Smart Tariff & Fee Assistant (STAFF)

CMPT-310

Milestone 1 — Progress Report (Oct 26) — Group 26

Group Members:

- Kazi Boni Amin (kba109@sfu.ca) — 301591325

- Ertugrul Yurtseven (eya38@sfu.ca) — 301588905

- Brayden Yee (@sfu.ca) — 301559654

1. What Has Been Accomplished So Far

Data & Files

Structured CSVs prepared: samples_clean.csv, samples_noisy.csv, hs_mapping.csv, rules_restricted.csv, rules_gift.csv.

EDA notebook (notebooks/01_EDA.ipynb) runs: schema checks, class balance, top tags, price/weight histograms (figures saved).

Preprocessing / Features

Implemented src/features.py to clean + engineer features:

tags → CountVectorizer (custom tokenizer in src/tokenizers.py).

price, weight → coarse bins.

origin, dest, gift → one-hot.

Saved artifacts: artifacts/embeddings/* and dataset-specific preprocessors (preprocessor_clean.joblib, preprocessor_noisy.joblib).

Models & Baselines

Implemented KNN (src/knn.py) with cosine or L2, Top-k outputs + raw weights.

Implemented ID3 (src/id3.py) with entropy/information gain and path explanations.

Training/eval pipeline (src/pipeline.py) saves models, metrics JSON, and confusion PNGs.

Baseline results (clean, KNN k=3 cosine):

- acc_valid = 1.00, acc_test = 0.50
- macro_f1_valid = 1.00, macro_f1_test = 0.33
- Accuracy@3_valid = 1.00, Accuracy@3_test = 0.50

CLI & Rules

CLI (src/cli.py) predicts Top-k for a single item using the matching preprocessor + model.

Rule engine (src/rules.py) loads CSV rules and emits Restricted/Gift flags (e.g., "perfume" → flammable; --gift 1 triggers gift notes).

Verified end-to-end examples (e.g., headphones, lithium battery pack, toy gift) with predictions and flags.

Documentation

Project README.md added with run instructions, file-by-file explanations, examples, and troubleshooting.

2. What Has Fallen Behind

BERT/text embeddings: deferred (TA guidance) to keep scope explainable and reproducible.

ID3 evaluation depth:

The Decision Tree (ID3) model has been implemented and runs correctly, but we haven't yet tested how different settings (hyperparameters) affect its performance.

Hyperparameters are like tuning knobs that control how the model learns — for example:

- max_depth: how many levels the tree can grow (too deep → overfits, too shallow → misses patterns)
- min_samples: minimum number of samples required to split further

  We've experimented with KNN using different k values (3, 5, 7), but we still need to run similar experiments for ID3 to compare how tree depth and splitting rules affect accuracy and Macro-F1 on both the clean and noisy datasets. These evaluations are planned for the next milestone.

Rules coverage: Basic rules are in place (e.g., perfume/gift); however, the restricted list needs to be expanded and validated (e.g., ensuring the lithium row is present/clean).

Challenges/blockers

Small dataset → some classes have <2 samples (non-stratified splits; unstable test accuracy).

CSV formatting pitfalls (quoting/mismatched columns) required cleanup for hs_mapping.csv / rules files.

3. Any Project Changes:

Scope change: shifted from free-text + BERT to structured inputs only (tags + numeric/categorical) using KNN/ID3.

- Why: TA feedback to avoid heavy NLP, prioritize explainability, reproducibility, and faster progress.

Output change: emphasize Top-k predictions (useful UX) and include Accuracy@3 in evaluation.

4. Evidence of Meaningful Progress

Reproducible pipeline with saved preprocessors, embeddings, models, metrics, and plots.

Working CLI demo producing Top-k HS codes with raw scores and compliance flags.

Baseline metrics reported; EDA completed; documentation shipped.