

Data and Artificial Intelligence

Cyber Shujaa Program

Week 1 Assignment

Web Scraping and Data Handling in Python

Student Name: Deborah Kwamboka Omae

Student ID: CS-DA02 -25075

Introduction

First week's assignment was about web scrapping which simply means automatically extracting data from a website. To achieve this, I used python programming language together with helpful libraries such as requests, BeautifulSoup and pandas. Requests is used to download the HTML content from the website, BeautifulSoup to parse through the HTML content and pandas to organize the data in a structured format. The website had structured data about Hockey team scores and these scrapping was done on Google Colab which is a free cloud-based platform provided by google that lets you write python code in your web browser, in my case Google Chrome. After extracting the data, it was saved as a csv file.

The objectives of the assignment were:

1. Practical Python coding on Jupiter Notebooks hosted on Google Colab
2. Use requests and BeautifulSoup to extract data from a web page.
3. Parse and clean the extracted data.
4. Store structured data into a Pandas DataFrame.
5. Export the final dataset to a .csv file.

Tasks Completed

Step 1: Importing of python libraries used for web scrapping.

I imported the libraries: Requests, BeautifulSoup and pandas to be able to collect data from the website and organize it in a structured format.

The code:

```
from bs4 import BeautifulSoup  
  
import requests  
  
import pandas as pd
```

These lines of code import the given libraries through the keyword 'import' in python

Step2: Fetching the HTML webpage content

The Code:

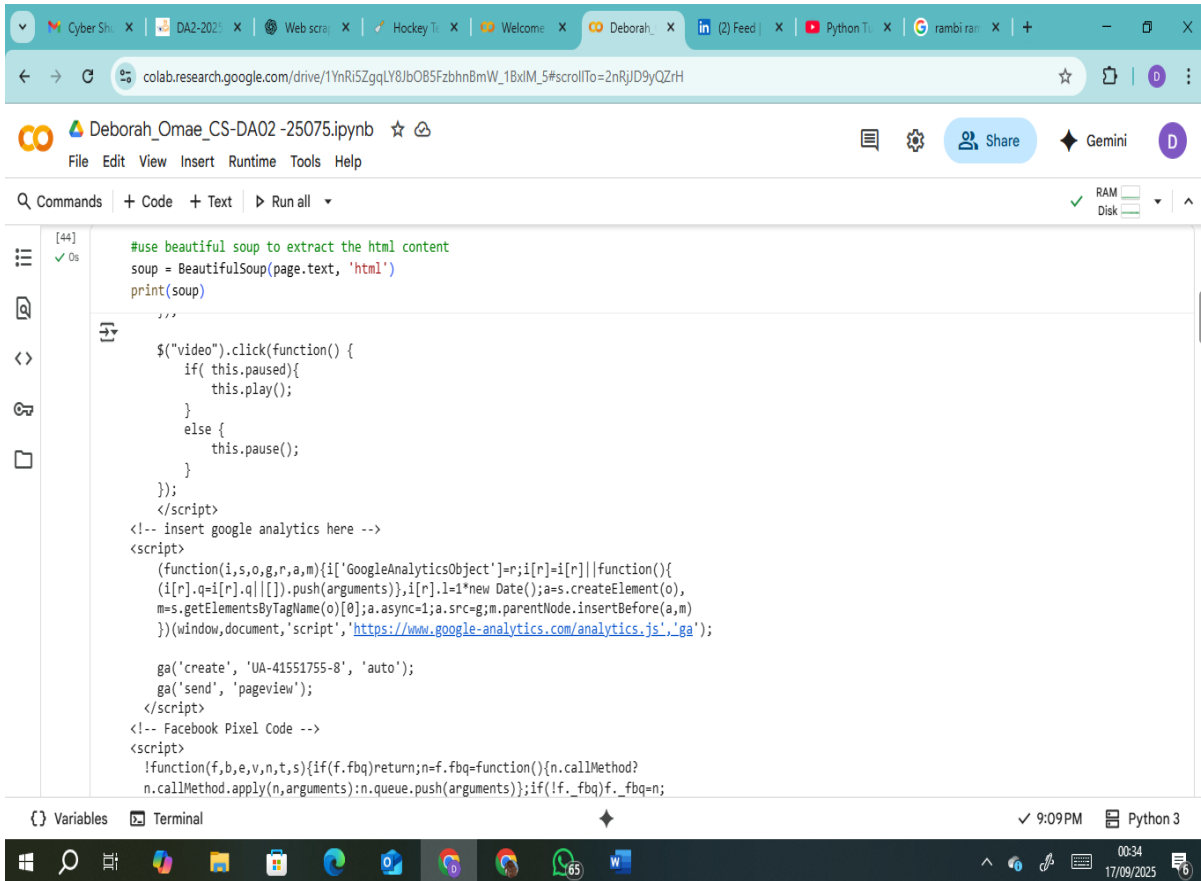
```
url = 'https://www.scrapethissite.com/pages/forms/'  
  
page = requests.get(url)
```

I set the URL of the website we are scrapping in a variable called url. To fetch the html content, the get method of requests is used and the object returned is stored in the variable page. That response object contains the HTML of the page (in .text), the HTTP status code (in. status code) and headers.

