Northeastern University

# Project 1

# Web Data Extraction / Web scraper

INFO 7390
Fall 2023

Presented by Nupoor Korde

# Introduction

The provided notebook addresses the task of web scraping eBay's laptop listings to collect essential product information, including titles, prices, conditions, and types. The primary objective is to automate the data extraction process and structure it into a CSV file for analysis and reference.

**Problem Definition:**

The objective is to collect extensive laptop listing data from eBay for the purpose of enabling price comparisons and market analysis, with a primary focus on extracting product details.

**Scope:**

The code's scope is limited to extracting and structuring product information such as titles, prices, conditions, and types, then exporting it to a CSV file.

**Objective:**

This notebook’s motivation is to automate data collection from eBay, reducing manual effort and offering a structured dataset for versatile applications like market research, price tracking, and trend analysis, enhancing efficiency and user convenience.

# Outline

- Dataset: The dataset includes laptop listings from eBay, comprising product titles, prices, conditions, and types. It's organized in a Pandas DataFrame, facilitating data analysis.

- Goal: Web scrape eBay's laptop listings to gather product details and then format and store this data in a CSV file for analysis and future reference.

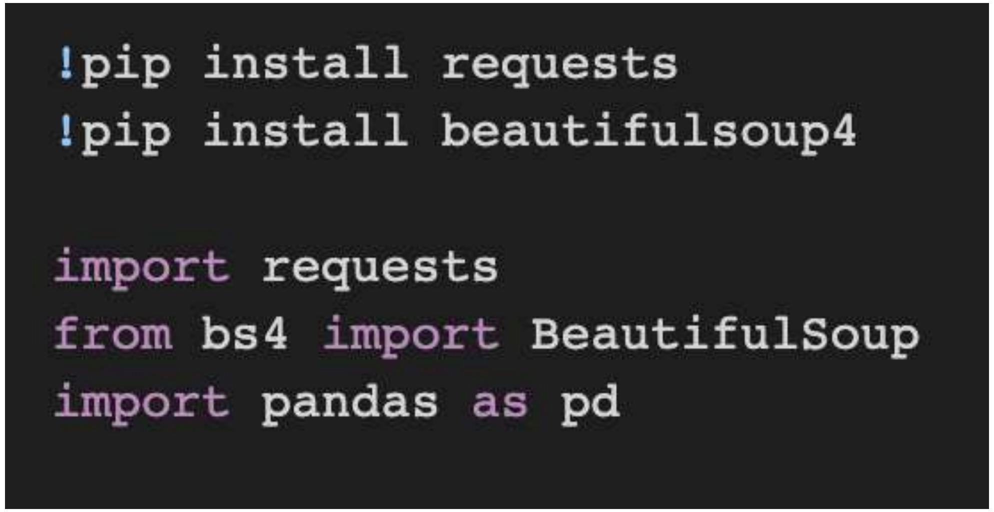
- Result: Successfully scraped eBay for product details, exported to 'ebay_laptop_listings.csv' with a download option, and displayed in a Pandas DataFrame for review.

- Created a notebook using google colab https://colab.research.google.com/

# Methodology

### Environment Setup:

Tool Installation: The code begins by installing the necessary Python libraries using pip, which include 'requests' for making HTTP requests and 'beautifulsoup4' for HTML parsing.

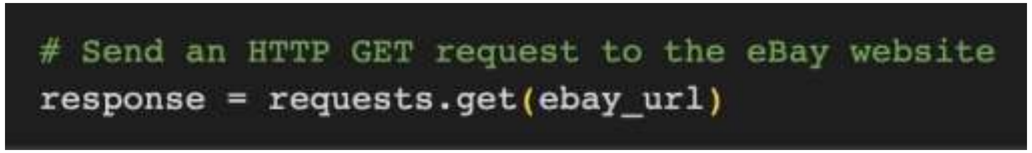


```
!pip install requests
!pip install beautifulsoup4

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

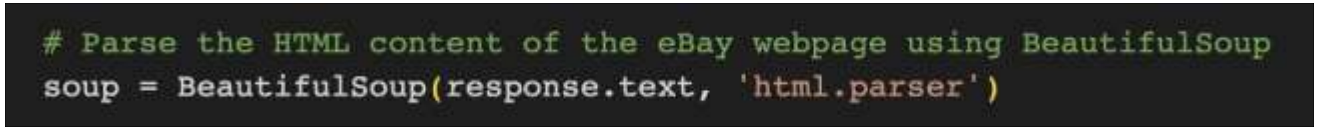
## Tools Used:

Requests: This Python library is employed to send HTTP GET requests to the eBay website, enabling the retrieval of webpage content.



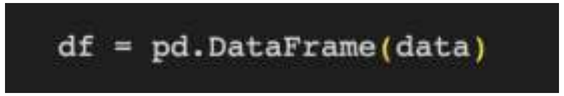
```
# Send an HTTP GET request to the eBay website
response = requests.get(ebay_url)
```

BeautifulSoup (from bs4 module): BeautifulSoup is used for parsing and navigating the HTML content of the eBay webpage, simplifying the extraction of specific data elements.



