Data Cleaning & Preprocessing: *cleaning data collected from saya.pk using selenium*

1. Handling Missing Values

When scraping certain fields such as price or availability were not present on a product page. In such cases I handled missing values by filling them with "Not Available" so that the dataset remained consistent and no rows had blanks. In the data I collected there is no missing values

```
cleaned_df = cleaned_df.fillna({
    "Title": "Not Available",
    "Price": "0",
    "Barcode": "Not Available",
    "Availability": "Out of Stock",
    "Available_Quantity": "0",
    "Link": "Not Available"
})
```

2. Data Type Conversion

The prices I scraped were in text format such as "Rs. 4,590 PKR". I converted them into numeric values 4590.00 to make calculations like averages and comparisons possible.

```
cleaned_df["Price"] = cleaned_df["Price"].astype(str).str.replace(",", "")
cleaned_df["Price"] = cleaned_df["Price"].apply(lambda x: re.sub(r"[^\d.]", "", x) if pd.notna(x) else None)
```

3. Text Cleaning

Some product titles contained unnecessary extra spaces promotional words. I cleaned the text by removing unwanted characters and standardizing spacing.

```
cleaned_df["Availability"] = cleaned_df["Availability"].str.strip().str.title()
cleaned_df["Availability"] = cleaned_df["Availability"].replace({"In Stock": "In Stock", "Out Of Stock": "Out of Stock"
```

4. Normalization & Standardization

The variations like "InStock", "instock", and "In Stock". inorder to normalized into a single standard format "In Stock" or "Out of Stock". Similarly, I standardized prices so they all followed one consistent structure.

```
cleaned_df["Availability"] = cleaned_df["Availability"].str.strip().str.title()
cleaned_df["Availability"] = cleaned_df["Availability"].replace({"In Stock": "In Stock", "Out Of Stock": "Out of Stock"})
```

5. Duplicates and Outliers Handling

Sometimes the scraper captured the same product more than once when a product appeared in different collections. I removed duplicate rows based on product name and barcode.

```
cleaned_df = cleaned_df.drop_duplicates(subset=["Barcode", "Title"], keep="first")
```

6. Encoding Categorical Data

For future machine learning tasks, categorical fields needed to be encoded. Therefore I converted "In Stock" to 1 and "Out of Stock" to 0. This numeric representation makes it easier to run computations.

```
cleaned_df["Availability_Encoded"] = cleaned_df["Availability"].map({"In Stock": 1, "Out of Stock": 0})
```

7. Feature Engineering

I also created some new feature that is Calculated word count of product titles to analyze product naming.Extracted collection type (stitched/unstitched) from product URLs.

```
cleaned_df["Title_Word_Count"] = cleaned_df["Title"].apply(lambda x: len(str(x).split()))
cleaned_df["Extracted_Collection"] = cleaned_df["Link"].apply(lambda x: x.split("/")[3] if isinstance(x, str) and "/" i
```

To handle extreme/high outliers I have used a filter to filter out product with extreme

```
cleaned_df = cleaned_df[cleaned_df["Price"] < 200000]
```

8. Merging Datasets

Since I scraped different categories like stitched, unstitched, kids, men, and accessories, I combined all these datasets into a single final dataset. This gave me one clean and complete dataset containing all the products.