Uniwersytet Warszawski
Wydział Nauk
Ekonomicznych

BIRRAPEDIA PROJECT

Webscrapping and Social Scraping
2400 – DS1WSMS
Lecturers: Anna Lewczuk and Przemyslaw Kurek
Student 1: Ayax Díaz Noriega
Student Number 1: 435047
Student 2: Behbud Hamzayev
Student Number 2: 436350

PROJECT DESCRIPTION

The Project that we develop takes as reference a Spanish craft beer website.

This website is extremely popular in the craft beer community in some Spanish-speaking countries as well as in some European ones. The popularity of the site is due to the information that it contains and the simplicity of it. You can find thousands of beers as well as equipment for beer brewing and glassware. The site is also divided by countries so users can find specific things and their country of origin.

For the matter of scraping information, on this site, we can find data regarding the style of beer, name of the beer, IBU (International Bitterness Units scale), degrees of alcohol, rating, etc.

SIMPLE ANALYSIS of GATHERING DATA

Regarding the extraction of data with different scrapers, we can conclude that the use of Beautiful Soup allowed obtaining the information in a less complicated way, writing the code took less time compared to Selenium and Scrapy, of the latter we have to say that writing the code was much more time consuming and complicated. For this website, we consider that Beautiful Soup is the most convenient tool to use given the degree of simplicity in terms of registering the code and time taken.

PERFORMANCE COMPARISON

Scrapy is the fastest, full-featured, and high-level data extraction tool and used for a wide range of purposes, from data mining to monitoring and automated testing. While, Selenium is a bit more browser automation tool for testing and data scraper although it is slow and resource-heavy and not easy to control.

In all scrapers, we start to create a function "PrintInfo" with a dictionary that prints out the data beer dictionary and handles the error/data loss with the IBU part.

At the end of the code, we created a scrapper and parser where we choose a div block and then divide it to the header that contains name and rating, images, and extra where will use them as the path for filling the beer dictionary and append it to the list of beers. Furthermore, we take care of the error in the list extra where NEIPA gave an error with IBU and text.

In addition to our task, we did a simple analysis of the scraped data for the top 10 beers and Top Beer that is served in most countries.

If we could make a comparison in terms of degrees of difficulty for data extraction of the three tools we used, starting from most troublesome to the easiest, Scrapy would be in the first place as the most complicated and time-consuming, then it would be Selenium and lastly Beautiful Soup as the simplest and least time-consuming. Beautiful Soup is a powerful library that parses specifically HTML and is extremely good for structured HTML, it is the most user-friendly of the three and very simple to use.

SCRAPPER MECHANICS AND TECHNICAL DESCRIPTION FOR THE OUTPUT

We scrap the same information from one website using 3 different scrappers. Each of the scrappers works uniquely. We notice some similarities and differences from those. All scrappers work perfectly for HTML Parsing and HTTP Programming, but only Selenium which has access to a specific software and we can simulate the human steps automatically.

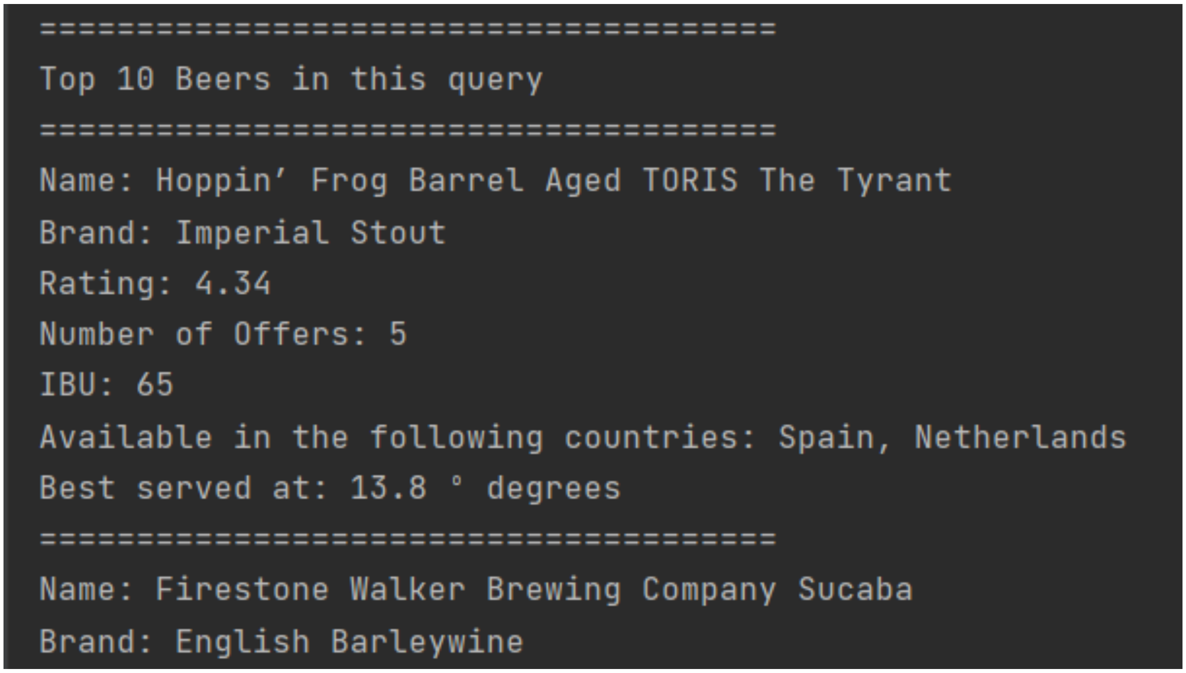
The other thing for the mechanism is about the text pattern matching. This function works in BeautifulSoup and Scrapy, but it's a bit hard on Selenium. Regarding the API connection, Selenium and Scrapy works easily with this functionality, in terms of code, it can be formed in a lean code.

