A Machine Learning Approach to Sentiment Analysis on Product Reviews

BT5153 Applied Machine Learning for Business Analytics - Group 07
Cui Zhixuan (A0295912W), Liu Kairui (A0296643R), Meng Fanchao (A0296014H), Shiho Tokuda (A0295949A), Wu Zixian (A0297539H)
Github Link: https://github.com/fanchaomeng/BT5153_Group7_2025

Abstract

This study presents a scalable NLP pipeline that transforms customer reviews into valuable insights through sentiment classification, topic clustering, and LLM-based summarization. Using a balanced Amazon Reviews dataset, a fine-tuned RoBERTa model achieved 81.5% accuracy in classification. Clustering of over 14,205 negative reviews using BERTopic with MPNet embeddings revealed 25 semantically coherent themes, such as delivery issues and product defects. Subsequently, Gemini 2.0 Flash, a high-performance generative model was used to synthesize cluster-specific summaries and actionable business recommendations.

1. Introduction

In today's data-saturated digital environment, businesses across industries face an unprecedented volume of customer-generated content, particularly in the form of product reviews. These reviews—distributed across platforms like Amazon, social media, and brand websites—contain rich sentiment, behavioral cues, and product-related feedback. However, due to their unstructured and heterogeneous nature, systematically extracting insights from such data remains a persistent challenge. The diversity in writing styles, use of informal language, and presence of domain-specific vocabulary further complicate the task of interpretation, making manual analysis both time-consuming and infeasible at scale.

This project aims to address these challenges by designing and implementing an end-to-end Natural Language Processing (NLP) pipeline that harnesses recent advancements in machine learning and large language models (LLMs). Our objective is not only to automate the sentiment classification of customer reviews into positive, neutral, and negative categories but also to mine deeper insights from negative feedback—traditionally the most informative yet complex to interpret. By clustering reviews with similar complaints and leveraging LLMs for summarization and recommendation generation, we aim to close the loop between passive sentiment detection and proactive business response.

The proposed solution is structured as a three-stage pipeline. First, it performs supervised sentiment classification to establish a high-accuracy foundation for downstream analysis. Second, it applies semantic clustering (via BERTopic) to group negative reviews into thematically coherent categories, surfacing dominant pain points and emergent issues. Third, it uses generative LLMs (Gemini 2.0 Flash) to synthesize cluster-specific summaries and generate actionable suggestions—bridging the gap between descriptive analytics and operational strategy.

This automated approach offers significant advantages over traditional feedback management systems. It enables businesses to monitor trends in real-time, prioritize high-impact issues, and act quickly on root causes, thereby improving product design, customer service protocols, and overall consumer satisfaction. More importantly, by systematically identifying patterns across thousands of user reviews, the pipeline supports strategic decision-making, enhances brand trust, and reinforces competitive differentiation in increasingly customer-centric markets.

2. Dataset & Exploratory Overview

The dataset used in this project originates from the McAuley Lab's Amazon Reviews 2023 corpus, a large-scale resource containing millions of user-generated reviews across a diverse range of product categories (McAuley Lab, 2023). For exploratory analysis, a 100,000-record sample was randomly drawn from the raw_review_* configurations, with each record containing review metadata (e.g., rating, title, helpful vote count) and review content. To ensure class parity for model training and evaluation, a balanced dataset was constructed by stratified sampling, selecting 1,000 reviews per sentiment class (positive, neutral, negative) across 34 source files. This process resulted in a balanced dataset of 102,000 records—closely matching the size of the original raw sample—while mitigating sentiment bias and preserving category diversity.

2.1 Sentiment Distribution

The raw dataset exhibits a class imbalance typical of real-world feedback, with ~74% positive, ~18% negative, and ~9% neutral reviews. To prevent class dominance in model learning, a balanced set was constructed by enforcing equal sample sizes across sentiment categories (Appendix A).

2.2 Text Length Distribution

Both raw and balanced datasets reveal a right-skewed distribution, with most reviews under 500 characters and a long tail extending beyond 3,000 characters. This necessitates token truncation, particularly for models constrained by input sequence lengths such as transformers.

2.3 Helpful Votes by Sentiment

Although positive reviews are more prevalent, negative reviews often accrue higher median helpfulness scores. This pattern suggests that critical feedback may be perceived as more informative by other users (Appendix A).

