Information Retrieval and Web Analytics

Final Project PART 1
Text Processing and Exploratory Data Analysis

Laura Naranjo & Laura Penalver & Aurora Pujols

October 24, 2025

TAG:	IRWA-2025-part-1
GitHub:	https://github.com/aurorapujols/irwa-search-engine

We separate this part 1 of the project in two parts:

a. **Data Preparation (1)** in which we load the corpus (product's articles) and pre-process the data.

b. **Exploratory Data Analysis (2)**: in which we **study our data** with statistics to understand the dataset.

1 PART 1: Data Preparation

In this section we explain how we decided to treat the data and pre-process it.

AI use

FIELDS: It gave us ideas on how to treat different types of fields. Mainly that some of them can be used for filtering and others for the index terms.

INDEX: In this part, we used ChatGPT to give us an explanation on what fields "indexed as separate fields in the inverted index" means. It showed us different ways of how the inverted index would look like and we chose a mix of the two best options we found (considering code efficiency and understanding, and for its future usage in the ranking).

CODE: All the code for this part is exclusively made by us, with the help of the code we already did in Practice Session 1. With exception of dataframe processing and plotting.

DEBUGGING: We also used AI tools for debugging, specially when dealing with compilation errors and JSON formatting.

We created a file named `data_prep.py` in which we defined all the functions related to the data preparation. The functions are used and tested in the next part (2) when doing the Exploratory Data Analysis.
The first step before processing the data is to load the corpus (1), and we used the function `load_corpus` (provided in the repository) in the function `load_corpus_from_json`. Then, we compute the inverted index (2) and store it in `index` (with some additonal variables, `info_index`, and `metadata`). And, finally, we store the index in a JSON document (3) because the index computation takes a few minutes and it is of easy and faster access to store it and upload it from a JSON.

1.1 Document Preprocessing

To preprocess the text (in `title`, and `description`), we created a function named `join_build_terms` that given one or more strings, it concatenates them, and then process them by doing the following steps:

a. Lowercasing all the text

b. Tokenizing the text

c. Removing punctuation marks

d. Removing stop words

e. Stemming

The function that performs the operations on the text is called `build_terms` (based on the Practice Session 1 code). It is important to notice how we added the concatenation of two strings before the pre-processing. We did it because in the following items, we will see that it is interesting to join different fields' texts.
For instance, if we treat the `title` and `description` as in the Practice Session 1, we can concatenate them and then build the terms (see in Figure 1).

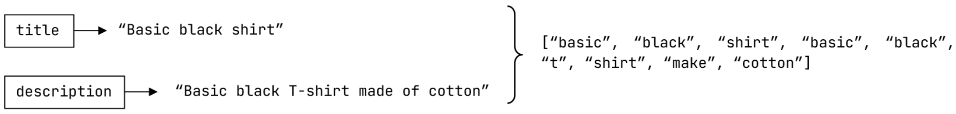

Figure 1: The result of "join_build_terms" function (not accurate regarding the text processing result).

The function `build_terms` is further explained in the README file in the repository.

1.2 Queries output

To take into account that for each retrieved document we need to show the information: `pid`, `title`, `description`, `brand`, `category`, `sub_category`, `product_details`, `seller`, `out_of_stock`, `selling_price`, `discount`, `actual_price`, `average_rating`, `url`; we decided to create two dictionaries from the `pid` of a document to the different data. As we will see in the following items, we are interested in keeping separate the categorical fields from the numerical fields, so we store in two dictionaries `info_index` and `metadata` as shown in Figure 2.

```
info_index[pid] = {
    "title": doc.title,
    "description": doc.description,
    "brand": doc.brand,
    "category": doc.category,
    "sub_category": doc.sub_category,
    "product_details": doc.product_details,
    "seller": doc.seller
    }
```

```
metadata[pid] = {
    "out_of_stock": doc.out_of_stock,
    "selling_price": doc.selling_price,
    "discount": doc.discount,
    "actual_price": doc.actual_price,
    "average_rating": doc.average_rating,
    "url": doc.url
    }
```

Figure 2: The format of the two info dictionaries. "doc" is the JSON Document for a single product's article.

