BỘ GIAO THÔNG VẬN TẢI
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI THÀNH PHỐ HỒ CHÍ MINH

DATA MINNG
BÁO CÁO CUỐI KỲ

PHÂN TÍCH NỘI DUNG TRÒ CHUYỆN TRÍCH XUẤT VẤN ĐỀ PHỔ BIẾN VỀ TÀI SẢN

GIẢNG VIÊN HƯỚNG DẪN:
HỌ VÀ TÊN SINH VIÊN THỰC HIỆN:
LỚP:
MÃ SỐ SINH VIÊN:

Thành phố Hồ Chí Minh, ngày .. tháng .. năm 2024

Analyzing Public Discussions for Product Insights

TEAM : Group 4
COURSE : UTH Data Mining — Final Report
PIPELINE : Reddit + Tiki collection → Parquet normalization → EDA → Topic modeling (TF-IDF → SVD → KMeans) + Aspect sentiment

Executive Overview

- We mine 16 hardware-centric Reddit communities (plus complementary Tiki reviews) to surface recurring product issues, pros/cons, and topic trends.
- Arctic Shift harvesting bypasses API caps to deliver multi-year coverage stored as harmonized Parquet.
- A CLI (`run_pipeline.py`) turns filtered corpora into TF-IDF/SVD features, KMeans clusters, and aspect-level sentiment artifacts.
- Planned transformer sentiment fine-tuning was documented but descoped after cost/benefit analysis; VADER backs the delivered pipeline.
- Outputs include reusable joblibs/JSON summaries, PNG dashboards, and presentation assets in `extra/`.

Data Sources & Collection

Reddit & Tiki rationale

- Reddit supplies rich, text-first, community-moderated threads; Tiki adds verified purchase feedback from the Vietnamese market.
- Facebook and TikTok were deprioritized due to API scarcity, bot noise, and media-first formats.
- Subreddit mix spans laptops, phones, audio, smart home, photography, PC building, and ergonomics—capturing both enthusiast and troubleshooting conversations.

Arctic Shift workflow

- Historical dumps (Academic Torrents) feed the Arctic Shift collector, eliminating PRAW's 1 000 post ceiling per subreddit.
- Tiki review dumps augment Reddit for cross-source validation when available.
- Data is normalized via Polars/Nushell from 32 JSONL exports to `posts.parquet` and `comments.parquet`, enforcing consistent schemas.

Corpus snapshot

Split	Rows	Columns	Unique subs
Posts	134121	27	16
Comments	1300190	16	16

Data Engineering & Integration

- Filtering uses case-insensitive alias matching (e.g. “hd600”, “fiio ft1”), with optional subreddit restriction.
- Brand/alias tokens are added to the stopword list to avoid dominating TF-IDF features.
- Post/comment parity is preserved by harmonizing key identifiers (`name`, `subreddit`, `link_id`, `parent_id`) before concatenation.
- `run_pipeline.py` logs reproducibility metadata (command-line, package versions) and writes all intermediate artifacts to the specified output directory.

Exploratory Data Analysis

- EDA assets reside under `eda/` (PNG + CSV) for quick reuse in slides and dashboards.

