COSC 482 – Data Science and Web Scraping

Lab Assignment 5

Scraping, Cleaning, and Analyzing eBay Tech Deals

Spring 2025

Dr. Roaa Soloh

CIS Department

Objective:

Your objective is to build a complete data pipeline that:

1. **Scrapes** live product data from eBay Global Tech Deals.
2. **Automatically updates** the raw data file using GitHub Actions (running every three hours) for nearly two days.
3. **Cleans and processes** the raw CSV data.
4. **Performs exploratory data analysis (EDA)** with visualizations on the cleaned data.

Task 1: Web Scraping with Selenium

- Create a Python script (scraper.py) that:
  - Uses Selenium to open https://www.ebay.com/globaldeals/tech.
  - Scrolls down the page to trigger lazy loading of all product listings.
  - Extracts the following details for every product on the page:
    - **timestamp:** The current date and time when the product is scraped.
    - **title:** The product title.
    - **price:** The discounted price.
    - **original_price:** The original price (if available).
    - **shipping:** Shipping details.
    - **item_url:** The product URL.
  - Saves the extracted data into a CSV file named ebay_tech_deals.csv, appending new data if the file already exists.
  - Do not impose any limit on the number of products to extract.

Task 2: Automation with GitHub Actions

- Configure your repository so that the scraper runs automatically every three hours.

    - **Note:** The automation phase should be scheduled to run for nearly two days before you begin the cleaning phase.
    - **Cron Expression:** Use the following cron schedule in your GitHub Actions workflow:
- cron: '0 */3 * * *'
    - This ensures that the scraper updates the CSV file (ebay_tech_deals.csv) at three-hour intervals, building a robust dataset over two days.

Task 3: Data Cleaning & Processing

- Create a Python script (clean_data.py) that:
    - Loads the raw CSV file (ebay_tech_deals.csv) with all columns as strings.
    - Cleans the price and original_price columns by removing "US $" and commas, and trims extra whitespace.
    - If original_price is missing (i.e., marked as "N/A" or empty), replaces it with the corresponding price.
    - Cleans the shipping column by replacing "N/A", empty strings, or strings containing only whitespace with the default message: "Shipping info unavailable".
    - Converts the price and original_price columns to numeric (float) values.
    - Creates a new column discount_percentage computed as:

$$\text{discount\_percentage} = \left(1 - \frac{\text{price}}{\text{original\_price}}\right) \times 100$$

        - (rounded to two decimal places, with missing values handled appropriately).
    - Saves the cleaned data as cleaned_ebay_deals.csv.

Task 4: Exploratory Data Analysis (EDA) & Visualization

Develop a Jupyter Notebook (EDA.ipynb) that uses the cleaned data (cleaned_ebay_deals.csv) to perform the following analyses:

1. **Time Series Analysis:**
   - Convert the timestamp column to datetime and sort the data.
   - Extract the hour from each timestamp and group the data by hour.
   - Plot a bar chart showing the number of deals per hour.
2. **Price and Discount Analysis:**
   - Plot a histogram and boxplot to visualize the distribution of product prices.
   - Create a scatter plot comparing original_price versus price.
   - Plot the distribution of the discount_percentage to analyze how discounts vary.
3. **Shipping Information Analysis:**
   - Count the frequency of different shipping options.
   - Plot a bar chart showing the frequency of shipping options.
4. **Text Analysis on Product Titles:**
   - Define a set of keywords (e.g., "Apple", "Samsung", "Laptop", "iPhone", "Tablet", "Gimbal").
   - Count how many times each keyword appears in the title column (case-insensitive).
   - Visualize the keyword frequencies using a bar chart (ensuring the palette is set correctly).
5. **Price Difference Analysis:**
   - Compute a new column for the absolute discount (i.e., original_price - price).
   - Plot a histogram of the price differences.
6. **Discount:**
   - Sort the dataset by discount_percentage in descending order and display the top 5 deals with the highest discounts.

Submission Requirements

- **Repository:**
  Push your project to your GitHub repository. The repository should include:
  - scraper.py
  - clean_data.py
  - The raw CSV file (ebay_tech_deals.csv) generated by the scraper.
  - The cleaned CSV file (cleaned_ebay_deals.csv).
  - EDA.ipynb (your Jupyter Notebook with EDA and visualizations).
  - A GitHub Actions workflow file (with the cron expression 0 */3 * * * for scheduling).
- **Documentation:**
  Include a README or short report that summarizes your methodology, key findings from the EDA, challenges faced, and potential improvements.