## Web Scraping- -Data Analysis-1.Business Problems - E-commerce- Whar data???? 2.Data collection--API(App programming Interface), Databases -SQL Web Scrapping-- Collect the data from Website, automatically • Salenium. Python - (BeatuifulSoup, Scrapy) 1. Ensure Ethical and Legal Compliance:

## Respect the terms of service to avoid any legal issues.

• Check the website's robots.txt file for allowed paths and restrictions.

- Use a Structured Scraping Process:
- 2.Identify the URL structure and relevant data fields.
- Implement a scraping script using Python and BeautifulSoup. • Introduce delays between requests to avoid overloading the server.

## **Project Structure**

- Introduction
- Web Scraping Data Cleaning and Preparation
- Data Visualization

Data Analysis

- Conclusion
- Web Scraping and Data Analysis of Book Prices

1. Introduction

1. Collect data using web scraping.

## **Web Scraping for Auction Data**

In [15]: import requests from bs4 import BeautifulSoup import pandas as pd

In this project, we will scrape data from the books.toscrape.com website to analyze book prices. We will:

import time

'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0

Step 2: Define the URL and Headers

2. Clean and prepare the data. 3. Analyze trends in book prices.

4. Visualize the data to uncover insights.

• Step 1: Import Necessary Libraries

- In [16]: base\_url = 'http://books.toscrape.com/catalogue/page-{}.html' headers = {
  - Step 3: Function to Scrape Auction Data

In [17]:

items = soup.find\_all('article', class\_='product\_pod') data = []

for item in items:

data.append({

def scrape\_auction\_data(page\_number): url = base url.format(page number)

if response.status\_code == 200:

response = requests.get(url, headers=headers)

title = item.h3.a['title']

for page in range(1, 6): # Example for first 5 pages all\_auction\_data.extend(scrape\_auction\_data(page))

4 Sapiens: A Brief History of Humankind 54.23

In [20]: auction\_df.to\_csv('auction\_data.csv', index=False)

'title': title, 'price': price

soup = BeautifulSoup(response.content, 'html.parser')

price = item.find('p', class\_='price\_color').text.strip()

time.sleep(1) # Adding a delay between requests to prevent overloading the server

title price

To identify trends in buyer behavior, item valuation, and market demand for an auction house, I would:

Gather historical auction data, including buyer information, item details, and auction outcomes.

title price

A Light in the Attic 51.77

Tipping the Velvet 53.74

Soumission 50.10

Sharp Objects 47.82

Soumission 50.10 Sharp Objects 47.82

Tipping the Velvet 53.74

print('Failed to retrieve data') return [] • Step 4: Scrape Data with Rate Limiting

all auction\_data = []

else:

In [18]: # Scrape multiple pages

}) return data

auction\_df['price'] = auction\_df['price'].str.replace('f', '').astype(float) print(auction\_df.head()) A Light in the Attic 51.77

Step 5:Save the data to a CSV file

**Data Collection:** 

In [19]: auction\_df = pd.DataFrame(all\_auction\_data)

Step 5:Convert to DataFrame

 Use statistical methods and visualization techniques to identify trends. import matplotlib.pyplot as plt In [21]:

import seaborn as sns

Step1:Load the data

auction data

0

2

3

In [22]:

In [23]:

Out[23]:

**Data Analysis:** 

Clean and preprocess the data.

auction data = pd.read csv('auction data.csv')

Layered: Baking, Building, and Styling Spectac... 40.11

Judo: Seven Steps to Black Belt (an Introducto...

Sapiens: A Brief History of Humankind 54.23 Lumberjanes Vol. 3: A Terrible Plan (Lumberjan... 19.92

98 99 In the Country We Love: My Family Divided 22.00

auction\_data.head()

In [24]:

Out[24]:

In [25]:

Out[25]:

100 rows × 2 columns

0 1

2

3

(100, 2)

to Solve Supper!

plt.ylabel('Item') plt.show()

In [ ]:

• Step5:Identify market demand trends (e.g., distribution of item prices)

• Step4:Plotting the average price by item In [31]: plt.figure(figsize=(15, 10)) plt.title('Average Price by Auction Item') plt.xlabel('Average Price (£)')

The Past Never Ends The Pioneer Woman Cooks: Dinnertime: Comfort Classics, Freezer Food, 16-Minute Meals, and Other Delicious Ways 56.41 Name: price, dtype: float64

The Death of Humanity: and the Case for Life 58.11

auction\_data['price'] = auction\_data['price'].astype(float)

Soumission 50.10 Sharp Objects 47.82 4 Sapiens: A Brief History of Humankind 54.23 auction\_data.shape

Tipping the Velvet 53.74

title price A Light in the Attic 51.77

Step2:Data Cleaning

 Step3:Identify trends in buyer behavior (e.g., average price by item) In [28]: average\_price\_by\_item = auction\_data.groupby('title')['price'].mean().sort\_values(ascending=False) print(average\_price\_by\_item.head())

Slow States of Collapse: Poems 57.31 Our Band Could Be Your Life: Scenes from the American Indie Underground, 1981-1991

sns.barplot(x=average\_price\_by\_item.values, y=average\_price\_by\_item.index)

L'Ayelded: Beawher, et January & Archydr Sheki Mesaerion: The Barrelass City (this San he Gutsy Girl: Escapades for Your Life Plain Dumb Mistalan

en City: The Majesty of Pigeons, the Discreet Charm of Spails & Other Wonders of the U Spark Joy: An Illustrated Master Class on the Art of Organizing

The Inefficiency Assassin: Time Management Tactics for Mooking The Age of Gentus: Time Seventee in Century and the Bile The Coming Woman: A Novel Based on Hardenens A Practical Suite to Perso The Life-Changing Magic of Tidying Wiri The IABARESE Art of Declutering and Grac Outcast, Vol. 1: A Darkness Surrounds Him Tolkie Mama Tried: Traditional Italian Cooking for the Screwed Fride in the Princess Jellyfish 2-in-1 Ompilities void with the Screwed Fride in the Princess Jellyfish 2-in-1 Ompilities of the Princess Jellyfish 2-in-1 Ompilities of the Princess Jellyfish World Awai

Judo: Seven Steps to Black Beit (an Introducto

Rat Queens, Vol. 3: Demons (Rat Queens, 1918) oot Diet olange in the schilling a Baye Beslaim Grange. And Eschbiglies Lumberjanes Vol. 2: Friendship to the Max Lumberiale Natural History of Us (The Hing a)

In [32]: plt.figure(figsize=(10, 6)) sns.histplot(auction\_data['price'], bins=20, kde=True, color='blue') plt.title('Distribution of Auction Item Prices') plt.xlabel('Price (£)') plt.ylabel('Frequency') plt.show() Distribution of Auction Item Prices 10 8 6 equency

Average Price by Auction Item

4 2 10 20 30 40 50 60 Price (£) Regards --Tejas Patil--