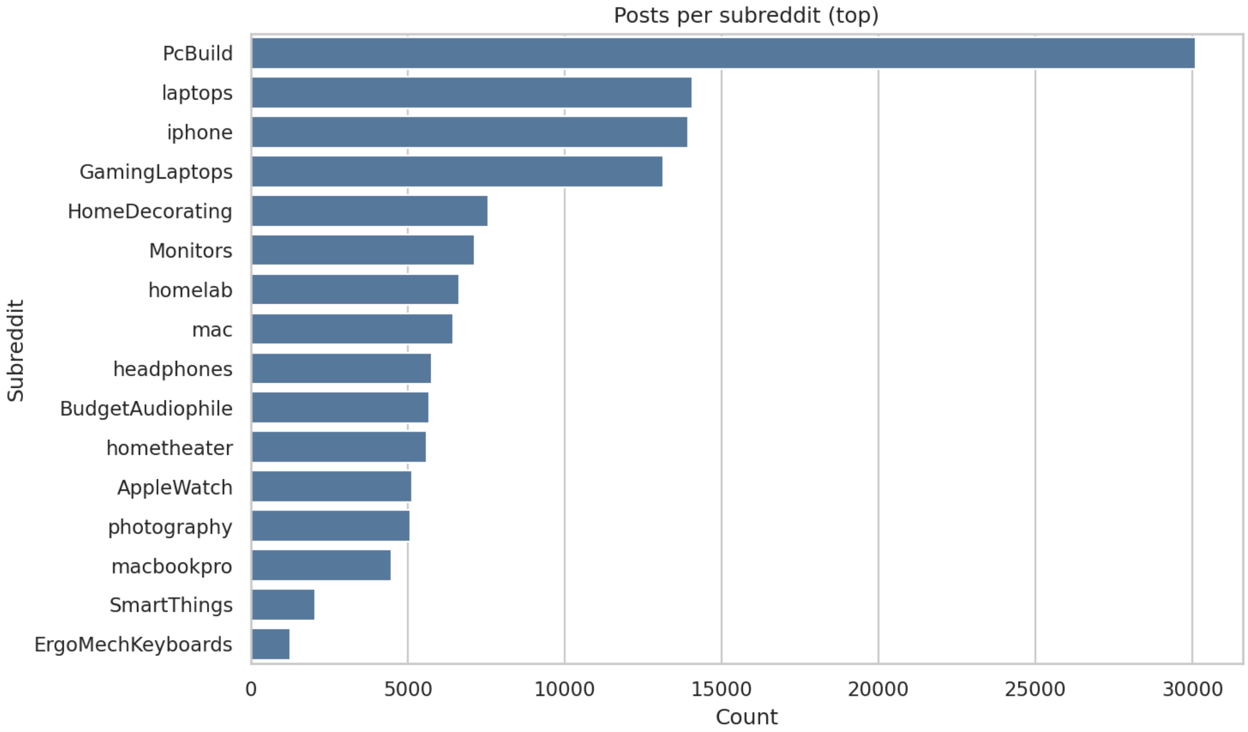

Figure 1: Post counts by subreddit — PcBuild, iPhone, and GamingLaptops dominate volume.