2.4 Keyword Analysis via TF-IDF & Word Clouds

TF-IDF and word cloud visualizations reveal semantic divergence by sentiment. Positive reviews frequently contain affirmatives such as “great,” “love,” and “easy,” whereas negative reviews are marked by terms like “waste,” “broke,” and “disappointed” (Appendix A). These lexical signals validate the viability of sentiment classification via supervised learning.

3. Data Preprocessing

Data preparation proceeded in stages to ensure integrity, balance, and compatibility with machine learning models. We streamed data from the Hugging Face version of the Amazon Reviews 2023 dataset (McAuley-Lab, 2023), selecting all records under the raw_review_* configuration. Reviews were assigned sentiment labels based on star ratings: 1–2 as negative, 3 as neutral, and 4–5 as positive. From each source file, a maximum of 5,000 reviews per class were extracted to ensure both computational feasibility and representativeness.

To construct a modeling-ready dataset, we then cleaned the balanced dataset. Entries lacking either the title, text, or sentiment fields were excluded. Each review was formed by concatenating the title and main review text into a single input string. Sentiments were encoded numerically using a manual mapping dictionary, assigning 0 to ‘negative’, 1 to ‘neutral’, and 2 to ‘positive’. Finally, a unique review_id was generated for each row to maintain alignment across models and facilitate traceability in ensemble evaluation.

We explored several text-cleaning strategies, including HTML tag removal and stopword filtering using NLTK. Although this approach enhanced keyword clarity during exploratory analysis (e.g., for TF-IDF and word clouds), we deliberately excluded stopword filtering from the model pipelines, as it risked removing negators like “not” that are critical to sentiment polarity (Jindal & Liu, 2008). Thus, stopword cleaning was retained for insight generation, not feature learning.

4. Sentiment Analysis

4.1 Model Selection and Justification

To capture the varying complexity and expressive power of sentiment-bearing text, we implemented three complementary models across classical, deep, and transformer-based paradigms, followed by an ensemble voting system for improved generalization.

Logistic Regression served as our classical baseline. It is fast, interpretable, and works effectively with TF-IDF features, making it ideal for benchmarking before transitioning to more sophisticated architectures.

Bi-LSTM with RoBERTa Embeddings was introduced as our deep learning solution. By combining pre-trained transformer embeddings with recurrent sequence modeling, this architecture balances expressiveness and training efficiency while being able to capture sequential dependencies in the review text.

Fine-tuned RoBERTa Transformer represents the state-of-the-art approach via transfer learning. We leveraged the HuggingFace roberta-base checkpoint with a classification head trained on our labeled dataset. The model jointly learns task-specific representations with large-scale language knowledge, achieving strong generalization with minimal tuning.

Each model relied on tokenized input representations: TF-IDF vectorization for the logistic regression model, RoBERTa token embeddings for the Bi-LSTM, and Hugging Face’s AutoTokenizer for the transformer. For the transformer model, we applied a maximum length of 512 tokens with truncation, attention masks, and dynamic padding.

Finally, we implemented a majority voting ensemble across all three models (Sagi & Rokach, 2018). This step aims to capitalize on the strength of each model while smoothing out individual prediction biases.

This layered model design provides a rigorous evaluation of sentiment classification methods while offering insight into tradeoffs between accuracy, interpretability, and scalability.

4.2 Training and Tuning

All models used a consistent, stratified 80/10/10 split across training, validation, and test sets, totaling 102,000 reviews from a manually balanced dataset (3,000 samples per sentiment across 34 source files). This ensured consistent evaluation and prevented data leakage.

TF-IDF for Classical Models: Logistic Regression used a combination of TF-IDF features extracted separately from review titles and texts. Title features were capped at 3,000 dimensions and text features at 5,000.

Feature matrices were horizontally concatenated and fed into scikit-learn's LogisticRegression with max_iter=1000. No deep tokenization or embeddings were used for this model

Bi-LSTM with RoBERTa Tokenization: For the Bi-LSTM, we concatenated titles and texts and tokenized them using the RoBERTa tokenizer with truncation to 64 tokens and padding. Tokenized pairs were passed through a frozen RoBERTa encoder followed by a bidirectional LSTM (hidden size = 128 per direction). A dense layer projected the final hidden states to the sentiment logits. Early stopping based on validation macro F1 was employed, terminating training after 4 epochs.

