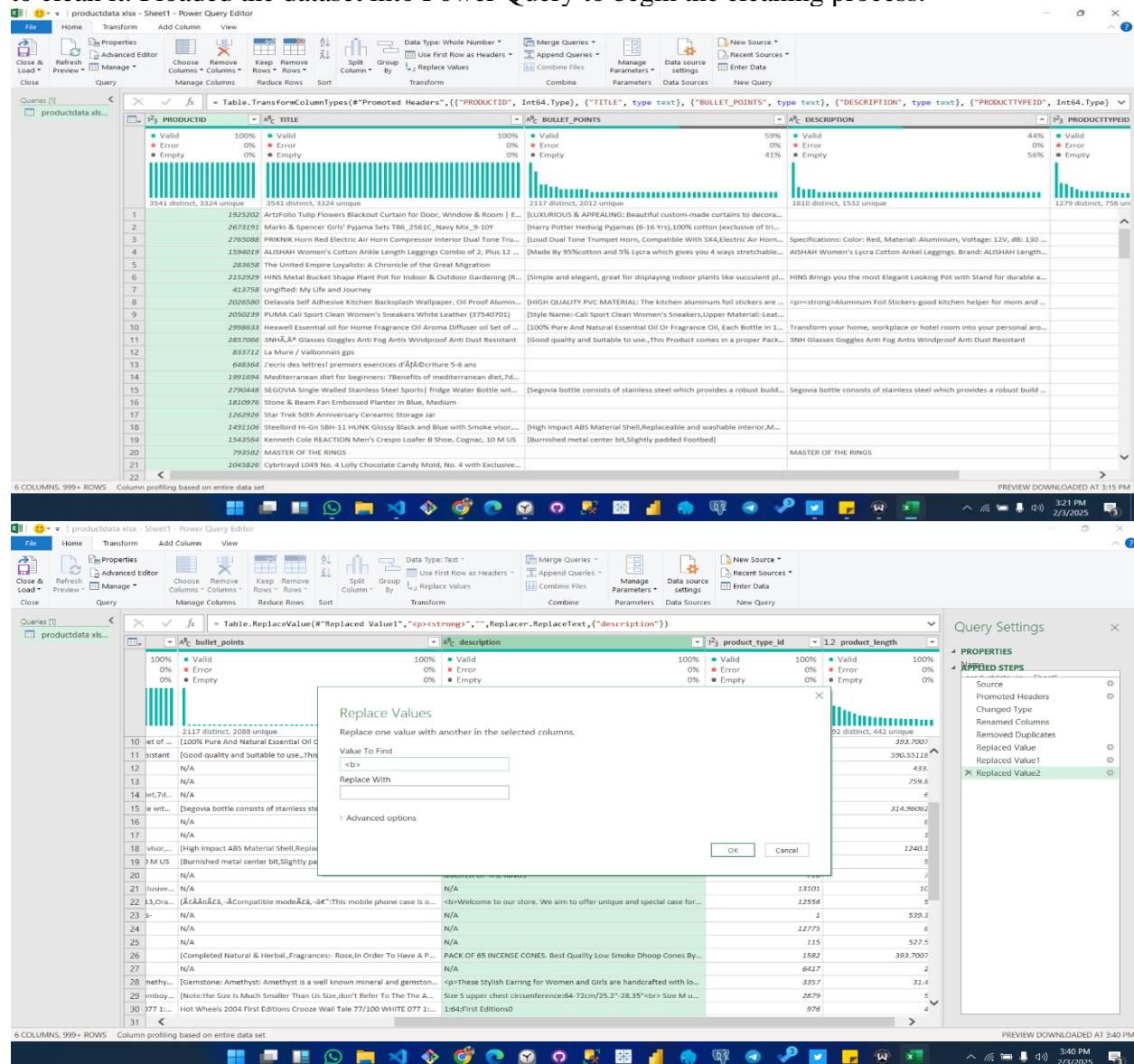


# Technical Report: Data Cleaning & Title Optimization By Adordev

## Introduction

The goal of this project was to clean raw marketing data, address data quality issues, and create a concise, SEO-optimized `short_title` feature to improve product discoverability.

