Universitat Pompeu Fabra Barcelona

Pau Reig Vaqué - u215101
Marc Rodríguez Vitolo - u215105
Clara Serra Borràs - u204635

REPORT

Part 1: Text Processing and Exploratory Data Analysis

PART 1: Data preparation

To start the project, we first downloaded the JSON document *fashion_products_dataset.json* and loaded it as a DataFrame to make all fields of each fashion product/document easily accessible.

For the first sub-section, following the instructions and the procedure from the first lab, we created a function to process the title and description fields for all documents. This allowed us to generate two new columns containing lists of tokens/words. In addition to the required preprocessing steps, we converted all text to lowercase and removed any special characters. We considered removing accents, but this was unnecessary since the text is in English.

Finally, as we had already merged the tokens from title and description in the practice, we created a column containing this combined information.

In the second sub-section, the focus is on the important product attributes that will be used in the future. Since the JSON is already stored as a DataFrame, these variables are fully accessible throughout the project. We also created a list of the names of these important attributes so that we can easily iterate over them or access their values directly. This setup also allows us to add the new columns containing tokens if needed.

For the third sub-section, we first separated product_details into individual columns to access detailed characteristics of each product: style_code, closure, pockets, fabric, pattern, and color. These attributes can be very useful for the relevance of future queries, and having them separated in different features rather than in a dictionary inside a single feature.

To decide how to handle the fields category, sub_category, brand, product_details (split into product_style_code, product_closure, product_pockets, product_fabric, product_pattern, product_color) and seller, we first looked at the number of unique values for each attribute to understand how specific they are and to assess how they should be treated. We also examined the actual values to better understand the context.

The results were as follows:

Number of unique categories: 4
Number of unique sub-categories: 24
Number of unique brands: 325
Number of unique sellers: 535
Number of unique style codes: 23263
Number of unique closures: 57
Number of unique pockets: 112
Number of unique fabrics: 244
Number of unique patterns: 118
Number of unique colors: 352

From these results, we can conclude that category is not very distinctive, while sub_category is slightly more distinctive. In the EDA part we also discovered that sub-categories are assigned to a category, we don't have more than one possible combination between category and sub_category, so merging them gives us the same information and leaves us with 24 categories. These two can be added to the combined title and description column without removing their individual columns, having a long phrase describing the product. Brand and seller are highly distinctive, which can affect the effectiveness of TF-IDF if added directly to the main text, therefore, we keep them separate for possible filtering.

Regarding the product details, style_code has too many unique values, so it is not suitable for merging into the main text. These values are like codes that hardly ever will be searched in queries. On the other hand, fabric, pattern, and color have a manageable number of unique values and are semantically relevant, so they can affect positively to TF-IDF. Pockets and closure have values that are not likely to appear in user queries, so we leave them separate.

Thus, we decided to create a new column that merges the title, description, category, sub_category, fabric, pattern, and color. This combined column will be very useful when applying TF-IDF in the next stage of the project.

Specific fields (brand, seller, product_style_code, closure, pockets) are kept separate and are intended for filtering or assigning special weights rather than for the current TF-IDF calculation that we are using.

This decision is based on the distinctiveness of the values, fields with few unique values can provide useful context without diluting the main text, while fields with many unique values could introduce noise if merged.

Finally, we processed the columns we considered important so that they could be merged with the title and description. Before merging, we ensured that their values were in the same format as the processed title and description. To achieve this, we applied lowercase conversion, removal of punctuation marks and special characters, tokenization, and stemming, following the same procedure as in the initial function. We also handled missing or empty values.

In the last subsection, we decided not to modify the variable out_of_stock, since it is a boolean that does not provide any semantic value for applying TF-IDF. However, we kept it because it can be very useful for future filtering operations.

Regarding the variables selling_price, actual_price, average_rating, and discount, which contain continuous numerical values, we converted them to INT or FLOAT types. These attributes do not carry semantic meaning as textual terms and would only introduce noise if indexed. On the other hand, keeping them as numerical values can be very helpful for performing filtering or sorting in next stages.

After performing the conversion we saw that some null values appeared where we had empty strings. We decided to look deeper in this because some values could be imputed and would help to get better results. We looked at the rows that had null values in the

discount column, and we realized that they also had the actual_price empty. We assumed that these products did not have a discount so the actual_price was the same as the selling price. To have the data more cleaned we assigned to the actual_price the value of the selling_price, and we set the discount to 0.

PART 2: Exploratory Data Analysis

This second section focuses on analyzing the data to better understand the context presented by the dataset. To do this, we divided the analysis according to the data type, first addressing textual variables and then numerical ones.

Regarding the textual variables, we focused our analysis on the word count distribution and determined the vocabulary size of the title and description once they were processed. These statistics provide insights into the characteristics of the text data and help us understand the values we will work with when applying future algorithms such as TF-IDF.

What we can see from the word count distribution is that the title has a mean of 6.2 words. For text based searching engines we might need more than that. Mixing the title and description will help to get better results with the queries as the description is around 18.5 words long. In great part of the cases (at least in more than 50%) this will be shorter so we will still have a short text for each line. More information should be added, like the color or the brand (as we did previously).

Finally, we also created a word cloud for both the title and the description. From this visualization, we can see that the most common words are those describing materials and characteristics. Some of the most frequent terms appear in both title and description. We can also notice that rather than having words alone what we see more often is two-concepts to define types of clothes or attributes of it.

Regarding the numerical variables, we first used boxplots to visualize their distributions and detect potential outliers, as well as distribution plots to show the frequency of values. The distribution of all four numerical attributes are more or less continuous, they don't have a clear discrete distribution. Almost every item in the dataset has a discount, fewer than 1,000 products have a 0% discount, and most are concentrated between 40% and 60%. Both actual and selling prices show wide ranges, but there is a clear clustering around moderate price levels.

Next, we identified the products with the highest ratings to understand which items receive the best user feedback. These top-rated products represent the most appreciated items in the catalog. However, we cannot determine how reliable or meaningful these ratings are, as there is no information about their source. We do not know how many individual reviews contribute to each average rating, whether the ratings reflect the overall purchase experience (involving the seller), or only the product quality (involving the brand). Therefore, we cannot draw definitive conclusions, although we can observe that some brands have multiple products with consistently good ratings.

In addition, we analyzed the brands with the lowest and highest selling prices to identify both the most affordable and the most expensive options. What we can conclude from this analysis is that there is a significant price variation among brands, some are considerably more expensive, while others offer much cheaper products. This is an important aspect to

keep in mind when developing the search engine, as it may be useful to assign different weights to certain types of brands depending on the search context.

Additionally, we identified the products with the highest and lowest discount percentages to understand the most prominent offers. The same brand provides the highest discounts, around 85%. However, at this stage, we cannot draw any further conclusions.

As we said before when justifying how we grouped some features, we did a crosstab to see how the categories and subcategories were grouped. There we could see that each sub category is attributed to a single category.

With the help of some tables, we could acknowledge some things about the sellers and the brands. We saw that there are big sellers in the dataset and that some of the items got null values in this column. Regarding the brands we could see that there are different types of brands, we see that the ones with more sales got a wide range of ratings, some of them with good ratings and others with lower ones. And the same when seeing the brands with better ratings, some of them are big companies with a lot of instances in the dataset, and others with just a few.

GitHub URL: https://github.com/mmarcrv/irwa-search-engine-G005
TAG: IRWA-2025-part-1
AI Usage: We used ChatGPT and Gemini to review our draft explanation of pre-processing decisions and to help us make some plots.