Crawling Global Economic Indicators

Group names:
Ahmed Abdelmaksoud - 454778
Hamed Ahmed Hamed Ahmed - 454827
Daniela Quintero Narváez - 456002

Detailed description which group participant wrote which part of the project:

- **Ahmed Abdelmaksoud:** was responsible for building the spider using **Selenium**
- **Hamed Ahmed Hamed Ahmed**: was responsible for for building the spider using **Scrapy**
- **Daniela Quintero Narvaez** : was responsible for building the spider using **Beautiful Soup**

Website to be scrapped: **UNData website**

Domain: http://data.un.org/

Description for the topic:
Global Economic Indicators refer to key statistical measures that provide insights into the overall economic performance and trends at a global level. These indicators offer essential information about various aspects of the economy, such as gross domestic product (GDP), GDP growth rate, inflation rate, employment levels, trade balances, and other relevant economic variables.

Description for the website:
The UN Data website is an online platform that provides access to various statistical databases maintained by the United Nations and its affiliated organizations. It offers a wide range of data on social, economic, environmental, and demographic indicators from around the world. The website serves as a central hub for accessing and exploring statistical information compiled by the United Nations system.

Short description of your scraper mechanics

The scraper is designed to extract economic indicator data from the UNData website for multiple countries. Here is a technical description of how the program works:

1- The program is built using the 3 different frameworks **Beautiful soup**, **scrapy** and **selenium** , powerful web scraping libraries in Python.

2- We have created 3 Spider, which serves as the main component for crawling and scraping data from the website.

3-The spider starts by sending a request to the main page of the UNData website.

4- the spider extracts the links to country-specific pages from the HTML response using XPath selectors.

5- The spider limits the number of links to scrape based on the ScrapMin100 flag and the max_links variable. If the flag is set to True and the maximum number of links has been reached, the spider stops.

6-For each country link, the spider sends a new request to that specific country's page, to fetch the economic indicators .

Here is a brief description of the data columns:

CountryName: The name of the country.

Year: The year to which the data corresponds.

GDP: Gross domestic product: The total value of goods and services produced within a country's borders.

GDP growth rate: The percentage change in a country's GDP from the previous year.

GDP per capita: The GDP divided by the population, representing the average economic output per person.

Economy: Agriculture: The percentage contribution of the agriculture sector to the country's Gross Value Added.

Economy: Industry: The percentage contribution of the industrial sector to the country's Gross Value Added.

Economy: Services and other activity: The percentage contribution of the services and other sectors to the country's Gross Value Added.

Unemployment: The percentage of the labor force that is unemployed.

Labor force participation rate: The percentage of the working-age population that is either employed or actively seeking employment.

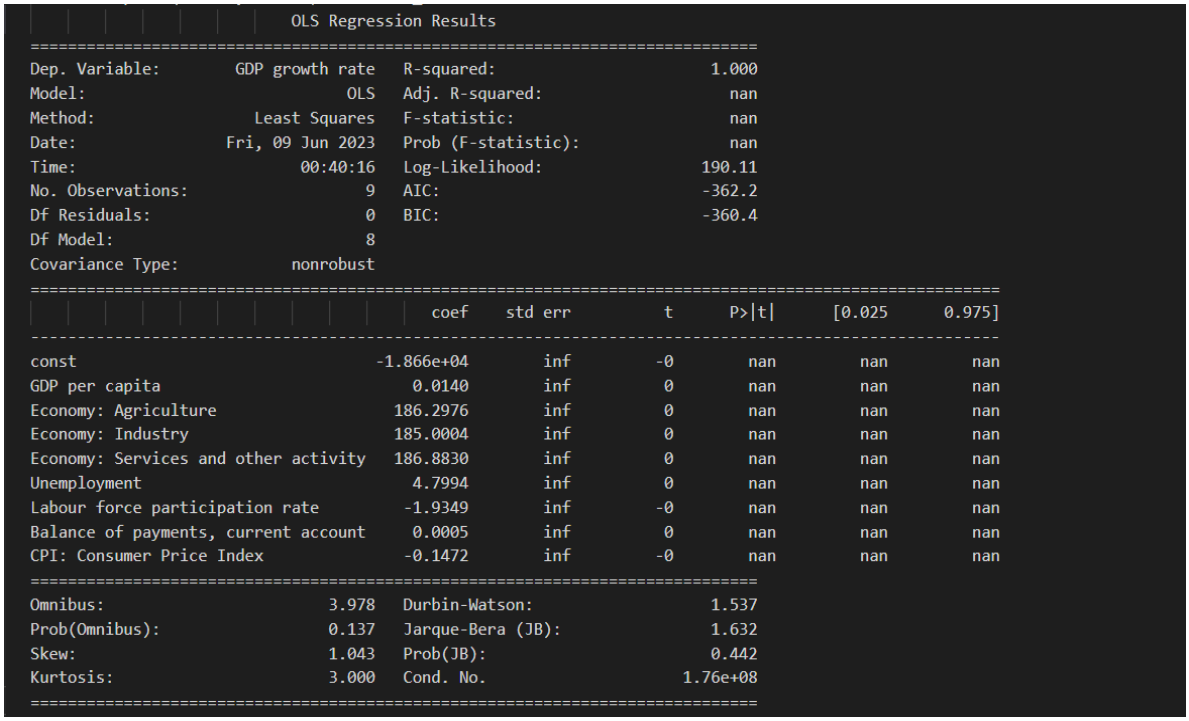
Balance of payments, current account: The difference between a country's total exports and imports of goods, services, and capital.

