Automated Customer Review Analysis

An NLP Pipeline for Sentiment Classification, Product Clustering & AI-Generated Summaries

Chandler Shortlidge

Data Science & AI Bootcamp | Ironhack Berlin

February 2026

1. Introduction

This report documents the development of an automated product review analysis pipeline. The project addresses the challenge of manually analyzing thousands of Amazon product reviews by applying NLP models to classify sentiment, cluster product categories, and generate summary recommendation articles using generative AI.

The pipeline processes 28,332 Amazon product reviews across 24 data columns, sourced from the Datafiniti dataset on Kaggle. The three core tasks are: (1) sentiment classification of review text, (2) product category clustering into meta-categories, and (3) AI-generated blog-style summaries with product recommendations and warnings.

2. Data & Preprocessing

2.1 Dataset Overview

The primary dataset contains 28,332 Amazon product reviews with fields including review text, star ratings (1-5), product names, and product category strings. The category strings were found to be long, comma-separated lists containing multiple overlapping tags (e.g., 'Fire Tablets,Tablets,All Tablets,Amazon Tablets,Computers & Tablets'), requiring significant preprocessing before use.

2.2 Sentiment Labeling

Star ratings were mapped to three sentiment classes: ratings of 1-2 were labeled as Negative, 3 as Neutral, and 4-5 as Positive. This mapping revealed a severe class imbalance: 90.16% of reviews were Positive, 5.58% Negative, and 4.26% Neutral.

2.3 Class Balancing

To address the class imbalance, downsampling was applied. The smallest class (Neutral) contained 1,206 samples, so all three classes were downsampled to 1,206 reviews each, producing a balanced dataset of 3,618 reviews. This ensures the classification model learns meaningful patterns across all sentiment classes rather than defaulting to the majority class.

Class	Before	After
Positive	25,568 (90.16%)	1,206
Negative	1,558 (5.58%)	1,206
Neutral	1,206 (4.26%)	1,206

3. Task 1: Sentiment Classification

3.1 Baseline: Pre-trained Model

An off-the-shelf sentiment model (cardiffnlp/twitter-roberta-base-sentiment) was used as a baseline. This model, trained on Twitter data, achieved only 64% accuracy on the Amazon review dataset. Neutral class recall was particularly poor at 26%, indicating significant domain mismatch between Twitter posts and product reviews.

3.2 Fine-tuned RoBERTa

To improve performance, RoBERTa-base was fine-tuned on the balanced review dataset using Hugging Face's Trainer API. The model was trained for 3 epochs with a batch size of 16 on a Google Colab T4 GPU. A 70/30 train-test split was used, resulting in 2,532 training and 1,086 test samples. A DataCollatorWithPadding handled variable-length review tokenization.

3.3 Results

Class	Precision	Recall	F1-Score	Support
Negative	0.84	0.85	0.85	382
Neutral	0.77	0.74	0.75	355
Positive	0.89	0.91	0.90	349
Overall Accuracy			**0.83**	**1,086**

Fine-tuning improved overall accuracy from 64% to 83%. The most significant improvement was in the Neutral class, where F1-score jumped from 0.35 to 0.75. This demonstrates the importance of domain-specific training: the pre-trained Twitter model could not capture the nuances of product review language, particularly the ambiguity inherent in 3-star reviews.

3.4 Training Dynamics

Training loss decreased across all three epochs (0.74, 0.47, 0.27), while validation loss showed a similar but slower decline (0.69, 0.54, 0.50). The widening gap between training and validation loss by epoch 3 suggests early signs of overfitting, indicating that additional epochs would likely not improve generalization. Three epochs proved to be an appropriate stopping point.

4. Task 2: Product Category Clustering

4.1 Initial Approach: Embeddings + K-Means

The initial approach used TF-IDF vectorization and Sentence Transformer embeddings combined with K-Means clustering (k=5) on the raw category strings. This produced noisy, overlapping clusters because dominant terms like 'Electronics' and 'Amazon' appeared across nearly all category strings, overwhelming the signal. Even after removing the top 10 most frequent terms, the clusters remained poorly separated.

4.2 Solution: LLM-based Classification

After K-Means failed to produce clean groupings, a local LLM (Qwen 2.5, run via Ollama) was used to classify each product into predefined meta-categories. The 60 unique category strings in the dataset were each sent to Qwen with a prompt instructing it to select from six target categories. The results were stored as a dictionary mapping and applied to all 28,332 rows via pandas .map().