We scrap these information:
Name: beer's brand
Rating: for all over the website
Number of Offers: how many times people ordering it thru website
IBU: Represents bitterness in beers, 0 means no bitterness, 100 means the most bitter
Available country: where we can order or find the particular beer
Best served at: Date of Expiry

Here is the result for the scrappers, and we get the Best 10 beers on the website. In this result, we can make further analysis on what beers are the most favourite one and the selling rate for each brand.

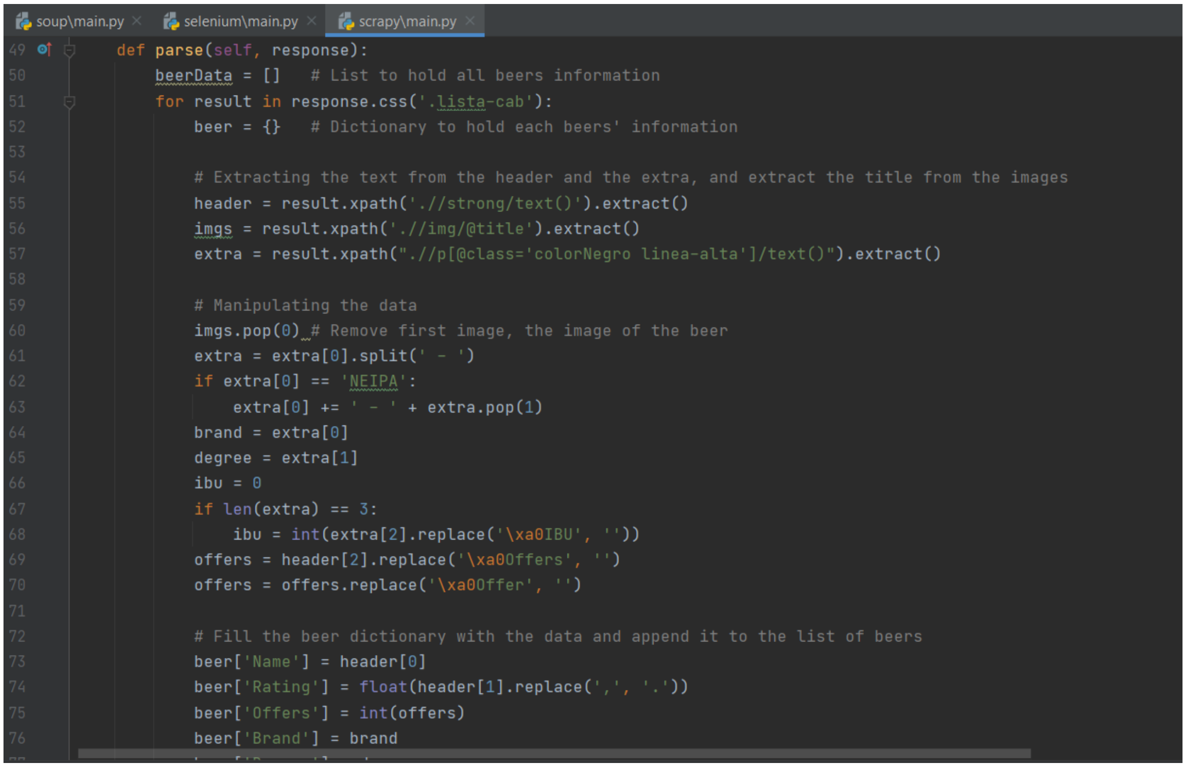
Here we can see the result from BS and Selenium, the code for each scraper is considered lean, but for Scrapy, the code it's more complicated and long. On the other hand, the result given is the same with effective running time.

