Assignment 2

Description

The goal of this project is for you to get familiar with data cleaning, schema matching, and data matching.

I. PART 1

This report outlines the **schema matching and data standardization** process for two datasets:

1. **tableA.csv** (BooksToScrape)
2. **tableB.csv** (OpenLibrary)

The objective is to ensure both tables have consistent attributes to enable further analysis and data cleaning.

Schema Analysis:

```
PS C:\Users\ADMIN\Desktop\HW> python processing/schema_matching.py

 🔹 Schema of Table A:
ID        object
Title     object
Price     object
Rating    object
Stock     object
dtype: object

 🔹 Schema of Table B:
ID         object
Title      object
Author     object
ISBN      float64
dtype: object

✅ Common Attributes (Same in Both Tables): ['ID', 'Title']
❌ Extra Attributes in Table A (Missing in Table B): ['Stock', 'Price', 'Rating']
❌ Extra Attributes in Table B (Missing in Table A): ['Author', 'ISBN']
```

Observations:

- The only common attributes between both tables were ID, Title and Author.
- Table A contained Price, Rating, and Stock, which were missing from Table B.
- Table B contained ISBN, which was missing from Table A.

II. PART 2

In this phase, I focused on selecting a standardized set of **attributes (S)** from **table A** and **table B**. The objective was to ensure that both tables contained the **same attributes**, making them suitable for further data cleaning and analysis.

After processing clean_tableA.csv and clean_tableB.csv, I identified the following attributes:

Table A - BooksToScrape

Attribute	Description
ID	Unique identifier for each book
Title	Book title
Price	Price of the book
Rating	Book rating (out of 5 stars)
Stock	Availability status (In stock / Out of stock)
Author	Name of the book's author

Table B - OpenLibrary

Attribute	Description
ID	Unique identifier (different from Table A)
Title	Book title
Author	Name of the book's author

Observations:

- Both tables contained ID, Title and Author as common attributes.
- Table A included Price, Rating, and Stock, which were missing from Table B.

=> S has only three attributes: ID, Title and Author.

III. PART 3

- **Missing values: report the percentage of missing values for X. For example, if there are 20 tuples in Table A, but X has value in only 5 of them, then the percentage of missing values for X in Table A is 75%. (Report both the fraction and the percentage.)**

```
PS C:\Users\ADMIN\Desktop\HW> python processing/missing_values_analysis.py

 🔹 Missing Values Report:
        Missing Count Missing Fraction Missing Percentage
ID                  0            0.00              0.00%
Title               0            0.00              0.00%
Price               0            0.00              0.00%
Rating              0            0.00              0.00%
Stock               0            0.00              0.00%
Author            125            0.12             12.50%

✅ Missing values report saved.
```

Observations

1. No Missing Values for ID, Title, Price, Rating, and Stock
   - These attributes are **fully available**, ensuring unique book identification and basic book properties remain intact.
2. **Author Has 12.50% Missing Values**
   - 125 out of 1,000 records are missing Author names.
   - "Unknown" values were considered **missing**, as they do not provide useful information.

- **You often have to fill in these missing values somehow for machine learning in subsequent steps. Discuss solutions you may use to fill in these missing values (you don't have to fill in these values; I'm only asking for a discussion of possible solutions).**

Possible solution:

1 - Fill Missing Author Values Using External Data.

- Use API to search for missing authors by title.

2 - Use Fuzzy Matching from Table B.

- Use fuzzy title matching to find books in Table B that have authors.
- If a close match is found, copy the author name table A.

3 - Use a Default Placeholder ("Unknown Author")

- Replace missing values with "Unknown Author".

4 - Drop Rows with Missing Author Values (very risky so I'm not recommended this)

- Remove records where Author is missing.

- **Classify the attribute X as numeric, textual, categorical, or boolean. If you can't classify, discuss why (e.g., an attribute has values 1, 3, and medium, so it's neither numeric nor categorical).**

In this part, I have analyzed and classified each attribute in the csv file as Numeric, Textual, or Categorical. Proper classification helps in data preprocessing, feature selection, and machine learning model training.

```
PS C:\Users\ADMIN\Desktop\HW> python processing/attribute_classification.py

 Attribute Classification:
   Attribute        Type
0         ID     Textual
1      Title     Textual
2      Price     Numeric
3     Rating  Categorical
4      Stock  Categorical
5     Author     Textual

Attribute classification report saved.
```

Textual Attributes

- **ID** : Stored as text, even though it contains numbers. IDs should not be used for calculations.
    - ID is a unique identifier, not a measurable value.