In this way, for any document given by `pid` we can retrieve the information we want easily and efficiently.
The function that gets this information and constructs the dictionaries is `get_articles_info`. To test it, in `data_prep.py` we have loaded the corpus, got the dictionaries from this function, and printed them in JSON files to see their content (with names `metadata_dict.json` and `info_index_dict.json` in the root folder).

1.3 Categorical fields

To handle the categorical fields (`category`, `sub_category`, `brand`, `product_details`, and `seller`), we considered the following:

- We might be interested in keeping some of the terms separate when creating the index by indexing with different fields. This will make it possible to consider different weights when adding to the `tf-idf` of a document. If, for instance, we consider a word in the query appearing in the `category` field more relevant than it appearing in the `description`, then the documents can be ranked accordingly.
- We might not keep all fields separate due to computation restrictions. Maybe less important fields like `product_details` and `seller` could be processed as a single new field in the index.

This is why we decided to consider the structure in Figure 3 for the index.

```
index = {
      “term1”: {
            “field1”: [(pid1, [positions]), (pid2, positions)]
            “field2”: [(pid3, [positions])]
      }
      “term2”: {
            “field1”: [(pid2, [positions])]
            “field3”: [[(pid2, [positions]), (pid3, positions)]
      }
}
```

Figure 3: Index with fields as subindex. "termX" is the terms in the pre-processed fields and "fieldX" are the fields of each document considered in the index ("title"+"document", "bran", "category", "sub_category", "product_details"+"seller").

For `title` and `description`, we will consider them of the same weight if a term appears on them, so we treat it as a single field for the index. The same happens for `product_details` and `seller`. But for the other three categorical fields (`brand`, `category`, and `sub_category`) we will take each of them as a term in the index because we consider them of relatively high and different importance.
In table 1 we can see what pros and cons this implementation has.

Pros	Cons
Helps give more importance to certain fields when ranking the documents.	Makes the indexing and searching process more complicated.
Lets us treat important fields (e.g. brand, category) differently than others we might consider less important (e.g. seller).	Takes more time and resources to build the index and ranking.
Allows more control over how search results are sorted.	Needs extra care to balance the weights between fields (maybe we use weights of important fields that make user preference worse).

Table 1: Caption

1.4 Numerical fields

To handle the numerical fields (`out_of_stock`, `selling_price`, `discount`, `actual_price`, and `average_rating`), we decided to use them as parameters for filtering. For example:

- Use `out_of_stock` as a filter for ranking out of all products that either are in stock or not.
- Use the `discount` as a variable to filter if the products need to have a discount or not.
- Use the `selling_price` and `actual_price` for sorting the final rankings (optional if the user wants to) or even filter out some products.
- Use `average_rating` also for filtering and/or sorting.

But all these terms are not gonna be used in the index, as it does not make sense to include numbers as text (they loose their meaning). And we only store them in the previously mentioned dictionary called `metadata`.

2 PART 2: Exploratory Data Analysis

In this section we provide an exploratory data analysis to describe the dataset.

AI use

EDA: We used ChatGPT to help us code the plots and with debugging.

STREAMLIT: We took the code we made in the jupyter notebook (`test.ipynb`) and asked it to adapt it to a Streamlit web interface. We asked for some further modifications to make it more efficient and understandable.

The first thing we did was take a look at the loaded dataframe (table of data) of the corpus (see in Figure 4.