BeautifulSoup



```
======================================
Top 10 Beers in this query
======================================
Name: Hoppin’ Frog Barrel Aged TORIS The Tyrant
Brand: Imperial Stout
Rating: 4.34
Number of Offers: 5
IBU: 65
Available in the following countries: Spain, Netherlands
Best served at: 13.8 ° degrees
======================================
Name: Firestone Walker Brewing Company Sucaba
Brand: English Barleywine
```

Scrapy



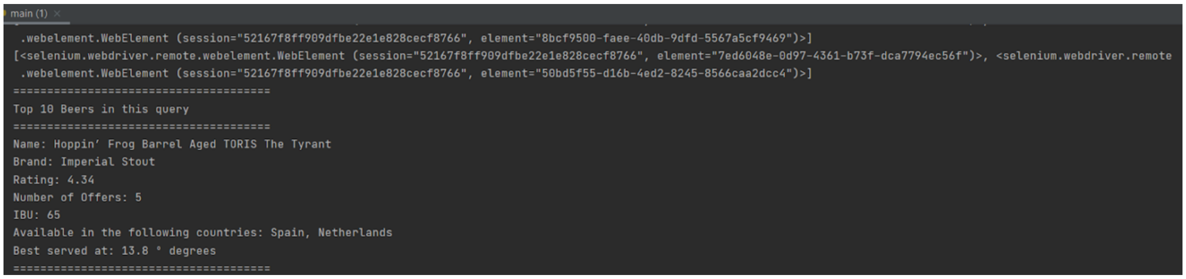
```python
    def parse(self, response):
        beerData = []   # List to hold all beers information
        for result in response.css('.lista-cab'):
            beer = {}   # Dictionary to hold each beers' information

            # Extracting the text from the header and the extra, and extract the title from the images
            header = result.xpath('.//strong/text()').extract()
            imgs = result.xpath('.//img/@title').extract()
            extra = result.xpath(".//p[@class='colorNegro linea-alta']/text()").extract()

            # Manipulating the data
            imgs.pop(0) # Remove first image, the image of the beer
            extra = extra[0].split(' - ')
            if extra[0] == 'NEIPA':
                extra[0] += ' - ' + extra.pop(1)
            brand = extra[0]
            degree = extra[1]
            ibu = 0
            if len(extra) == 3:
                ibu = int(extra[2].replace('\xa0IBU', ''))
            offers = header[2].replace('\xa0Offers', '')
            offers = offers.replace('\xa0Offer', '')

            # Fill the beer dictionary with the data and append it to the list of beers
            beer['Name'] = header[0]
            beer['Rating'] = float(header[1].replace(',', '.'))
            beer['Offers'] = int(offers)
            beer['Brand'] = brand
```

Selenium



```
 .webelement.WebElement (session="52167f8ff909dfbe22e1e828cecf8766", element="8bcf9500-faee-40db-9dfd-5567a5cf9469")>]
[<selenium.webdriver.remote.webelement.WebElement (session="52167f8ff909dfbe22e1e828cecf8766", element="7ed6048e-0d97-4361-b73f-dca7794ec56f")>, <selenium.webdriver.remote
 .webelement.WebElement (session="52167f8ff909dfbe22e1e828cecf8766", element="50bd5f55-d16b-4ed2-8245-8566caa2dcc4")>]
======================================
Top 10 Beers in this query
======================================
Name: Hoppin’ Frog Barrel Aged TORIS The Tyrant
Brand: Imperial Stout
Rating: 4.34
Number of Offers: 5
IBU: 65
Available in the following countries: Spain, Netherlands
Best served at: 13.8 ° degrees
======================================
```

DATA ANALYSIS

With this data, we can perform many analyses, for example, trying to classify different types of beers in the world. We can also see which are the best-ranked beers, besides those with many degrees of alcohol and those that are more bitter, etc.

While using these data, we could advise supermarkets or even bar owners to always stock up on the most popular beers, with this, customers looking for unique types of beer could lead these same bar owners and supermarkets owners to increase sales and profit since they would have a variety of beers that possibly they did not know before.

DETAILED PARTICIPANT

Beautiful Soup was done by Ayax Fabian Díaz Noriega while Selenium was performed by Behbud Hamzayev. Regarding Scrapy, as it was the hardest, we both helped each other.