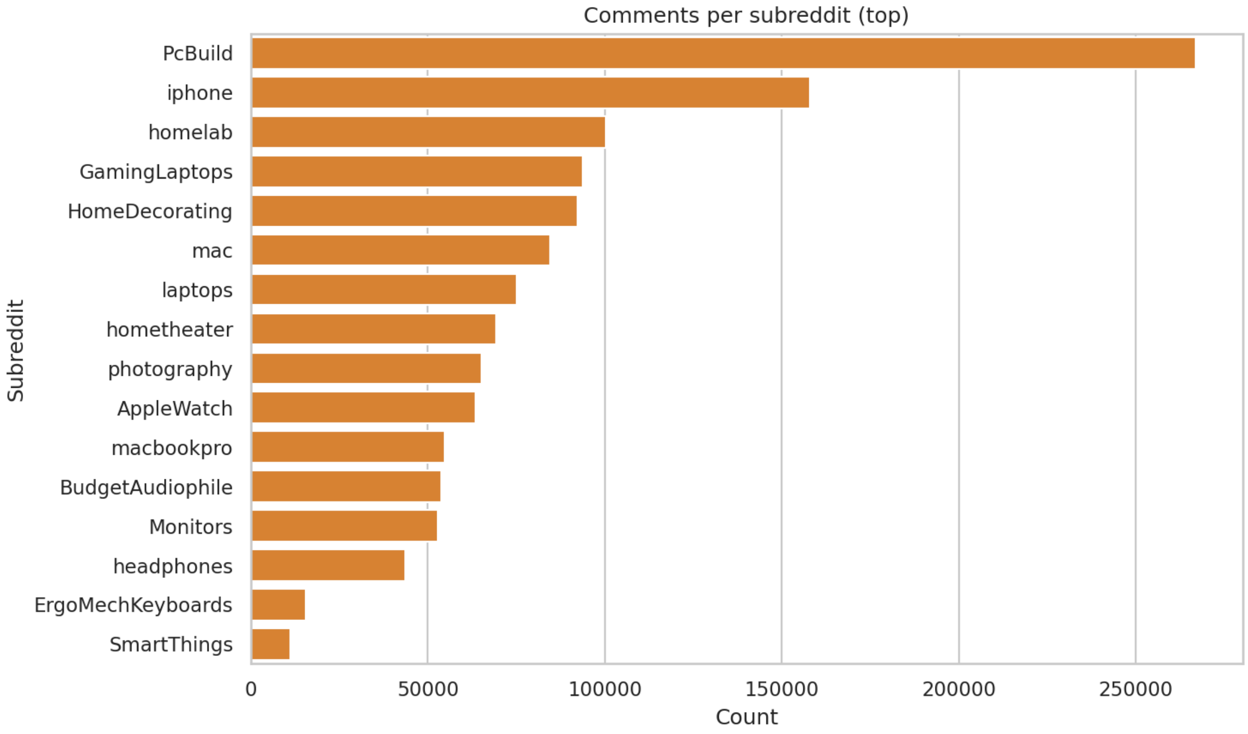

Figure 2: Comment counts: troubleshooting-heavy communities (PcBuild, homelab) lead engagement.

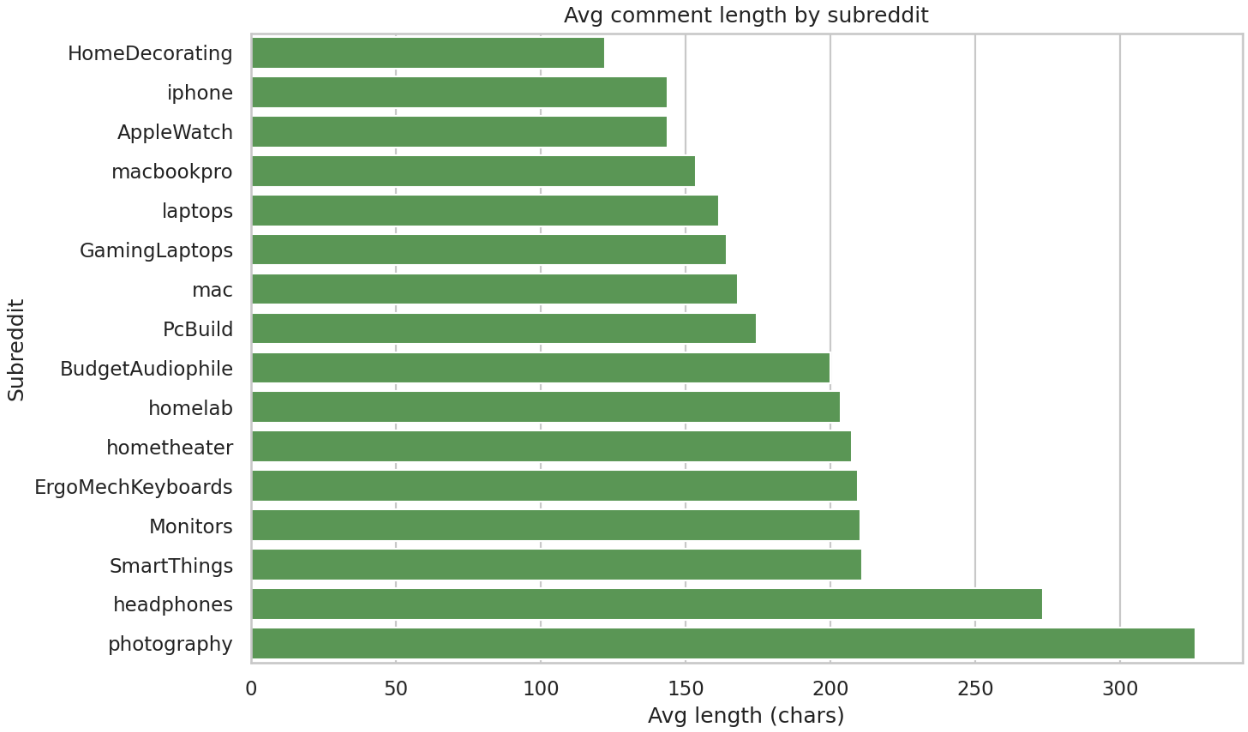

Figure 3: Average comment length highlights deep technical discussions in r/macbookpro and r/AppleWatch.

