Introduction to Data Science

Project Documentation

Project Topic:

Collecting Data from Multiple Bookstores, Including Preprocessing and Standardization

Github Project Repository:

https://github.com/KTalia/I2DS_Project

Student: Talia Kastrati – 191527

Mentor: Dimitar Peshevski

May, 2025

1. Introduction

This project focuses on the collection, preprocessing, and standardization of product data from four different online bookstores: Literatura.mk , SakamKnigi.mk, Ikona.mk and AkademskaKniga.mk. The goal was to automate the data collection process, clean and organize the extracted information, and prepare it for future analysis or application in other systems.

Each bookstore presents its own layout, structure, and way of displaying product information. To accommodate these differences, data from each website was scraped, preprocessed, and stored separately. The scraping process focused on retrieving the book title, author, price, category, and the time the data was retrieved. In cases where books were on sale, both the original and sale prices were extracted, and a new column was computed to show the discount percentage.

The retrieved_at timestamp was included in each record to support potential future use cases such as time-series tracking or price trend analysis. This adds an additional dimension to the datasets and allows for temporal insights if the data collection is repeated over time.

The objectives of the project were:

- To extract relevant book data from **four independent bookstores**, each with a different structure.
- To **handle and clean inconsistencies** such as missing authors, irregular price formats, and differing category labels.
- To **standardize the data within each dataset**, ensuring that fields like price and category are consistent and readable.
- To **calculate sale percentages** for books with discounts, enabling more meaningful comparison and analysis.

Some challenges included handling dynamically loaded content, inconsistent formatting of prices and categories, and missing author information. These challenges were addressed using a combination of manual corrections and automated cleaning techniques.

This project demonstrates how data science techniques can be used to transform unstructured web data into structured , usable formats — preparing the way for further analysis, visualization, or integration into other systems.

2. Technologies and Tools Used

To implement the scraping, preprocessing, and data storage tasks in this project, I used a combination of programming languages, libraries, and development tools. The choice of tools was guided by their robustness, ease of integration, and suitability for data science workflows.

- **Programming Language:** Python
- **Database:** SQLite
- **Libraries and Modules:**
    - **selenium** – for automated web scraping from dynamic websites
    - **pandas** – for data manipulation and preprocessing
    - **numpy** – for numerical operations
    - **sqlite3** – for interacting with the SQLite database
    - **csv** – for reading and writing CSV files
    - **missingno** – for visualizing and identifying missing data
      datetime, **time**, **os**, **re** – for time tracking, delays, file operations, and regular expressions

3. Data Sources (Bookstores)

This project is based on data collected from four Macedonian online bookstores: Literatura.mk, SakamKnigi.mk, Ikona.mk and AkademskaKniga.mk. Each website was analyzed and scraped individually due to differences in structure, product layout, and the way information is presented.

1. Literatura.mk

- **Website:** https://www.literatura.mk
- **Description:** One of the largest and most popular online bookstores in Macedonia, offering a wide range of books across categories such as fiction, non-fiction, children's books, and academic titles.
- **Structure:** Products are listed in a grid format with clearly defined elements such as title, author, price, and category.

2.SakamKnigi.mk

- **Website:** https://www.sakamknigi.mk
  Description: A well-known bookstore focusing on various genres and often featuring promotions and seasonal offers.

- **Structure:** Books are displayed in either list or card layouts depending on the selected category. Sale prices are included when available. However, author names were missing in approximately 75% of the scraped entries, as they were not present in the HTML structure.

3. AkademskaKniga.mk

- **Website:** https://www.akademskakniga.mk
- **Description:** Specialized in academic and professional literature, this bookstore targets students, researchers, and professionals.
- **Structure:** The site is more minimalistic and less standardized. Some product details like categories or discounts were harder to locate or sometimes missing.

