

Data science

Section 8D



**21F-9111**

**21F-9181**

**21F-9248**

**Milestone 2**

**Scraping Scripts & Data Snapshots**

**Project Title:** Scraping Bicycle-Related Business Data from Google Maps in Denmark

**1. Introduction**

This report documents Milestone 2 of a project focused on collecting comprehensive data on bicycle-related businesses in Denmark via web scraping from Google Maps. The objective is to compile a dataset containing details such as business names, addresses, phone numbers, ratings, and geolocation data for an in-depth analysis of Denmark’s cycling infrastructure and services.

**Deliverables for this Milestone:**

* Complete web scraping scripts with comprehensive documentation and inline comments.
* Data snapshots (sample records) of the raw data collected.
* Detailed documentation of preprocessing steps and challenges encountered during the data collection process.

The report provides a professional, end-to-end overview of the scraping process, addressing challenges, sample data output (approximately 2,000 records), and preprocessing techniques applied to ensure data consistency.

**2. Web Scraping Scripts**

**Overview**

The web scraping approach utilizes a Python script that leverages the Selenium library to automate interactions with Google Maps. The script systematically searches for bicycle-related businesses across Denmark, extracts key data points from the listings, and exports the results into a CSV file.

**Key Features**

* **Search Terms:**  
  Combines a set of base keywords (e.g., "Cykelbutik i Danmark", "Bicycle shop Denmark") with over 200 Danish locations, resulting in more than 11,600 unique search terms.
* **Data Points Extracted:**
  + Keyword
  + Business Name
  + Business Type
  + Address
  + Phone Number
  + Plus Code
  + Rating
  + Website
  + Reviews Count
  + Latitude
  + Longitude
  + Map URL
* **Output:**  
  The data is consolidated and saved into a CSV file named denmark\_bicycle\_shops.csv.
* **Documentation:**  
  The script is fully commented to detail the setup process, helper functions, and overall scraping workflow. This ensures clarity and reproducibility for future iterations or audits.

**Handling Google Maps Complexity**

The script is designed to manage Google Maps’ dynamic interface and built-in anti-scraping measures. Techniques include:

* **Dynamic Content Loading:**  
  A scrolling mechanism that continuously loads additional listings until all available data (up to a maximum of 100 listings per keyword) are fetched.
* **Error Handling:**  
  Robust try-except blocks and helper functions (e.g., safe\_extract) return "Not Found" for missing elements to maintain script stability and prevent crashes.
* **Rate Limits and CAPTCHA Avoidance:**  
  Randomized delays between actions (3-8 seconds) mimic human behavior, reducing the risk of detection and interruption by CAPTCHAs.
* **Incremental Data Saving:**  
  Data is saved incrementally to the CSV file after each listing, ensuring that progress is preserved even if the script is interrupted.

**3. Challenges Encountered During Data Collection**

**3.1 Dynamic Content Loading**

* **Challenge:** Google Maps loads listings dynamically as users scroll, meaning the initial page load may not display all available data.
* **Solution:** Implemented a scrolling mechanism that repeatedly scrolls to the bottom of the results feed until no new listings appear.

**3.2 Variability in Data Structure**

* **Challenge:** Not all listings include the same details; some may lack phone numbers, websites, or ratings.
* **Solution:** Incorporated error handling with try-except blocks and fallback functions to assign "Not Found" where data is missing.

**3.3 Extraction of Specific Data Points (e.g., Phone Numbers)**

* **Challenge:** Phone numbers are presented in varied formats and locations on the page.
* **Solution:** Used multiple extraction methods (e.g., tel: links, aria-labels, regex-based text parsing) and standardized phone formats (including adding the +45 country code where necessary).

**3.4 Rate Limits and CAPTCHAs**

* **Challenge:** Google Maps enforces rate limits and may trigger CAPTCHAs to block automated access.
* **Solution:** Introduced random delays between actions to mimic human interaction, thereby reducing the likelihood of triggering anti-scraping mechanisms.

