**Report on Myntra Women’s Clothing Data Collection & Analysis**

**1. Introduction**

The goal of this project was to **collect, clean, analyze, and visualize data** from Myntra’s women’s clothing section, specifically kurtas.  
The project demonstrates the end-to-end data pipeline:

* Automated **web scraping** of product information.
* **Data cleaning & preprocessing** to prepare structured datasets.
* **Exploratory data analysis (EDA)** to uncover insights about pricing, ratings, brands, and discounts.
* **Data visualization** to present findings in a clear and interpretable manner.

This report summarizes the workflow, findings, challenges, and conclusions from the project.

**2. Data Collection Process**

Data was collected directly from Myntra using **Selenium WebDriver** in Python. The scraping process was divided into two parts:

1. **Collecting product URLs:**

* Extracted product links using CSS selectors.
* Handled pagination by dynamically clicking the “Next” button.
* Stored the extracted links in a text file (myntra\_products.txt).

1. **Scraping product details:**

* For each product URL, details such as **brand, product name, price, MRP, rating, number of reviews, category, and URL** were extracted.
* Regular expressions were used to clean numeric fields (price, MRP, reviews).
* Browser automation tricks (e.g., disabling automation flags, setting user-agents) were applied to avoid detection.
* Data was saved incrementally into a CSV file (myntra\_products.csv).

This pipeline ensured a structured dataset that could be fed into the next stage: data cleaning & analysis.

**3. Data Cleaning & Preparation**

**Dataset Overview**

* **Total records (products):** 2,313
* **Columns (features):** 8 → Brand, Product\_Name, Price, MRP, Rating, Reviews, Category, URL
* **Unique brands:** 237
* **Categories:** 1 (focused only on *Women’s Clothing – Kurtas/Kurtis/Suits*)

Before analysis, several preprocessing steps were performed:

* **Duplicate removal:** Ensured no repeated product entries.
* **Handling missing values:** Replaced with NaN where data was unavailable.
* **Numeric conversions:** Converted Price, MRP, and Rating into numeric datatypes.
* **Standardized brand names:** Ensured consistency.
* **Discount percentage calculation:** Added a derived column:

\text{Discount %} = \frac{MRP - Price}{MRP} \times 100

This produced a clean dataset ready for statistical and visual analysis.

**4. Key Findings from Analysis**

**4.1 Descriptive Statistics**

* **Prices**: Mean price was significantly lower than MRP, reflecting heavy discounting.
* **Ratings**: Most products clustered between 4.0–4.5, indicating generally positive customer feedback.

**4.2 Brand Analysis**

* Certain brands (e.g., Sangria, Libas, Anouk – depending on dataset size) dominated in terms of product count.
* Top 5 brands together contributed a significant share of total listings.

**4.3 Discount Analysis**

* Some brands consistently offered **higher discounts (40–70%)**, positioning themselves as “value-for-money.”
* Others maintained lower discounts, possibly relying on brand reputation or premium positioning.

**4.4 Category Insights**

* **Price ranges varied by category:**
  + - Kurtas showed the widest spread of prices.
    - Kurta sets and suits were priced higher on average.
* **Ratings were stable across categories**, with only slight variation.
* Box plots revealed **outliers**, suggesting luxury/premium items co-exist with budget-friendly options.

**4.5 Visualizations**

* **Histogram**: Prices followed a right-skewed distribution , most products in the ₹500–₹2000 range.
* **Bar Chart**: Clear differentiation in discount strategies among brands.
* **Box Plot**: Categories exhibited different price spreads; kurtas had lower medians than sets.
* **Scatter Plot**: Weak correlation between discount percentage and ratings , meaning discounts did not strongly influence customer satisfaction.

**5. Challenges and Solutions**

1. **Dynamic content loading (JavaScript):**
   * Challenge: Elements did not load instantly.
   * Solution: Used WebDriverWait with explicit conditions to wait for elements.
2. **Anti-bot detection by Myntra:**
   * Challenge: Website occasionally blocked automated browsing.
   * Solution: Added custom **user-agent**, disabled Selenium automation flags, and introduced time delays between requests.
3. **Pagination handling:**
   * Challenge: “Next” button selectors changed across pages.
   * Solution: Implemented multiple fallback selectors and JavaScript-based clicks.
4. **Data inconsistencies:**
   * Challenge: Missing values in MRP, ratings, or reviews.
   * Solution: Replaced with NaN and standardized formats during cleaning.

**6. Conclusion**

This project successfully demonstrated the **end-to-end workflow of a data science pipeline** on e-commerce data:

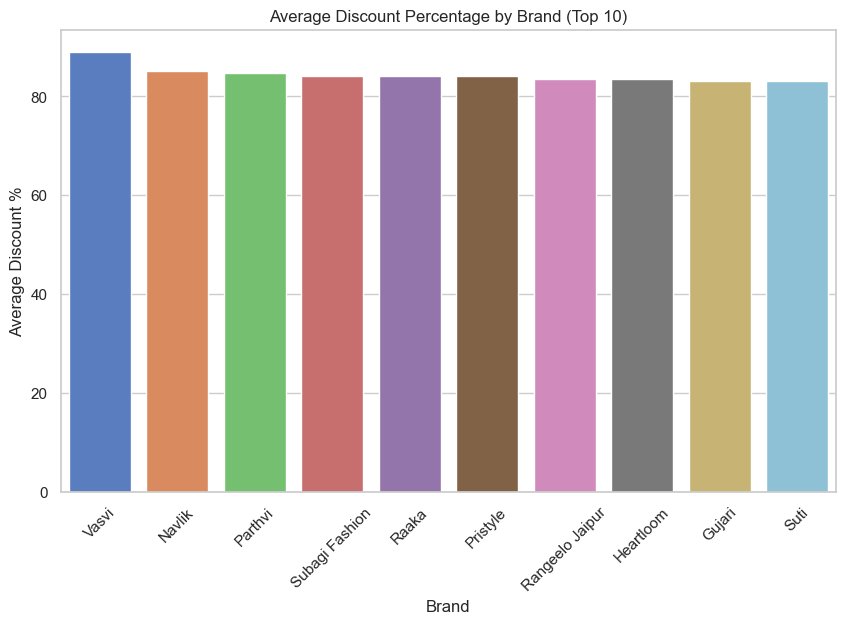
* **Web scraping** enabled the automated collection of product-level data.
* **Data cleaning & preprocessing** ensured high-quality structured datasets.
* **Exploratory data analysis & visualization** uncovered meaningful insights into brand strategies, pricing, discounts, and customer ratings.
  1. **Data Visualization**