    - Example: "Book_001" and "Book_002" are different books, but arithmetic operations don't make sense on these values.
- **Title** : Free-text book title, varies in length. Used for search and categorization.
- **Author** : Names of authors, often containing multiple words or names.

Numeric Attributes

- **Price** : Converted to Numeric after removing £ currency symbols.
    - Used for price analysis, trend detection, and machine learning models.

Categorical Attributes

- **Rating** : Although ratings are numbers (e.g., 1-5), they are not continuous values; instead, they are ordinal categories.
    - Ratings are not continuous numerical values.
    - Example: "One Star", "Two Stars" : These are better treated as categories, not numerical values.
- **Stock** : Has limited values ("In stock", "Out of stock"), making it a categorical variable.

- **If the attribute X is textual, report the average length of its values, report the minimal and the maximal length of its values (length is measured in the number of characters in the value).**

```
Text Attribute Length Report:
        Min Length  Max Length  Avg Length
ID               6           9        7.89
Title            2         205       39.12
Author           5          38       12.79

Text length analysis completed.
```

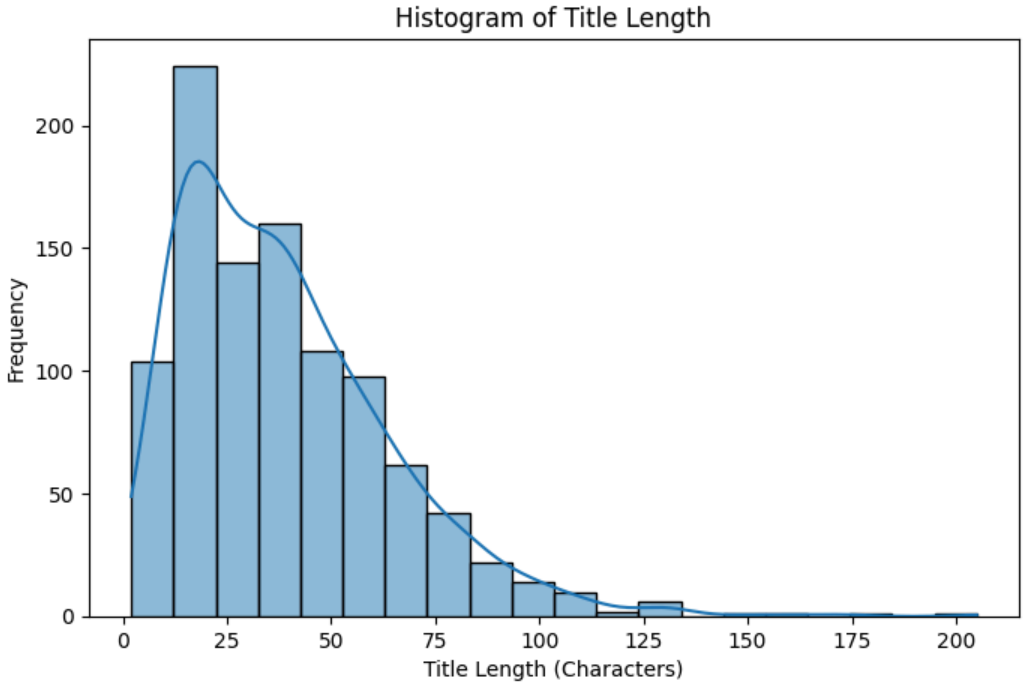
ID (Min: 6, Max: 9, Avg: 7.89)