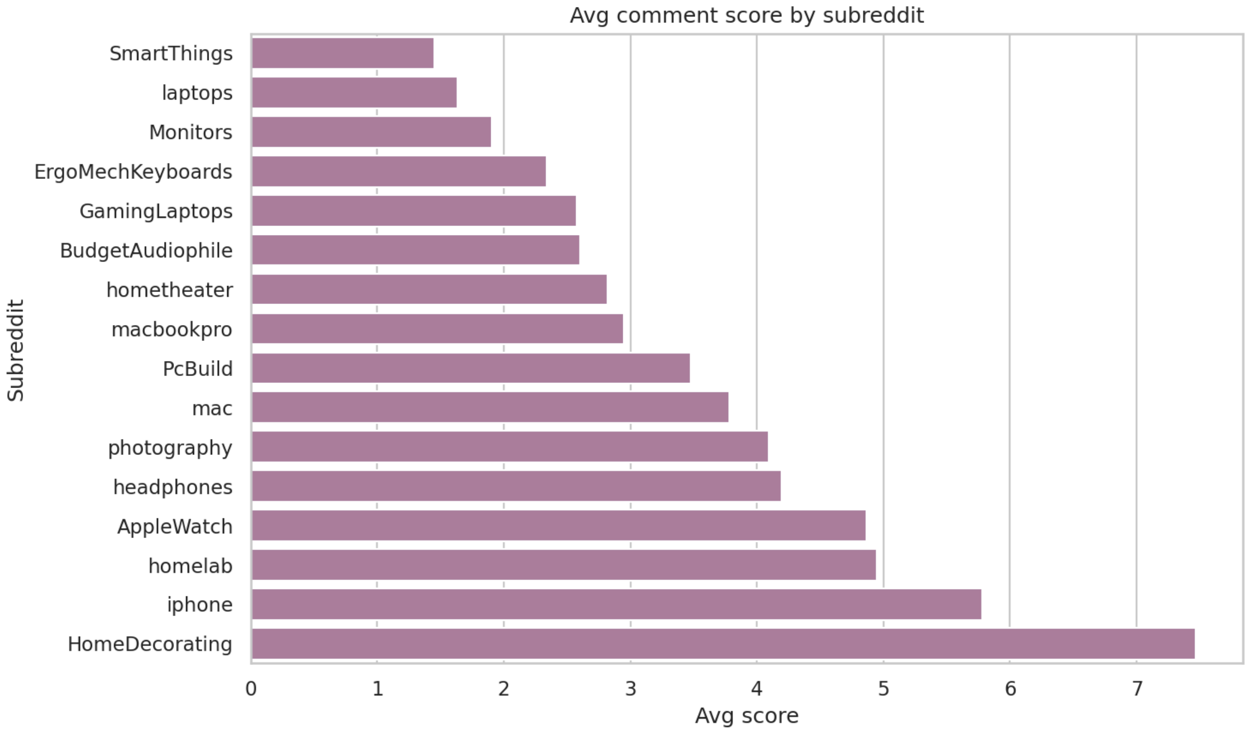

Figure 4: Mean comment scores remain low overall (≤2.5), reinforcing the need for textual insight beyond karma.

