## 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). Using the elbow method, I estimated that the optimal number of clusters is either 3 or 5. However, I was unable to confirm this result since the true number of categories is unknown.

**Result:** Produced less meaningful groupings because forcing a fixed *k* led to arbitrary boundaries that did not reflect the natural structure of the data.

**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**. Reducing min\_cluster\_size to 30 and min\_samples=10 resulted in **27 clusters** and **601 outliers**. 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.

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

### 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 28 clusters were combined into 22 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 profiles, Steel pipes, and Steel bars were all consolidated under the broader category **Steel Products.** As a result, we have **8 categories** including **'Metal Products', 'Steel Products', 'Hardware 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.