Part 1: Text Processing and Exploratory Data Analysis

Fashion E-Commerce Search Engine

IRWA Project

October 2025

1 Introduction

This report documents the preprocessing and exploratory analysis of a fashion e-commerce dataset containing 28,080 product documents. The goal is to prepare the data for building an effective search engine by applying text preprocessing techniques and understanding the corpus characteristics through statistical analysis. Each product document contains textual fields (title, description, product details), categorical metadata (brand, category, seller), and numerical attributes (price, rating, stock status).

2 Text Preprocessing

2.1 Core Processing Pipeline

Our preprocessing pipeline transforms raw text into normalized tokens suitable for indexing. The process applies five main operations: lowercase conversion, punctuation removal, tokenization, stop word removal, and stemming. We process three textual fields per document: title, description, and product details.

The implementation combines these fields and applies transformations sequentially:

Listing 1: Text preprocessing function

```
def build_terms(document):
    stemmer = PorterStemmer()
    stop_words = set(stopwords.words('english'))

    # Combine text fields
    text = (document['title'] + '␣' +
            document['description'] + '␣' +
            extract_product_details(document['product_details']))

    # Lowercase and remove punctuation
    text = text.lower()
    text = ''.join(char if char.isalnum() or char.isspace()
                   else '␣' for char in text)

    # Tokenize and filter
    text = text.split("␣")
    text = [term for term in text if term not in stop_words]
    text = [term for term in text if term != '']

    # Stem
    text = [stemmer.stem(term) for term in text]
    return text
```

The lowercase conversion ensures case-insensitive matching, so "Cotton" and "cotton" are treated identically. Punctuation removal eliminates non-alphanumeric characters while preserving numbers, which often carry semantic meaning in product descriptions (e.g., "100% cotton"). Stop word removal filters common words like "the", "and", "is" using NLTK's English stop word list, reducing index size and improving precision. The Porter Stemmer reduces words to root forms, allowing "running", "runner", and "runs" to match the same stem "run".

For product details, we extract all attribute values since they contain important searchable information like colors, materials, and patterns:

Listing 2: Product details extraction

```
def extract_product_details(details):
    values = []
    for category in details:
        values.extend(v for v in category.values())
    return "␣".join(values)
```

2.2 Hybrid Approach for Non-Text Fields

A critical design decision involves handling categorical and numerical fields. We implement a hybrid approach that separates searchable text from structured metadata. The preprocessing function returns three components:

Listing 3: Hybrid document structure

```
def preprocess_document(document):
    tokens = build_terms(document)

    metadata = {
        'category': document.get('category', '').lower().strip(),
        'sub_category': document.get('sub_category', '').lower().strip
            (),
        'brand': document.get('brand', '').lower().strip(),
        'seller': document.get('seller', '').lower().strip()
    }

    original = {
        "pid": document["pid"],  # Critical for evaluation
        "title": document["title"],
        "description": document["description"],
        "brand": document["brand"],
        "category": document["category"],
        # ... all other fields preserved
    }

    return {
        "searchable_text": tokens,
        "metadata": metadata,
        "original": original
    }
```

This structure provides three key benefits. First, searchable text contains fully preprocessed tokens with stemming applied, maximizing recall through morphological conflation. Second, metadata fields (brand, category, sub-category, seller) are normalized to lowercase but not stemmed, preserving exact names for filtering operations. Stemming "Nike" or "Adidas" would create nonsensical forms that don't match user expectations for brand names. Third, the original component maintains all fields in their raw form, including the PID required for evaluation and numerical fields needed for filtering and ranking.

2.3 Treatment of Numerical Fields

Numerical fields (selling_price, actual_price, discount, average_rating, out_of_stock) are deliberately not indexed as textual terms. This decision reflects their fundamentally different role in retrieval. Indexing price "921" as text provides no search value, as users don't search for exact prices but rather filter by price ranges. Similarly, searching for rating "3.9" as text is meaningless when users actually want products above a rating threshold. These fields remain in the original structure for numerical comparison operations during filtering and ranking.

2.4 Validation Query Analysis

Our preprocessing approach is validated against the two provided queries. Query 1, "women full sleeve sweatshirt cotton", combines categorical terms (women), descriptive attributes (full sleeve), product type (sweatshirt), and material (cotton). Our system indexes "cotton" from descriptions and product details (Fabric attribute), while "women" may appear in both text and category metadata. Query 2, "men slim jeans blue", similarly benefits from our product details extraction, as "blue" typically appears in the Color attribute rather than free-text descriptions.