**3.5 Managing a Large Volume of Search Terms**

* **Challenge:** Over 11,600 search terms required careful management to avoid data loss and ensure complete data collection.
* **Solution:** The script saves data incrementally, ensuring that progress is maintained even if the operation is interrupted.

**3.6 Geolocation Data Extraction**

* **Challenge:** Latitude and longitude data are embedded within dynamically generated URLs.
* **Solution:** Parsed the URL to extract coordinates from the /place/ segment, with fallback values provided in case of extraction errors.

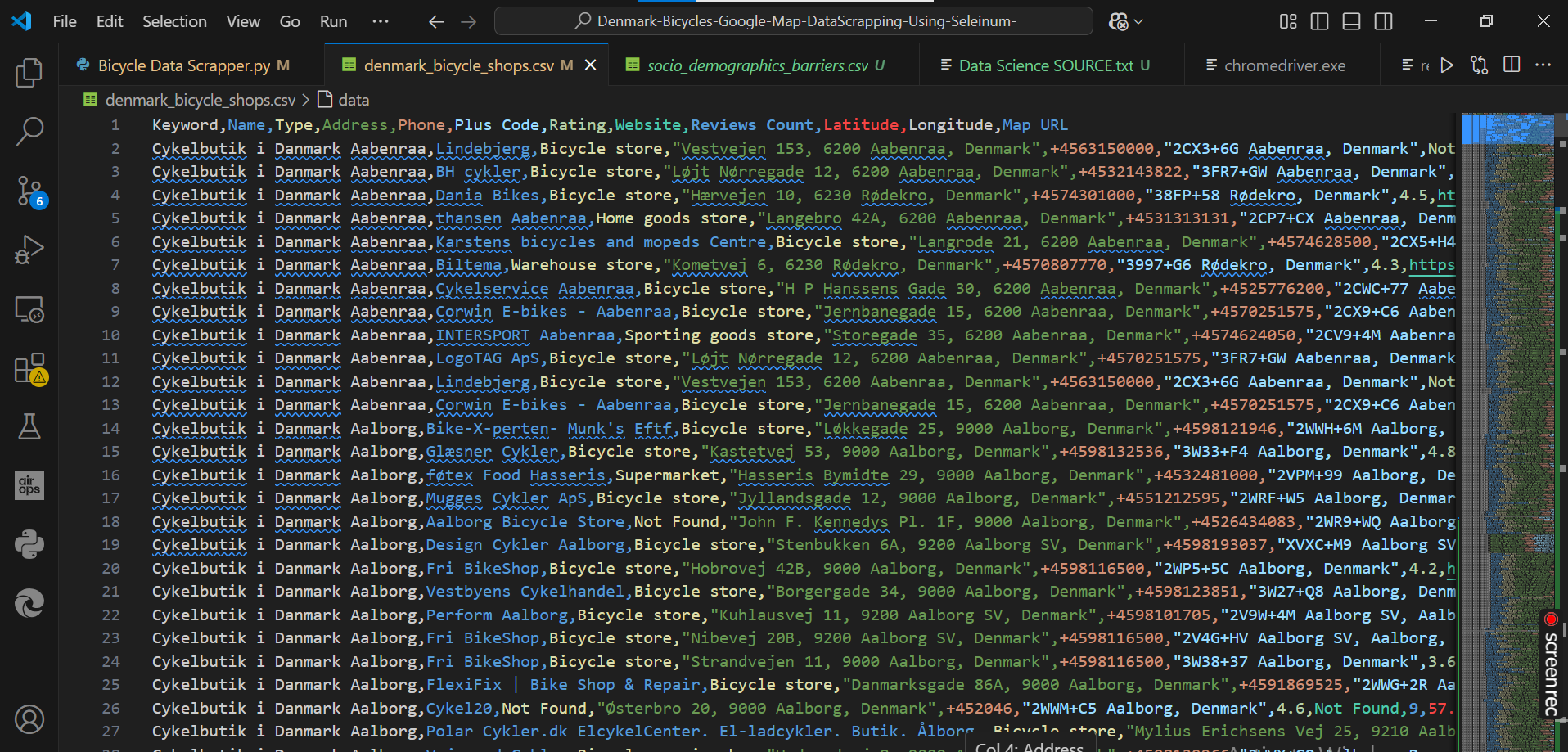
**4. Data Snapshots**

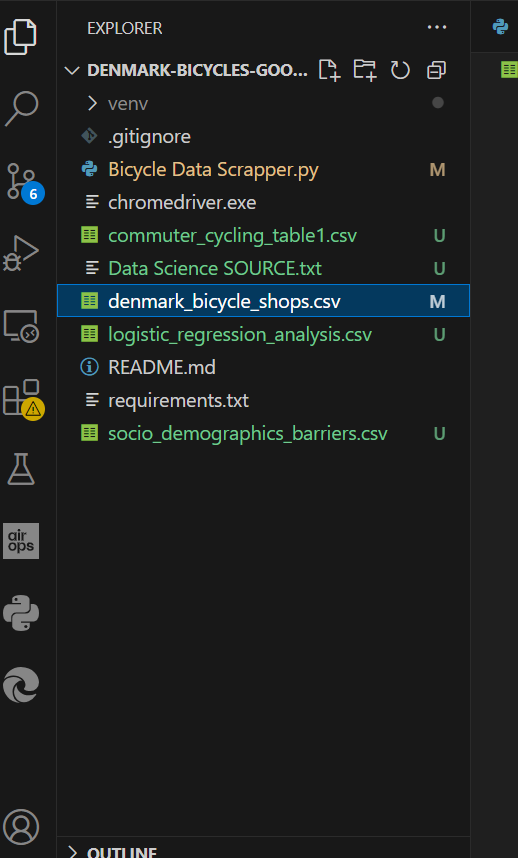
The scraping process successfully collected data for approximately 2,000 bicycle-related businesses across Denmark. The resulting CSV file, denmark\_bicycle\_shops.csv, includes the following columns:

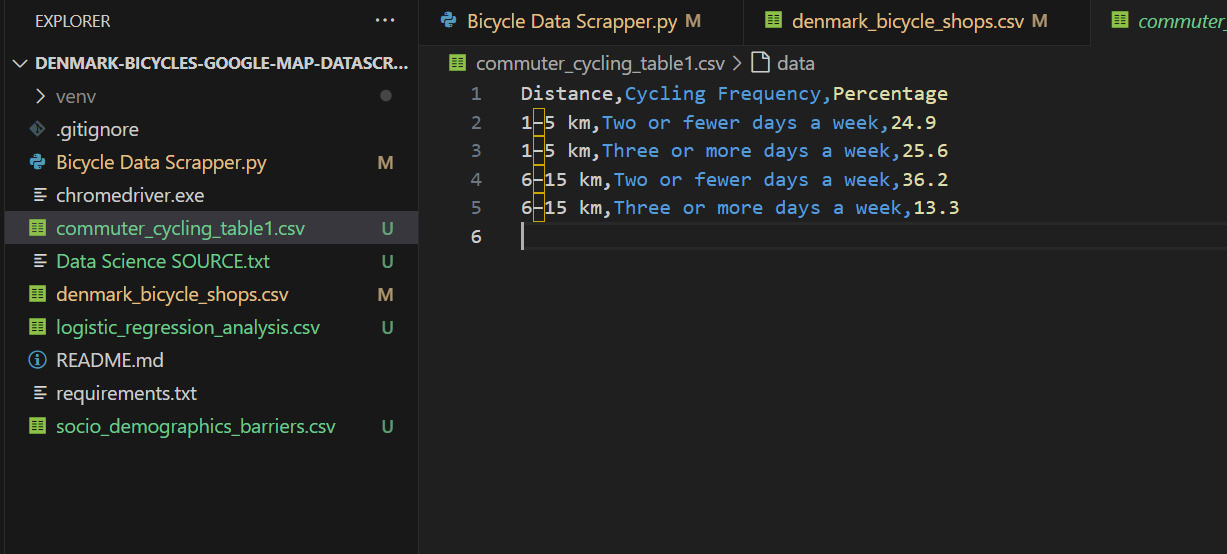
* Keyword
* Name
* Type
* Address
* Phone
* Plus Code
* Rating
* Website
* Reviews Count
* Latitude
* Longitude
* Map URL

