Project Report

For readers books are a multi-hour commitment of learning and leisure while from the author's and publisher's perspectives, books are mostly a way of living. In both cases, knowing which factors explain and predict great books will save both time and money. Due to different people have different tastes and values, finding out how a book is rated in general is a great starting point.

Subject: Machine Learning Prediction of the Book's Rating using Python

Group Members: Mohamed Hamiche, Vincent Lamirault, Tetiana Shchudla

Cohort: S23

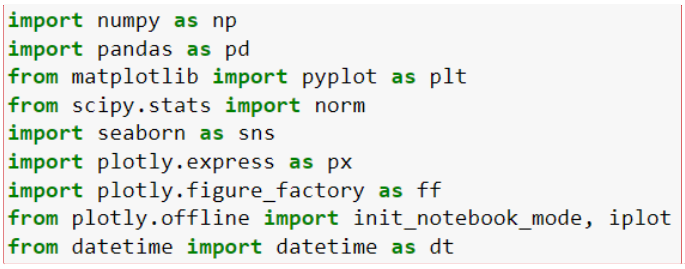
Date: 31.08.2023

Dataset: Books CSV file – a curation of Goodreads books based on real user information (contains 11 127 book ids). Additionally, we decided to scrape some data from the Goodreads website to extend it by adding 5 new columns.

Objective: Using the provided dataset and scraped data to train a model that predicts a book's rating

Environment, packages, and libraries

We worked in a virtual environment using Python 3.9 by importing the list of packages and libraries as mentioned below:



```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from scipy.stats import norm
import seaborn as sns
import plotly.express as px
import plotly.figure_factory as ff
from plotly.offline import init_notebook_mode, iplot
from datetime import datetime as dt
```

Commented [TS1]: Should be modified after we finish the notebook

Remark

You also can reach our works on the following Github repository :

https://github.com/mohamed06H/dsti-ml-book-ratings

Below is the information regarding the dataset attributes we have (both provided and scraped one):

	Variable		Description
provided	1	bookID	A unique identification number for each book.
	2	title	The name under which the book was published.
	3	authors	The names of the authors of the book. Multiple authors are delimited by "/".
	4	average_rating	The average rating of the book received in total.
	5	isbn	Another unique number to identify the book, known as the International Standard Book Number.
	6	isbn13	A 13-digit ISBN to identify the book, instead of the standard 11-digit ISBN.
	7	language_code	Indicates the primary language of the book. For instance, "eng" is standard for English.
	8	num_pages	The number of pages the book contains.
	9	ratings_count	The total number of ratings the book received.
	10	text_reviews_count	The total number of written text reviews the book received.
	11	publication_date	The date the book was published.
	12	publisher	The name of the book publisher.
scraped	13	quotes	The total number of quotes from the book.
	14	discussions	The number of readers` discussions about the book on the site.
	15	questions	Overall number of questions asked by readers on the site.
	16	written_books	The total number of author's written books.
	17	followers	The quantity of author`s followers on the Goodbooks website.

Introduction :

Our main challenge is to build a set of data strong enough to keep the continuous aspect of our target value (average_rating of a book, with two decimals). Our advantage is that we have many titles in the original file (more than 11.000). However, some provided features can seem of too little or no help.

As we want to be as accurate as possible, we've chosen to scrape some more numerical features (number of books written by the author, number of followers of the author, number of quotes / discussions / questions about the book).

At that point, we project using some specific model adapted for continuous values.

First steps

We began with some brainstorming to check the provided features. We agreed that we would surely drop some (isbn13 for sure for instance), but decided to keep the dataset as a whole until we would reach the preparation phase.

Meanwhile we started to scrape datas. It took several hours scraping and rescraping missing values until almost each rows were filled. Honestly, there are some datas we would have liked to scrape but were not able to reach technically (some books figure in the 'all_time_favorite' in bookshelves ; some books have received 'literary_awards'…)

During the time we scraped datas, we all began to check how to clean information.

Dealing with features

This part really was intense and time consuming. After we scraped, we had about 15 features to clean, transform, plot and so on.

This also leads to questions needing decisions to be taken (for instance : 'authors' contains information we want to keep, but some rows give 40 different authors ; should we keep all the names, but then we would have many features with a lot of empty values? shall we keep only the first one? Shall we count the number of contributors?)

There are too many questions to appear in this report, but they are treated inside the notebook provided. We had no real technical issue on how to work on this, our decisions mainly came from reflexion and plottings.

Modeling

We made a point verifying that the notebook was running well until this.

Then we had to process each feature so that they could fit into models. At the very beginning, some of us had tested models more adapted to classification, which lead to very poor results. When downgrading the accuracy of the target value (rounding to nearest .5 value), results were good, but we wouldn't have been very relevant, since most of the books would have been rated under only 3 or 4 different ratings (3.5; 4; 4.5).

Then we processed the data in a way we could use it in a model adapted for continuous target values (Linear regression, KNeighborsRegressor).

Results

Unfortunately, we did not manage to produce any convincing model. We even are far from acceptable average result.

The fact is that we surely have to work again on our features. Seeking for more outliers, better categorizing some others, and maybe try a less demanding way of rating (maybe with one decimal instead of two).

At that point, we still have ideas, but our time ran off.

By the way, we've learned a lot, this is a positive result