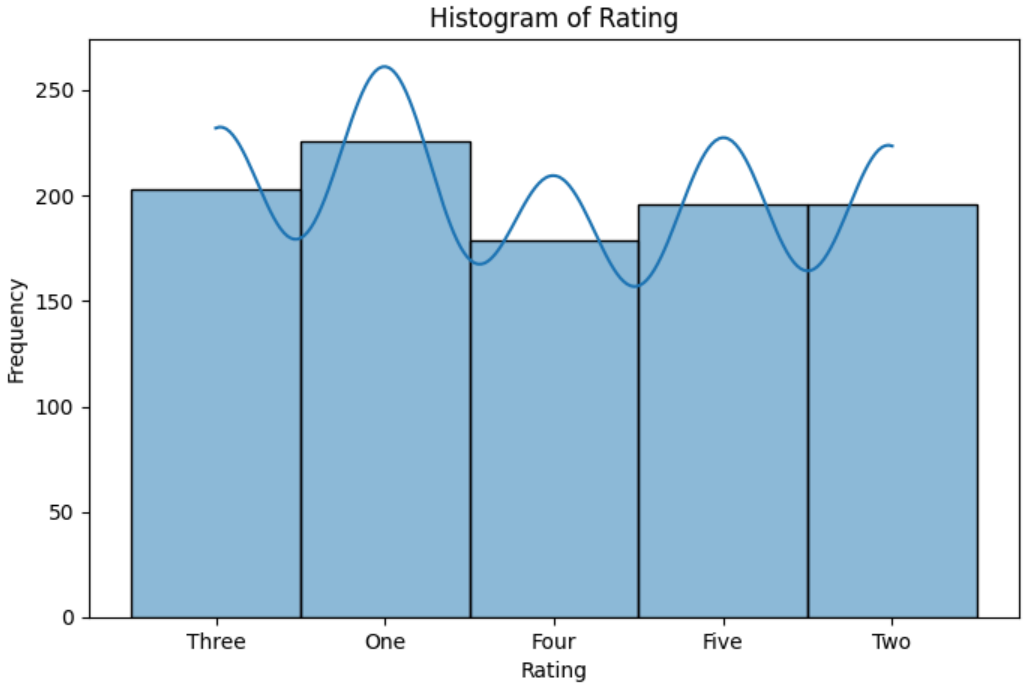
- The ID field has a consistent length range (6-9 characters).
- This suggests a uniform ID structure, likely following a specific format (e.g., Book_001).
- Since it's not used for analysis, its length does not impact data quality.

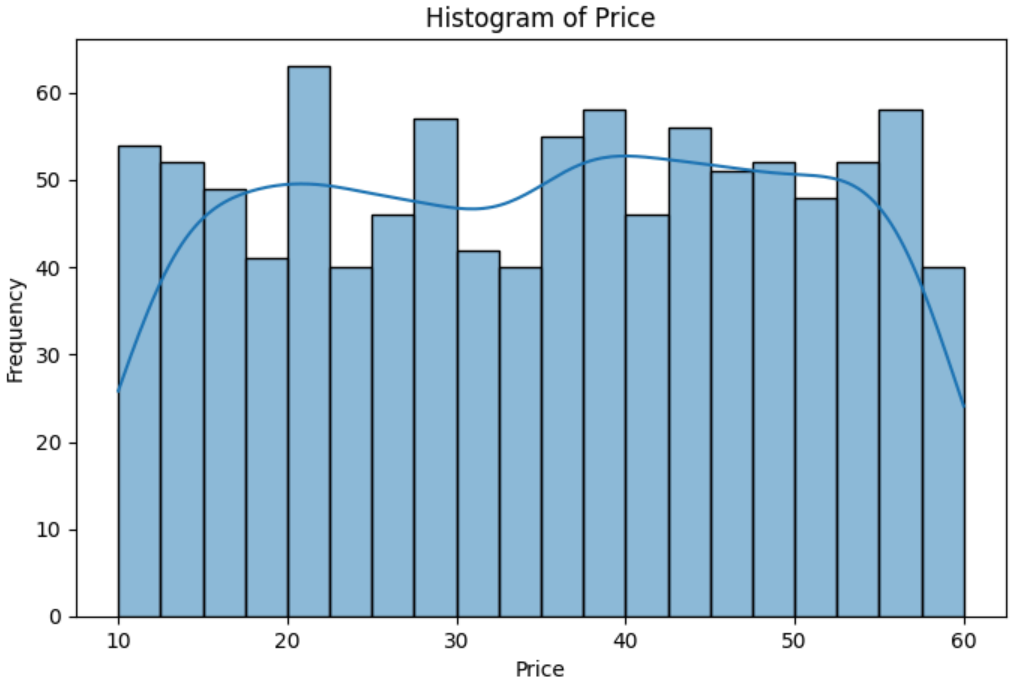
Title (Min: 2, Max: 205, Avg: 39.12)

