## Problem Overview

The task is to work with historical purchase order data that contains multilingual item descriptions (primarily in English and Arabic). The nature of the data reflects real-world procurement challenges, such as inconsistent formatting, varying levels of detail, and the coexistence of multiple languages within the same dataset.

**Objective:**

1. Clean and preprocess item descriptions.
2. Group items into logical categories
3. Analyze spend distribution and extract insights from categorized data.

## Data Processing

### 2.1 Loading

I loaded the Excel file using **pandas**. Pandas handled the English and Arabic text perfectly. Data shape is (3150, 11) which are 11 features and 3150 records.

### Cleaning

During data inspection, **missing values were found across several features** **(e.g., Item Name, Product ID, Tax ID, Project ID**). All missing records are dropped, and Project ID feature is dropped as well. I found that **Product ID** and **Tax ID** are **Float64,** so I converted them to **Integer**. After cleaning it has 10 features and 2845 records.

## Item Categorization

### 3.1 My Methodology

The goal of item categorization is to group messy, multilingual purchase order item names into logical, meaningful categories.  
My methodology is hybrid approach follows four steps:

1. **Multilingual embeddings** to represent item names in a unified semantic space.
2. **Clustering** to automatically group similar items without requiring predefined categories.
3. **Cluster labeling with LLM** to generate human readable category names.
4. **Rule-based refinement** to refine generated categories to general categories.

This approach balances automation with interpretability, and it is particularly effective for messy, multilingual procurement data.

### 3.2 Multilingual Emeddings

I used **multilingual sentence embeddings** (LaBSE model) to convert item names into fixed-length vectors. They capture semantic similarity, meaning like “16mm steel bar” and “حديد تسليح 16 ملم” are mapped close together in vector space.

**Advantages:**

* Handles both English and Arabic.
* Robust against spelling variations and word order differences.
* No need for manual keyword dictionaries.

This ensures that semantically similar items cluster together regardless of language.

### 3.3 Clustering Approach

During experimentation, I compared two clustering approaches: **K-Means** and **HDBSCAN**.

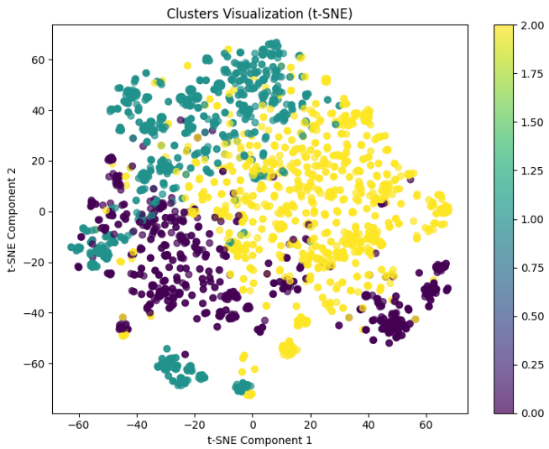
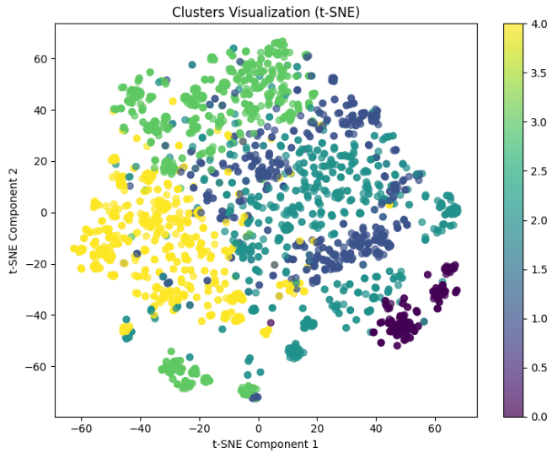
**K-Means**

**Strength:** Simple and efficient.

**Limitation:** Requires pre-specifying the number of clusters (*k*), which is not feasible in this case since the optimal number of categories is unknown.