```
# Parse the HTML content of the eBay webpage using BeautifulSoup
soup = BeautifulSoup(response.text, 'html.parser')
```

Pandas: The Pandas library is utilized to create a structured DataFrame, which serves as the storage format for the extracted product data.

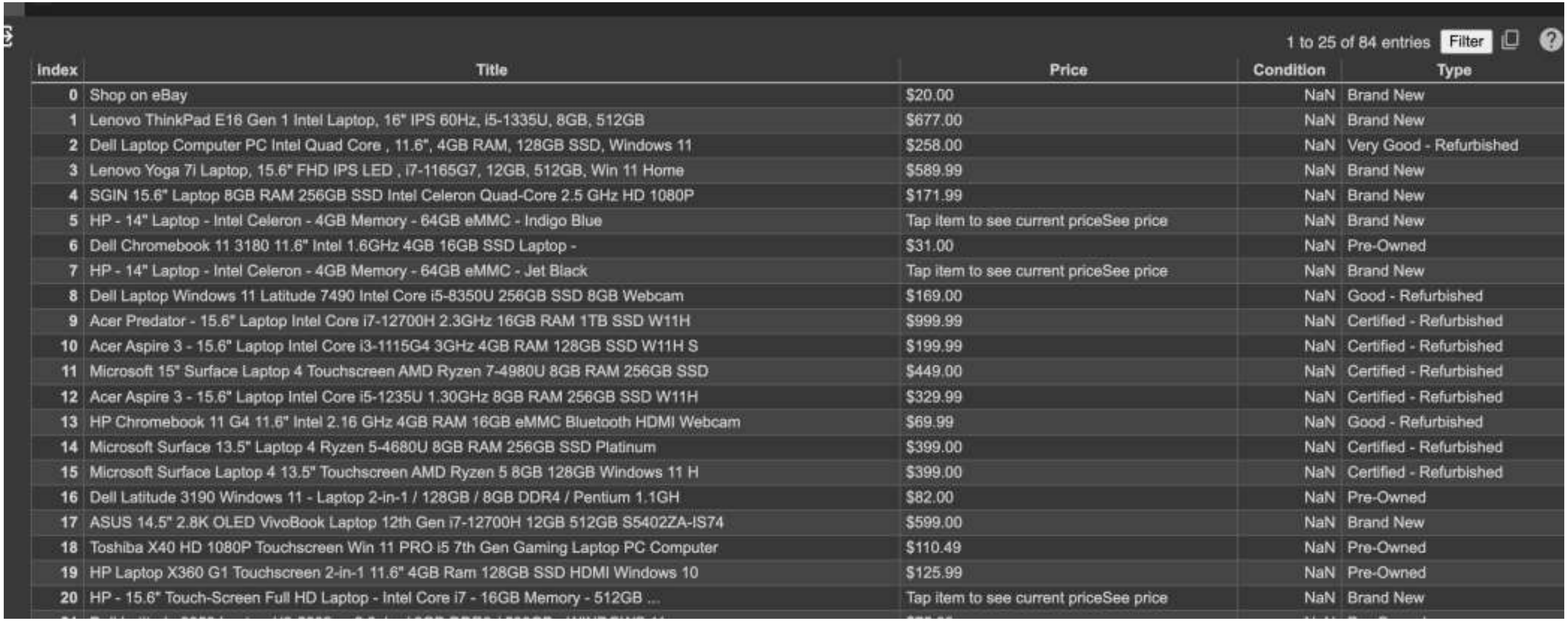