Step 3: parsing the html content into a navigable structure

The code:

```
#Use BeautifulSoup to extract the HTML content  
  
soup = BeautifulSoup(page.text, 'html')  
  
print(soup)
```



```
[44] ✓ Os
#use beautiful soup to extract the html content
soup = BeautifulSoup(page.text, 'html')
print(soup)

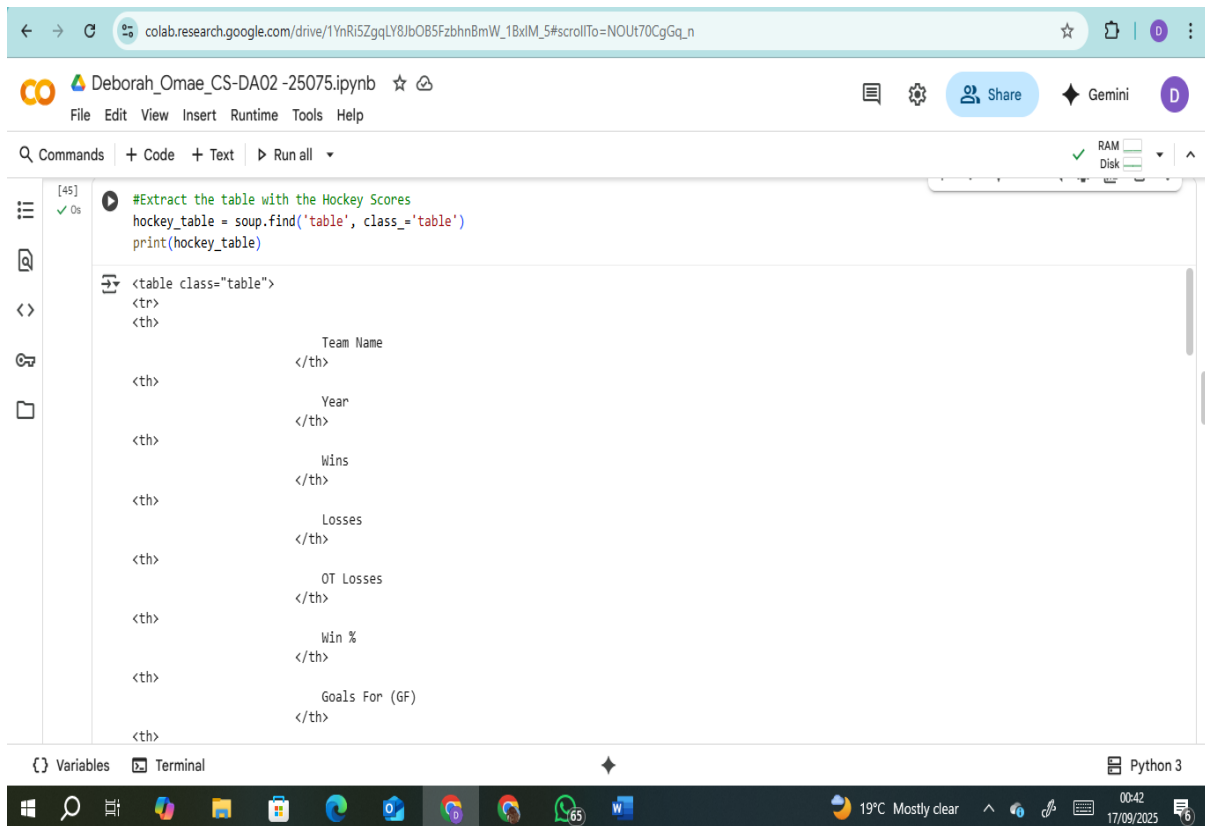
'''
$("video").click(function() {
  if( this.paused){
    this.play();
  }
  else {
    this.pause();
  }
});
</script>
<!-- insert google analytics here -->
<script>
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
})(window,document,'script','https://www.google-analytics.com/analytics.js','ga');

ga('create', 'UA-41551755-8', 'auto');
ga('send', 'pageview');
</script>
<!-- Facebook Pixel Code -->
<script>
!function(f,b,e,v,n,t,s){if(f.fbq)return;n=f.fbq=function(){n.callMethod?
n.callMethod.apply(n,arguments):n.queue.push(arguments)};if(!f._fbq)f._fbq=n;
```

