

# NLP Business Case

## Automated Customer Reviews Analysis

---

Project 3 — IronHack AI Engineering Bootcamp

Group 4: Krzysztof Giwojno & Felipe Doria

February 2026

Live Demo: <https://giwojno.pl/review-analyzer/>

Component	Method	Key Result
Classification	RoBERTa fine-tuned	95.48% accuracy, 0.791 macro F1
Clustering	TF-IDF + K-Means	6 clusters, silhouette 0.2364
Summarization (API)	Claude Sonnet API	39 summaries, \$0.43 total
Summarization (Local)	Flan-T5-base (250M)	6 summaries, \$0 cost
Deployment	Static site, OVH hosting	giwojno.pl/review-analyzer

# **Table of Contents**

1. Data Exploration
2. Sentiment Classification
3. Product Clustering
4. Review Summarization — API
5. Review Summarization — Local Model
6. Web Deployment
7. Key Learnings & Conclusion

# 1. Data Exploration

The dataset contains 28,332 customer reviews for 65 Amazon-branded products, sourced from the Datafiniti Amazon Consumer Reviews dataset on Kaggle. Reviews span 2009–2019 and cover products including Fire Tablets, Kindle E-Readers, AmazonBasics batteries, Echo smart speakers, and various accessories.

## Dataset Overview

Metric	Value
Total reviews	28,332
Unique products	65 (all Amazon-branded)
Columns (original)	24 (14 retained, 10 dropped)
Mean rating	4.51 / 5
Median review length	17 words / 87 characters
Duplicate review texts	10,164 (35.87%) — kept as legitimate

## Sentiment Class Distribution

Star ratings were mapped to three sentiment classes. The resulting distribution is heavily imbalanced — Positive dominates at 90.2%, creating a major challenge for classification.

Class	Star Rating	Count	Percentage
Positive	4–5	25,545	90.2%
Negative	1–2	1,581	5.6%
Neutral	3	1,206	4.3%

## Key Decisions

- Kept duplicate review texts — they are legitimate short reviews from different users (0 full row duplicates)
- Dropped 10 columns with >99% null values or metadata with no predictive value
- Used **categories** column (rich, multi-label) for clustering instead of primaryCategories (only 2 dominant values)
- No minimum review length filter — even single-word reviews carry sentiment signal
- No text preprocessing for transformers — models handle raw text; heavy cleaning can hurt performance

## 2. Sentiment Classification

We fine-tuned a RoBERTa-base transformer in two stages: first pre-trained on 650K Yelp reviews (3-class sentiment), then fine-tuned on the Amazon dataset with class weights to handle the severe class imbalance. The v2 model achieved 95.48% accuracy on a held-out test set of 5,667 reviews.

### v1 vs v2 Comparison

The v1 model used Yelp-pretrained weights without Amazon fine-tuning or class weights. The v2 model added stratified splitting, class weights (Negative 6.0x, Neutral 7.8x, Positive 0.4x), and early stopping (patience 3). Best checkpoint was found at step 3,200 (epoch 2.4 of 5).

Metric	v1	v2	Improvement
Accuracy	87.23%	95.48%	+8.25%
Macro F1	0.635	0.791	+0.156
Negative F1	0.699	0.822	+0.124
Neutral F1	0.269	0.569	+0.300
Positive F1	0.937	0.982	+0.045

### Per-Class Results (v2 Test Set)

Class	Precision	Recall	F1	Support
Positive	0.985	0.979	0.982	5,110
Negative	0.804	0.842	0.822	316
Neutral	0.550	0.589	0.569	241

### Error Analysis

Only 256 errors on the test set (4.5% error rate). Most errors occur at the Positive/Neutral boundary — genuinely ambiguous reviews like 3-star reviews with mild praise. The worst error type (Negative predicted as Positive) is very rare at only 12 cases. Sample errors reveal sarcasm, mixed sentiment, and likely mislabeled training data.

### 3. Product Clustering

---

We clustered 65 products into 6 meta-categories using TF-IDF vectorization on product names combined with cleaned category labels, followed by K-Means clustering. Features: unigrams + bigrams, 200 max features, English stop words removed.

#### Cluster Selection

Tested K=2 to K=10. Best overall silhouette was K=9 (0.277), but the project required 4–6 clusters. Within this range, K=6 achieved the best silhouette score (0.2364).

