Data ScienceTech Institute

BOOK RATING PREDICTION MODEL

Developing an end-to-end Machine Learning Pipeline

PYTHON
Machine Learning LAB

A22 Cohort
MSc in Data Analytics

Instructed by:
Professor Hanna Abi Akl

Performed by:
Mahdi Khalsi
Ainur Iagafarov
Leonardo Bertocco

01/03/2023

TABLE OF CONTENTS

TOPIC INTRODUCTION

INTRODUCTION

The project is presented as a Jupiter notebook. The work includes four blocks of the ML Pipeline path: Problem (detected question or problem statement), Data (exploratory data analysis), Features (feature engineering) and Model (model training and evaluation). Imperative way of programming is used. Data source: provided CSV file. The target column (average_rating) is detected for training the model. The dataset provided is a curation of Goodreads books based on real user information. Not everything (just key things) was included in the report. Proper work with necessary comments is located in the Jupyter "**Project-PythonML.ipynb**" file.

PROBLEM STATEMENT

Building a model based on provided data to predict certain book's rating.

Data overview

EXPLORATORY DATA ANALYSIS

The data set is a table which consists of 11123 rows and 12 columns, representing the information related to each book: unique ID, name, authors (multiple authors are delimited by "/"), average rating of the book received in total, International Standard Book Number (ISBN), a 13-digit ISBN, primary language of the book, number of pages, total number of ratings the book received, total number of written text reviews, publication date and name of publisher.
ting, shows that the largest number of books published was in the year 2006.

Data types: Integers, Floating-Point Numbers, Strings.

Figure 1	The highest number of reviews is around the 2000s.
Figure 2	Rated books appeared in the dataset several times.
Figure 3	The average rating of all books is at around 4.
Figure 4	The most popular language used is English.
Figure 5	Top ten most popular books (mostly rated).
Figures 6 & 7	The top ten (highly rated) authors and publishers (respectively) with the highest number of books met in the dataset.
Figure 8	The plotting, shows that the largest number of books published was in the year 2006.
Figure 9	The pie chart showing the most reviewed books.
Figure 10	The extent to which values of each column are linearly related (correlation). Here, we can see that the strongest relations are between text_reviews_count and ratings_count, and between num_pages and average_rating.
Figure 11	Dependencies between publication_year and both average_rating and text_reviews_count.

Feature engineering

DATA PREPARATION

In order to make the dataset ready for training a model, some preparation and data cleaning is performed.

Figure 12 Example of the outlier's detection.

Figure 13 Removing outliers using ratings_count as an example is shown below.

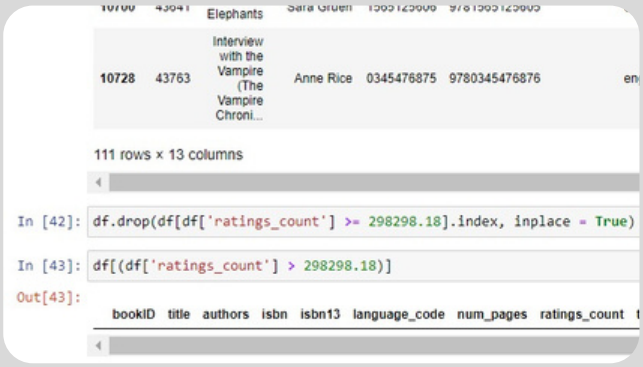
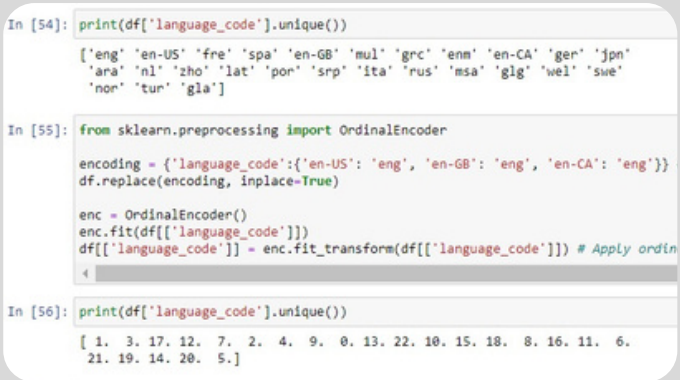


Figure 14 To be able to train the model, the string type of data ('object' in the language_code column) transformed into integers.



MODELING

NORMALIZATION FUNCTION

Normalization usually means scaling a variable to have values between 0 and 1. The process of transforming the columns in a dataset to the same scale is referred to as normalization. It is only required when the ranges of characteristics are different.

CLEAN DATAFRAME

After dropping the last columns, our dataframe is clean and ready to be implemented with a model. We kept only the columns with relevant data for our analysis, such as: **title**, **authors**, **language_code**, **num_pages**, **ratings_count**, **text_reviews_count**, **publisher**.
We also dropped the column "**average_rating**" which will be used as a target value for our prediction model. After this, we split the Dataframe into 70% - 30%. To determine which models meet our issue the best, we tested a variety of them.

SELECTION OF PARAMETERS

Before diving into the description of the different modelling technique that we used, we should spend few words regarding how we optimized the selection of the parameters.
Different models require different parameters, and every parameters need to be set with the best possible values. In order to do so, we combined the results of two functions: **RandomizedSearchCV** and **GridSearchCV**.

MODELS

AdaBoostRegressor Using DecisionTreeRegressor
Linear regression
Ridge
Extra-trees regressor
Random forest regressor
Gradient Boosting for regression
XBG Regressor

We selected many regression models because they fit better our problem of predicting continuous numbers.

METRICS OF PERFORMANCE

R2_score: regression score function.

MSE: The mean squared error (MSE) is largely used as a metric to determine the performance of an algorithm.

Conclusion

COMPARING THE MODELS

To sum up, we compared all the models and their corresponding Test Score in a summary table in the end of our project file. As we can see, the "**XGB Regressor**" model is the most accurate for our problem statement.

	Model	Test Score
0	AdaBoost Regression	3.3439571749784736
1	Linear Regression	2.2628389144321748
2	Ridge Regression	2.2690707582287306
3	ExtraTrees Regression	17.646239640237017
4	Random Forest Regression	18.354814101914986
5	XGB Regression	23.08333527007729