#Extract the table with the Hockey Scores

hockey_table = soup.find('table', class_='table')

print(hockey_table)



The screenshot shows a Google Colab notebook interface. The browser address bar displays a URL from colab.research.google.com. The notebook title is "Deborah_Omae_CS-DA02 -25075.ipynb". The code editor shows a Python snippet that uses BeautifulSoup to find a table by class and print its HTML structure. The output shows the HTML for a table with 11 columns: Team Name, Year, Wins, Losses, OT Losses, Win %, and Goals For (GF). The status bar at the bottom indicates "Python 3" and the system clock shows 00:42 on 17/09/2025.

```
[45] ✓ Os
#Extract the table with the Hockey Scores
hockey_table = soup.find('table', class_='table')
print(hockey_table)

<table class="table">
<tr>
<th>
Team Name
</th>
<th>
Year
</th>
<th>
Wins
</th>
<th>
Losses
</th>
<th>
OT Losses
</th>
<th>
Win %
</th>
<th>
Goals For (GF)
</th>
<th>
```

#Extract the column headings

table_titles = hockey_table.find_all('th')

hockey_table_title = [title.text.strip() for title in table_titles]

print(hockey_table_title)

```

<td class="ga">
278
</td>

[46] ✓ 0s
#Extract the column headings
table_titles = hockey_table.find_all('th')
hockey_table_title = [title.text.strip() for title in table_titles]
print(hockey_table_title)

['Team Name', 'Year', 'Wins', 'Losses', 'OT Losses', 'Win %', 'Goals For (GF)', 'Goals Against (GA)', '+ / -']

[47] ✓ 0s
#Save the column headings onto a Pandas DataFrame
df = pd.DataFrame(columns=hockey_table_title)
df

```

Step 4: storing the rows into a table like structure using pandas' data frame

The code:

#Save the column headings onto a Pandas DataFrame

df = pd.DataFrame(columns=hockey_table_title)

df

#Extract the data row by row. First get all rows, then loop through each while stripping and saving data into the DataFrame

table_data = hockey_table.find_all('tr')

for row in table_data[1:]:

raw_data = row.find_all('td')

each_raw_data = [data.text.strip() for data in raw_data]

print(each_raw_data)

#saving each row data as it is generated into the pandas data frame

length = len(df)

df.loc[length] = each_raw_data

#Inspect the resulting DataFrame

df

The screenshot shows a Google Colab notebook with the following code and output:

```
[47]
#Save the column headings onto a Pandas DataFrame
df = pd.DataFrame(columns=hockey_table_title)
df
```

The output of the above code is a Pandas DataFrame with the following columns: Team Name, Year, Wins, Losses, OT Losses, Win %, Goals For (GF), Goals Against (GA), and + / -. The first three rows of data are:

Team Name	Year	Wins	Losses	OT Losses	Win %	Goals For (GF)	Goals Against (GA)	+ / -
Boston Bruins	1990	44	24		0.55	299	264	35
Buffalo Sabres	1990	31	30		0.388	292	278	14
Calgary Flames	1990	46	26		0.575	344	263	81

```
[48]
#Extract the data row by row. First get all rows, then loop through each while stripping and saving data into the DataFrame
table_data = hockey_table.find_all('tr')
for row in table_data[1:]:
    raw_data = row.find_all('td')
    each_raw_data = [data.text.strip() for data in raw_data]
    print(each_raw_data)

#saving each row data as it is generated into the pandas data frame
length = len(df)
df.loc[length] = each_raw_data

#Inspect the resulting DataFrame
df
```

The output of the above code shows the first three rows of data being printed and then added to the DataFrame:

```
['Boston Bruins', '1990', '44', '24', '', '0.55', '299', '264', '35']
['Buffalo Sabres', '1990', '31', '30', '', '0.388', '292', '278', '14']
['Calgary Flames', '1990', '46', '26', '', '0.575', '344', '263', '81']
```

I extracted the data row by row then looped through each while stripping and saving the data into a data frame. The `.loc` in pandas is used to access or insert rows and columns by their labels, and in the above code it's adding each new scraped row into the DataFrame at the next index. The data is now in a structured format.

Step 5: storing the data frame to disk (.csv file)

The code:

df.to_csv(r'./Hockey.csv')

Link to Code:

https://colab.research.google.com/drive/1YnRi5ZgqLY8JbOB5FzbhnBmW_1BxIM_5?usp=s
haring

Conclusion

This first week I have had an insightful introduction to the history of AI and how data plays an important role in the emergence and development of AI. Job roles in the field of data and AI were highlighted. Starting my first project on web scrapping is an eye opener of the data science methodology in particular, the fetching of data and organizing it into a well-structured format. I have posted my writeup on my blog and I look forward to building a portfolio that I can showcase on my CV as I look for jobs in Data and AI.