	pid	title	description	brand	category	sub_category	actual_price	discount	selling_price	average_rating	out_of_stock	product_details	seller
0	TKPFCZ9EA7H5FYZH	Solid Women Multicolor Track Pants	Yorker trackpants made from 100% rich combed c...	York	Clothing and Accessories	Bottomwear	2,999	69% off	921	3.9	False	[{'Style Code': '1005COMBO2'}, {'Closure': 'El...	Shyam Enterprises
1	TKPFCZ9EJZV2UVRZ	Solid Men Blue Track Pants	Yorker trackpants made from 100% rich combed c...	York	Clothing and Accessories	Bottomwear	1,499	66% off	499	3.9	False	[{'Style Code': '1005BLUE'}, {'Closure': 'Draw...	Shyam Enterprises
2	TKPFCZ9EHFCY5Z4Y	Solid Men Multicolor Track Pants	Yorker trackpants made from 100% rich combed c...	York	Clothing and Accessories	Bottomwear	2,999	68% off	931	3.9	False	[{'Style Code': '1005COMBO4'}, {'Closure': 'El...	Shyam Enterprises
3	TKPFCZ9ESZZ7YWEF	Solid Women Multicolor Track Pants	Yorker trackpants made from 100% rich combed c...	York	Clothing and Accessories	Bottomwear	2,999	69% off	911	3.9	False	[{'Style Code': '1005COMBO3'}, {'Closure': 'El...	Shyam Enterprises
4	TKPFCZ9EVXKBSUD7	Solid Women Brown, Grey Track Pants	Yorker trackpants made from 100% rich combed c...	York	Clothing and Accessories	Bottomwear	2,999	68% off	943	3.9	False	[{'Style Code': '1005COMBO1'}, {'Closure': 'Dr...	Shyam Enterprises

Figure 4: Dataframe with the fields of a document. First 5 articles.

From the dataframe, we can see how the numerical fields need to be pre-processed before doing an EDA with them, because they can't be converted to floats with ",", and the discounts are not in percentage (0.XX) format. But we will se more of that later.

In the report, we will now see what we observe in the data through different fields of the dataframe.

NOTE: we did the EDA with the help of a Jupyter notebook (`test.ipynb`), and visualized it better through a `streamlit` web interface (shown how to visualize it in the README file).

2.1 Word Count Distribution

For the fields `title` and `description`, we can count how many words they have and plot their distribution (see Figure 5.

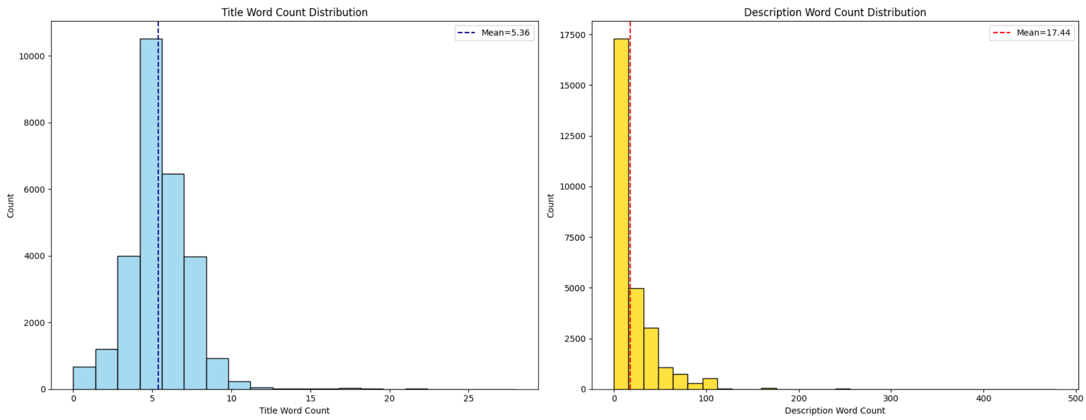

Figure 5: Word count distribution for title (blue) and description (yellow). The dotted line is the mean for the respective distributions.

From the distributions we can see how the `title` word count has a similar distribution to the Gaussian (with most documents having a title of around 5 words), while the `descriptions` seem to have between 0 and ~100 words but most of them have very limited descriptions (around 0 and 20).

In conclusion, `titles` seem to have a very consistent titling style. But most products have very short `descriptions` which means maybe this field is not very significant for most products or they are very general products that are hard to differentiate from the rest.

2.2 Average Sentence Length

For the `descriptions`, we can also look at the average sentence length (see Figure 6).