Transformer Fine-Tuning: We fine-tuned the full Roberta-base transformer model using HuggingFace's Trainer API. Tokenization followed the same process as above, but with a maximum sequence length of 512 tokens. We used a batch size of 16, a learning rate of 1e-5, and a weight decay of 0.05. Training ran for up to 5 epochs with early stopping enabled (patience = 2). Validation was conducted every epoch, and the best model checkpoint was restored for test evaluation.

Ensemble via Majority Voting: Each model generated prediction files with review_id, true_label, and predicted labels. These were merged on review_id, and the final ensemble label was assigned via majority voting among the three models. This approach retained model independence while improving overall robustness.

4.3 Evaluation

Although positive reviews are more prevalent, negative reviews often accrue higher median helpfulness scores. This pattern suggests that critical feedback may be perceived as more informative by other users (Appendix A).

Model performance was assessed using accuracy, macro-averaged F1, and negative-class recall, reflecting both overall performance and ability to detect problematic reviews.

Logistic Regression achieved 74.3% accuracy and 0.742 macro F1, with a negative recall of 0.758. It performed best on the positive class but struggled with neutral sentiment, reflecting the limitations of sparse features in capturing nuanced expression.

Bi-LSTM improved to 77.6% accuracy and 0.775 macro F1, with a slight gain in negative recall (0.773). The recurrent architecture enabled better sequence modeling, especially for longer reviews. However, training was more time-consuming and required tuning the embedding truncation length to avoid GPU memory overflow.

Transformer (Fine-tuned RoBERTa) achieved the highest standalone performance with 81.5% accuracy, 0.816 macro F1, and 0.815 negative recall. RoBERTa's pre-trained contextual embeddings proved highly effective in distinguishing sentiment nuances, particularly improving neutral-class classification (precision = 0.82, recall = 0.81). This model's success confirms the value of transfer learning for language understanding tasks.

Majority Voting Ensemble offered balanced performance across all classes, achieving 80.3% accuracy and 0.800 macro F1. Although it did not outperform RoBERTa in absolute terms, it reduced variance across classes. For instance, neutral-class F1 improved from 0.67 (LSTM) and 0.64 (Logistic) to 0.70 in the ensemble (Appendix C). Negative recall also remained high (0.79), indicating effective identification of dissatisfaction.

The ensemble thus emerges as a stable and interpretable deployment choice, even when individual model performances fluctuate across subsets.

4.4 Insights and Impact

The sentiment classification results highlight several practical and theoretical contributions. Among the models, the fine-tuned RoBERTa transformer consistently achieved the highest performance, demonstrating the advantage of pre-trained language representations in understanding user-generated content. Its strength lies in capturing context and subtle sentiment cues, especially in ambiguous or mixed reviews.

Despite being a simpler model, Logistic Regression delivered reasonable performance and offered advantages in speed and interpretability. This makes it suitable for constrained environments or early-stage prototyping. The Bi-LSTM model, combining deep learning with RoBERTa embeddings, balanced efficiency with improved recall, particularly for underrepresented sentiments.

A key insight is the consistent ability of all models to detect negative sentiment, with negative recall exceeding 0.75. In practical settings, this supports early flagging of user dissatisfaction and helps guide interventions such as customer support or quality control.

The structured prediction format—based on unique review identifiers and standardized text fields—enables easy integration into real-world applications. Examples include automated feedback monitoring, alert systems, or input for recommender engines. While this study did not incorporate explainability tools, the current framework is compatible with SHAP or LIME, which could enhance transparency in future work (Ribeiro et al., 2016).

Together, these findings support the use of sentiment models as a scalable, adaptable tool for consumer insight and review moderation.

5. Clustering Negative Reviews

Table 1. Method Explored

Method	Embedding / Vectorization	Clustering Model	Silhouette Score	Result
TF-IDF + KMeans	TF-IDF	KMeans	~0.016	Basic separation; generic, hard-to-label topics
BERT Embedding + KMeans	Sentence Transformer (MPNet)	KMeans	~0.039	Semantically improved; low interpretability
BERTopic (MPNet, Reduced)	MPNet Contextual Embedding	BERTopic	N/A (qualitative)	Final method; clear themes and useful outputs

5.1 Feature Engineering

To explore recurring themes in customer dissatisfaction, we utilized a dataset of over 14,205 product reviews that had been predicted as negative by the Majority Voting Ensemble sentiment classifier. Each review contained a title and text field, which were concatenated into a single text_combined field after standard preprocessing.

