A3 – CUJ Runthrough + Demo

Group Name: TemporAI

- Mohamed Sbeinati – 1009163717
- Zherui Zhang - 1008865122

Submission Date: 2025/09/24

TL;DR

We researched how SMBs struggle with messy inventory data, then tested OpenRefine and ChatGPT on Smell & Smile's legacy export. OpenRefine handled simple clustering/normalization; complex joins and domain conversions failed. We switched to LLM-assisted transforms via prompt iterations to finish the job. Key insight: the best system mixes automation with user input, ultimately feeding a multi-source, LLM-powered data platform.

User Goal

As a small retailer with a messy legacy inventory system, I want to clean, normalize, and categorize my product data, flagging errors and low stock, so I can reorder with confidence and prepare for an AI-assisted data platform that unifies all my sources.

Persona Description

- **Role / Background:** Owner-operator of a small fragrance retailer (Smell & Smile); responsible for inventory, purchasing, and sales.
- **Experience:** Comfortable with Excel; first time using OpenRefine or LLMs for data work.
- **Context:** Time-constrained, domain knowledge heavy, limited technical background.

Tools Used

- **OpenRefine** (clustering, faceting, regex transforms)
- **Google Sheets** (quick checks, buy list filters)
- **ChatGPT** (LLM-assisted regex, column mapping, free-text → structured rules; prompt iteration)
- **Excel** (initial preview/export)

Summary of Findings

Research & Problem Framing

Before testing tools, the user researched typical SMB inventory issues: duplicate SKUs, inconsistent units (ml/ML/mL/tola), mixed item types (raw materials, finished products, packaging), negative stock, and missing costs. The success metric was: produce a clean, categorized dataset with low-stock flags and explainable steps.

Phase 1 — OpenRefine: quick wins, clear limits

What worked well:

- **Clustering:** Normalized “Oud 6ml,” “OUD 6 ML,” and “Oud – 6mL” into one label.
- **Simple unit fixes:** Standardized ml / ML / mL to mL; pcs/PCS → pcs.

- **Facets:** Surfaced anomalies like negative stock and blank costs quickly.

Where it failed:

- **Unit conversions:** Handling tola → mL was brittle across typos and spacing.
- **Compound parsing:** Extracting size + unit from messy free-text names often broke.
- **Rule collisions:** Categorization rules (“tester bottle” vs “bottle of tester oil”) became messy chains that were hard to maintain.

Result: OpenRefine handled straightforward normalization, but struggled with domain-specific conversions and edge cases, demanding heavy manual effort.

Phase 2 — Switching to ChatGPT (LLM-assisted transforms)

To go beyond OpenRefine, the user integrated ChatGPT to generate transformations:

- **Regex parsing:** Extracted `size_value` and `size_unit` from strings like “12 ml,” “.41 oz,” “1 tola.”
- **Unit conversions:** Proposed consistent logic: map tola → 11.66 mL, handle L/cL → mL, leave pcs/kg.
- **Categorization rules:** Built a keyword table with synonyms/negatives. An `_override` column was added for human review.

Prompt iteration: Early prompts failed on decimals/hyphens. Adding **dataset-specific examples** improved accuracy. Asking the LLM to generate explanations + test cases produced more reliable patterns.

Outcome: ChatGPT accelerated regex and logic authoring, but results still required checking. Lesson: automation + domain input outperforms either alone.

Phase 3 — Cleaned Data & Actionable Outputs

Combining both tools, the business owner produced a structured, decision-ready dataset:

- **Cleaned columns:** Standardized names, normalized units (mL, pcs, kg), extracted fields like `size_value`.

- **Custom flags:** Added `NEG_QTY`, `MISSING_COST`, and `REORDER` for low stock.
- **Categorization:** Grouped items into Raw, Packaging, Finished, clarifying what was low or inconsistent.
- **Buy list:** Exported to Google Sheets, filtering on `REORDER` produced a supplier-ready restocking list.

Key Insight

SMB data cleaning is socio-technical: tools automate repetitive work, but only business owners can validate messy, domain-specific edge cases. Sustainable workflows must combine automation with lightweight human review.