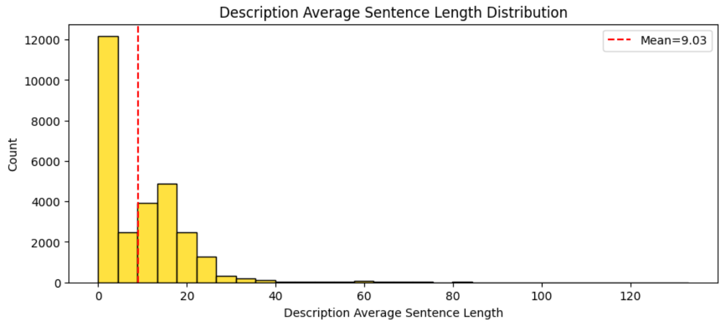

Figure 6: Average sentence length in of the field description.

From the histogram we can see how the sentences in the description are very short and concise. This can mean that the sentences are keyword based and few have detailed descriptions about the product.

2.3 Word Dictionary and Word Cloud

Regarding the categorical fields, we can also look at the vocabulary in our corpus. We can try to look at the total number of words in the documents.

It is important to notice that we are talking about the words after filtering all the fields with `build_terms`, that is, without the stop words and stemming (reduces the amount of words in the dictionary considerably).

We find that the total number of *unique words* is **16,829**. We know there are 28,080 documents so it means that the vocabulary is very limited and specific to the context we are in (clothes, shopping, accessories, etc.). This makes the few words in a document very relevant for its retrieval when trying to match them with the user query.

From all the words in the corpus (not only taking unique words), we can also build a word cloud to visualize the most common words. In the Jupyter notebook (`test.ipynb`) we printed the top 20 words and the amount of times they appear throughout the corpus. An interesting one is the number '1'. This number is very much meaningless when taking into consideration the queries of the users (we do not know what it is referring to: âĆň, cm, etc.). Therefore, we thought maybe it would be necessary to take numbers out in the `build_terms` function. Latter, we will discuss why we do not do it in the end.

Then, we constructed the word cloud in Figure 7.
From the word cloud we can see predominant themes like: "cloth", "wear", "topwear", "neck", "round", "western", "accessori", etc. Which indicates that the corpus focuses heavily on apparel types, styles, and garment features. Also, terms like "casual", "polo", "regular", "winter wear", "full-sleev", "half-sleeve", reflect how the style, types and seasonal descriptions of clothes are very common. In addition, the materials are also common, for instance, "cotton".

2.4 Out of Stock Distribution

Switching to the numerical variables, that we saw could be interesting for filtering, we can first take a look at the out of stock distribution (Figure 8).
In the distribution, we can see how filtering out the products that are out of stock (is interesting for a user as they cannot buy these products), we would eliminate 5.85% of the documents. When taking into account the processing, this number is not as big, but could increase the computation time in some extreme cases. Most of the products are in stock (94.15%).

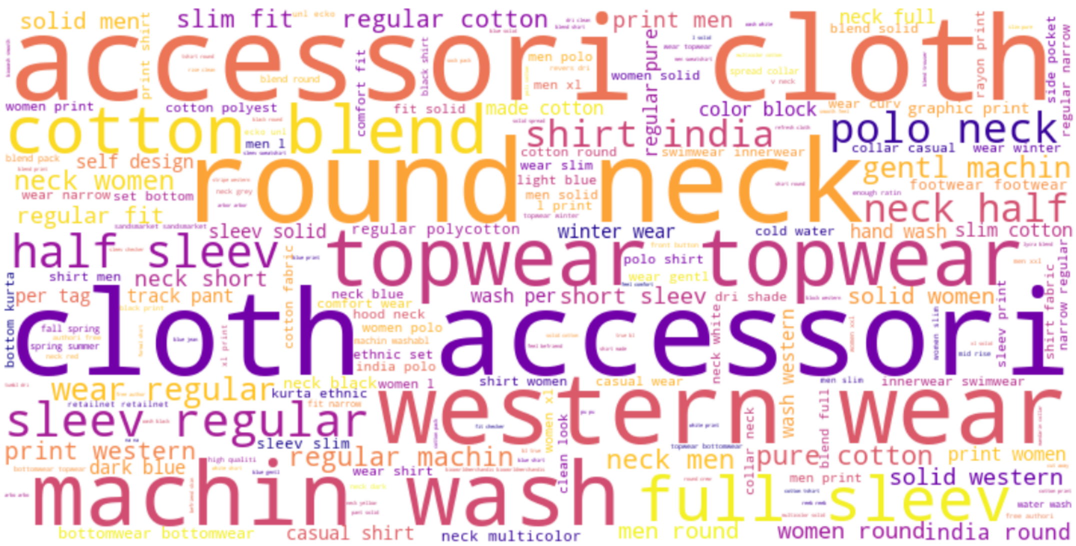

Figure 7: Word cloud from the processed categorical fields in the corpus.