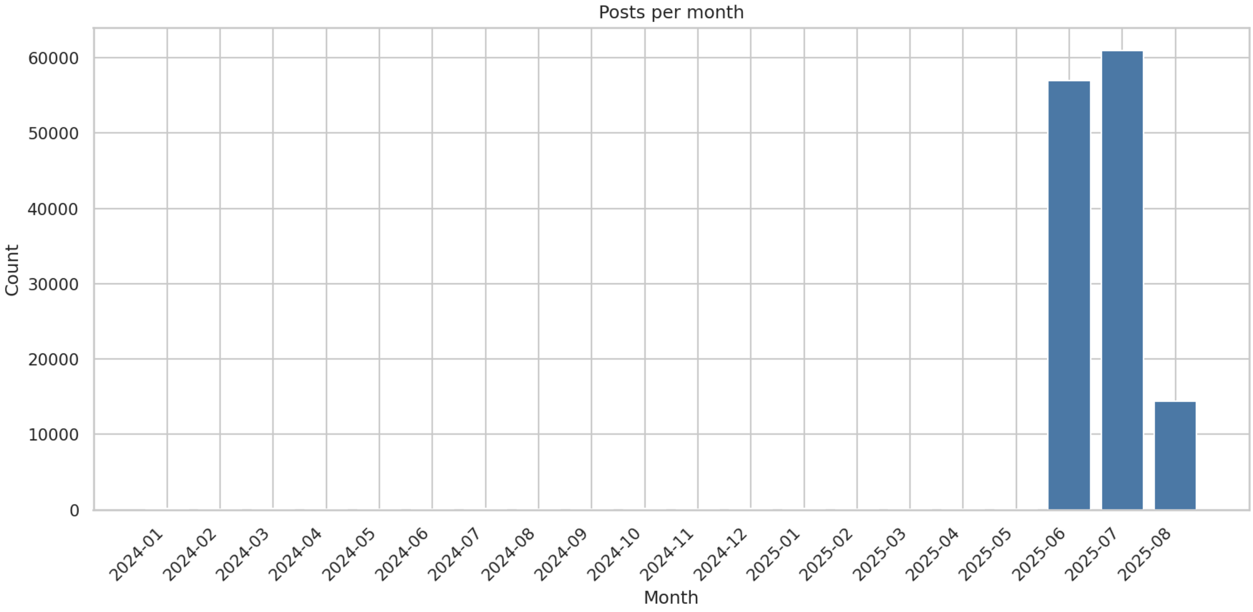

Figure 5: Monthly posting cadence detects seasonal hardware launch spikes.

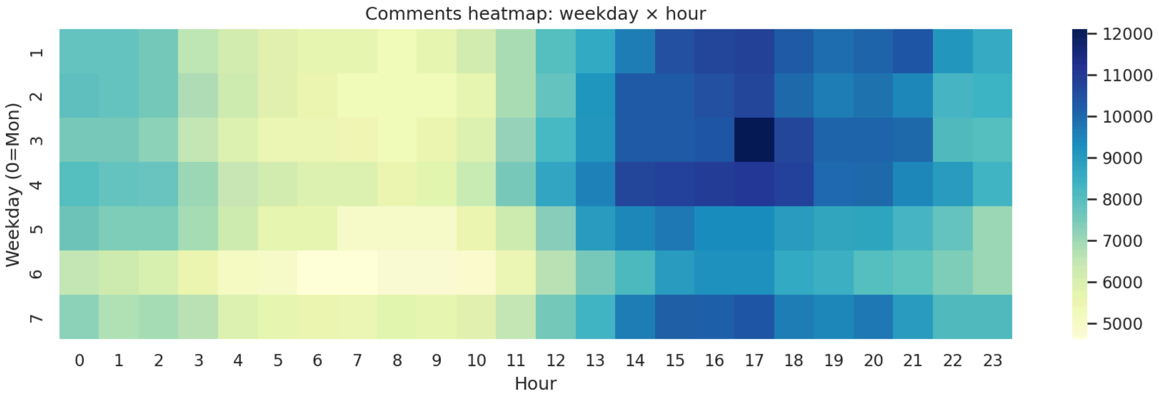

Figure 6: Weekly/hourly heatmap of activity — evenings and weekends drive most conversations.

- Additional visuals: word clouds, score-versus-length scatterplots,

author leaderboards, and CSV exports for deeper analysis (`eda/*.csv`).

Sentiment Analysis — Initial Plan & Decision

- Intended flow: fine-tune `lxyuan/distilbert-base-multilingual-cased-sentiments-student` on a stratified, Gemini-assisted label set (target

≥5 k annotations for calibration and evaluation).

- Obstacles: GPU rental costs, rate-limited labeling throughput, and

limited marginal benefit vs. leveraging unsupervised topic insights.

- Delivered approach: VADER (rule-based) for sentence polarity during

aspect aggregation, retaining the documented plan for future enhancement.

Topic Modeling & Aspect Pipeline

Pre-processing

- Normalize Unicode (NFC), strip URLs/code fences, lowercase, remove non-

alphabetic characters while preserving apostrophes.

- Token filtering retains words longer than two characters and excludes

expanded stopwords (including brand aliases).

- Aliases/subreddit filters yield a focused corpus per product run.

Feature Selection & Dimensionality Reduction

- TF-IDF (`ngram_range = 1–2`) with adaptive `min_df` (3 if N ≥ 300 else

2) and `max_df = 0.95`; vocabulary capped at `min(2000, 3N)` when auto mode is used (`run_pipeline.py:545`).

- TruncatedSVD clamps components to `min(round(0.25N), 200)`

with a floor of 50, logging explained variance for transparency (`run_pipeline.py:560`).