#### 6 Clusters

Cluster	Products	Reviews	Avg Rating	% Positive
Fire Tablets	20	14,396	4.56	93.2%
Batteries & Household	7	12,086	4.45	86.1%
E-Readers	10	1,049	4.66	95.2%
Smart Speakers	9	632	4.54	91.9%
Accessories	11	138	4.31	82.6%
Media & Home	8	31	4.39	83.9%

#### Key Observations

- Fire Tablets and Batteries dominate with 93% of all reviews combined
- E-Readers have the highest satisfaction — 95.2% positive, only 1.5% negative
- Batteries & Household has the highest negative rate (9.5%) among major clusters
- Accessories and Media & Home are small clusters with limited review material

## 4. Review Summarization — API

We used the Anthropic Claude Sonnet API to generate recommendation articles from customer reviews. For each product, we sampled the most informative reviews (5 positive, 5 negative, 3 neutral — prioritized by length) and sent them with aggregate statistics to the LLM via structured prompts.

Metric	Value
Model	Claude Sonnet (claude-sonnet-4-20250514)
Temperature	0.3 (consistent, factual output)
Total API calls	39
Cluster summaries	6 (~600 words each)
Product summaries	33 (products with 20+ reviews)
Total cost	\$0.43 USD
Cost per summary	~\$0.011

Each cluster summary includes: category overview, top 3 recommended products with evidence, common complaints (3–5 recurring issues), worst-rated product with reasoning, and buying recommendation. Product summaries include an overall verdict (Buy/Consider/Avoid), top 3 strengths, top 3 weaknesses, and target buyer profile.

## 5. Review Summarization — Local Model

In parallel, we ran a fully local summarization pipeline using Google's Flan-T5-base (250M parameters) on an NVIDIA RTX 4050 GPU. The approach builds structured 'evidence briefs' per category with product rankings and TF-IDF complaint extraction, then prompts T5 to generate articles.

### Data Pipeline Highlights

- Product ranking:  $\text{avg\_rating} \times \log(1 + \text{review\_count})$  — balances quality with volume
- Minimum 20 reviews required for worst-product selection
- TF-IDF complaint extraction from negative reviews (unigrams + bigrams)
- Complaint caching to avoid redundant computation

### API vs Local Comparison

Aspect	API (Claude Sonnet)	Local (Flan-T5-base)
Output quality	Full recommendation articles	1–2 sentences, factual
Structure followed	All sections present	Failed to produce sections
Complaint analysis	Integrated into narrative	Raw TF-IDF terms
Cost	\$0.43	Free
Infrastructure	API key + internet	GPU (RTX 4050)

Flan-T5-base (250M parameters) lacks the capacity for structured multi-paragraph generation. It excels at short factual QA but cannot follow complex output formatting or write coherent articles. Flan-T5-large (780M) or Flan-T5-xl (3B) would likely perform significantly better. The API-generated summaries are used in the web deployment.

## 6. Web Deployment

The project is deployed as a static single-page web application at <https://giwojno.pl/review-analyzer/>, hosted on OVH shared hosting. No backend, database, or API keys are needed at runtime — all results are pre-computed and served as JSON files.

### Architecture

Component	Technology
Frontend	Single HTML file, vanilla CSS + JS
Data	4 JSON files generated from project outputs
Fonts	Google Fonts (DM Sans + DM Serif Display)
Hosting	OVH shared hosting
Deployment	SFTP via deploy.sh (credentials in .env)
Theme	Dark editorial design, responsive

### Features

- **Dashboard** — stats overview, sentiment distribution, rating charts, reviews by category
- **Categories (API)** — browse Claude-generated recommendation articles per cluster
- **Categories (Local)** — browse Flan-T5 generated summaries for comparison
- **Products** — filterable table of all 65 products, click for AI summaries
- **Reviews** — explorer with filters (category, sentiment, rating, text search)
- **About** — model details, project info

## 7. Key Learnings & Conclusion

---

### Technical Learnings

- **Class weights are critical** for imbalanced datasets — Neutral F1 jumped from 0.269 to 0.569
- **Early stopping** prevented overfitting and found the best checkpoint automatically (epoch 2.4 of 5)
- **Model size matters for generation** — Flan-T5-base (250M) cannot handle structured article generation; API models produce vastly superior output at minimal cost (\$0.43)
- **Static sites work** — pre-computing results removes backend infrastructure needs, making deployment trivial on shared hosting
- **Simpler ranking formulas** can be more defensible — `avg_rating x log(review_count)` is clean and effective

### Project Outcome

The project delivers a complete end-to-end NLP pipeline: from raw Amazon reviews to a deployed web application with sentiment classification (95.48% accuracy), product clustering (6 meaningful categories), and AI-generated recommendation articles. Both API-based and local summarization approaches were implemented and compared, demonstrating the trade-offs between cost, quality, and infrastructure requirements. The live demo at [giwojno.pl/review-analyzer](http://giwojno.pl/review-analyzer) makes all results interactively browsable.