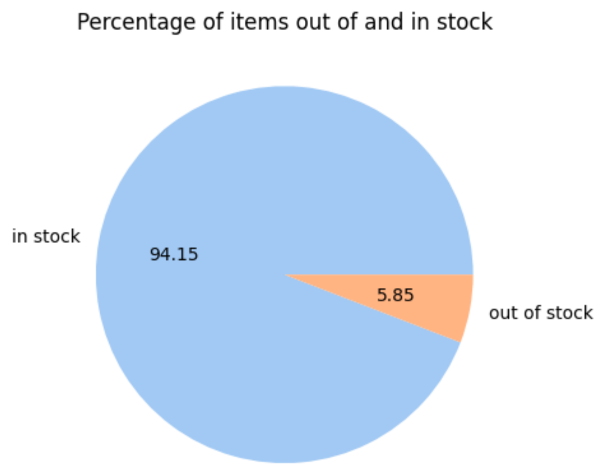

Figure 8: Out of stock distribution.

2.5 Rankings

After processing the numerical data, we can start treating the numbers and creating rankings based on `average_rating`, prices and `discount`.

In Figure 9 we can see the rankings based on `actual_price`. From Subfigure 9a we see how the products with lowest actual price, it is 0, but actually have a selling price. This can indicate errors in the data. In Subfigure 9b we see that the products with largest prices are either big clothing pieces or they include more than one product. This also shows us that maybe having the numbers in the index can be a good indicator of products that might interest the user (contradicting what we found in the word cloud).

In Figure 10 we can see the rankings based on `selling_price`. From these we can see that the ones with lowest prices seem to be accessories (which makes sense), and that the most expensive are of full tracksuits, possibly dresses, coats, and big clothing pieces.

Finally, in Figure 11 we can see the products with the highest discounts (most seem to be T-shirts) and the ones with no discount. And in Figure 12 we can see the top `brands` and `sellers`. It is interesting to notice that the top brand and seller do not actually have a value. However, we can still see that the top brand is **ECKO Uni** and the top seller **RetailNet**.

	pid	title	actual_price	discount	selling_price	url
1705	TSHFZ3JEEZC6KTVB	Solid Women Polo Neck Blue T-Shirt	0.0	0.0	1099.0	https://www.flipkart.com/reebok-solid-men-polo...
1734	SOCFH2UDUMG6GMSR	Men Striped Ankle Length	0.0	0.0	499.0	https://www.flipkart.com/reebok-men-striped-an...
1891	TKPFZ3JRBVZD3AKM	Solid Women Grey Track Pants	0.0	0.0	1499.0	https://www.flipkart.com/reebok-solid-men-grey...
1922	SWSFJY5ZBJAVWWJX	Full Sleeve Solid Men Sweatshirt	0.0	0.0	2399.0	https://www.flipkart.com/reebok-full-sleeve-so...
1949	TKPFZ3JRYR599GGY	Solid Men Grey Track Pants	0.0	0.0	1499.0	https://www.flipkart.com/reebok-solid-men-grey...
1950	TSHFK3W7AZQYSWGF	Solid Men Polo Neck Green T-Shirt	0.0	0.0	1299.0	https://www.flipkart.com/reebok-solid-men-polo...
1953	JCKFJY5A7Q7XCMHK	Full Sleeve Solid Men Sports Jacket	0.0	0.0	3699.0	https://www.flipkart.com/reebok-full-sleeve-so...
1958	TSHFZ3JDCZQHVU8G	Solid Men Polo Neck Green T-Shirt	0.0	0.0	1599.0	https://www.flipkart.com/reebok-solid-men-polo...
1964	SOCFYY5RGZHT3AZF	Original Cotton Half Cushion Women Ankle Lengt...	0.0	0.0	399.0	https://www.flipkart.com/reebok-original-cotto...
2020	TSHFK3W7JGJUXFUT	Printed Women Round Neck Dark Blue T-Shirt	0.0	0.0	3999.0	https://www.flipkart.com/reebok-printed-men-ro...