During the feature engineering phase, we experimented with different representations of review content. We applied TF-IDF vectorization separately on title, text, and their combination. Each version was evaluated as input for clustering, with the goal of identifying which encoding best preserved topic-level distinctions. We used standard TF-IDF parameters (min_df=5, max_df=0.9) and applied KMeans to assess clustering quality. The best-performing representation from this approach—TF-IDF on combined title + text—was used as a baseline before exploring embedding-based methods.

5.2 Methods Explored

We evaluated three major clustering strategies to group the transformer-predicted negative product reviews into semantically coherent themes (*Table 1*). These methods were compared in terms of clustering quality, interpretability, and practical usability.

Our baseline approach used TF-IDF vectorization combined with KMeans clustering. We experimented with different text inputs, including text alone, title alone, and a concatenated title + text format. Each representation was vectorized using standard TF-IDF parameters and clustered across a range of values for K (from 2 to 14). The best-performing setup was based on the combined title + text input, which offered marginally better separation than text alone. However, the Silhouette Scores remained low—with a maximum of only ~0.016—indicating weak cluster boundaries. Furthermore, the resulting clusters were often dominated by high-frequency, generic words such as "product," "not," or "use", and failed to capture meaningful semantic themes. Despite attempts to tune vectorization thresholds or apply dimensionality reduction via SVD, the TF-IDF based clusters lacked both depth and interpretability.

To improve semantic representation, we transitioned to embedding-based clustering using Sentence-BERT models, specifically all-MiniLM-L6-v2 and all-mpnet-base-v2 from the SentenceTransformers library. These models encode full sentences into dense, contextual embeddings that better preserve the underlying meaning of review text. KMeans was then applied to these embeddings. Compared to TF-IDF, this approach provided modestly improved cohesion in clustering, as reflected by slightly higher Silhouette Scores (up to ~0.039). However, due to the high dimensionality and lack of inherent explainability in the clusters, it remained difficult to interpret the output. Without a mechanism to extract representative keywords or summarize clusters, this method lacked practical clarity and was not well suited for human labeling or business analysis.

The most effective approach was BERTopic, applied with all-mpnet-base-v2 as the embedding model. BERTopic leverages sentence-level contextual embeddings to group semantically similar reviews, and it automatically generates interpretable topic descriptors by extracting top keywords per cluster. Importantly, we used BERTopic's reduce_topics() function to merge semantically overlapping clusters and constrain the final output to 25 distinct topics. This not only simplified the output but also significantly improved the interpretability, distinctiveness, and coherence of each topic. Unlike earlier methods, BERTopic provided a clear representation of each cluster's thematic focus—for instance, delivery delays, refund issues, product defects, or customer service problems—along with representative sample reviews. As a result, this approach proved highly effective and was selected as our final clustering method.

5.3 Final Choice and Justification

After systematically evaluating all clustering approaches, we ultimately selected BERTopic with MPNet embeddings, combined with topic reduction to 25 themes, as our final and most effective method. This decision was grounded in both qualitative and structural advantages observed during the analysis.

In contrast to KMeans, which suffered from the sparsity of TF-IDF vectors and struggled to form semantically meaningful clusters, BERTopic successfully leveraged contextual sentence embeddings to identify nuanced semantic patterns across the review corpus. These embeddings captured deeper relationships between reviews that shared similar sentiment or subject matter, even when the surface vocabulary differed. As a result, clusters generated by BERTopic exhibited substantially better cohesion, with topics that were not only more compact but also more interpretable.

To further enhance clarity and reduce noise, we applied BERTopic's built-in reduce_topics() function, which merged semantically overlapping clusters based on distance in embedding space. This step yielded a clean and distinct set of 25 topics, eliminating redundancy without losing topical diversity. Each topic was summarized by its top-ranked keywords and linked to a representative review—the most confidently assigned sample—providing a strong anchor for human interpretation and business application.