**Histogram – Distribution of product prices**

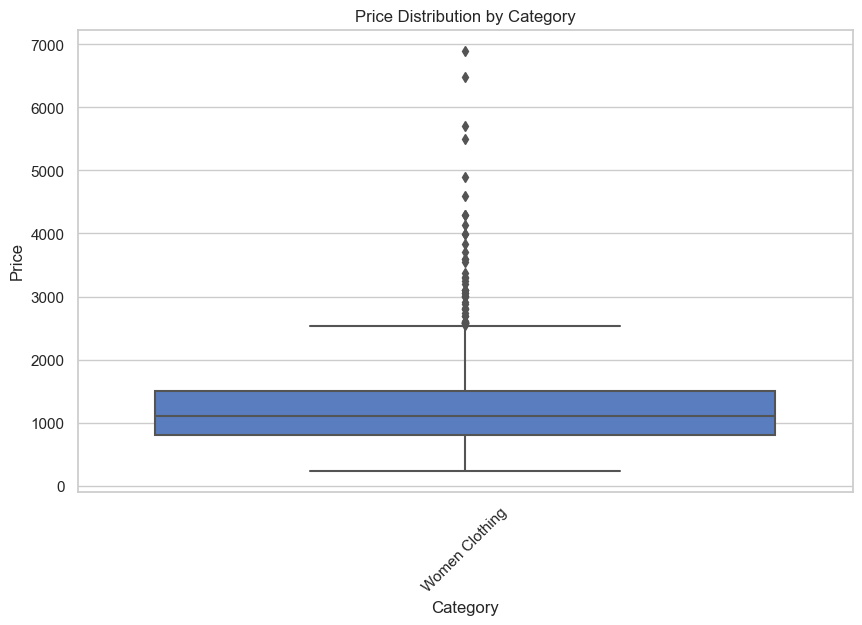
**A graph showing a distribution of product prices

Description automatically generated**

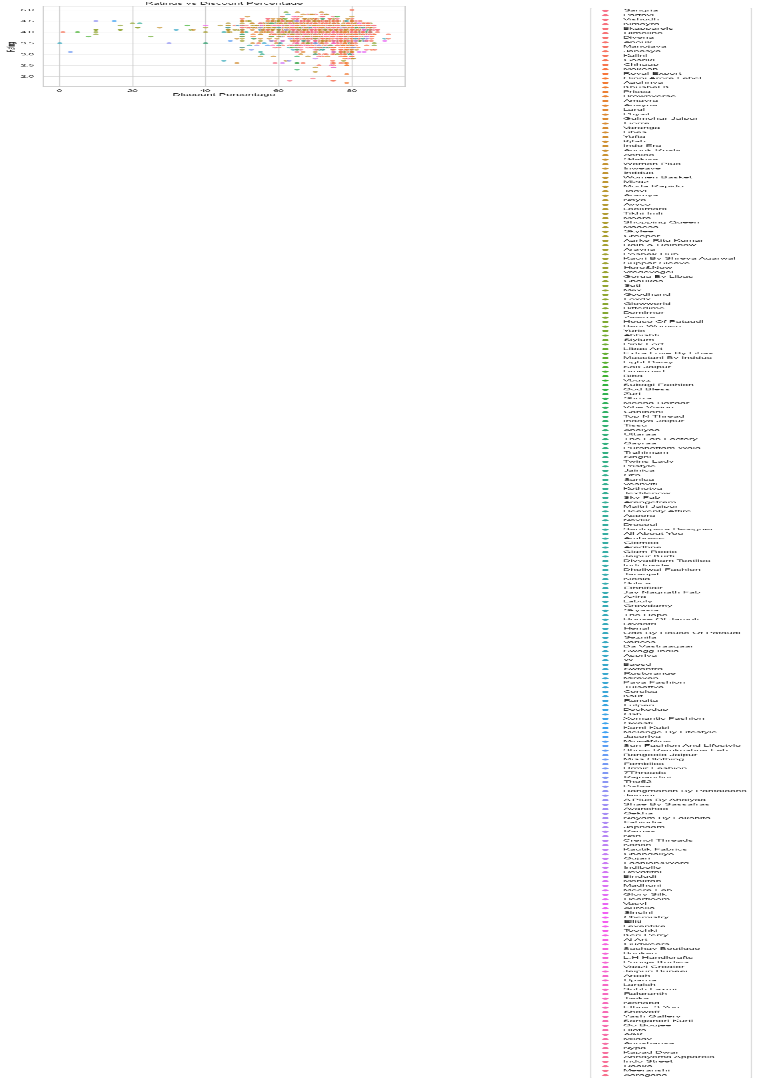
**Bar chart – Average discount percentage by brand (Top 10 brands)**

****

**Box plot – Price distribution across categories**

****

**Scatter plot – Ratings vs Discount Percentage**

****