I initially downloaded the dataset with the intention of using **Excel**, specifically **Power Query**, to clean it. I loaded the dataset into Power Query to begin the cleaning process.



However, as the dataset grew larger and the cleaning process became more complex, it became tedious to handle solely in Excel. I realized that using **Python** would be more efficient for automating repetitive tasks. I wrote a Python script to handle the bulk of the data cleaning process, which significantly improved both speed and accuracy.



The dataset contained the following key columns:

- **product\_id**
- **title**
- **bullet\_points**

- **description**
- **product\_type\_id**
- **product\_length**

The primary focus was on cleaning the text-based columns (`title`, `bullet_points`, and `description`) and optimizing product titles for marketing purposes.

---

## Data Cleaning Process

### 1. Handling Missing Values

- **Issue Identified:** Missing values were present in the `bullet_points` and `description` columns.
- **Solution:** Replaced missing values (`NaN`) with empty strings to ensure smooth processing.
- **Code Used:**

```
data.fillna('', inplace=True)
```

---

### 2. Removing Duplicates

- **Issue Identified:** Potential duplicate product entries were identified.
- **Solution:** Removed all duplicate records to maintain data integrity.
- **Code Used:**

```
data.drop_duplicates(inplace=True)
```

---

### 3. Standardizing Data Formats

- **Actions Taken:**
  - Removed unwanted characters such as `/`, `(`, `:`, and `)`.
  - Removed special characters like `é`, `ü`, etc., instead of converting them to ASCII.
  - Unescaped HTML entities to clean up the text.
- **Code Used:**

```
text = re.sub(r'[\\"/()]:]', '', text) # Remove unwanted
characters
text = re.sub(r'^\x00-\x7F]+', '', text) # Remove non-ASCII
characters
```

---

## Creating the `short_title` Feature

### Objective:

Generate concise product titles (30–50 characters) that retain essential information for SEO and readability.

### Methodology:

1. **Removed Redundant Phrases:**
  - Phrases like "Set Of 2", "Pack Of 4", and "Classic Reprint" were removed.
2. **Extracted Key Attributes:**
  - Attributes such as product size (e.g., 9–10Y), color (e.g., Navy), and quantity (e.g., 2 PCS) were retained.
3. **Intelligent Truncation:**
  - Applied smart truncation to ensure titles did **not cut off mid-word**.
4. **Fallback Mechanism:**
  - If the cleaned title became too short, the script reverted to the original title (up to 50 characters).

---

## Example Transformations

Original Title	Short Title
ALISHAH Women's Cotton Ankle Length Leggings Combo 2, Plus 12 Colors_L	ALISHAH Women's Cotton Ankle Length Leggings
PosterHub Pink Floyd Wall Poster Matte Finish Paper Print 12 x18 Inch Multicolor HS - P076	PosterHub Pink Floyd Wall Poster Matte Finish
Oxza Universal Foldable Stand Holder Mount Bracket Tablet, Cell, Mobile Phone Table Stand Mobile Holder	Oxza Universal Foldable Stand Holder Mount

---

## Clean Dataset Overview

Metric	Before Cleaning	After Cleaning
Total Rows	3,847	3,541 (duplicates removed)
Missing Values	Present	Handled (replaced with empty strings)
Duplicate Entries	Present	Removed
Unwanted Characters	Present	Cleaned
short_title Feature	Not Present	Added

---

## Conclusion

In conclusion, the data cleaning process involved:

- Addressing missing values
- Removing duplicates
- Standardizing data formats
- Optimizing product titles for SEO

The `short_title` feature improves readability and search performance while retaining essential product information.