CPI: Consumer Price Index: A measure of the average change in prices of goods and services over time, reflecting inflation or deflation.

A simple comparison between 3 scrapers:

Criteria	Beautiful Soup	Scrapy	Selenium
Performance	Moderate: BeautifulSoup is not very efficient for large-scale scraping, because it depends on the requests library to make HTTP requests, which are blocking and slow.	High: Scrapy is faster than BeautifulSoup due to its asynchronous handling of requests.	Low: Selenium is generally slower because it requires the loading of an entire webpage, including JavaScript execution. It's like automating a full-fledged web browser.
Scalability	Low: BeautifulSoup is ideal for small to medium-scale projects due to its simplicity, but it might be slower for larger projects because it doesn't handle concurrent requests.	High: Scrapy can handle large-scale scraping due to its asynchronous architecture, which allows for concurrent requests and high throughput.	Low: Selenium doesn't handle concurrent requests natively, and it uses a lot of resources because it loads the entire webpage. It's not very scalable for large-scale scraping.

A simple analysis for the collected data:



OLS Regression Results

Dep. Variable:	GDP growth rate	R-squared:	1.000
Model:	OLS	Adj. R-squared:	nan
Method:	Least Squares	F-statistic:	nan
Date:	Fri, 09 Jun 2023	Prob (F-statistic):	nan
Time:	00:40:16	Log-Likelihood:	190.11
No. Observations:	9	AIC:	-362.2
Df Residuals:	0	BIC:	-360.4
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-1.866e+04	inf	-0	nan	nan	nan
GDP per capita	0.0140	inf	0	nan	nan	nan
Economy: Agriculture	186.2976	inf	0	nan	nan	nan
Economy: Industry	185.0004	inf	0	nan	nan	nan
Economy: Services and other activity	186.8830	inf	0	nan	nan	nan
Unemployment	4.7994	inf	0	nan	nan	nan
Labour force participation rate	-1.9349	inf	-0	nan	nan	nan
Balance of payments, current account	0.0005	inf	0	nan	nan	nan
CPI: Consumer Price Index	-0.1472	inf	-0	nan	nan	nan

Omnibus:	3.978	Durbin-Watson:	1.537
Prob(Omnibus):	0.137	Jarque-Bera (JB):	1.632
Skew:	1.043	Prob(JB):	0.442
Kurtosis:	3.000	Cond. No.	1.76e+08

The regression results indicate that the model has a perfect fit to the data, with an R-squared value of 1.000. However, there are some issues with the statistical tests and significance levels, as indicated by the "nan" values and lack of p-values for the coefficients.

The coefficient estimates for the predictors (independent variables) cannot be interpreted in this case due to the non-informative statistical output. The "inf" values in the standard errors and t-statistics indicate some problems with the estimation process.

The model's constant term (intercept) is estimated to be -18,660, indicating a negative baseline value for GDP growth rate. However, the lack of statistical significance and unreliable estimates prevent drawing meaningful conclusions about the impact of the predictors on GDP growth rate.

Overall, based on the available regression results, it is not possible to interpret the relationship between the predictors (GDP per capita, Economy: Agriculture, Economy: Industry, Economy: Services and other activity, Unemployment, Labour force participation rate, Balance of payments, current account, and CPI: Consumer Price Index) and the GDP growth rate accurately.