CustomsEdge

Building a Hybrid ML/LLM Classification System

For HTS Codes

From CBP Rulings to Production Pipeline

Zachary Gillett | Customs Edge | February 2026

CustomsEdge

The Problem: HTS Classification

- Every international shipment requires HTS classification
- 10-digit codes determine duties, compliance, and entry
- Manual process: 4+ hours per manifest
- Error rates: 15%+ causing delays, penalties, audits

HTS Code Structure:

```
8525.80.4000
  85    = Chapter (Electrical machinery)
  8525  = Heading (Transmission apparatus)
  8525.80 = Subheading (Television cameras)
  8525.80.40 = US Line (Digital cameras)
```

CustomsEdge

The ML Challenge: Why This is Hard

- 18,000+ target classes with sparse training data
- Domain mismatch: Legal ruling text ≠ product descriptions
- Extreme hierarchical class imbalance
- GRI edge cases require legal reasoning, not just pattern matching
- Real-world validation: UFC pilot launching April 1st

4+ hours → 15 minutes

87ms inference | 70% Top-1 on 18K classes | 85% Top-3

CustomsEdge

Data Procurement: CBP CROSS Database

- 100K+ customs rulings publicly available at rulings.cbp.gov
- Challenge: Legal ruling text != product descriptions
- Scraping approach: Playwright + targeted chapter filtering
- Focus chapters: 84 (machinery), 85 (electronics), 90 (optical), 94 (lighting)

```
commodity_groupings = ['Electronics', 'Machinery']
scraper = CBPScraper(headless=True)
total = await scraper.run(
    headings,
    commodity_groupings=commodity_groupings,
    use_pdf=True  # Extract FACTS sections
)
```

CustomsEdge

Synthetic Data: LLM Codification

- Problem: Rulings say 'the article is a device designed for...'
- Need: Retail-style descriptions for real-world matching
- Solution: Claude API transforms legal text to product descriptions
- Result: 3,400+ labeled training samples

Before (CBP Ruling):

"The merchandise at issue is a wireless stereo headset with microphone and USB dongle transceiver..."

After (Retail Description):

"Sony PlayStation Wireless Stereo Headset - Gaming headset with built-in microphone"

CustomsEdge

Feature Engineering: 513 Dimensions

Feature Type	Count	Why This?
SBERT Embeddings	384	Semantic similarity; handles synonyms/paraphrasing
Material Keywords	20	Explicit signals embeddings miss (HTS cares about material)
Product Type Keywords	50	Domain priors + interpretability for debugging
Attribute Keywords	46	Customs-relevant flags (wireless, professional, etc.)
Brand + Numeric	13	Price/weight often determines HTS subheading

Key insight: SBERT alone got 62% — adding keyword flags pushed to 70%

```
embedding = sbert_model.encode(description)  # 384 dims (semantic)
keyword_flags = extract_domain_keywords(text)   # 129 flags (explicit)
features = np.concatenate([embedding, keyword_flags])  # hybrid approach
```

CustomsEdge

4-Stage Hierarchical Classification

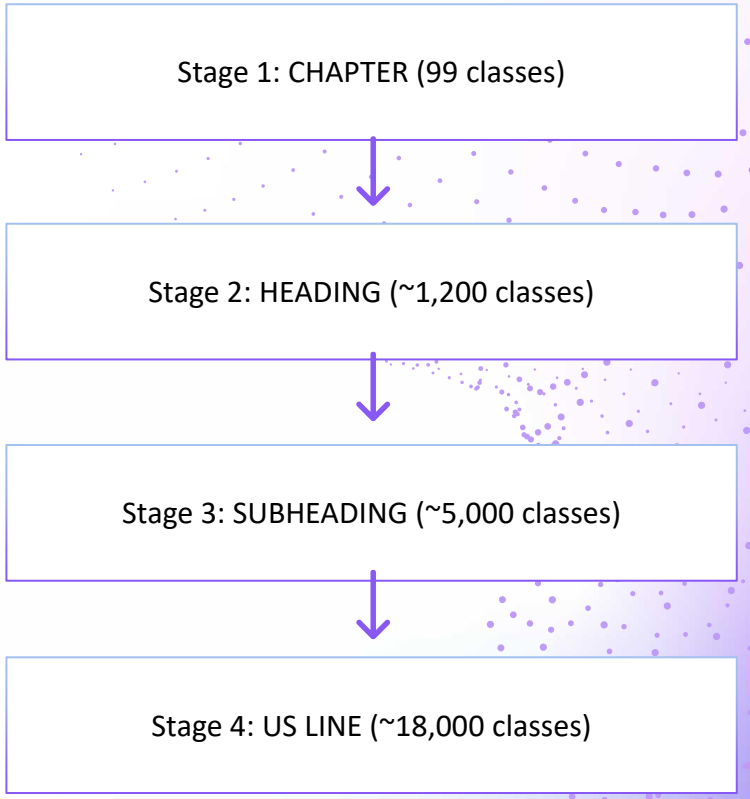
Why hierarchical beats flat classification:

HTS structure is naturally hierarchical

Fewer classes per level = higher accuracy

Previous predictions constrain next level

Each stage adds previous probabilities as features



CustomsEdge

Model Performance by Level

Why hierarchical beats flat classification:

Level	Classes	Top-1 Accuracy	Top-3 Accuracy
Chapter	99	95%	99%
Heading	1,200	84%	93%
Subheading	5,000	76%	90%
US Line	18,000	70%	85%

Context:

Random baseline on 18K classes = 0.005%

Training set: 3,400 samples (actively expanding)

Top-3 at 85% is operationally useful — human reviews top 3 candidates

End-to-end inference: ~87ms

CustomsEdge

Hybrid Approach: ML + LLM

ML Handles:

- Speed (87ms inference)
- Consistency across batches
- Bulk classification

LLM Handles:

- Edge cases and ambiguous products
- Low-confidence verification
- Written justification (GRI rules)

Result: Fast AND Explainable

Confidence threshold triggers LLM review

CustomsEdge

5-Layer Verification Pipeline

Layer 1: Format Validation	`4-10 digits, XXXX.XX.XXXX format`
Layer 2: Local Database Lookup	`35,859 valid HTS codes from USITC`
Layer 3: Online API Fallback	`USITC API verification (5s timeout)`
Layer 4: ML Cross-Check	`Ensemble agreement validation`
Layer 5: LLM Verification	`Claude review for low confidence`

CustomsEdge

Product Enrichment Pipeline

Problem: Manifests often have sparse product data

Multi-source enrichment strategy:

Priority 1: B&H Photo API (professional equipment)

Priority 2: Best Buy API (consumer electronics)

Priority 3: SerpAPI web search (fallback)

Priority 4: Equipment database (catalog lookup)

Enriched fields: Full specs, dimensions, materials, country of origin

Better input = Better classification

CustomsEdge

Current System Status

Working demo: Processes real customs manifests

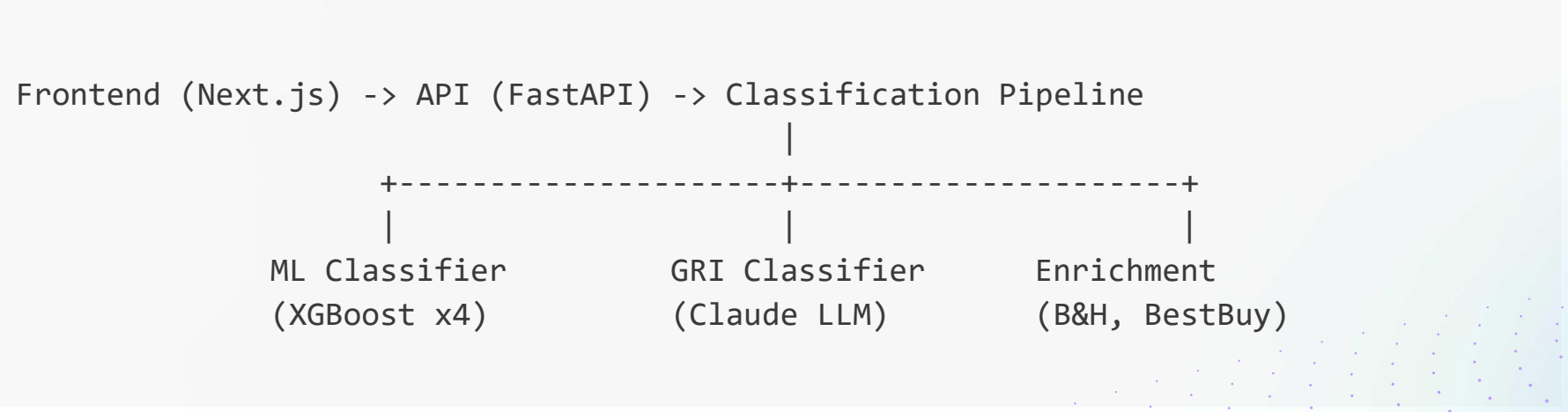
Tech stack: FastAPI + Next.js + XGBoost + Claude

10-country export templates (Mexico, Canada, Brazil, China...)

Job tracking and progress monitoring

OCR intake: PDF/image manifests via Tesseract

35,859 HTS codes in local database



CustomsEdge

Roadmap & Open Problems

April 1st: UFC Pilot Launch

Real-world validation with 300-400 manifests/year

Technical Roadmap:

Expanding training data beyond Chapters 84/85/90/94

Handling ambiguous GRI edge cases (Rule 3 disputes)

Multi-language description support (Spanish, Mandarin)

Confidence calibration improvements for cascading classifiers

Open Problems (would love input):

Hierarchical classification with extreme class imbalance

Calibrating confidence across cascading models

CustomsEdge

What's Next + Collaboration

Building in public — happy to share war stories

What exists today:

Working demo processing real customs manifests

3,400+ training samples, 4-stage hierarchical pipeline

UFC pilot launching April 1st

Open to:

Collaborators and feedback on the approach

War stories on hierarchical classification

Technical co-founder (if you're interested)

Happy to chat after or grab coffee this week

CustomsEdge

Google Calendar

zackgillett@customsedge.com

Let's Connect