Recommendations

For Products

- **Schema-Aware Import Layer:** Auto-detects common SMB schemas and maps columns for review.
- **Domain Rule Packs:** Modular packs. Stored in JSON/YAML for reuse.
- **LLM Transforms w/ Confidence:** Suggestions tagged with confidence. High-confidence (>90%) apply automatically; others flagged for override.
- **Sample-Based Previews:** Let users apply transforms on sample rows and see before/after previews to refine rules.
- **Override Columns:** Auto-generate `_override` columns to preserve human judgment.
- **Snapshots & Sandboxing:** Version data cleaning steps like Git commits; easy rollback if errors occur.
- **Provenance & Audit:** Log every transformation (who, when, confidence). Allow replay on new datasets.
- **Dashboards:** Out-of-the-box connectors for inventory health, buy lists, and error monitoring.

For Future Users

- Start with a small slice of data to test.
- Use OpenRefine for simpler tasks; switch to LLMs for messy parsing.
- Always provide **examples** to LLMs, not vague prompts.
- Keep override columns and save snapshots, mistakes are inevitable.
- Expect context switching; keep notes on steps and timing.

Highlights & Lowlights Table

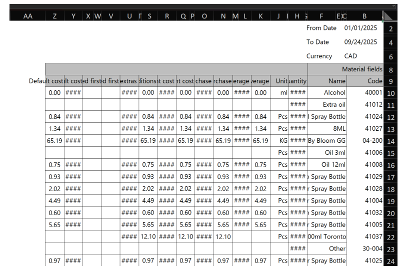
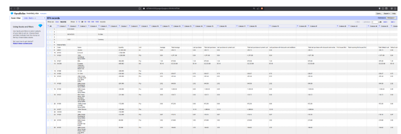
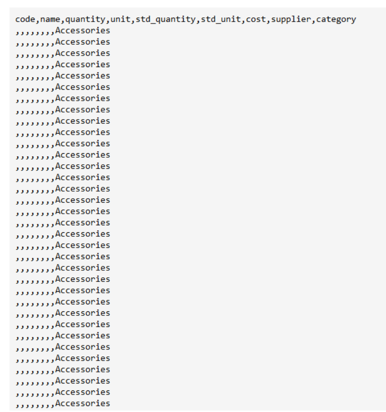
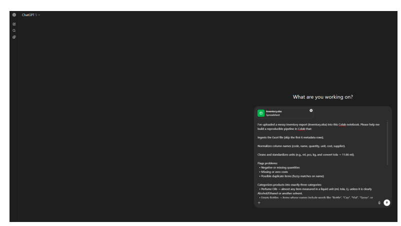
Task / Moment	Severity	Notes
OpenRefine clustering/unit fixes	Great	Quick normalization; surfaced anomalies fast.
Complex parsing in OpenRefine	Severe	Struggled with edge cases; required custom logic.
LLM regex & rule generation	Great	Faster + more accurate when fed examples.
Categorization & buy list export	Moderate	Needed overrides, but produced actionable output.

CUJ Overview Table

Task	Time	Switches
Open Excel export & inspect messy inventory	5 min	1
Load into OpenRefine, try clustering/facets	15 min	2
Attempt GREL transforms (unit normalization, regex) → docs/tutorials → back to tool	30 min	3
Switch to ChatGPT for regex/unit conversion help; iterate prompts with examples	40 min	3
Test ChatGPT-generated logic back in OpenRefine & Sheets	20 min	2
Move to Google Colab with ChatGPT-generated Pandas pipeline	50 min	2
Debug file import/export & generate clean CSVs	30 min	1
Export flagged data + buy list to Sheets/Data Studio for visualization	25 min	1

Total Time: ~3 hours 35 minutes
Total Context Switches: 15

Full CUJ Table (Step-by-Step Documentation)

