Data Project Case Study

Nike Clothing Web Scraping & Analysis

Gitta Ataeimanesh

1. Project Overview

Goal: Analyse Nike clothing product data from the web to identify trends in pricing and availability.

→

Scope: Web scraping → Data cleaning → Analysis → Visualisation.

2. Tools & Technologies

- Python

- BeautifulSoup & Selenium (Web Scraping)

- Pandas & NumPy (Data Wrangling)

- Power BI & Tableau (Visualisation)

- Jupyter Notebook

- GitHub

3. Step-by-Step Process

1. Scraped product data (title, price, stock) from Nike website.

2. Cleaned and structured the data using Pandas.

3. Conducted exploratory data analysis (EDA).

4. Created dashboards in Power BI and Tableau.

5. Interpreted trends and documented findings.

4. Challenges & Solutions

Challenge: Website structure changed during scraping.

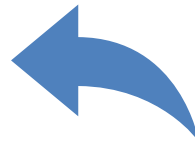Solution: Adapted script using Selenium to handle dynamic content.

Challenge: Duplicate entries & missing data.

Solution: Used Pandas functions to clean and validate datasets.

5. Results & Insights

- Identified popular price ranges for specific clothing categories.

- Highlighted inventory gaps based on product availability.

- Suggested pricing strategies based on seasonal patterns.
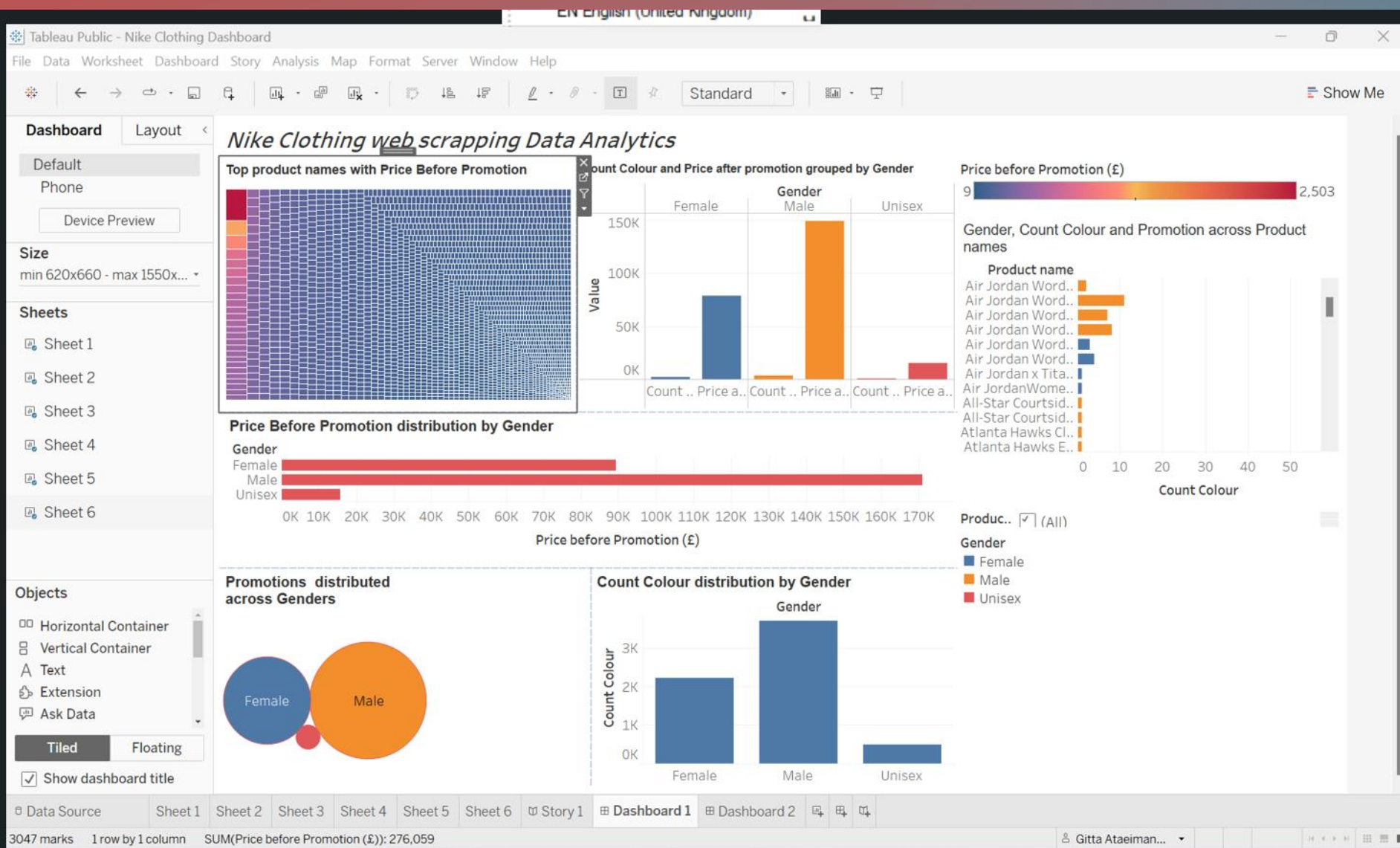
6. What I'd Improve

- AUTOMATE DATA SCRAPING WITH SCHEDULED JOBS.

- INTEGRATE DATA FROM ADDITIONAL SOURCES (E.G., COMPETITORS).

- BUILD PREDICTIVE MODELS FOR SALES FORECASTING.

```python
import time
import csv # Import the csv module
from selenium import webdriver
from bs4 import BeautifulSoup

# Set up the Selenium driver
driver = webdriver.Chrome()  # You need to have the Chrome driver installed

url = "https://www.nike.com/gb/w/clothing-3rauvz5e1x6z6ymx6znik1"
driver.get(url)

# Scroll to the bottom of the page to trigger content loading
scroll_pause_time = 2
scroll_height = 0
while True:
    driver.execute_script(f"window.scrollTo(0, {scroll_height});")
    time.sleep(scroll_pause_time)
    scroll_height += 1000  # Adjust the value based on your preference and page behavior

    # Break the loop when you've scrolled to the end of the page
    if scroll_height >= driver.execute_script("return Math.max( document.body.scrollHeight, document.body.offsetHeight, document.documentElement.clientHeight, document.documentElement.scrollHeight,
document.documentElement.offsetHeight );"):
        break

# Get the updated page source after scrolling
html_content = driver.page_source
driver.quit()

# Parse the updated HTML content using BeautifulSoup
soup = BeautifulSoup(html_content, "html.parser")

# Extract clothing details
clothing = []
for item in soup.find_all("div", class_="product-card__body"):
    product_name = item.find("div", class_="product-card__titles").text
    count_colour = item.find("div", class_="product-card__count-item").text
    price = item.find("div", class_="product-card__price").text
    gender = item.find("div", class_="product-card__subtitle").text

    print(product_name,count_colour, price, gender)

    clothing.append([product_name, count_colour, price, gender])


 # Write the extracted data to a CSV file
with open("Nike_clothing.csv", "w", newline="", encoding="utf-8") as csvfile:
    csvwriter = csv.writer(csvfile)
    csvwriter.writerow(["Product_name", "Count_Colour", "Price", "Gender"])
    csvwriter.writerows(clothing)

print("Data extracted and saved to Nike_clothing.csv")
```