3 Exploratory Data Analysis

3.1 Vocabulary Statistics

Processing all 28,080 documents produces 1,393,344 total tokens with a vocabulary of 20,906 unique terms. This yields an average of 49.62 tokens per document (median: 38), indicating moderate-length product descriptions. The vocabulary-to-token ratio of 1.5% shows expected repetition in a domain-focused corpus. Before preprocessing, titles average 6.46 words while descriptions average 29.77 words, reflecting the concise nature of product names versus detailed specifications.

The term frequency distribution exhibits the expected Zipfian pattern. The top 15 terms are dominated by fashion-specific vocabulary:

Rank	Term	Frequency (%)
1	shirt	48,293 (3.47%)
2	neck	42,330 (3.04%)
3	cotton	34,429 (2.47%)
4	wear	29,152 (2.09%)
5	round	28,535 (2.05%)
6	women	28,060 (2.01%)
7	regular	27,430 (1.97%)
8	men	27,237 (1.95%)
9	wash	26,719 (1.92%)
10	sleev	26,316 (1.89%)

Table 1: Top 10 most frequent terms after preprocessing

The dominance of "shirt" (3.47%) reflects the prevalence of shirt products in the corpus. Gender terms ("women", "men") and garment features ("neck", "sleev", "round") appear prominently, confirming that our preprocessing successfully retains domain-relevant vocabulary while removing non-content stop words.

3.2 Categorical Distribution

The metadata analysis reveals 321 distinct brands and only 4 top-level categories. The category distribution is highly concentrated: "clothing and accessories" accounts for approximately

27,000 products (96%), with "footwear", "bags wallets and belts", and "toys" comprising the remainder. This concentration indicates a primarily apparel-focused dataset, aligning well with the validation queries targeting clothing items.

Brand distribution shows greater diversity but remains concentrated among a few major brands. The top 10 brands (ecko unltd, free authority, arbo, reeb, pu, true blue, keo, amp, black beat, vims rai) account for a substantial portion of the catalog, suggesting a mix of major labels and regional brands.

3.3 Price and Rating Analysis

Price distribution exhibits heavy right-skewness typical of e-commerce fashion. Most products cluster in the affordable range, with an average selling price around $Rs.$1,100 and median around $Rs.$600. Prices range from under $Rs.$200 to over $Rs.$7,000, representing basic items through premium products. The presence of discounts is pervasive, with many products showing 40-70% discounts, reflecting aggressive online retail pricing strategies.

Rating distribution is left-skewed with concentration in the 3.5-5.0 range. The average rating of 3.95 indicates generally positive customer sentiment. The modal rating falls between 4.0-4.5, with relatively few products below 3.0. This pattern suggests either rating inflation (satisfied customers are more likely to rate) or selection bias (poorly-rated products may be delisted).

Stock availability is high at 94.1% in-stock versus 5.9% out-of-stock. This availability rate simplifies search system requirements, as most retrieved products will be purchasable, though real-time stock checking would be necessary in production.

4 Key Insights and Design Validation

Several findings validate our preprocessing decisions. The vocabulary size of 20,906 terms is manageable for efficient indexing while capturing product description diversity. The concentration of frequency in domain-specific terms like "shirt", "cotton", and "women" confirms that our stop word removal targets generic function words while preserving content-bearing fashion terminology.

The categorical structure's focus on clothing validates the relevance of our validation queries (sweatshirts and jeans). The brand diversity supports brand-specific queries, though concentration in major brands means result counts will vary significantly across brand queries. The pricing and rating distributions indicate a corpus biased toward affordable, well-rated products, typical of online fashion marketplaces emphasizing volume and value.

The high stock availability simplifies filtering requirements for this dataset snapshot, though production systems would need real-time inventory tracking. The pervasive discounting suggests that ranking algorithms should consider both actual and selling prices to accurately reflect value propositions.

5 Conclusion

This preprocessing and analysis phase establishes a foundation for fashion product search. Our preprocessing pipeline successfully applies tokenization, stop word removal, punctuation elimination, and stemming while incorporating domain-appropriate enhancements like number preservation and product detail extraction. The hybrid approach to non-textual fields enables both flexible full-text search through stemmed tokens and precise categorical filtering through preserved metadata, while keeping numerical fields separate for appropriate filtering and ranking operations.

The exploratory analysis quantifies our corpus characteristics: 20,906 unique terms from 1,393,344 tokens across 28,080 documents, concentrated in clothing items from 321 brands across

4 categories, with prices averaging $Rs.$1,100 and ratings averaging 3.95/5.0. These insights validate our preprocessing strategy and inform subsequent indexing and retrieval algorithm design. With preprocessing complete and corpus characteristics understood, we are prepared to proceed with inverted index construction and query processing in the next project phases.