<table>
<tr><th>Step</th><th>Notes (What + Why)</th><th>Screenshot</th></tr>
<tr><td>1</td><td>Opened messy Excel export in OpenRefine to inspect inventory. Skipped metadata rows and previewed columns. First goal: see duplicates and blanks.</td><td>
<table>
<tr><th>AA</th><th>Z</th><th>Y</th><th>X W</th><th>V</th><th>U T S</th><th>R</th><th>Q P</th><th>O</th><th>N</th><th>M L</th><th>K</th><th>J I</th><th>H</th><th>F</th><th>E C</th><th>B</th><th></th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>From Date</td><td>01/01/2025</td><td></td><td>2</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>To Date</td><td>09/24/2025</td><td></td><td>4</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>Currency</td><td>CAD</td><td></td><td>6</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>Material fields</td><td></td><td>8</td></tr>
<tr><td>Default cost</td><td>lt cost</td><td>ed first</td><td>d first</td><td>extras</td><td>ditions</td><td>nt cost</td><td>nt cost</td><td>chase</td><td>rchase</td><td>erage</td><td>verage</td><td>Unit</td><td>uantity</td><td>Name</td><td>Code</td><td></td><td>9</td></tr>
<tr><td>0.00</td><td>####</td><td></td><td></td><td>####</td><td>0.00</td><td>####</td><td>0.00</td><td>####</td><td>0.00</td><td>####</td><td>0.00</td><td>ml</td><td>####</td><td>Alcohol</td><td>40001</td><td></td><td>10</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>####</td><td>Extra oil</td><td>41012</td><td></td><td>11</td></tr>
<tr><td>0.84</td><td>####</td><td></td><td></td><td>####</td><td>0.84</td><td>####</td><td>0.84</td><td>####</td><td>0.84</td><td>####</td><td>0.84</td><td>Pcs</td><td>####</td><td>l Spray Bottle</td><td>41024</td><td></td><td>12</td></tr>
<tr><td>1.34</td><td>####</td><td></td><td></td><td>####</td><td>1.34</td><td>####</td><td>1.34</td><td>####</td><td>1.34</td><td>####</td><td>1.34</td><td>Pcs</td><td>####</td><td>8ML</td><td>41027</td><td></td><td>13</td></tr>
<tr><td>65.19</td><td>####</td><td></td><td></td><td>####</td><td>65.19</td><td>####</td><td>65.19</td><td>####</td><td>65.19</td><td>####</td><td>65.19</td><td>KG</td><td>####</td><td>By Bloom GG</td><td>04-200</td><td></td><td>14</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>Pcs</td><td>####</td><td>Oil 3ml</td><td>41006</td><td></td><td>15</td></tr>
<tr><td>0.75</td><td>####</td><td></td><td></td><td>####</td><td>0.75</td><td>####</td><td>0.75</td><td>####</td><td>0.75</td><td>####</td><td>0.75</td><td>Pcs</td><td>####</td><td>Oil 12ml</td><td>41008</td><td></td><td>16</td></tr>
<tr><td>0.93</td><td>####</td><td></td><td></td><td>####</td><td>0.93</td><td>####</td><td>0.93</td><td>####</td><td>0.93</td><td>####</td><td>0.93</td><td>Pcs</td><td>####</td><td>Spray Bottle</td><td>41029</td><td></td><td>17</td></tr>
<tr><td>2.02</td><td>####</td><td></td><td></td><td>####</td><td>2.02</td><td>####</td><td>2.02</td><td>####</td><td>2.02</td><td>####</td><td>2.02</td><td>Pcs</td><td>####</td><td>r Spray Bottle</td><td>41028</td><td></td><td>18</td></tr>
<tr><td>4.49</td><td>####</td><td></td><td></td><td>####</td><td>4.49</td><td>####</td><td>4.49</td><td>####</td><td>4.49</td><td>####</td><td>4.49</td><td>Pcs</td><td>####</td><td>r Spray Bottle</td><td>41004</td><td></td><td>19</td></tr>
<tr><td>0.60</td><td>####</td><td></td><td></td><td>####</td><td>0.60</td><td>####</td><td>0.60</td><td>####</td><td>0.60</td><td>####</td><td>0.60</td><td>Pcs</td><td>####</td><td>Spray Bottle</td><td>41032</td><td></td><td>20</td></tr>
<tr><td>5.65</td><td>####</td><td></td><td></td><td>####</td><td>5.65</td><td>####</td><td>5.65</td><td>####</td><td>5.65</td><td>####</td><td>5.65</td><td>Pcs</td><td>####</td><td>r Spray Bottle</td><td>41005</td><td></td><td>21</td></tr>
<tr><td></td><td></td><td></td><td></td><td>####</td><td>12.10</td><td>####</td><td>12.10</td><td>####</td><td>12.10</td><td></td><td></td><td>Pcs</td><td>####</td><td>00ml Toronto</td><td>41037</td><td></td><td>22</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>####</td><td>Other</td><td>30-004</td><td></td><td>23</td></tr>
<tr><td>0.97</td><td>####</td><td></td><td></td><td>####</td><td>0.97</td><td>####</td><td>0.97</td><td>####</td><td>0.97</td><td>####</td><td>0.97</td><td>Pcs</td><td>####</td><td>r Spray Bottle</td><td>41025</td><td></td><td>24</td></tr>
</table></td></tr>
<tr><td>2</td><td>Used clustering and faceting in OpenRefine to merge similar product names. Also normalized simple units.</td><td>
</td></tr>
<tr><td>3 (switch)</td><td>Tried to handle complex cases but OpenRefine transforms broke down. Switched to ChatGPT to draft regex + unit conversion logic. Iterated prompts with sample rows.</td><td>
code,name,quantity,unit,std_quantity,std_unit,cost,supplier,category
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories
,,,,,,,,Accessories</td></tr>
<tr><td>4</td><td>Took the regex and categorization rules generated by ChatGPT, applied them in OpenRefine/Sheets. Created derived columns: size_value, size_unit, standardized_unit. Some mismatches still needed override.</td><td>
</td></tr>
</table>