4.3 Final Meta-Categories

Meta-Category	Description
Electronics	Audio, video, smart home, cameras
Tablets & E-readers	Fire tablets, Kindle devices
Health & Beauty	Batteries, personal care, health products
Home & Kitchen	Kitchen appliances, storage, organization
Office Supplies	Laptop stands, desk accessories, filing
Pet Supplies	Dog crates, cat litter, pet carriers

The LLM classifications were validated against the dataset's existing 'primarycategories' column, confirming strong alignment between Qwen's predictions and the ground truth labels. This approach proved more effective than unsupervised clustering for this particular dataset.

5. Task 3: Review Summarization

5.1 Approach

For each meta-category, the top 3 highest-rated products and the worst-rated product were identified using grouped aggregation (mean rating and review count). A minimum review threshold was applied to ensure statistical significance. Review texts for these products were then fed to Qwen 2.5 via Ollama with a structured prompt requesting a blog-style summary article.

5.2 Output Structure

Each generated article includes: the top 3 products with their average ratings, number of reviews, and key differentiators; the top customer complaints for each product; and the worst-rated product in the category with specific reasons to avoid it. The summaries were generated for all six meta-categories and saved as structured JSON for reproducibility.

5.3 Example Output

For the Electronics category, the model identified the Fire HD 8 Tablet (4.60 avg, 2,443 reviews), Fire Kids Edition Tablet in Pink (4.53 avg, 1,676 reviews), and Fire Kids Edition Tablet in Blue (4.53 avg, 1,425 reviews) as the top products. The generated article highlighted key differences in display size, parental controls, and battery life, while identifying the Amazon Kindle Replacement Power Adapter (2.8 avg, 5 reviews) as the worst product due to durability and international shipping issues.

6. Technical Stack

Component	Tools
Data Processing	pandas, NumPy
Visualization	matplotlib, seaborn
Sentiment Model	Fine-tuned RoBERTa-base (Hugging Face Transformers)
Fine-tuning Infrastructure	Hugging Face Trainer API, Google Colab T4 GPU
Category Classification	Ollama / Qwen 2.5 (local LLM)
Blog Generation	Ollama / Qwen 2.5 (local LLM)
Evaluation	scikit-learn (classification_report, confusion_matrix)

7. Challenges & Learnings

7.1 Challenges

Class imbalance: The original dataset was 90% positive reviews. Without downsampling, any model could achieve 90% accuracy by always predicting positive, learning nothing useful about negative or neutral sentiment.

Domain mismatch: The pre-trained CardiffNLP model was trained on Twitter data and struggled with product review language. Neutral reviews were particularly difficult, as the text often contradicted the star rating (e.g., negative-sounding text with a 3-star rating).

Category noise: The raw category strings contained overlapping, redundant tags that made unsupervised clustering ineffective. Dominant terms like 'Electronics' appeared in nearly every product, preventing K-Means from finding meaningful separations.

7.2 Key Learnings

Fine-tuning matters: Domain-specific fine-tuning improved accuracy from 64% to 83%, with the biggest gains on the hardest class (Neutral F1: 0.35 to 0.75). This underscores the value of training on task-specific data rather than relying on general-purpose models.

LLMs as classifiers: When traditional clustering failed, using a local LLM (Qwen 2.5) as a zero-shot classifier produced clean, accurate category labels. This approach combines the flexibility of language understanding with the structure of predefined categories.

Iterative problem-solving: Each failure informed a better approach. The final pipeline combines multiple AI techniques (fine-tuned transformers, local LLMs, prompt engineering) chosen based on what worked rather than theoretical preference.

8. Future Improvements

Aspect-level sentiment: Moving beyond overall sentiment to detect per-feature opinions (e.g., battery life, screen quality, durability) within individual reviews would provide more granular insights for product recommendations.

Expanded dataset: Including reviews from additional platforms (Best Buy, Walmart) would improve model robustness and provide broader product coverage.

Real-time pipeline: Deploying the system as a live service that continuously ingests new reviews and updates recommendations would demonstrate production-readiness.

Interactive dashboard: Building a user-facing web application where users can explore sentiment trends and product comparisons visually would add practical value.