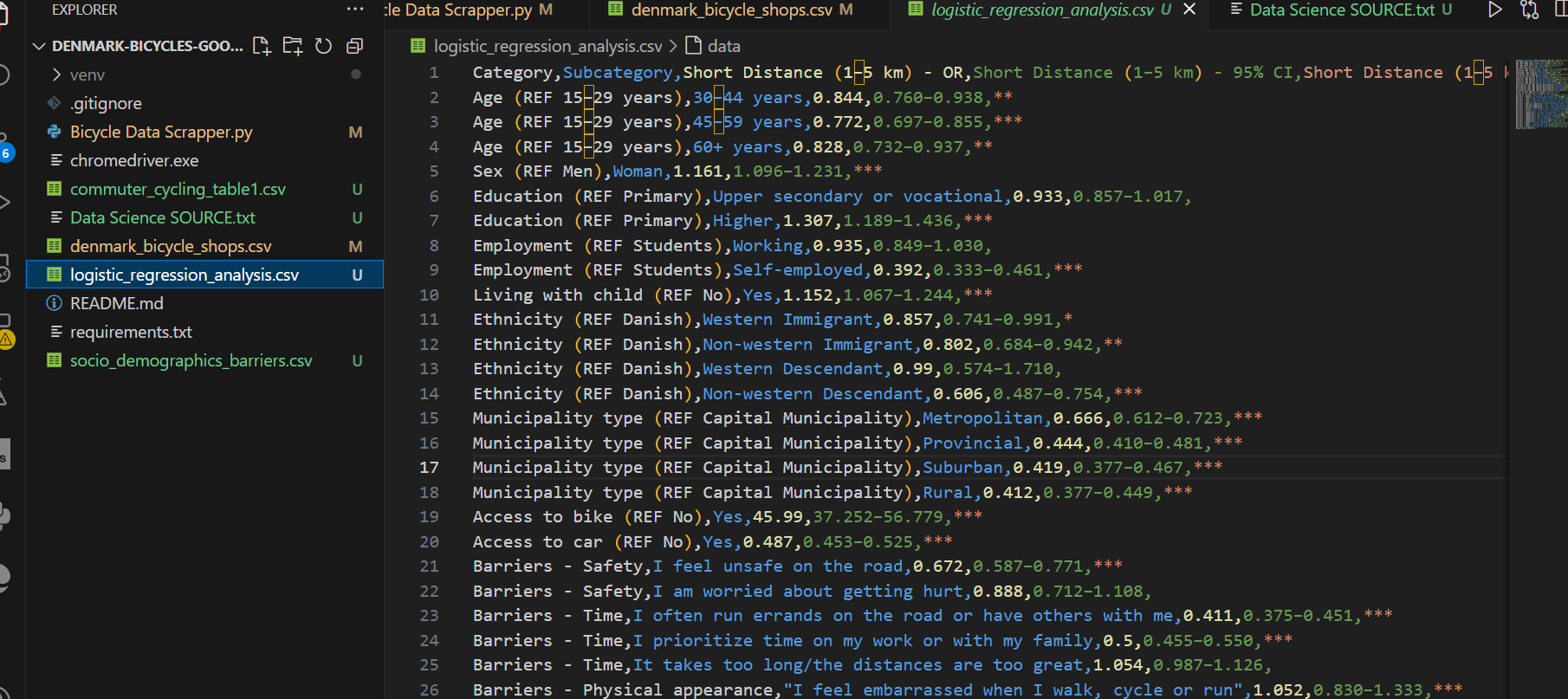
**Sample Data Snapshot**