```
df = pd.DataFrame(data)
```

# Dataset

- The dataset contains information related to laptop listings available on eBay.
- It includes columns such as 'Title' (product titles), 'Price' (listing prices), 'Condition' (product conditions), and 'Type' (laptop types or secondary information).
- The data is organized in a structured tabular format using a Pandas DataFrame, making it suitable for data analysis.
- The dataset is intended to facilitate tasks such as price comparison, market analysis, and trend tracking for laptop listings on eBay. Users can export this data to a CSV file for future reference or analysis.

- Following is the snippet for the generated dataset in my notebook.



| index | Title | Price | Condition | Type |
|---|---|---|---|---|
| 0 | Shop on eBay | $20.00 | NaN | Brand New |
| 1 | Lenovo ThinkPad E16 Gen 1 Intel Laptop, 16" IPS 60Hz, i5-1335U, 8GB, 512GB | $677.00 | NaN | Brand New |
| 2 | Dell Laptop Computer PC Intel Quad Core , 11.6", 4GB RAM, 128GB SSD, Windows 11 | $258.00 | NaN | Very Good - Refurbished |
| 3 | Lenovo Yoga 7i Laptop, 15.6" FHD IPS LED , i7-1165G7, 12GB, 512GB, Win 11 Home | $589.99 | NaN | Brand New |
| 4 | SGIN 15.6" Laptop 8GB RAM 256GB SSD Intel Celeron Quad-Core 2.5 GHz HD 1080P | $171.99 | NaN | Brand New |
| 5 | HP - 14" Laptop - Intel Celeron - 4GB Memory - 64GB eMMC - Indigo Blue | Tap item to see current priceSee price | NaN | Brand New |
| 6 | Dell Chromebook 11 3180 11.6" Intel 1.6GHz 4GB 16GB SSD Laptop - | $31.00 | NaN | Pre-Owned |
| 7 | HP - 14" Laptop - Intel Celeron - 4GB Memory - 64GB eMMC - Jet Black | Tap item to see current priceSee price | NaN | Brand New |
| 8 | Dell Laptop Windows 11 Latitude 7490 Intel Core i5-8350U 256GB SSD 8GB Webcam | $169.00 | NaN | Good - Refurbished |
| 9 | Acer Predator - 15.6" Laptop Intel Core i7-12700H 2.3GHz 16GB RAM 1TB SSD W11H | $999.99 | NaN | Certified - Refurbished |
| 10 | Acer Aspire 3 - 15.6" Laptop Intel Core i3-1115G4 3GHz 4GB RAM 128GB SSD W11H S | $199.99 | NaN | Certified - Refurbished |
| 11 | Microsoft 15" Surface Laptop 4 Touchscreen AMD Ryzen 7-4980U 8GB RAM 256GB SSD | $449.00 | NaN | Certified - Refurbished |
| 12 | Acer Aspire 3 - 15.6" Laptop Intel Core i5-1235U 1.30GHz 8GB RAM 256GB SSD W11H | $329.99 | NaN | Certified - Refurbished |
| 13 | HP Chromebook 11 G4 11.6" Intel 2.16 GHz 4GB RAM 16GB eMMC Bluetooth HDMI Webcam | $69.99 | NaN | Good - Refurbished |
| 14 | Microsoft Surface 13.5" Laptop 4 Ryzen 5-4680U 8GB RAM 256GB SSD Platinum | $399.00 | NaN | Certified - Refurbished |
| 15 | Microsoft Surface Laptop 4 13.5" Touchscreen AMD Ryzen 5 8GB 128GB Windows 11 H | $399.00 | NaN | Certified - Refurbished |
| 16 | Dell Latitude 3190 Windows 11 - Laptop 2-in-1 / 128GB / 8GB DDR4 / Pentium 1.1GH | $82.00 | NaN | Pre-Owned |
| 17 | ASUS 14.5" 2.8K OLED VivoBook Laptop 12th Gen i7-12700H 12GB 512GB S5402ZA-IS74 | $599.00 | NaN | Brand New |
| 18 | Toshiba X40 HD 1080P Touchscreen Win 11 PRO i5 7th Gen Gaming Laptop PC Computer | $110.49 | NaN | Pre-Owned |
| 19 | HP Laptop X360 G1 Touchscreen 2-in-1 11.6" 4GB Ram 128GB SSD HDMI Windows 10 | $125.99 | NaN | Pre-Owned |
| 20 | HP - 15.6" Touch-Screen Full HD Laptop - Intel Core i7 - 16GB Memory - 512GB ... | Tap item to see current priceSee price | NaN | Brand New |

# Solution

**Handling of HTML and CSS:**

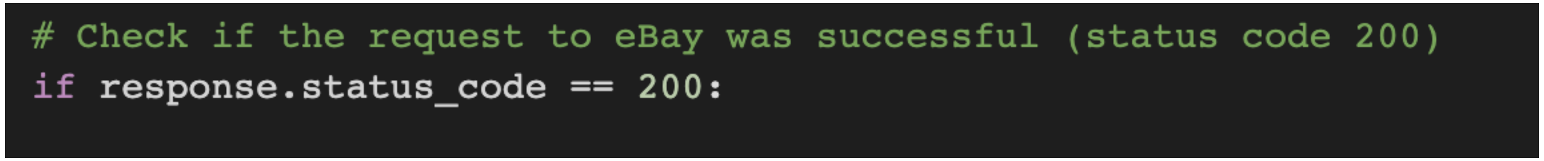
- The code utilizes the 'BeautifulSoup' library to parse and navigate the HTML content of the eBay webpage. It targets specific HTML elements with defined classes ('class_') to extract product details such as titles, prices, conditions, and types.

**Pagination or Dynamic Content:**

- The code focuses on scraping a single eBay page; however, eBay listings can span multiple pages. To handle pagination or dynamic content, additional code would be required to iterate through multiple pages, updating the URL accordingly.

## Handling Errors or Rate Limit Issues:

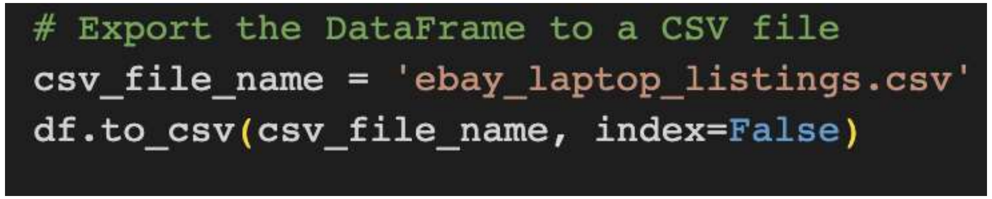
- The code checks the HTTP response status code (status_code) to ensure a successful request to eBay (status code 200). If an error occurs or the request is unsuccessful, it prints an error message along with the status code.