**Experiment:** I first applied UMAP for dimensionality reduction, but the results were unsatisfactory. Therefore, I proceeded by applying K-Means clustering directly on the multilingual embeddings and experimented with different values of *k* (3, 4, 5, 7, and 9). As shown in **Figure 1** Using the elbow method, I estimated that the optimal number of clusters is either 3 or 5 as shown in **Figures 2-3**. However, I was unable to confirm this result since the true number of categories is unknown.

A graph with a line

AI-generated content may be incorrect.**Result:** Produced less meaningful groupings because forcing a fixed *k* led to arbitrary boundaries that did not reflect the natural structure of the data.

**Figure 1** elbow method **Figure 2** k=3 **Figure 3** k=5

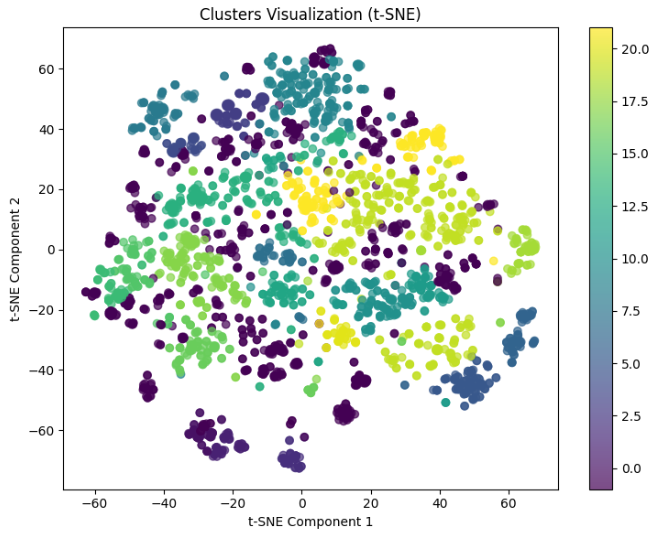
**HDBSCAN (Selected Approach)**

**Strength:** A density-based algorithm that does not require the number of clusters to be set in advance. Automatically discovers clusters of varying shapes and densities. Also, Handles multilingual item names effectively.

**Limitation:** Some items are classified as noise (label -1). While this avoids misclassification, these outliers require manual review or a secondary handling strategy.

**Experiment:** Firstly, I reduced the high-dimensional multilingual embeddings from 768 to 15 dimensions using UMAP, making the data more suitable for density-based clustering. Next, I applied HDBSCAN with different parameter settings. With min\_cluster\_size=40 and min\_samples=10, the algorithm produced **22 clusters** and **763 outliers** as shown in **Figure 4**. Reducing min\_cluster\_size to 30 and min\_samples=10 resulted in **27 clusters** and **601 outliers** as shown in **Figure 5**. Based on this comparison, I selected the (30, 10) as it provided a better balance between the number of meaningful clusters and the proportion of items classified as noise.

A diagram of a cluster of dots

AI-generated content may be incorrect.**Result:** Generated clusters that were more coherent and interpretable compared to K-Means, allowing natural categories to emerge without forcing items into arbitrary groups.

**Figure 4** for (40, 10) **Figure 5** for (30, 10)

### 3.4 Cluster Labeling using LLM

Once clusters were formed, the next step was to assign **meaningful category labels**. For each cluster, I randomly selected 10 items as a representative sample. These samples were then passed to a large language model (DeepSeek) with the following prompt:  
*“Your task is to identify the single, most general category that a given list of items belongs to. Respond only with the category name in English, without any additional text or commentary.”.* As a result of this labeling step, the initial 27 clusters and 1 uncategorize were combined into 21 meaningful categories.

**Advantages:**

* Automates category naming without manual effort.
* Works across languages and mixed-language clusters.
* Produces human readable, interpretable results for business users.