This final output addressed the core limitations observed in previous methods:

- It resolved the high-dimensional noise and generic term dominance seen in TF-IDF + KMeans;
- It improved interpretability through automated topic labeling;
- It delivered insights that closely aligned with real-world customer pain points, including issues related to product quality, delivery problems, misleading descriptions, and refund experiences.

Moreover, the structure of BERTopic's output made it well suited for downstream applications such as manual labeling, issue tracking, customer support optimization, and even LLM prompt construction. By combining semantic depth with practical interpretability, BERTopic ultimately provided the most valuable foundation for clustering negative product reviews in our analysis.

6. LLM Interpretation & Recommendation

6.1 LLM Model Selection

We selected Gemini 2.0 Flash from Google's Generative AI suite to generate tailored business insights from clustered negative reviews, based on its strong alignment with our operational needs. First, it offers exceptional speed and cost-efficiency, making it well-suited for scalable, production-level deployments. Second, the model excels at prompt-driven tasks such as summarization, root cause analysis, and recommendation generation, consistently producing structured and business-relevant outputs that require minimal post-processing. Most importantly, Gemini Flash demonstrates a high degree of consistency with minimal hallucination, ensuring that the insights derived from customer feedback remain factually grounded and actionable. By choosing Gemini, we achieved an effective balance between latency, output quality, reliability, and cost—key pillars for a robust and scalable customer insights platform.

6.2 Prompt Engineering

Our system sends a representative sample (20 reviews per cluster) to Gemini along with structured instructions. The prompt is started with concise instruction, and changes iteratively based on the result. Although we thought about giving a few shot examples, as the identified cluster might be different every time, we decided to not give examples and keep the instruction simple and concise.

Final prompt is as below:

""" You are an experienced customer success expert. Based on the information provided below, do the following: 1. Provide a concise summary of what the negative feedback is mainly about. 2. For each of the categories provided, provide potential root cause and short recommendation to address the issue. Be clear and concise. input: {input_json} """ .

6.3 Performance & Evaluation

The performance of Gemini 2.0 Flash was manually evaluated by comparing its outputs against the original review clusters. The evaluation focused on three key aspects: relevance, actionability, and consistency. Specifically, we checked whether the generated summaries accurately reflected the main issues within each cluster, whether the recommendations were practical and could inform real business decisions, and whether the responses were coherent and non-repetitive across clusters.

After conducting ten rounds of testing, we observed that the model consistently identified core pain points—such as missing parts, product defects, or size mismatches—and translated them into actionable suggestions like quality control improvements or packaging enhancements.

However, we also noted some limitations. In a few cases, the output included vague or overly generalized statements, and the overall quality of recommendations proved highly dependent on the diversity and richness of the input reviews.

Despite these drawbacks, Gemini 2.0 Flash remains a robust, fast, and cost-effective solution for LLM-powered review analysis. It adds a layer of strategic, human-like reasoning that enhances our automated sentiment and clustering pipeline, making customer feedback more interpretable and actionable for businesses.

7. Result and Discussion

7.1 Sentiment Classification Performance

The first stage of the pipeline establishes a robust foundation for downstream tasks through accurate and reliable sentiment classification. Among the three evaluated models, the fine-tuned RoBERTa transformer achieved the strongest results, attaining 81.5% accuracy and a macro-F1 score of 0.816 on the test set. These results indicate the superior ability of pre-trained contextual embeddings to capture subtle variations in sentiment expression, especially in ambiguous or mixed-polarity reviews. In comparison, the Bi-LSTM model achieved 77.6% accuracy and the TF-IDF-based Logistic Regression baseline reached 74.3%, confirming the performance gap between shallow models and modern transformer-based approaches.

Importantly, all models maintained negative-class recall above 0.75, indicating consistent detection of user dissatisfaction—a vital feature for risk mitigation in customer experience workflows. The majority-vote ensemble further improved model robustness by boosting neutral-class F1 to 0.70, while maintaining overall accuracy at 80.3%, thus balancing class-specific biases. This ensembling strategy proves particularly useful in production scenarios where sentiment distribution is non-stationary or data drift is expected.

7.2 Sentiment Classification Performance

In the second stage, we focused on structuring customer dissatisfaction via semantic clustering of 14,205 reviews predicted as negative. A conventional TF-IDF + K-Means baseline yielded weak topical separation, with low silhouette scores (~0.04) and vague term-frequency-driven clusters that lacked actionable specificity.