```python
# Check if the request to eBay was successful (status code 200)
if response.status_code == 200:
```

## Data Storage and Format:

- The extracted product data is stored in a structured format using a Pandas DataFrame. This format is suitable for data manipulation and analysis.
- The code exports the DataFrame to a CSV file ('ebay_laptop_listings.csv') for storage and future reference.



```python
# Export the DataFrame to a CSV file
csv_file_name = 'ebay_laptop_listings.csv'
df.to_csv(csv_file_name, index=False)
```

Overall, the notebook efficiently scrapes eBay laptop listings, handles basic error checking, and stores the data in a readily usable CSV format.

# Results

- The code effectively extracts essential product information, including titles, prices, conditions, and types, from eBay's laptop listings page using web scraping techniques.
- It processes the data and creates a structured dataset, organized into a Pandas DataFrame.
- The dataset is exported to a CSV file named 'ebay_laptop_listings.csv,' providing a convenient format for further analysis or archival purposes.
- While the code includes an option to download the CSV file, the primary focus is on data extraction and organization rather than advanced analysis or visualization.
- Lastly, the code successfully reads and displays the extracted data in a Pandas DataFrame within the notebook, offering immediate insights into the eBay laptop listings.

# Challenges and Outlook

**Challenges Faced:**

- HTML Structure Changes: One potential challenge is that eBay's HTML structure or class names may change over time, causing the code to break. Regular maintenance and updates may be needed to adapt to such changes.
- Pagination Handling: The code currently focuses on a single eBay page. To address pagination, future enhancements could include automating the process of iterating through multiple pages of listings.
- Dynamic Content: If eBay's website utilizes more dynamic content loading techniques, the code may need adjustments to handle this efficiently.

- Error Handling: While basic error handling is in place, more robust error handling could be implemented to manage various types of HTTP responses or connection issues.

### How Roadblocks Were Addressed:

- The code currently relies on the HTML structure found during development. Regular monitoring and updating may be required to handle any HTML structure changes.

- For pagination or dynamic content loading, further development would be necessary to navigate through multiple pages or handle dynamic content.

- To enhance error handling, more comprehensive error-checking mechanisms can be integrated to address various potential issues, ensuring robustness.

## Future Enhancements:

- Pagination Handling: Implementing a mechanism to navigate through multiple pages of eBay listings automatically would make the solution more comprehensive.
- Error Handling: Enhancing error handling to cover a wider range of potential issues and providing informative error messages for troubleshooting.
- Data Storage: Exploring options beyond CSV, such as databases, to efficiently manage and access larger datasets.
- User Interaction: Adding user-defined search parameters and options to customize the web scraping process for different eBay searches.
- Scheduled Scraping: Implementing scheduled scraping to regularly update the dataset with fresh eBay listings.

# Conclusion

- In summary, the provided code accomplishes the task of scraping eBay's laptop listings and structuring the data into a dataset. It effectively utilizes Python libraries for web scraping and data manipulation. Future improvements could enhance its capabilities in handling pagination, dynamic content, error management, and data storage options. This code serves as a valuable foundation for web scraping projects and data collection from eCommerce websites like eBay.

# References

- https://www.crummy.com/software/BeautifulSoup/bs4/doc/
- BeautifulSoup + Requests | Web Scraping in Python