30-09-2025

Project NLP | Business Case: Automated Customer Reviews

Motivation / Problem Statement

- Why analyzing customer reviews matters (e.g., too many reviews, hard for marketing teams to extract insights).
- Business need → “We want to know how customers feel and which products are best/worst.”

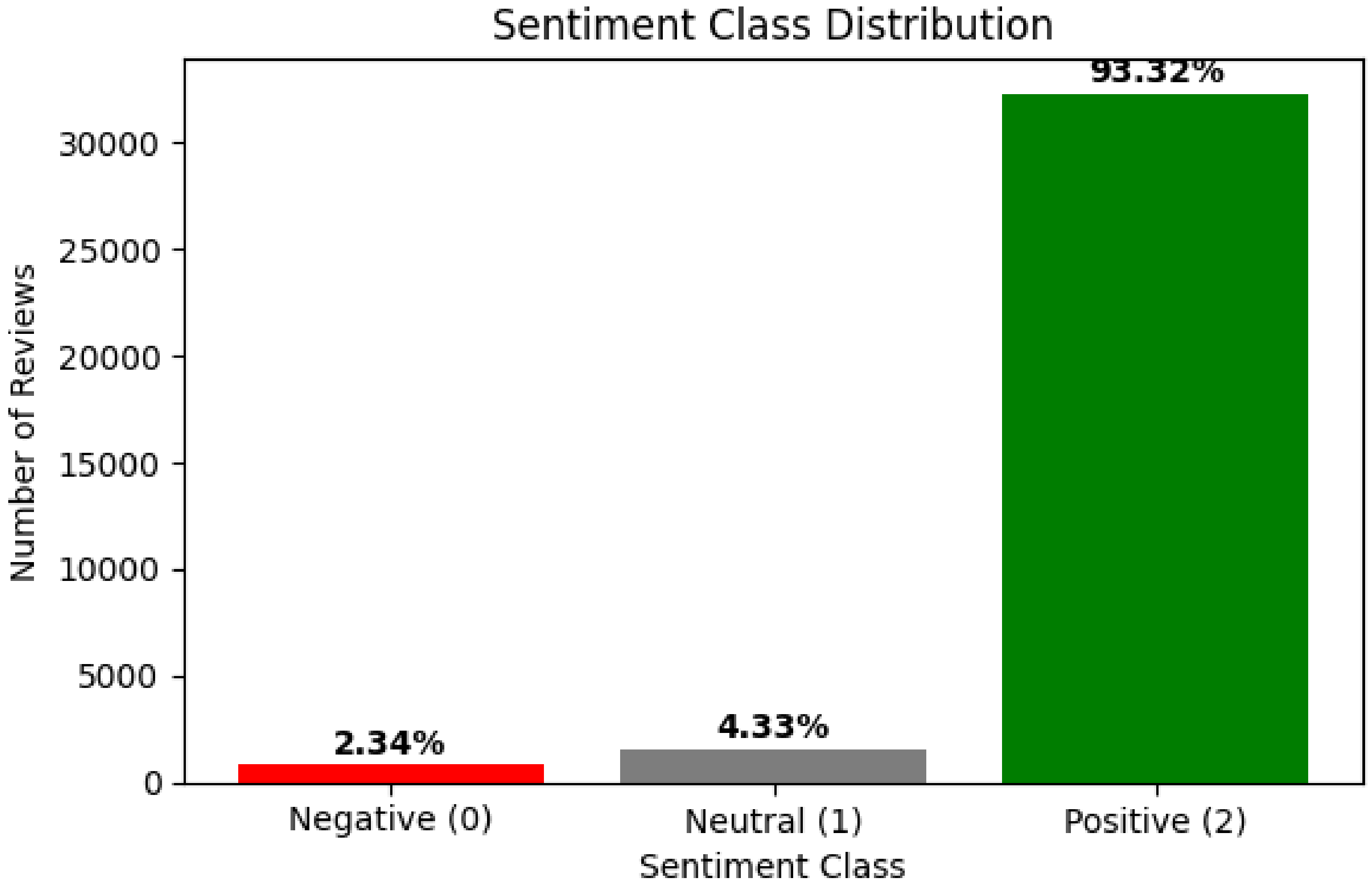
Project Overview

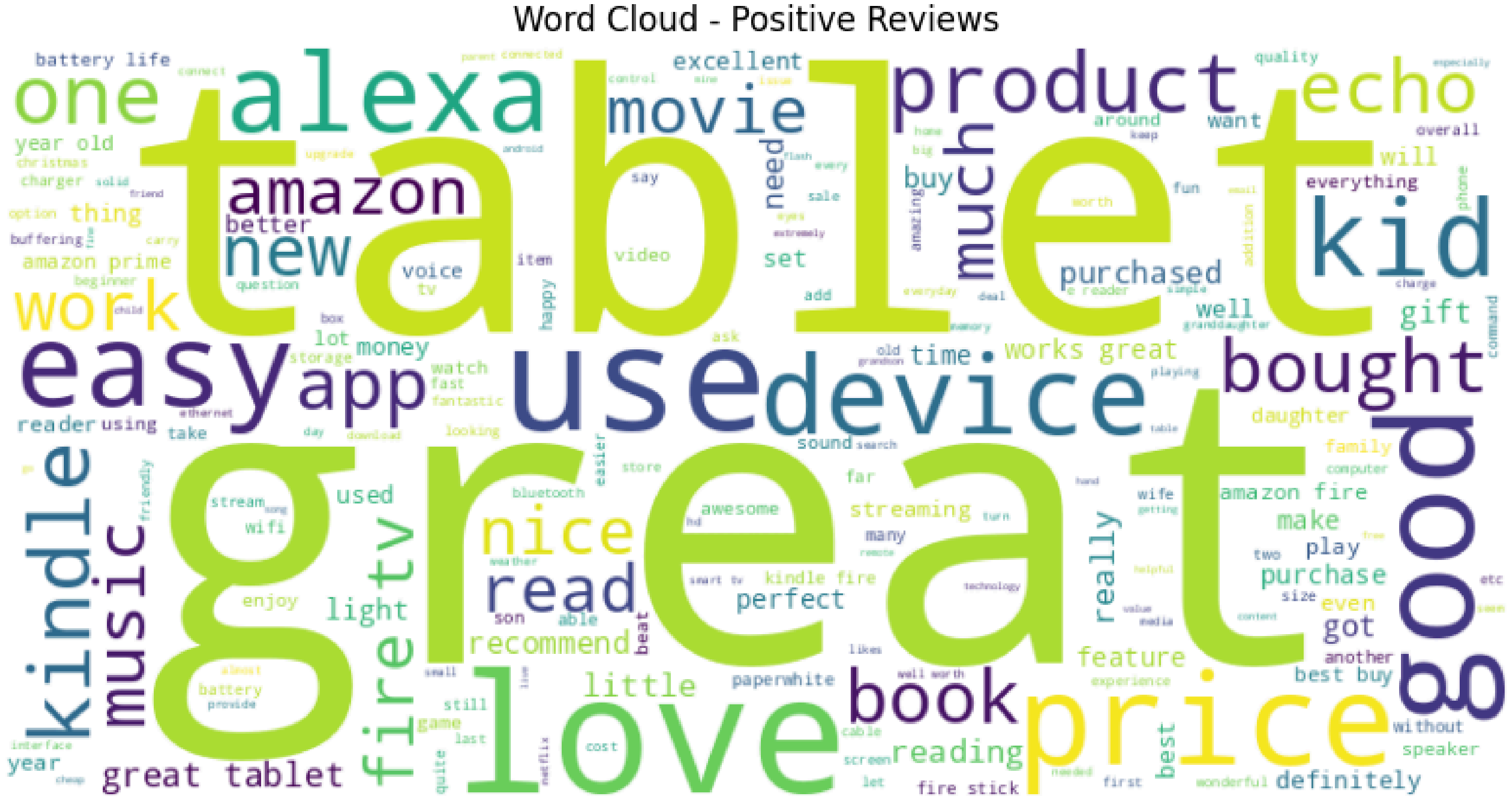
- Task 1: Sentiment Classification (Positive/Neutral/Negative)
- Task 2: Product Category Clustering (meta-categories)
- Task 3: Review Summarization (blog-style articles)
- Final Goal: A dashboard/website for the Marketing team