### 3.5 Rule-based Refinement

After generating initial category labels using the LLM, I applied rule-based refinement to improve generalization and consistency across clusters. This step groups closely related clusters under broader, more meaningful categories and corrects minor inconsistencies. For example, clusters such as **Steel reinforcement bars, Steel rebar**, and Steel bars were all consolidated under the broader category **Steel Products.** As a result, we have **7 categories** including **'Metal Products', 'Steel Products', 'Electrical Products', 'Construction Materials', 'Plumbing Products', 'Automotive Products', and 'Uncategorized'**.

### 3.6 Comparison with Alternative Approaches

While developing my item categorization methodology, I considered several alternative approaches and weighed their pros and cons against the chosen hybrid approach:

1. **Clustering Text Embeddings Alone**

* **Description:** Use embeddings and clustering algorithms (like KMeans or HDBSCAN) without any human-in-the-loop labeling.
* **Pros:** Fully automated; scalable to large datasets; requires minimal manual intervention.
* **Cons:** Cluster labels are not interpretable; clusters may be overly specific or inconsistent; multilingual data may cause semantic misalignment.
* **Reason for not choosing:** Pure clustering does not provide meaningful category names, making the output hard to use in practice.

1. **LLM-Based Categorization Alone**

* **Description:** Use a Large Language Model to directly assign categories to items.
* **Pros:** Can leverage LLM understanding for messy or multilingual text, interpretable labels.
* **Cons:** Computationally expensive for large datasets, may produce inconsistent categories without structured guidance, less scalable without.
* **Reason for not choosing:** While effective labeling, using LLMs alone on large datasets is expensive and inefficient and may result in overlapping or redundant categories.

1. **Rule-Based Categorization Alone**

* **Description:** Define manual rules (keywords, regex, or dictionaries) to classify items.
* **Pros:** Transparent and interpretable, fast for known categories, predictable results.
* **Cons:** Difficult to maintain, struggles with messy, multilingual, or new/unseen items, high initial effort, not appropriate for unknown categories, does not scale well to large or dynamic datasets.
* **Reason for not choosing:** Pure rule-based systems are brittle, require extensive domain knowledge, and cannot easily handle new or unseen categories, making them unsuitable for large-scale, dynamic, multilingual procurement data.

1. **Why the Hybrid Approach Was Chosen**

The chosen hybrid methodology combines the strengths of clustering, LLMs, and rule-based refinement.

* **Clustering multilingual embeddings** provides automated grouping, scalability to large datasets, reduces manual effort, and effectively handles multiple languages by representing items in a unified semantic space.
* **LLM labeling** gives interpretable, human readable category names while handling multilingual and messy text, though it adds some computational cost.
* **Rule-based refinement** ensures consistency, generalization, and practical usability, with minimal additional cost once rules are defined.

This combination strikes a balance between automation, interpretability, robustness, and cost-efficiency. It scales well to large, dynamic datasets and is particularly effective for messy, multilingual procurement data, where neither pure automation nor manual rules alone would be sufficient or cost-effective.

**Limitation:** Despite its strengths, the methodology leaves 601 outlier items uncategorized. These outliers still require human review or an alternative solution to be fully classified.

## 4. Analysis & Insights

### 4.1. Spend Concentration

A graph of a bar graph

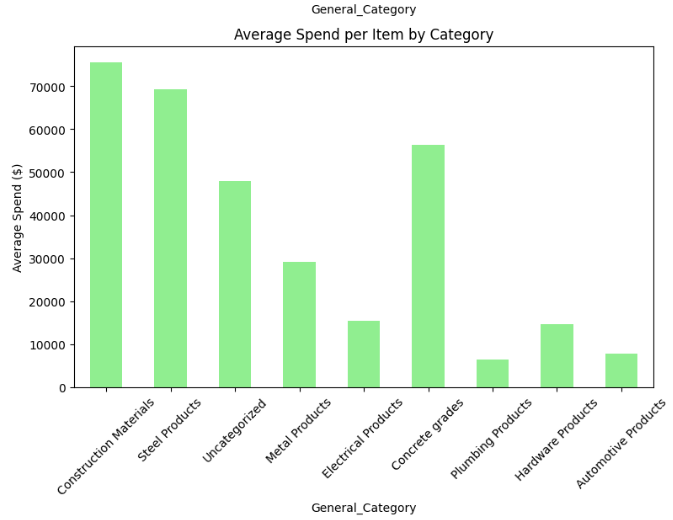
AI-generated content may be incorrect.A pie chart with numbers and text