4. Ikona.mk

- **Website**: https://www.ikona.mk
- **Description**: A Macedonian online bookstore known for its curated selection of books in literature, psychology, self-help, and spirituality. Ikona.mk often features local authors and includes thematic collections or recommended reading lists.
- **Structure**: Books are presented in a grid layout with essential information like the title and price consistently displayed. Author names are not visible in the grid view and only appear on the individual product detail pages. Some listings include short descriptions or promotional tags, but category labels are not uniformly applied across the site.

Selection Criteria

The bookstores were selected based on:

- **Relevance and popularity** in the Macedonian book market
- **Availability of diverse product data**, including pricing, authorship, and categories
- **Variety in website structure**, which allowed the project to demonstrate adaptability in scraping and preprocessing across different layouts

4. Database Design

To efficiently manage and query the collected book data, all scraped datasets are stored in a single SQLite database (books.db). Within this database, separate tables were

created for each bookstore (literatura, sakamknigi, akademskakniga, and ikona). This design choice simplifies data management and enables easy cross-source querying or integration without the overhead of handling multiple databases. The full project structure, including scripts and data organization, is available in the GitHub repository.

5. Web Scraping Process

This section describes the general workflow for scraping book data from four Macedonian online bookstores: **Literatura**, **SakamKnigi**, **Ikona.mk** and **Akademska Kniga**. While each scraper targets a different website with its own HTML structure and navigation, they share common patterns in data extraction, storage, and export.

1. Initialization

Each scraper uses Selenium WebDriver to automate Chrome in headless mode for efficient scraping.

Browser options include --headless, --no-sandbox, and --disable-dev-shm-usage for stability and performance.

A local SQLite database is created or opened to store scraped data persistently.

Each scraper defines its own table schema tailored to the site's data fields (e.g., title, author, price, category, retrieval date).

2. Category and URL Management

Each scraper has a predefined list of categories with corresponding URLs to traverse.

Categories represent different book genres or academic disciplines.

Some scrapers handle specific categories that contain only a single page differently to optimize scraping.

3. Pagination Handling

For categories with multiple pages, the scrapers detect the total number of pages via pagination controls or URL patterns.

The scraper then iterates through each page by adjusting URL parameters or query strings.

Pagination ends when there are no further pages or no more book elements found.

4. Book Data Extraction

For each book element found on a page, the scrapers extract:

- **Title** — usually from a prominent heading or link element.
- **Author** — extracted when available; defaults to "N/A" if missing.
- **Price** — includes real price and sale price if discounted; numeric values are cleaned for consistency.
- **Category** — assigned based on the current scraping context.
- **Date Retrieved** — timestamp of when the data was scraped.

Site-specific adaptations were necessary due to differences in HTML structures. Notably:

- **Ikona.mk** required a two-step process: while book listings are shown in a grid on category pages, the author name is not visible in the main grid view. Therefore, for every book listed, the scraper opened the individual book detail page to extract the author. This added overhead but ensured accurate and complete author data.

- On **SakamKnigi.mk**, roughly 75% of the books had missing author names, not due to scraping error but because the author information was not present in the HTML source.

- **AkademskaKniga.mk** had a more minimalistic and inconsistent layout. Some fields like category or price were occasionally missing or harder to locate.

5. Data Storage

Extracted data is inserted into the respective SQLite tables with schema designed to support:

- Unique identifiers
- Textual fields (title, author, category)
- Numeric and boolean fields (prices, sale status)
- Date fields (retrieved_at)

6. Data Export

After scraping all categories, data from the SQLite database is exported to a CSV file. Each scraper outputs to the data/original_datasets/ folder.

The CSV contains headers for all fields and is encoded in UTF-8 for compatibility.

7. Error Handling and Robustness

The scrapers employ try-except blocks to gracefully handle missing data or DOM changes.

Timeouts and waits are used to ensure elements are loaded before scraping.

The scripts log errors or skip problematic entries without stopping the entire process.

8. Resource Cleanup

At the end of scraping, the browser sessions are closed and database connections are safely terminated. This prevents resource leaks and allows for repeated runs without corruption.

6. Data Preprocessing and Standardization

This section outlines the key steps taken to clean, structure, and standardize the raw book datasets for reliable analysis and modeling.

