

Tagging Summary Report

1. Tagging Approach & Logic

Approach Overview:

To systematically tag the free-text fields - Complaint, Cause, and Correction - with the categories provided in the Taxonomy sheet, we followed a semi-automated, logic-driven approach using Python (Pandas + Fuzzy Matching via fuzzywuzzy). The task focused on identifying and mapping relevant terms or phrases to predefined categories under five main tags:

1. Symptom_Condition
2. Symptom_Component
3. Root_Cause
4. Fix_Condition
5. Fix_Component

We began by cleaning the taxonomy and task datasets, including handling inconsistent whitespace, lowercasing all text, and removing special characters. We then applied fuzzy matching logic with a scoring threshold to map each free-text description to its closest taxonomy category. If the match score was high (>85), we confidently assigned it; if the score was moderate (65-85), we flagged it for manual review.

To maintain transparency and critical thinking, we logged confidence levels in a separate "Observations" column. We also cross-validated our matches against two provided ground truth examples to ensure at least 50% tagging accuracy.

Potential Insights (Food for Thought):

- Patterns in Root Causes and Symptom Components can help prioritize high-impact issues or common failures across products.
- Mapping the most frequent Fix Components against Root Causes may reveal engineering gaps or repetitive fault trends.
- A semantic search model (embedding-based) or domain-specific NLP model could further improve tagging

Tagging Summary Report

accuracy beyond lexical similarity.

- Building a dashboard using this tagged data could help visualize category distributions and assist in faster resolution planning.

Overall, this tagging strategy balances automation and human logic to meet the required accuracy threshold and provides a strong foundation for deeper analysis.