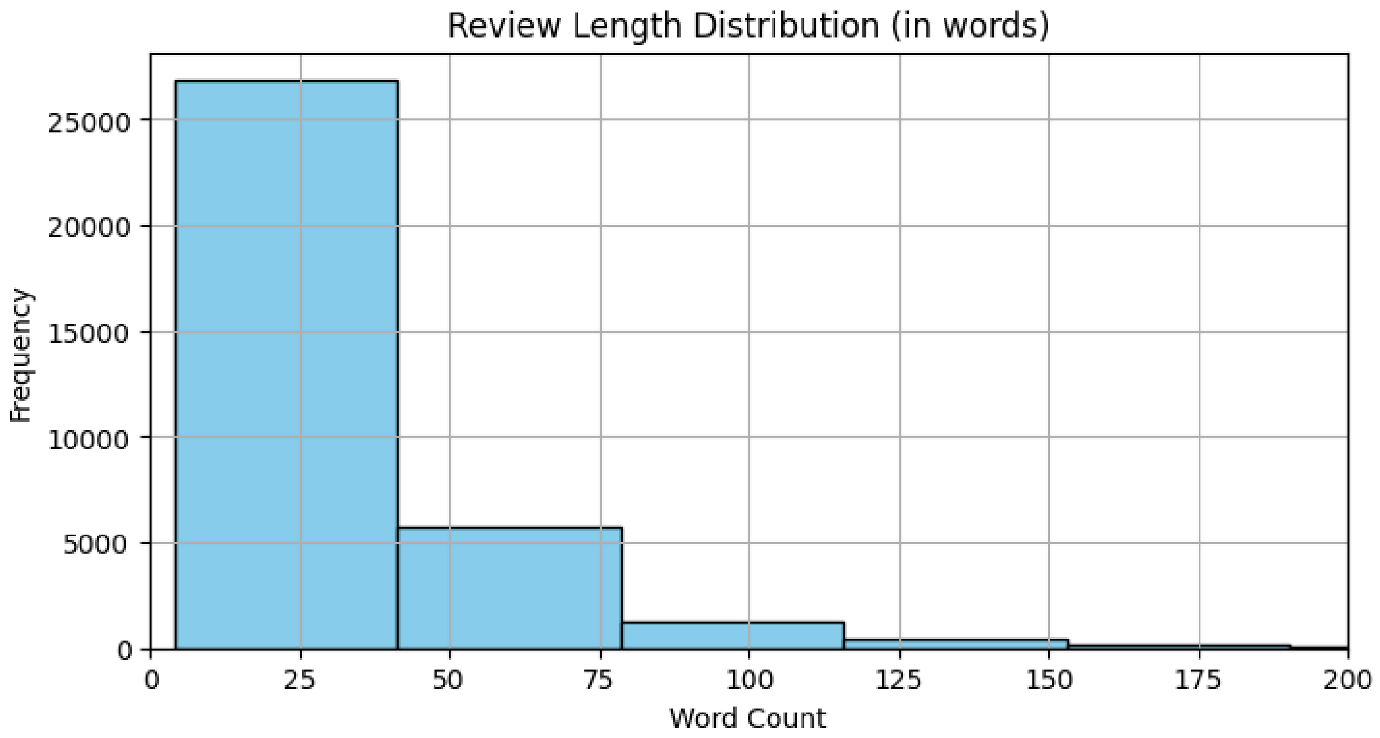
Dataset and cleaning

- Source: Kaggle (Amazon product reviews dataset) - 1429_1.csv
- Combines review.text + review.title - why?
- Text processing - duplicates/symbols/blanks/very short reviews (noise)

Data Visulization







Task 1: Review Classification

Sentiment Mapping

- Explain mapping:
- ⭐ 1–2 → Negative, ⭐ 3 → Neutral, ⭐ 4–5 → Positive.
- Why: converts numeric stars into 3 simple classes.

Train/Test Split + Tokenization

- Train/test/split
- model - nlptown/bert-base-multilingual-uncased-sentiment
- Tokenization

