"Web Scraping Financial Data from Wikipedia: Insights into the Largest U.S Companies"

COMPANY NAME :SHARPENED MIND TECHNOLOGIES PRIVATE LIMITED

DOMAIN:DATA ANALYST

MENTOR:MR.MADHAVAN SIVA

DATE:29-09-2024

SUBMIITED BY:

ABDUL ZUHAIL M

**Table of Contents**

1.Introduction

1.1 Project Overview

1.2 Objectives

2. Project Requirements

2.1 Software Requirements

2.2 Libraries Required

3. Technologies Used

3.1 Python

3.2 Jupyter Notebook

3.3 BeautifulSoup and Requests

3.4 Pandas

3.5 Matplotlib

3.6 Seaborn

4. Methodology

4.1 Data Source

4.2 Steps Involved

5. Results

5.1 Data Extracted

5.2 Sample Output

5.3 Data Visualization

6. Challenges and Solutions

6.1 HTML Structure Changes

7. Conclusion

8. References

**1. Introduction**

**1.1 Project Overview**

This mini-project aims to develop a Python script to scrape data from the Wikipedia page listing the largest companies in the United States by revenue. The purpose is to collect information such as company names, revenue, industry, and other relevant details, and store it in a structured format for further analysis.

**1.2 Objectives**

* To understand and implement web scraping techniques using Python.
* To extract structured data from a public website.
* To analyze and visualize the collected data.

**2. Project Requirements**

**2.1 Software Requirements**

* Python 3.x
* Jupyter Notebook (for code development and documentation

**2.2 Libraries Required**

* **requests:** To send HTTP requests to the website.
* **BeautifulSoup**: To parse and extract data from HTML content.
* **pandas:** For data storage and manipulation.
* **matplotlib:** For data visualization and plotting the extracted data.
* **seaborn:** For enhanced and more aesthetic data visualization.

**3. Technologies Used**

**3.1 Python**

* Used for scripting and data processing.

**3.2 Jupyter Notebook**

* Provides an interactive environment to write, test, and document the code.

**3.3 BeautifulSoup and Requests**

* BeautifulSoup is used to parse the HTML content, while Requests fetches the web page content.

**3.4 Pandas**

* Used to store the extracted data in a structured format (DataFrame) and perform data manipulations.

**3.5 Matplotlib**

* Used to visualize the extracted data through charts and graphs, making it easier to analyze trends and distributions.

**3.6 Seaborn**

* Used for creating advanced and more aesthetic visualizations, such as heatmaps and pair plots, to gain deeper insights into the data.

**4. Methodology**

**4.1 Data Source**

The data is sourced from the Wikipedia page: List of largest companies in the United States by revenue.

**4.2 Steps Involved**

1. **Fetch the HTML Content**: Use the requests library to retrieve the HTML content of the Wikipedia page.

2**. Parse the HTML:** Use BeautifulSoup to parse the HTML and locate the table containing the desired data.

3. **Extract Information:** Loop through the rows of the table to extract details such as company rank, name, industry, revenue, and headquarters.

4. **Store the Data:** Store the extracted data in a pandas and DataFrame for easy manipulation and analysis.

5. **Visualize the Data:** Use matplotlib to create visual representations, such as bar charts and pie charts, to summarize the data effectively.

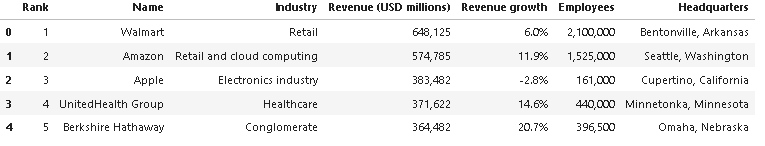
**5. Results**

**5.1 Data Extracted**

The script successfully extracts data on the top 50 companies, including:

1. Company Rank
2. Name
3. Industry
4. Revenue (in millions of USD)
5. Revenue growth
6. Employees
7. Headquarters

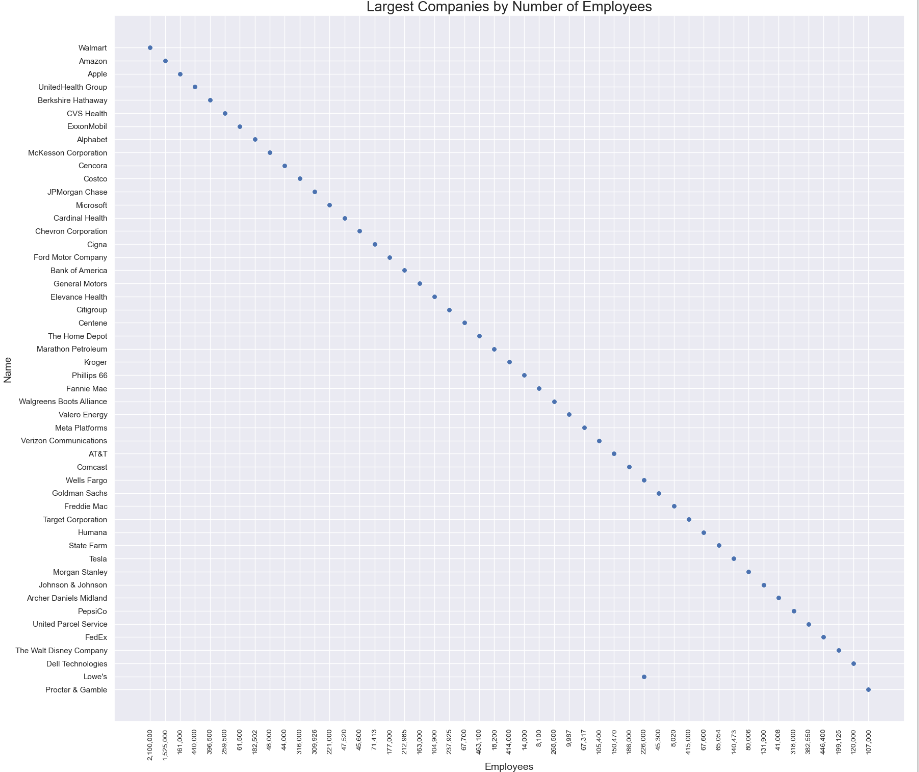
**5.2 Sample Output**



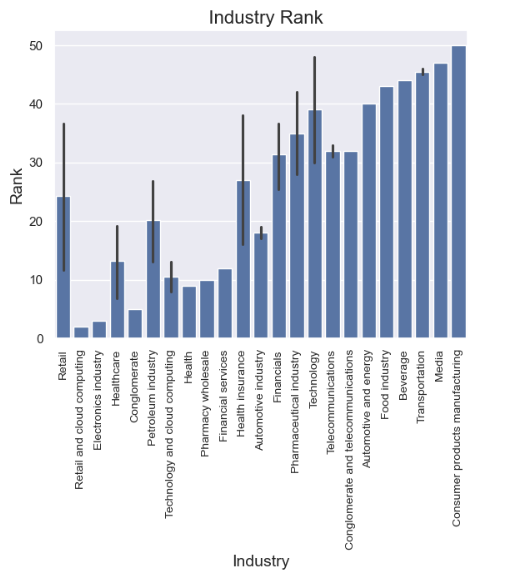
**5.3 Data Visualization**

The top companies by revenue are visualized using the following chart, showing their respective revenue figures. This helps in understanding the distribution of revenue among the largest companies.