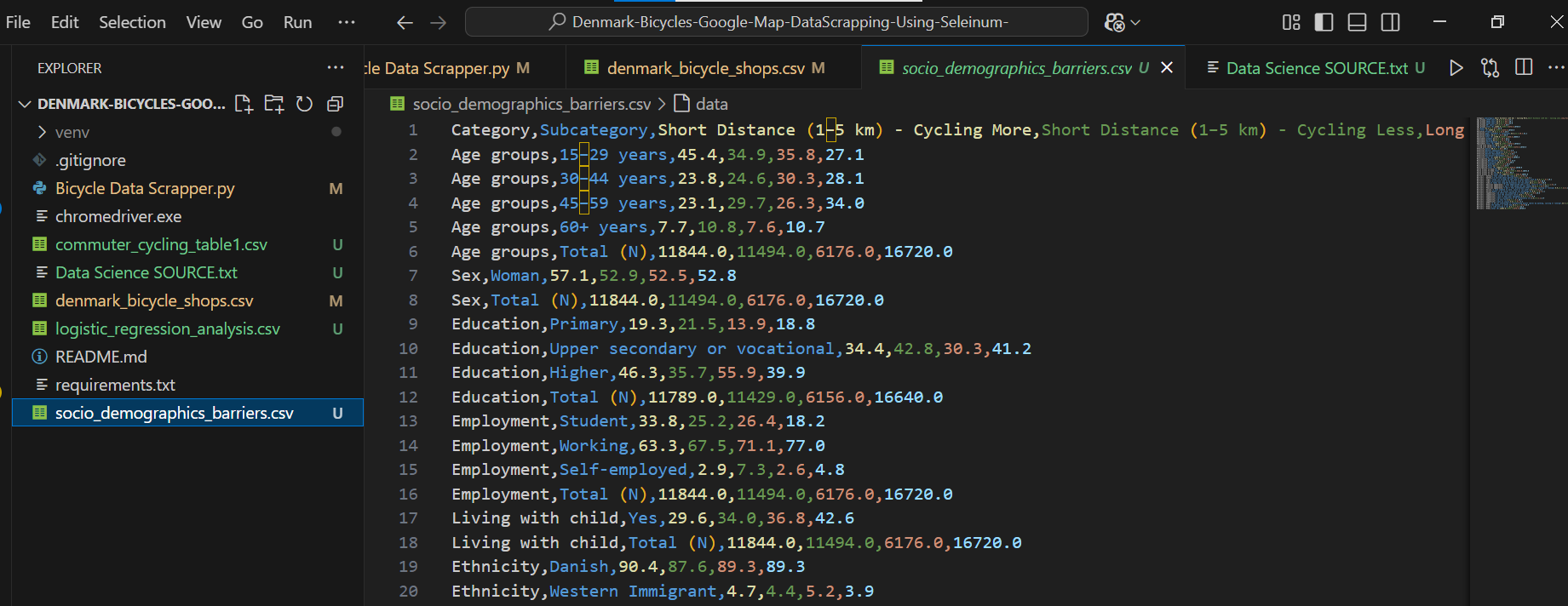
Below is an example of the first three rows of the raw data:

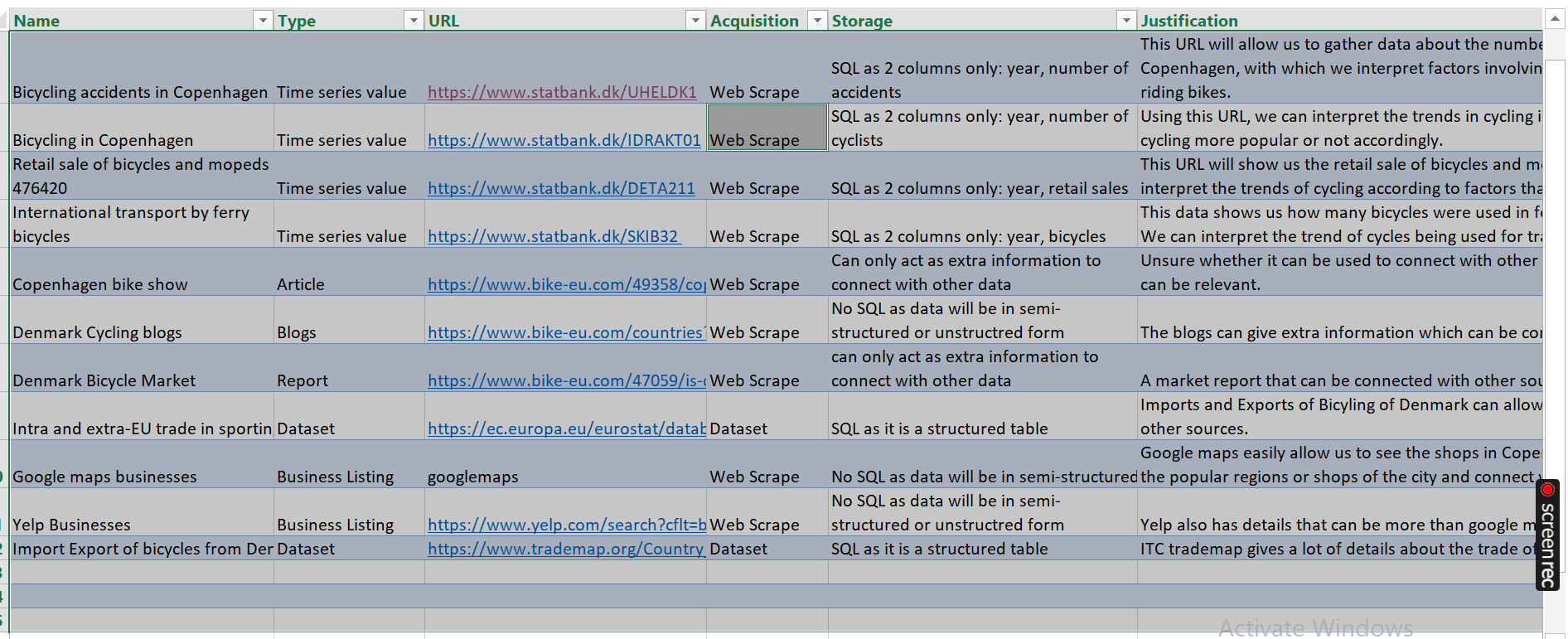
**Screenshot Placeholder:**  












**5. Preprocessing Steps**

**5.1 Address and Plus Code Cleaning**

* **Description:**  
  Addresses and plus codes occasionally contain non-alphanumeric characters or extra spaces.
* **Action:**  
  Applied a clean\_text function to remove any unwanted characters, ensuring consistent formatting across records.

**5.2 Phone Number Standardization**

* **Description:**  
  Phone numbers were extracted in various formats.
* **Action:**  
  Utilized multiple extraction methods, removed non-numeric characters, and standardized the numbers by prefixing with the country code +45 if absent.

**5.3 Handling Missing Data**

* **Description:**  
  Some listings were missing fields such as website or rating.
* **Action:**  
  Employed error-handling functions to assign a default value ("Not Found") to missing fields, thereby maintaining the integrity of the dataset.

These preprocessing steps ensured that the raw data was transformed into a structured, standardized format that facilitates subsequent analysis.

**6. Conclusion**

This milestone successfully delivered a comprehensive dataset of approximately 2,000 bicycle-related businesses in Denmark, gathered via a robust web scraping script targeting Google Maps. Despite facing challenges related to dynamic content, variable data structures, and anti-scraping measures, the script effectively extracted key data points and applied preprocessing to standardize the information.

**Next Steps:**

* Clean and deduplicate the dataset to further improve quality.
* Perform exploratory data analysis to uncover insights into Denmark’s cycling infrastructure and business ecosystem.