1. Handling Missing and Corrupted Data

- **Author Column:** A significant number of entries are missing or contain corrupted characters (e.g., ?). These are identified and set to NaN.
- Missingness is quantified by counting missing vs. non-missing values and calculating percentages to understand the scope.
- No generic imputation is applied, but dataset-specific corrections are made (e.g., assigning "J.K. Rowling" as the author for all Harry Potter books in the Sakamknigi dataset).

2. Text Cleaning and Normalization

- **Whitespace Trimming:** Leading/trailing spaces are removed from all textual fields such as Title, Author, and Category.
- **Prefix Removal:** The "од:" prefix in the Author field is stripped to standardize names.

- **Parentheses Removal:** To clean author names, all content enclosed in parentheses—including the parentheses themselves—was removed. This eliminated descriptors such as (author), (editor), (illustrator), and incomplete brackets, ensuring only clean author names remain.
- **Author Name Normalization:** Names in "LastName, FirstName" format are converted to "FirstName LastName".
- Authors embedded within the Title field (e.g., in the Sakamknigi dataset) are extracted and moved to the Author column.
- Multiple authors separated by commas or by "and" are split into separate columns to improve structure.
- Extracted authors embedded in the title field.

3. Price Cleaning and Conversion

- **Format Correction:** The "ден" currency symbol and commas (used as thousand separators) are removed from Real Price and Sale Price.
- Type Conversion: Cleaned price values are converted to float.
- Invalid values are coerced to NaN to prevent processing errors.

4. Column Renaming and Type Standardization

- **Column Renaming:** Sale → IsOnSale for improved clarity.
- **IsOnSale:** Converted from 0/1 integers to boolean (True/False).
- **Retrieved At:** Converted from string to datetime for time-based operations.
- **Price fields:** Converted from string to float

After preprocessing, each dataset is saved to the data/original_datasets/ folder.

7. Preprocessing Output Summary

The table below summarizes key dataset characteristics after preprocessing for each bookstore dataset. These metrics reflect the scope and scale of the cleaned data ready for analysis:

Bookstore	Number of Categories	Number of Books Scraped	Number of Discounted Books
Literatura.mk	99	20188	0
SakamKnigi.mk	11	2167	11
AkademskaKniga.mk	37	21393	19228
Ikona.mk	13	1100	32

Number of Categories: Represents the unique book categories identified after cleaning and standardization.

Number of Books Scraped: Total number of book records successfully scraped and processed from each bookstore source.

Number of Discounted Books: Indicates books flagged as discounted after price cleaning and sale status standardization.

Notably, AkademskaKniga.mk has the highest volume of discounted books, which may reflect more active promotional pricing compared to the other sources. Meanwhile, Literatura.mk has no discounted books recorded, suggesting either no promotions or missing discount data.

This summary illustrates the dataset's integrity and organization, enabling reliable insights and informed decision-making in later stages.

8. Conclusion

This project successfully demonstrated the complete data science pipeline, from web scraping to preprocessing and standardization, applied to real-world data from four different Macedonian online bookstores. Despite the structural

inconsistencies and missing information across websites, the scraping process was adapted to each store's unique layout, ensuring comprehensive and accurate data extraction.

The preprocessing phase addressed critical challenges such as missing author names, inconsistent price formats, and non-standardized category labels. By applying both automated and manual cleaning techniques, the final datasets were transformed into clean, consistent, and structured formats, ready for future analysis, visualization, or integration into other systems.

The modular and reusable design of the scraping scripts, combined with the use of SQLite and CSV exports, ensures scalability and maintainability, allowing future updates or expansions to be easily implemented. Additionally, the inclusion of a timestamp on each record opens the possibility for time-based analytics, such as monitoring price trends or tracking inventory changes over time.

Overall, the project illustrates how data science tools and techniques can be applied to heterogeneous web data sources, turning raw and unstructured inputs into valuable, well-organized datasets that serve as a foundation for deeper insights and applications.