<table>
<tr>
<td>5
(switch)</td>
<td>Moved to Google Colab with ChatGPT-generated Pandas script for reproducibility. Cleaned dataset, applied conversion, flagged negatives (NEG_QTY), missing costs, and reorder threshold.</td>
<td>I've uploaded a messy inventory export (`Inventory.xlsx`) into this Colab notebook. Please help me build a **reproducible pipeline in Colab** that:

1. Ingests the Excel file (skip the first 6 metadata rows).

2. Normalizes column names (`code, name, quantity, unit, cost, supplier`).

3. Cleans and standardizes units (e.g., `ml`, `pcs`, `kg`, and convert `tola` → `11.66 ml`).

4. Flags problems:
• Negative or missing quantities
• Missing or zero costs
• Possible duplicate items (fuzzy matches on name)

5. Categorizes products into exactly **three categories**:
• **Perfume Oils** → almost any item measured in a liquid unit (ml, tola, L), unless it is clearly Alcohol/Ethanol or another solvent.
• **Empty Bottles** → items whose names include words like "Bottle", "Cap", "Vial", "Spray", or "Atomizer".
• **Accessories** → everything else (e.g., packaging parts, tools, non-liquid items).

6. Generates three CSV outputs into the Colab environment:
• `items_clean.csv` with normalized data
• `flags.csv` with all detected issues
• `BuyList_YYYY-MM-DD.csv` for *Perfume Oils* below a stock threshold of 20 units, grouped by supplier

