Paula Ceprián (u198630), Judit Viladecans (198724), Berta Noguera (u199893)

Final Project - Part 1

Text Processing and Exploratory Data Analysis

GitHub: https://github.com/juditvribe/IRWA_2025_G14_FinalProject.git
TAG: IRWA-2025-part-1

PART 1: Data preparation

To pre-process the documents, we implemented the function called **build_terms()** that performs lowercasing, punctuation and number removal, stop-word filtering, tokenization and stemming.

We observed that the field **product_details** contained a list of dictionaries, each representing key-value attribute pairs (e.g., {"Fabric":"Cotton Blend"}). In order to make these attributes searchable for future queries, we implemented a function called **flatten_product_details()** that flattens the list into a space-separated string. This ensures that attribute values remain available for indexing in the future.

After flattening, we applied the **build_terms()** function to all relevant textual fields such as: brand, category, description, seller, title, product_details and sub_category. In our implementation, all these fields were pre-processed independently of each other rather than merged into a single text field:

```
for col in fashion_df_text.columns:
  fashion_df_text[col] = fashion_df_text[col].apply(lambda line : build_terms(line))
```

We intentionally kept these fields separate so that they could later be **indexed as independent fields in the inverted index and optionally combined or weighted differently during retrieval.** We chose this approach because preserving the distinct semantics of each field allows us to assign higher importance to some fields, like for example title, than others such as seller or description. With that approach we'll have disadvantages such as slightly higher storage and indexing cost or requiring a retrieval engine that supports field handling.
The other approach might have a simpler implementation or might ensure that all tokens are searchable without field management but at the end it has some limitations as for example: semantic separation is lost, field weights cannot be applied, or that there is a high risk of noise (less important terms being overweighted).

To take also into account the numeric or non-textual fields such as: pid, out_of_stock, selling_price, discount, actual_price, average_rating and url, we decided to separate them into another DataFrame to not index them as textual terms so that we can preserve them for future quantitative ranking or filtering without being affected by text preprocessing. Then both subsets were concatenated and ordered to facilitate readability and usage.

PART 2: Exploratory Data Analysis

We conducted an Exploratory Data Analysis on the fashion products dataset to examine its textual and numerical characteristics, distribution patterns, and possible data-quality issues.

Word Counting Distribution

We computed global word frequencies across the fields title and description only, since metadata such as brand or seller contains very limited vocabularies and would not be informative.
RESULTS: The most frequent tokens corresponded to generic product categories such as “tshirt”, “women”, “men”, “neck”, “print”, “cotton”, “shirt”, “solid”, “round” and “fit”. Their high frequency confirms that the dataset is dominated by apparel items and supports our previous choice to apply inverse-document weighting to counteract this imbalance.

Average Sequence Length

Average token length indicates how concise product titles are compared to descriptions and informs text-normalization or indexing strategies. Typically, titles are short but carry key discriminative information, while descriptions are longer and noisier.
RESULTS:

- Average length of the title: 6.05 words
- Average length of the description: 18.18 words

The large gap between titles and descriptions justifies assigning higher retrieval weight to titles in later experiments.

Vocabulary Size

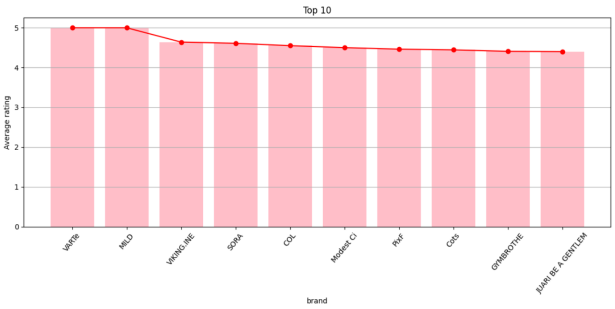
The vocabulary size provides an estimate of lexical diversity after preprocessing.
RESULTS: The corpus contained approximately 5778 unique tokens, with an average of 0.21 unique terms per product. This confirms a balanced but manageable vocabulary for inverted-index construction.

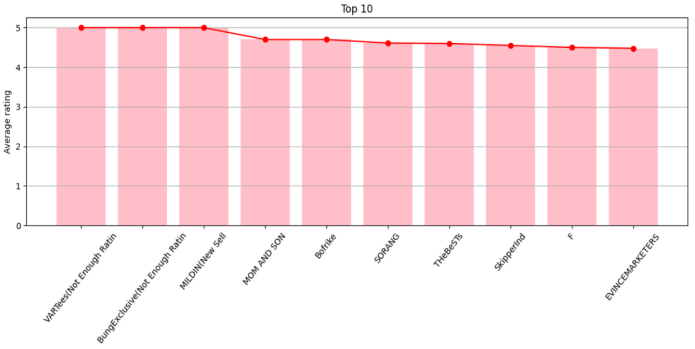
Ranking of products

We examined how numerical attributes differ across sub-categories. Each attribute was converted to a numeric type and cleaned (removing symbols such as “%”, “off”, or commas). By plotting the top 10 sub-categories for each metric, we visualized which product types tend to have higher prices, discounts, or ratings.
RESULTS: Sub-categories such as Crocks Club Clothing and Accessories showed higher selling prices, while Brand Trunk Bags, Wallets & Belts displayed larger discounts.
These variations will later help justify field-specific or attribute-based ranking adjustments (e.g., boosting highly rated items).

Top Brands and Sellers

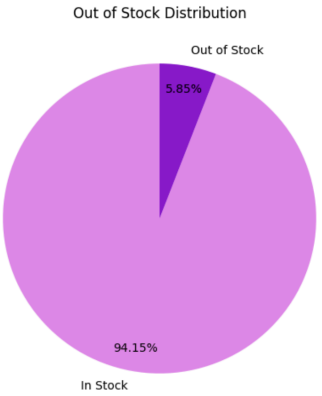
This chart provides insight into product-quality perception and can serve as a reputational signal for ranking brands or sellers.
RESULTS: Both “VARTe” and “MILD” brands dominate the top ratings when it comes to brands, suggesting that brand reputation significantly influences perceived quality. Seller ratings are more uniformly distributed, but there is no significant difference when comparing the top 10 rated brands and sellers.





Out-of-stock distribution

We assessed product availability, as a high out-of-stock proportion might bias retrieval evaluation if unavailable products are retrieved.
RESULTS: Approximately 94.15% of items are in stock and 5.85% are out of stock. This supports the later decision to down-weight or exclude out-of-stock products during ranking.



Word clouds

Word clouds provide an intuitive visualization of the most common tokens after preprocessing. They highlight high-frequency terms.
RESULTS: The most prominent words relate to clothing types confirming that apparel dominates the corpus.

Entity Recognition

To detect named entities such as brands, locations, and product types within the titles.
Although the generic spaCy model is not domain-specific, it helps assess whether off-the-shelf NER can capture meaningful entities.
RESULTS: At the moment there are a lot of false positives due to domain-specific naming conventions. Future work could include training or fine-tuning a custom NER model specialized in product and brand recognition.