parameter-efficient fine-tuning (PEFT) using LoRA

- Why?? → comverting the original 5 class classification into 3 classes

Balancing the dataset

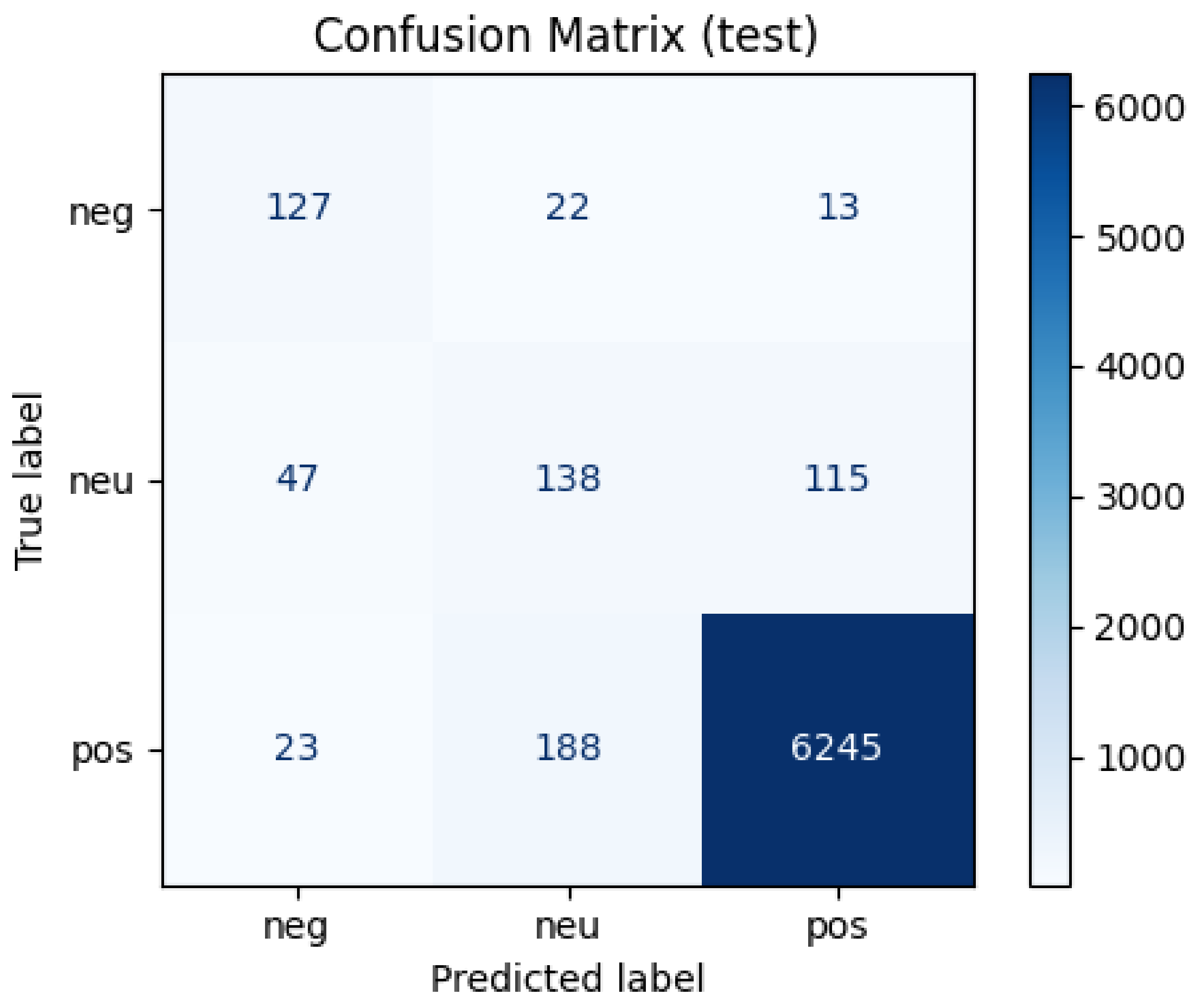
- to handle Positive-heavy imbalance
- Using a formula that gives larger weights to minority classes

Model Training

Task 1: Review Classification

Model evaluation

```
*** Eval metrics ***
eval_loss: 0.7039
eval_accuracy: 0.9410
eval_precision_macro: 0.6737
eval_recall_macro: 0.7371
eval_f1_macro: 0.7023
eval_precision_weighted: 0.9468
eval_recall_weighted: 0.9410
eval_f1_weighted: 0.9436
eval_precision_neg: 0.6447
eval_recall_neg: 0.7840
eval_f1_neg: 0.7075
eval_precision_neu: 0.3966
eval_recall_neu: 0.4600
eval_f1_neu: 0.4259
eval_precision_pos: 0.9799
eval_recall_pos: 0.9673
eval_f1_pos: 0.9736
eval_runtime: 12.9364
eval_samples_per_second: 534.7720
eval_steps_per_second: 16.7740
epoch: 3.0000
```



Task 1: Review Classification

Inference

- Makes 3 classifications after fine tuning

Outputs

Task 2: Product Category Clustering

- Problem: Too many specific product names/categories.
- Goal: Group products into 4–6 meta-categories (E-readers, Batteries, Accessories, Non-electronics, etc.).
- Approach: TF-IDF vectorization + KMeans clustering.

TF -IDF

- On “asins” and “categories” column

Picking Number of Clusters

- Used silhouette score for k=4,5,6 → chose best k.
- If we used a sample, we find each sampled row’s closest cluster and compute the silhouette score (how well-separated the clusters are). Higher silhouette = better separation.

Top terms per cluster

- to name clusters based on top words
- Picking the top words per cluster
- Assign clusters to products & reviews - create id for each cluster and join it with the reviews and an auto-generated human label based on top terms.

Output

- product_id, product_name, normalized review text, rating

Task 3: Summarization

- Goal: Generate short articles for each category:
- Top 3 products + key differences
- Common complaints
- Worst product
- Used BART (facebook/bart-large-cnn) model for text generation.

Evidence → Article

- Evidence bundle built per category: avg stars, ratios, top phrases (pros/cons).
- Fed into generative model → blog-style summary.
- Example snippet: “For E-Readers, Kindle Paperwhite leads with long battery life…”

Final Dashboard / Website

- Built with Streamlit.
- Features:
- Live Review Sentiment Classifier
- Category insights (distribution, best/worst product)
- Generated summaries per category

Conclusion & Next Steps

- ✅ Built pipeline: Sentiment → Clusters → Summaries → Dashboard.
- Benefits: Saves time for marketing, shows competitive products.
- Future work:
- Real-time review uploads
- Smarter cluster labeling
- Deploy on cloud (AWS/GCP).