(a) Actual Price Sorting (ASCENDING).

	pid	title	actual_price	discount	selling_price	url
10272	SUIFNNPF3W8GEHAB	3 Piece Solid Women Suit	12999.0	0.60	5199.0	https://www.flipkart.com/true-blue-3-piece-sol...
10287	SUIFPDS2DEZNSKTH	2 Piece Self Design Women Suit	12999.0	0.60	5199.0	https://www.flipkart.com/true-blue-2-piece-sel...
10315	SUIFNMK2FQDWYTUZ	2 Piece Solid Men Suit	12999.0	0.60	5199.0	https://www.flipkart.com/true-blue-2-piece-sol...
25423	JCKFQF5K72AT2JDC	Full Sleeve Solid Women Casual Jacket	12999.0	0.50	6499.0	https://www.flipkart.com/puma-full-sleeve-soli...
25815	JCKFQF5KMJJ349H8	Full Sleeve Solid Women Casual Jacket	12999.0	0.40	7799.0	https://www.flipkart.com/puma-full-sleeve-soli...
26089	SWSFUMFGQFKVZGYH	Full Sleeve Printed Men Sweatshirt	12999.0	0.40	7799.0	https://www.flipkart.com/puma-full-sleeve-prin...
6895	JEAF8S4GWU5YKQTF	Skinny Men Blue Jeans	12990.0	0.40	7794.0	https://www.flipkart.com/gas-skinny-men-blue-j...
25569	JCKFW8EFUXMSBHMZ	Full Sleeve Solid Men Padded Jacket	10999.0	0.45	6049.0	https://www.flipkart.com/puma-full-sleeve-soli...
6870	JCKF8SWBNSGY5TPX	Full Sleeve Self Design Women Casual Jacket	10990.0	0.52	5188.0	https://www.flipkart.com/gas-full-sleeve-self-...
6901	JEAF65G3MZYG3BQM	Maxx Regular Men Black Jeans	10990.0	0.39	6692.0	https://www.flipkart.com/gas-maxx-regular-men-...

(b) Actual Price Sorting (DESCENDING).

Figure 9: Actual Price Sorted Documents.

	pid	title	actual_price	discount	selling_price	url
19482	SOCFYNKS7XZ3YUKZ	Men Printed Calf Length (Pack of 5)	0.0	0.00	0.0	https://www.flipkart.com/foot-fix-men-printed-...
27574	TKPFZAGJYK9YGRAA	Striped Men Black Track Pants	0.0	0.00	0.0	https://www.flipkart.com/ravilka-striped-men-b...
16485	SOCET7QRNHYG9HHB	Women Mid-Calf/Crew (Pack of 2)	199.0	0.50	99.0	https://www.flipkart.com/welwear-men-mid-calf-...
20435	BDAFUBD2EJHFCRNC	Men Printed Bandana	199.0	0.50	99.0	https://www.flipkart.com/t10-sports-men-printe...
7654	SOCFFGA2FYZQBFXT	Women Color Block Ankle Length (Pack of 3)	499.0	0.76	118.0	https://www.flipkart.com/your-shopping-store-m...
24437	SOCFZAGJC3VUFQU9	Women Solid Ankle Length (Pack of 3)	399.0	0.69	120.0	https://www.flipkart.com/ina-group-men-solid-a...
24438	SOCFZ7GFAZGYZGR7	Men Solid Ankle Length (Pack of 3)	399.0	0.69	120.0	https://www.flipkart.com/ina-group-men-solid-a...
24439	SOCFZ7JX39ZEW8GE	Women Solid Ankle Length (Pack of 3)	399.0	0.69	120.0	https://www.flipkart.com/ina-group-men-solid-a...
20253	CAPEX5YHPH3MSGFC	Cotton 5 panel baseball Cap	249.0	0.50	124.0	https://www.flipkart.com/t10-sports-cotton-5-p...
16402	SUSECSFFVNKG5VGG	Brand Trunk Y- Back Suspenders for Men (Black)	499.0	0.74	125.0	https://www.flipkart.com/brand-trunk-y-back-su...
906	TSHF5FRXKGF6A4FH	Printed Women Round Neck White T-Shirt	998.0	0.87	128.0	https://www.flipkart.com/jack-royal-printed-me...
25325	SOCFPR9UF8Q4FCHG	Men Ankle Length (Pack of 3)	499.0	0.73	132.0	https://www.flipkart.com/puma-men-ankle-length...