Clustering workflow

- KMeans sweep across `k_min..k_max` (default 3–8) guided by silhouette score; cluster sizes, top TF-IDF terms, and representative comments are persisted as JSON (`run_pipeline.py:603`).

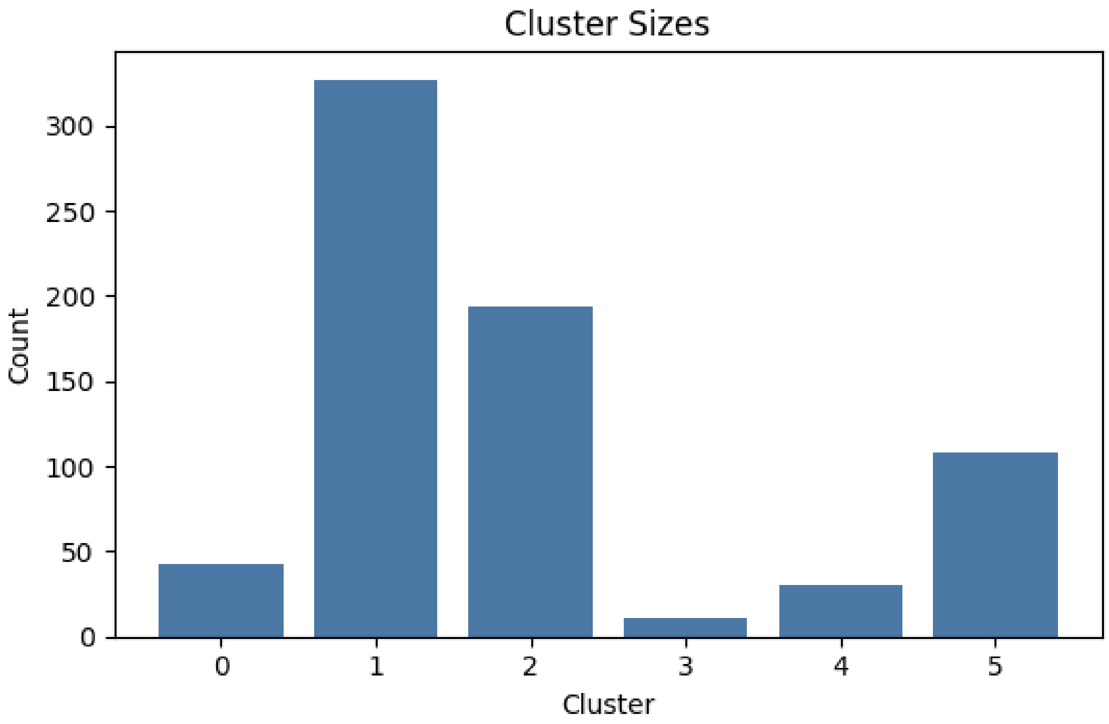

Figure 7: Cluster size distribution (Fiio FT1 example) after SVD-reduced KMeans.

Aspect extraction & sentiment fusion

- Subreddit membership auto-selects aspect seed dictionaries (battery,

thermals, comfort, etc.); optional TF-IDF expansion adds corpus-specific terms.

- Sentence-level pattern matching assigns hits; VADER polarity populates

mentions, average sentiment, and curated quote lists.