AI-generated content may be incorrect.The spend distribution across categories shows a significant concentration in a few key areas. Steel Products alone accounts for **48.4%** of total spend, followed by Construction Materials at **23.5%**, and Uncategorized items at **21.1%**. Together, these three categories make up **over 92%** of the total procurement spend, indicating that most of the budget is focused on a small set of categories.

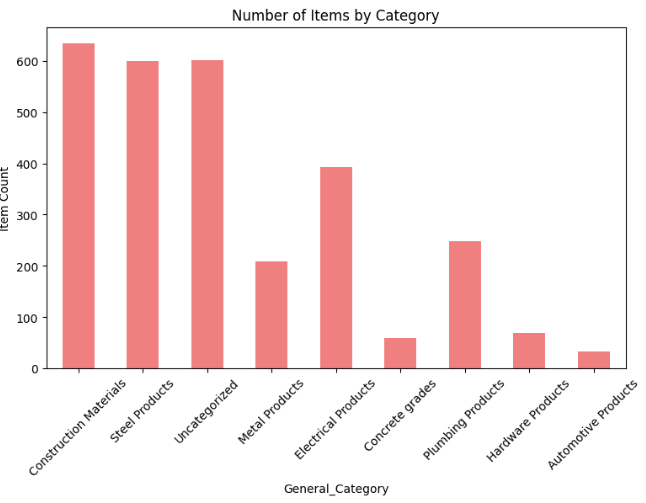
**Figure 6** spend percentage **Figure 7** total spending

### 4.2. Average Spend Per Item

The average spend per item varies widely across categories. Steel Products lead with an average of **$71,154.51 per item**, followed by Construction Materials at **$54,071.97**. In contrast, lower-cost categories such as Plumbing Products and Automotive Products have average spends of **$4,135.86** and **$7,762.70 per item**, respectively. Electrical Products sit in the middle with an average spend of **$15,358.50 per item**.

**Figure 8** average spending per item

### 4.3. Item Count Distribution

The number of items per category also reflects concentration patterns. Steel Products have the highest item count (**928 items**), followed by Construction Materials (**594 items**) and Uncategorized items (**601 items**). Smaller categories such as Metal Products (**117 items**), Plumbing Products (**179 items**), and Automotive Products (**33 items**) represent a small fraction of the total portfolio.

**Figure 9** item count

### 4.4. Uncategorized Spend

A significant portion of spend, **$28,789,212.17 (21.1%)**, falls under Uncategorized items. This highlights the importance of improving categorization and classification processes, as over one-fifth of procurement spend lacks a defined category, potentially affecting analysis and strategic decision-making.

### 4.5. Overall Portfolio view

The total procurement portfolio amounts to **$136,428,467.01** across **2,845 items**. The data reflects a highly skewed spend profile where a few categories dominate both in spend and item count. This insight can help focus cost optimization efforts on the highest-impact categories while also addressing gaps in categorization to improve portfolio visibility.

## 5. Future Improvements

For a production system, several enhancements can be made to increase scalability, accuracy, and usability:

* **Active Learning for Outliers** builds a feedback loop where procurement experts review uncategorized items, and the system learns from these corrections.
* **Vector Database Integration** store embeddings in a vector database (like FAISS or Pinecone) for real-time categorization and similarity search.
* **LLM Fine-Tuning** fine-tune or distill models on procurement-specific data to make labeling more consistent and cheaper at scale.
* **Interactive Dashboard** provides a user interface for business users to explore clusters, refine categories, and audit decisions.