(a) Selling Price Sorting (ASCENDING).

	pid	title	actual_price	discount	selling_price	url
2067	TKTFZ3YGGMMNBYEZ	Solid Women Track Suit	9999.0	0.20	7999.0	https://www.flipkart.com/reebok-solid-men-trac...
11010	BZRFVAX2QGTEGHRH	Checkered Single Breasted Party Women Full Sle...	7999.0	0.00	7998.0	https://www.flipkart.com/true-blue-checkered-s...
25815	JCKFQF5KMJJ349H8	Full Sleeve Solid Women Casual Jacket	12999.0	0.40	7799.0	https://www.flipkart.com/puma-full-sleeve-soli...
26089	SWSFUMFGQFKVZGYH	Full Sleeve Printed Men Sweatshirt	12999.0	0.40	7799.0	https://www.flipkart.com/puma-full-sleeve-prin...
6895	JEAF8S4GWU5YKQTF	Skinny Men Blue Jeans	12990.0	0.40	7794.0	https://www.flipkart.com/gas-skinny-men-blue-j...
11008	BZRFVDGUJHTQHDAX	Self Design Single Breasted Party Women Full S...	6999.0	0.00	6998.0	https://www.flipkart.com/true-blue-self-design...
6898	JEAF8S4GE8PKH7H3	Regular Fit Men Dark Blue Cotton Blend Trousers	10990.0	0.36	6925.0	https://www.flipkart.com/gas-regular-fit-men-d...
6901	JEAF65G3MZYG3BQM	Maxx Regular Men Black Jeans	10990.0	0.39	6692.0	https://www.flipkart.com/gas-maxx-regular-men-...
25423	JCKFQF5K72AT2JDC	Full Sleeve Solid Women Casual Jacket	12999.0	0.50	6499.0	https://www.flipkart.com/puma-full-sleeve-soli...
6855	JEAFEN3WGHH3ZEFY	Skinny Men Blue Jeans	9990.0	0.35	6493.0	https://www.flipkart.com/gas-skinny-men-blue-j...
14194	JEAFQFGWXSVSMRWC	Slim Women Dark Blue Jeans	7999.0	0.20	6399.0	https://www.flipkart.com/levis-slim-men-dark-b...

(b) Selling Price Sorting (DESCENDING).

Figure 10: Selling Price Sorted Documents.

	pid	title	actual_price	discount	selling_price	url
8158	VESFVYNGDZBZTDCF	VIP Women Vest (Pack of 8)	0.0	0.0	784.0	https://www.flipkart.com/vip-men-vest/p/itm8bd...
8826	BRFFYQJAFQFUGHVD	Women Brief	0.0	0.0	149.0	https://www.flipkart.com/juari-gentleman-men-b...
8827	BRFFYQJA8TNJSFEZ	Women Brief	0.0	0.0	149.0	https://www.flipkart.com/juari-gentleman-men-b...
8833	CAPF9FVSYWE8W2FU	Self Design Baseball Cap Cap	0.0	0.0	190.0	https://www.flipkart.com/roy-self-design-baseb...
8834	CAPEG3ANT9UHJGZW	Self Design REGULAR Cap	0.0	0.0	191.0	https://www.flipkart.com/roy-self-design-regul...
16403	SOCFMKXAK3FWDVGT	Women Mid-Calf/Crew (Pack of 5)	0.0	0.0	653.0	https://www.flipkart.com/jspcsons-men-mid-calf...
16404	VESFZHNUMJQZHYZM	jspcsons Men Vest (Pack of 6)	0.0	0.0	1416.0	https://www.flipkart.com/jspcsons-men-vest/p/i...
2985	TSHF73YH4XEGEBZG	Printed Men Round Neck Multicolor T-Shirt (Pa...	0.0	0.0	695.0	https://www.flipkart.com/axmann-printed-men-ro...
8522	TRKFC33FMARCJQRM	Women Trunks	735.0	0.0	730.0	https://www.flipkart.com/vip-men-trunks/p/itmf...
2986	TSHF742RH3JHH8VG	Printed Men Round Neck Multicolor T-Shirt (Pa...	0.0	0.0	695.0	https://www.flipkart.com/axmann-printed-men-ro...