Author (Min: 5, Max: 38, Avg: 12.79)

- **Find and report possible outliers and anomalies among the attribute values. For example, if attribute price typically has values in the range $1-20, and then there is a value of $200, then this value is an outlier and can also be an anomaly (that is, an incorrect value in this case). You can detect outliers by creating a histogram on some property of the attribute values. For example, a histogram on price values will help detect price outliers. As another example, if an attribute is textual, then a histogram on the length of the values (as measured by the number of characters in the value) can help detect very long or short values. Show at least two histograms that you have created(I am not asking for two histograms per attribute; I am just asking for at least two histograms; they could be for two attributes X and Y, for example).**


Histogram of Title Length
Frequency
Title Length (Characters)
0
50
100
150
200
0
25
50
75
100
125
150
175
200


Histogram of Rating
Frequency
Rating
0
50
100
150
200
250
Three
One
Four
Five
Two



- If the attribute value is supposed to follow a certain format (e.g., dates), then discuss if all values follow the same format or if there is some problem with the format, and we will have to standardize the formats later.

- **Are there synonyms among attribute values? For example, an attribute "book-type" may have values "softcover" and "paperback", which are synonyms**.

The scraped data from BooksToScrape and OpenLibrary already follows consistent naming conventions; there is no variation in values.
The most common synonym issues arise in book formats (hardcover, paperback), but my dataset only contains ID, Title, Author, Price, Rating (Numeric), and Stock, the likelihood of synonyms is very rare.

- **Sometimes, attribute values are "sprinkled" all over the item. For example, a book may have an attribute "publisher", but its value is missing.**

Instead, the book title contains the publisher (e.g., "Principles of Data Integration by Springer"). Do you have this problem with this attribute?

Yes, I identified at least one case where the Author was embedded in the Title.

```
PS C:\Users\ADMIN\Desktop\HW> python processing/sprinkled_values_check.py

 Books with Extracted Authors from Titles:
                                          Title         Extracted Author   Author
422  Ship Leaves Harbor: Essays on Travel by a Reco...  a Recovering Journeyman  Unknown

 Found 1 books where the author was embedded in the title.

 ✅ Author extraction check completed.
```

- The Author column is empty (Unknown), while the correct author is inside the Title.
- This affects data integrity, making it harder to analyze books by author.
- Standardizing the dataset becomes challenging when important attributes are sprinkled into other columns.

- **Do you see any other data quality problems with this attribute?**

```
PS C:\Users\ADMIN\Desktop\HW> python processing/sprinkled_values_check.py

Data Quality Issues in `Author` Attribute:

125 books have missing or 'Unknown' authors:
                                            Title   Author
10     Starving Hearts (Triangular Trade Trilogy, #1)  Unknown
21  Miss Peregrineâs Home for Peculiar Children ...  Unknown
24  Library of Souls (Miss Peregrineâs Peculiar ...  Unknown
27  Hollow City (Miss Peregrineâs Peculiar Child...  Unknown
29  Full Moon over Noahâs Ark: An Odyssey to Mou...  Unknown

1 authors found with inconsistent `Last, First` format:
          Author
732  Dyckman, Ame

1 authors contain non-author information (Dr., PhD, etc.):
        Author
645  Dr. Seuss

✅ Author attribute verification completed.
```

Yes, besides authors being embedded in titles, I have identified other data quality problems with the Author column.

Missing or "Unknown" Authors:

- 125 books have "Unknown" or blank values in the Author column.
- This makes author-based analysis difficult (e.g., finding the most popular authors).

Author Representations

- 1 authors found with inconsistent `Last, First` format (Dyckman, Ame should be Ame Dyckman)
- This makes it hard to match books by the same authors.

Presence of Non-Author Information

- 1 authors contain non-author information (Dr., PhD, etc.): Dr. Seuss

IV. PART 4 (Tools & Libraries Used)

I have used the following Python libraries to process the dataset:

Library	**Purpose**
pandas	Data manipulation, reading & writing CSV files.
re (Regex)	Pattern matching for cleaning text and detecting format inconsistencies
matplotlib	Visualization (histograms for numeric/text length analysis).
seaborn	Enhanced plotting of histograms and data distribution analysis.
numpy	Numerical computations for missing value handling.