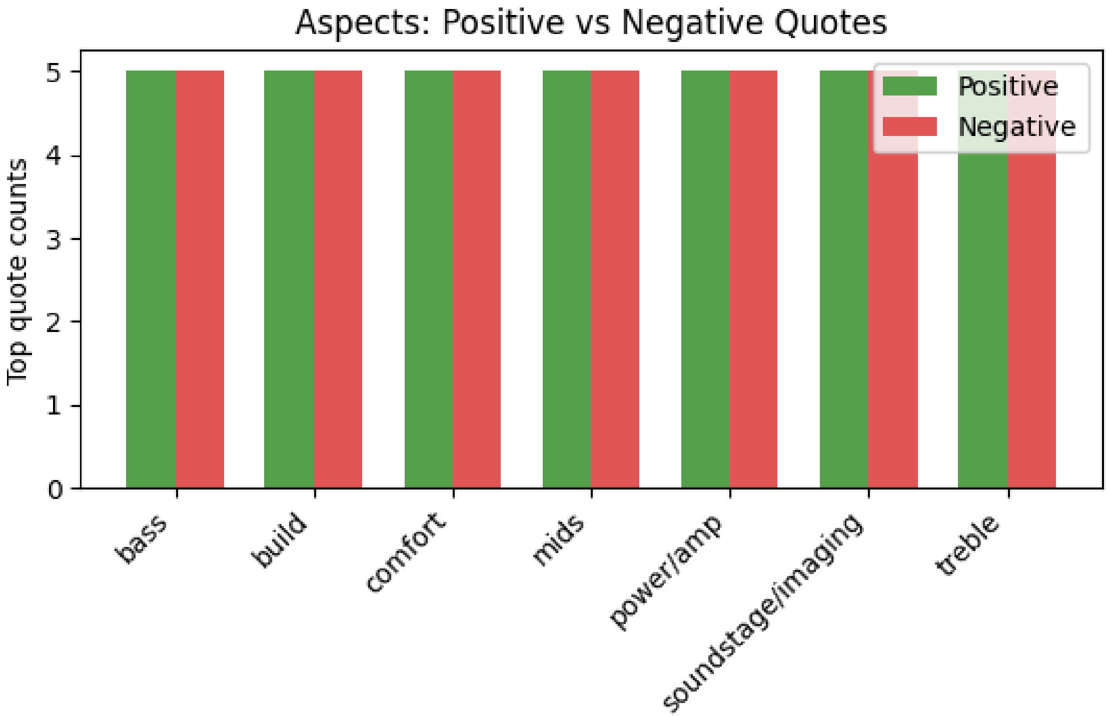

Figure 8: Aspect sentiment summary (Fiio FT1 run): positive vs. negative quote counts.

Key artifacts

- `tfidf_vectorizer.joblib`, `svd_model.joblib`, `svd_explained_variance.json` — reproducible feature pipeline.
- `kmeans_clusters.json`, `cluster_representatives.json`, `cluster_topics.json` — interpretable topic exports.
- `assignments.jsonl`, `aspect_summary.json`, `aspect_summary_by_subreddit.json` — document-level clusters with aspect sentiment.
- Optional PNGs: `cluster_sizes.png`, `aspects_pos_neg.png`.

Deliverables & Usage

- CLI entry point: `run_pipeline.py --data <parquet> --product <name> --aliases <comma-separated> --out <dir> [options]`.
- Configuration knobs control vectorization (`--min-df`, `--max-df`, `--max-feat`, `--ngram-*`), dimensionality (`--svd`), clustering (`--method`, `--k-min`, `--k-max`), and aspect behavior (`--aspect-category`, `--expand-seeds`, `--min-aspect-freq`).
- Artifacts for Fiio FT1, Sennheiser HD600, and Sony WF-1000XM4 are archived under `extra/artifacts_ft1`, `extra/artifacts_hd600`, and `extra/artifacts_m4`.

Limitations & Future Work

- Sentiment remains rule-based; executing the documented fine-tuning plan would handle sarcasm and domain-specific jargon better.

- The NMF/LDA branch is scaffolded but not productized—adding it would support overlapping topic mixtures.
- Broader Tiki integration and multilingual expansion would capture non-English sentiment more faithfully.
- Additional evaluation (topic coherence, stability under resampling, human-in-the-loop validation) should accompany the next iteration.

Appendix

Core assets

- `plan.md` — detailed architecture, alternatives, evaluation heuristics.
- `eda/` — exploratory plots and CSV summaries.
- `run_pipeline.py` (root & `extra/`) — configurable topic + aspect sentiment CLI.
- `extra/bai_thuyet_trinh.typ` — Typst slide deck presented in class.

Example console summary (abridged)

```
N=742, V=1625, min_df=3, max_df=0.95, max_features=2000
SVD components=150, cumulative explained variance=0.72
Chosen K=5 with silhouette=0.41
Aspect battery POS: “Battery life has been excellent…” (+0.68)
Aspect battery NEG: “Battery drains fast when streaming…” (-0.52)
```

End of Report

Next steps (optional):

1. Run typst compile final_report.typ final_report.pdf.
2. Re-run run_pipeline.py for any new products and swap image paths if you

generate additional artifacts.

3. If you revisit sentiment fine-tuning, add the new evaluation results to

the “Limitations & Future Work” section.