

Intro to Data Science

Assignment # 1

Question 1

Write code for a web scraper in Python (preferably a Jupyter Notebook) to extract the 'title', 'year', 'duration', and 'IMDB rating' for all-time top 250 movies from the IMDB website (URL given below). Once you have the data you need, export it to a CSV file (tabular format).

Solution:

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
url = 'https://www.imdb.com/chart/top/'
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36',
    'Accept-Language': 'en-US,en;q=0.9',
}
response = requests.get(url, headers=headers)
html_content = response.content
soup = BeautifulSoup(html_content, 'html.parser')
movies_data = []
movies_container = soup.find('ul', class_='ipc-metadata-list ipc-metadata-list--dividers-all sc-cvbbAY cVhFZB compact ipc-metadata-list--base')
if movies_container:
    movie_items = movies_container.find_all('li', class_='ipc-metadata-list-summary-item sc-10233bc-0 iherUv cli-parent')
    for movie in movie_items:
        title = movie.find('h3', class_='ipc-title__text').text if movie.find('h3', class_='ipc-title__text') else 'N/A'
        metadata_items = movie.find_all('span', class_='cli-title-metadata-item')
        if metadata_items:
            year = metadata_items[0].text if len(metadata_items) > 0 else 'N/A'
            duration = metadata_items[1].text if len(metadata_items) > 1 else 'N/A'
        else:
            year, duration = 'N/A', 'N/A'
        rating = movie.find('span', class_='ipc-rating-star--imdb').text.strip() if movie.find('span', class_='ipc-rating-star--imdb') else 'N/A'
        movies_data.append([title, year, duration, rating])
    df_movies = pd.DataFrame(movies_data, columns=['Title', 'Year', 'Duration', 'IMDB Rating'])
    df_movies.to_csv('imdb_top_250_movies_updated.csv', index=False)
    print("Scraping completed and saved to imdb_top_250_movies_updated.csv")
else:
```

```
print("Movies container not found.")
```

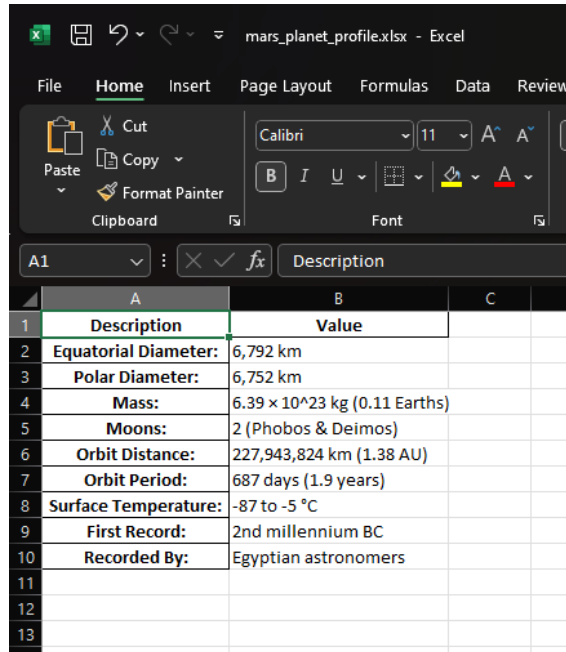
Question 2

Create a web scraper in Python (preferably a Jupyter Notebook) to fetch the 'Mars Planet Profile' data from the URL given below. You must save the data in an Excel format (tabular data).

Solution:

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
url = 'https://space-facts.com/mars/'
response = requests.get(url)
if response.status_code == 200:
    soup = BeautifulSoup(response.text, 'html.parser')
    mars_table = soup.find('table')
    df_list = pd.read_html(str(mars_table))
    mars_df = df_list[0]
    mars_df.columns = ['Description', 'Value']
    mars_df.set_index('Description', inplace=True)
    mars_df.to_excel('mars_planet_profile.xlsx', engine='openpyxl')
    print("Mars Planet Profile data has been saved to\n'mars_planet_profile.xlsx'.")
else:
    print("Failed to fetch the webpage. Status code:", response.status_code)
```

Output:



The screenshot shows an Excel spreadsheet titled "mars_planet_profile.xlsx". The ribbon is set to "Home", and the font settings are Calibri, size 11. The table has two columns: "Description" and "Value". The data rows are as follows:

	A	B	C
	Description	Value	
1			
2	Equatorial Diameter:	6,792 km	
3	Polar Diameter:	6,752 km	
4	Mass:	6.39×10^{23} kg (0.11 Earths)	
5	Moons:	2 (Phobos & Deimos)	
6	Orbit Distance:	227,943,824 km (1.38 AU)	
7	Orbit Period:	687 days (1.9 years)	
8	Surface Temperature:	-87 to -5 °C	
9	First Record:	2nd millennium BC	
10	Recorded By:	Egyptian astronomers	
11			
12			
13			