ratings than those published in the summer season. The number of pages can also potentially affect the book rating, as readers may have different preferences for longer or shorter books. Additionally, the number of ratings a book receives can be an indication of its popularity and can help to identify books that are widely read, and so on.

Identifying such patterns or trends can be useful in creating predictive models to identify which books are likely to receive high or low ratings.

Therefore, we created some new features to include in our predictive model:

* For *“****publication\_date****”*, we opted to split the data by season[[2]](#footnote-2) and by century with corresponding numerical indications. The 'season' and 'century' columns could provide valuable insights into temporal patterns in book publishing and help identifying which periods and seasons are most popular for book releases.
* We found ‘***languages’*** with the values of “*en-us*”, “*en-gb*”, and “*en-ca*”, and grouped them into the “*language\_code\_ENgroup*” category. This column will be useful to see which English-speaking countries are most active in publishing books.
* For the *“****rating\_count****”* column, we transformed this into categories with a log function. We defined the bins using a logarithmic scale since the distribution of the “*ratings\_count\_updt*” column was highly skewed, making logarithmic bins a good choice for creating more evenly distributed categories. We then defined the labels for each category as ['very low', 'low', 'medium', 'medium-high', 'high', 'very high'], assigned a number (from 1 to 6) to each category, and created a new column with these category labels.
* Another new feature we created was to transform *“****max\_pages****”* into categories by using a classic splitting method. We decided to define bin ranges as [0, 50, 200, 400, and >= 1000], with corresponding labels of ['very low', 'low', 'medium', 'high', 'very high'].
* We transformed also the *“****average\_rating***” into discrete and more meaningful categories of ['very bad', 'bad', 'medium', 'good', 'very good'] - using a logarithmic function we didn’t use it at the end.
* We identified the ‘***genres’*** using keywords found within the *'Subject\_list'* value. For any NA values, we filled them with *'not specified*'. We grouped all those whose genre couln't be classified into the group *'mixed'.* The 'genre' column in particular seems interesting as it was created by extracting subjects and topics dealt in the book and using them to identify the genres using some key words. This could help grouping and comparing books based on their genres and see which genres are most popular and successful.

We re-verified the dataset. As a result, we obtained a well-prepared dataframe with 10283 observations and 60 columns with no missing values. The dataframe contains a lot of useful information about books and can be used to analyze various aspects of the book industry such as the prediction of book ratings.

Some observations at this stage :

* After removing duplicates, we have 10 283 unique ‘*titles’*;
* In terms of *‘authors’*: we now have 6 252 unique values in the ‘*authors’* column and 4164 unique values in the ‘*main\_author’* column. Stephen King was also the author with the most books in the dataset;
* We observed 27 unique ‘*languages’* in the dataset, with English books dominating (8908 books). We created an additional column to group languages related to English, and now there are 9736 English books in the dataset;
* There were 2262 unique ‘*publishers’*, with vintage being the most common (318 works). Even after processing the data, 'vintage' still appears to be the most frequent publisher with 293 observations;
* We found 31 unique *genres*, with *'novel-narrative'* being the most popular.
* Most books (3002) had a ‘*rating\_count’* that fell under the 'medium high' category, whereas the most frequent category for *'text\_reviews'* was 'low' with 4448 observations. If we consider the total number of reviews, the most popular category is 'medium high' with 3046 observations.
* The difference between *'num\_pages'* and *'max\_pages'* was very low. The most frequent category for both columns was *'medium*,' with 4 865 and 4 897 observations, respectively.
* Finally, with 9384 observations, ‘*good’* was the most frequently occurring *rating category* in the dataset.

We have selected the following parameters to address the challenge of predicting the average ratings of a book: title, main author, rating category, language code, number of pages, categorical ratings count, categorical text reviews count, publication date by season and year, and publisher.

DESCRIBE FEATURE SELECTION WITH RF CLASSIFIER AND OPTIMAL NUMBER OF FEATURES TO INCLUDE IN THE MODEL

1. **Model training and evaluation**

After doing the data cleaning and feature selection. We have decided to test models in order to predict the ratings of a book from different inputs. One of the main transformations that were performed was the transformation of the ‘*rating’* columns into 3 different categories. It would be interesting too to compare models that predict the initial ‘*average\_rating’*, when using SMOTE algorithm in order to balance the data.

First of all, we split the dataset into two parts: 20% for testing and the remaining 80% for training.

In terms of algorithms to choose for prediction of a variable, we have chosen the classic way of fitting data with a Linear Model - Ordinary Least Squares (OLS)[[3]](#footnote-3). It assumes that the relationship between the dependent variable and the independent variables is linear. OLS aims to find the line of best fit that minimizes the sum of the squared errors between the predicted and actual values. This line can then be used to make predictions for new values of the independent variables.

For classification purposes, we have selected the Random Forest & Adaboost algorithms[[4]](#footnote-4). Random Forest is a type of decision tree algorithm that combines multiple decision trees to improve the accuracy of the classification and the final prediction will be based on a majority vote of the individual tree predictions. Adaboost, on the other hand, is a boosting algorithm that combines multiple weak classifiers to create a strong classifier. It is particularly useful when dealing with imbalanced datasets or when the base classifiers are simple.

ADD SOME GRAPHS

1. **Deployment**

(to be completed)

1. **Results and recommendations**

(to be completed)

1. Goodreads is an American social cataloging website and a subsidiary of Amazon that allows individuals to search its database of books, annotations, quotes, and reviews. [↑](#footnote-ref-1)
2. The seasons were defined as follows: Winter (1), Spring (2), Summer (3) and Autumn (4) [↑](#footnote-ref-2)
3. Other options : **Polynomial Regression** (used when the linear model is not sufficient and a polynomial could help), **Elastic Net** (a linear model that takes into account features of both Lasso and Ridge Regressions), **Bayesian Regression** (which uses random variables to obtain a fully probabilistic model, where the output is assumed to be Gaussian distributed around Xw), and **Stochastic Gradient Descent** (useful when the number of features and samples is large). [↑](#footnote-ref-3)
4. For classification purposes, we have Gaussian Naive Bayes, Logistic Regression, AdaBoost, and Random Forest. [↑](#footnote-ref-4)