Please propose the steps first, then generate **deterministic Pandas code cells** I can run in Colab. Keep LLM use minimal (only for categorization suggestions if needed); all math and conversions must be reproducible. Also include any Colab-specific snippets (e.g., `files.upload()` for file import, `files.download()` for export).</td>
</tr>
<tr>
<td>6</td>
<td>Exported cleaned CSVs to Google Sheets. Used filters to generate a buy list for low-stock oils, grouped by supplier. The final dataset now had standardized names, normalized units, and actionable flags.</td>
<td>supplier,code,name,quantity,unit,std_quantity,std_unit,cost,stock_threshold
,03-003,(1) Aqua Serenade - Inspired By Acqua Di Gio AR,0.423,kg,0.423,kg,103.37824436906348,20
,03-052,(1) Floral Radiance - Inspired By 1 Million P.R,0.2,kg,0.2,kg,155.0673665535952,20
,01-135,(1) Ivory Journey - Inspired By White L,0.9990000000000001,kg,0.9990000000000001,kg,67.61834813549177,20
,04-272,(2) Aqua Serenade - Inspired By Acqua Di Gio AR,1.0,kg,1.0,kg,112.58813331381702,20
,05-540,(2) Floral Radiance - Inspired By 1 Million P.R,2.214,kg,2.214,kg,131.01049567244957,20
,02-099,(2) Ivory Journey - Inspired By White L,0.912,kg,0.912,kg,61.598721653895296,20
,41037,100ml Toronto,-165.0,pcs,-165.0,pcs,0.0,20
,23-012,10ML Glass syringe Italian,4.0,pcs,4.0,pcs,11.42731978297175,20
,23-024,10ML Plastic Syringe,2.0,pcs,2.0,pcs,3.8091065943239166,20
,23-013,20ML glass Syringe Italian,15.0,pcs,15.0,pcs,9.858193110586082,20
,40002,27L Plastic Gallon,3.0,pcs,3.0,pcs,37.59111107720269,20
,08-026,AL Qurashi Musk,0.968,kg,0.968,kg,114.27319782971749,20
,15-013,Abu Saud Oud,0.9615,kg,0.9615,kg,593.4138836426231,20
,26-199,Accord Particulier Givenchy   2.997,اكورد بارتيكولير جيفينشي,kg,2.997,kg,120.21367162877849,20
,07-018,African Sunset - Inspired By Bal d'Afrique BY,2.847,kg,2.847,kg,212.54170079172835,20
,05-004,Al Habshi Musk,0.965,kg,0.965,kg,73.53804894164116,20
,27-245,Alloy Extreme - Inspired By L'Homme Le Parfum YSL - Eco,2.0,kg,2.0,kg,43.9990868956395,20
,03-048,Almond Smoothie - Inspired By Pegasus PDM,0.8330000000000001,kg,0.8330000000000001,kg,153.99680413473823,20
,01-144,Almond Soleil - Inspired By L'Homme Ideal G,0.939,kg,0.939,kg,114.27319782971749,20
,26-028,Amber Aura - Inspired By Ambre Absolute T.F,1.45,kg,1.45,kg,122.55046493373372,20
,05-103,Amber Bloom - Inspired By More Than Words X,0.9760000000000001,kg,0.9760000000000001,kg,192.0362804543032,20
,27-169,Amber Cacao - Inspired By Choco Musk A.R - Eco,2.0,kg,2.0,kg,43.9990868956395,20
,26-040,Amber Celestial - Inspired By Elysium R.P,1.894,kg,1.894,kg,197.99589103037775,20
,01-236,Amber Dawn - Inspired By Burberry Black BB,0.9300000000000002,kg,0.9300000000000002,kg,72.50851644492654,20
,05-065,Amber Eclipse - Inspired By Interlude AM,0.7760000000000001,kg,0.7760000000000001,kg,205.6917560934915,20
,27-099,Amber Eclipse - Inspired By Interlude AM - Eco,2.0,kg,2.0,kg,43.9990868956395,20
,02-438,Amber Embrace - Inspired By SWY Amber AR,1.092,kg,1.092,kg,143.26943519211673,20
,03-095,Amber Glow - Inspired By Narcisco Ambrée N.R,0.8910000000000001,kg,0.8910000000000001,kg,114.27319782971749,20
,02-252,Amber Kick - Inspired By Grand Soir MFK,2.454,kg,2.454,kg,110.90205116823073,20
,05-044,Amber Legacy - Inspired By Epic Man AM,0.9430000000000001,kg,0.9430000000000001,kg,222.34734545021746,20
,02-014,Amber Musk - Inspired By Body BB,0.9320000000000002,kg,0.9320000000000002,kg,61.5987216538953,20
,27-006,Amber Musk - Inspired By Body BB - Eco,2.0,kg,2.0,kg,43.9990868956395,20
,27-064,Amber Musk - Inspired By Molecule 02 E.M - Eco,2.0,kg,2.0,kg,43.9990868956395,20
,27-056,Amber Nectar Oud - Inspired By Shay Oud A.D - Eco,2.0,kg,2.0,kg,43.9990868956395,20
,27-162,Amber Rose - Inspired By Touch Me NB - Eco,2.0,kg,2.0,kg,43.9990868956395,20
,01-298,Amber Smoke - Inspired By Aoud Amber Mon,1.462,kg,1.462,kg,89.0980803312808,20
,27-240,Amber Spice - Inspired By Alexandria II Xerjoff - Eco,2.0,kg,2.0,kg,43.9990868956395,20
,02-496,Amber Sultan - Inspired By Al Fareed AF,0.9710000000000001,kg,0.9710000000000001,kg,142.99703241082838,20
,26-053,Amber Sunrise - Inspired By Amber Oud AJ,0.969,kg,0.969,kg,114.27319782971749,20
,27-289,Amber Tango - Inspired By Amber Romance V.S - Eco,2.0,kg,2.0,kg,43.9990868956395,20
,02-482,Amber Tree - Inspired By Opus XV King Blue AM,0.9360000000000002,kg,0.9360000000000002,kg,208.99566275428762,20</td>
</tr>
</table>