(a) Discount Sorting (ASCENDING).

	pid	title	actual_price	discount	selling_price	url
906	TSHF5FRXKGF6A4FH	Printed Women Round Neck White T-Shirt	998.0	0.87	128.0	https://www.flipkart.com/jack-royal-printed-me...
903	TSHFMFT7VASAHBH3	Printed Women Round Neck White T-Shirt	999.0	0.86	136.0	https://www.flipkart.com/jack-royal-printed-me...
902	TSHFMFXGFJ7G2ABK	Printed Women Round Neck Grey T-Shirt	999.0	0.86	136.0	https://www.flipkart.com/jack-royal-printed-me...
18249	TSHFGH6T3CVGDXS9	Printed Men Round Neck Multicolor T-Shirt (Pa...	2999.0	0.85	449.0	https://www.flipkart.com/yellowvibes-printed-m...
91	CTPFVZTBN4GRZKXH	nu-Lite Satin Tie & Cufflink (Red)	3299.0	0.84	499.0	https://www.flipkart.com/nu-lite-satin-tie-cuf...
18017	TSHFKHRYJYMEMZHK	Printed Men Mandarin Collar Blue T-Shirt	1800.0	0.84	282.0	https://www.flipkart.com/yellowvibes-printed-m...
18093	TSHFGFNBYVKZBQ2M	Printed Men Collared Neck Multicolor T-Shirt	1500.0	0.84	230.0	https://www.flipkart.com/yellowvibes-printed-m...
9811	CAPE9YWMSVSZPM2K	Solid Balclava Cap	1499.0	0.84	228.0	https://www.flipkart.com/graceway-solid-balcla...
18016	TSHFHQNCHJJUQYVQ	Printed Women Round Neck Blue T-Shirt	1800.0	0.84	280.0	https://www.flipkart.com/yellowvibes-printed-m...
3102	TSHFVM4PQPRY2CRZ	Color Block Women Round Neck Green T-Shirt	1499.0	0.84	228.0	https://www.flipkart.com/refro-color-block-men...

(b) Discount Sorting (DESCENDING).

Figure 11: Actual Price Sorted Documents.

	count
brand	
	2009
ECKO Unl	951
Free Authori	860
ARBO	806
REEB	802
Pu	798
True Bl	793
Keo	660
Amp	585
Black Beat	548
vims rai	503
yellowvib	492
PixF	429
Oka	414
Gracew	405
TEE BUD	393
Shoef	358
Marca Disa	353
V	343
CupidSto	338

(a) Top brands.

	count
seller	
	1643
RetailNet	1411
SandSMarketing	887
BioworldMerchandising	842
ARBOR	783
Keoti	660
AFFGARMENTS	587
Black Beatle	548
AMALGUS ENTERPRISE	477
Tayab Manch Fashions	436
KAPSONSRETAILPVTLTD	415
GRACEWAY	408
T-SHIRT EXPRESS	393
OKANE	386
WHITE SKY	371
SHOEFLY	358
ArvindTrueBlue	338
ModaElementi	333
Marca Disati	330
CupidStoreIN	329

(b) Top sellers.

Figure 12: Distributions of word counts in titles and descriptions.

From the EDA we have gained some information on the corpus, its content and the information we are gonna work with. This will help us understand how we can make an Information Retrieval system that provides the user with relevant products to their queries.