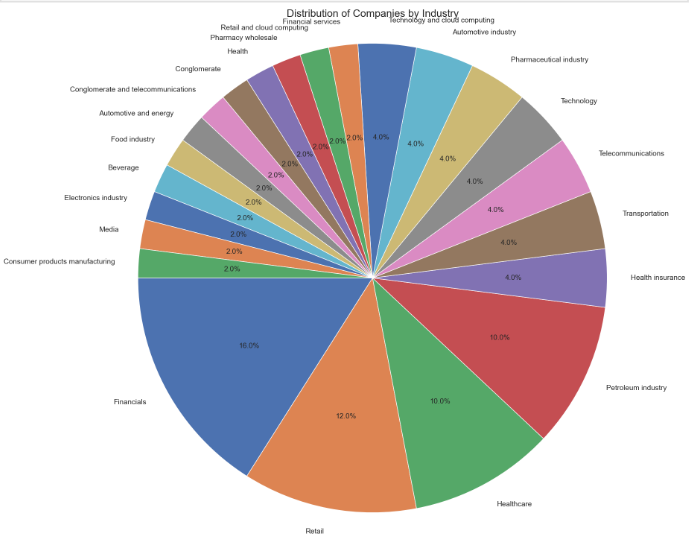
**1.Scatterplot:**



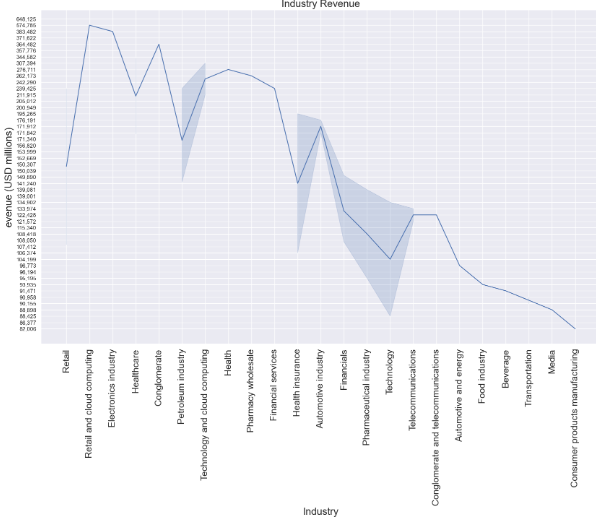
**2.Barplot**



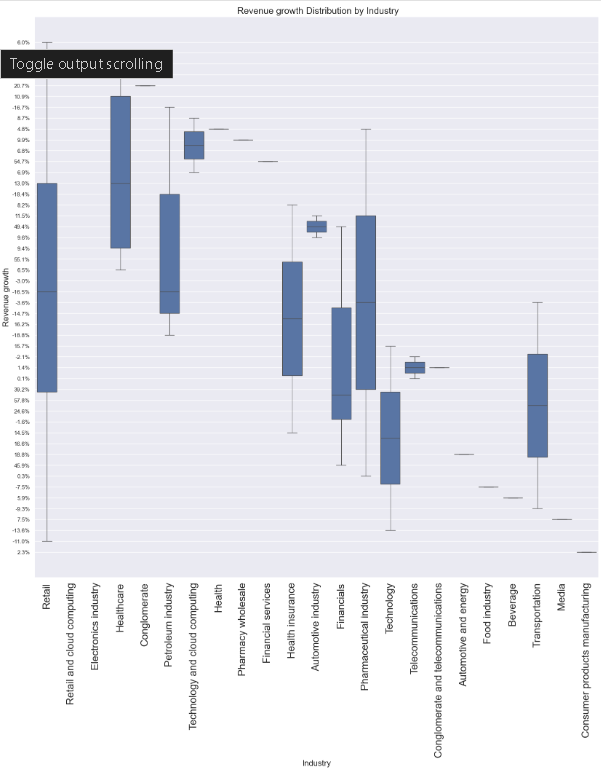
**3.Piechart**

****

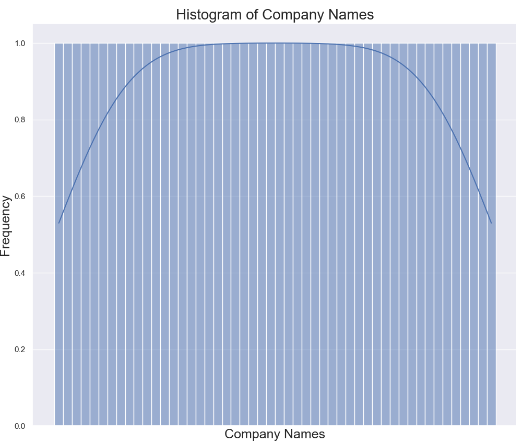
**4.Lineplot**

****

**5.Boxplot**

****

**6.Histogram**



**7. Challenges and Solutions**

**7.1 HTML Structure Changes**

**Issue:** Changes in the website’s HTML structure can break the script.

**Solution:** Use flexible selectors and include error handling to catch parsing issues.

**8. Conclusion**

The project demonstrates a simple yet effective way to scrape and analyze data from a public website. It provides insights into the financial performance of major U.S. companies and uses visualization to enhance understanding.

**9. References**

* BeautifulSoup [Documentation](https://beautiful-soup-4.readthedocs.io/en/latest/)
* Requests [Documentation](https://requests.readthedocs.io/en/latest/)
* Matplotlib [Documentation](https://matplotlib.org/stable/index.html)
* Seaborn [Documentation](https://seaborn.pydata.org/tutorial/introduction)
* Wikipedia: [List of largest companies in the United States by revenue](https://en.wikipedia.org/wiki/List_of_largest_companies_in_the_United_States_by_revenue)
* GitHub repository: <https://github.com/abdulzuhail/web-scrap/blob/main/Mini_Project.ipynb>