In contrast, our adopted BERTopic pipeline with MPNet embeddings and topic reduction uncovered 25 distinct, interpretable themes, including issues like logistics delays, missing components, and premature breakage. These clusters aligned well with common business pain points and were validated via manual inspection, revealing greater intra-cluster homogeneity and thematic clarity than traditional clustering approaches. The use of contextual embeddings played a key role in capturing semantic nuance, while the reduce_topics() function improved interpretability by merging redundant subclusters into cohesive categories.

This phase bridges the gap between unstructured sentiment and operational themes, enabling product managers and CX teams to move from "what's wrong?" to "what kind of issues are most frequent?"

7.3 LLM-Based Summary and Recommendation Generation

The third and final stage of our pipeline bridges data analytics with actionability. We employed Gemini 2.0 Flash, a fast and cost-effective generative LLM, to synthesize each topic cluster into a concise summary and generate actionable recommendations. For example, clusters on "missing components" were paired with suggestions like stricter outbound quality control, while clusters on "fit issues" led to pre-purchase sizing guidance.

Evaluation across ten prompt iterations revealed high consistency in generating root-cause summaries and targeted interventions. The generated outputs were not only coherent and domain-relevant but also concise enough to be consumed by product or operations teams without further curation. While the process still involves human validation, Gemini significantly reduced turnaround time and bridged the final step from detection to decision.

7.4 Operational Implications

Collectively, these results confirm that our pipeline fulfills the primary technical goal ($\geqslant$ 80% accuracy in classification) while also delivering on business needs. The ensemble classifier can be seamlessly integrated into real-time feedback monitoring dashboards, offering early alerts for shifts in customer sentiment. Meanwhile, the 25-topic map serves as a prioritized insight structure that enables issue triaging, resource allocation, and strategic intervention planning.

The pipeline not only quantifies sentiment but diagnoses its root causes and suggests responses, offering organizations a strategic advantage in customer experience optimization. Its modular architecture ensures adaptability across domains, while its empirical validation supports deployment confidence in operational settings.

8. Limitations and Future Improvements

8.1 Model Accuracy & Data Sufficiency

While our fine-tuned RoBERTa classifier exceeded the 80% accuracy threshold, its macro-F1 (0.816) and negative-class recall (0.815) fall slightly below the ideal benchmarks of 0.85–0.90 often cited in high-stakes industrial NLP applications. A key limitation stems from the inherent complexity of neutral reviews, which often exhibit subtle sentiment signals, sarcasm, or mixed opinions. These are harder to disambiguate even with advanced transformer architectures.

To address this, future work should expand the dataset with more diverse and ambiguous edge-case samples, especially targeting underrepresented classes. Manual annotation of borderline cases or semi-supervised bootstrapping using weak labels could increase both coverage and label consistency. From a modeling perspective, larger backbone models like DeBERTa-v3-large or LLaMA-3 could be explored to capture deeper linguistic nuance. Leveraging gradient accumulation and mixed-precision training would allow such architectures to be trained efficiently on commodity GPUs. Additionally, synthetic data augmentation via back-translation, adversarial rephrasing, or LLM-generated paraphrases could increase robustness by exposing models to wider sentiment expression patterns..

8.2 Topic-Model Robustness

Although BERTopic successfully surfaced 25 coherent themes from over 14,205 negative reviews, its unsupervised nature presents limitations in topic purity and generalizability. Some low-frequency complaints were grouped into overly generic "catch-all" clusters, which hinders downstream interpretation. This is particularly problematic in business contexts where prioritization depends on theme specificity.

We propose the incorporation of cluster quality metrics such as normalized mutual information (NMI), topic coherence, and stability indices to quantitatively evaluate cluster validity. Moreover, integrating human-in-the-loop workflows—where analysts iteratively relabel or refine topics—can bridge the gap between algorithmic output and business semantics. Alternatively, semi-supervised models such as Guided LDA or ZeroShotTopicModeling could be explored to anchor clusters around predefined operational categories (e.g., delivery delays, refund issues, safety complaints).

8.3 LLM Evaluation & Cost Management

Gemini Flash showed strong performance in generating meaningful summaries and actionable suggestions. However, current evaluations are qualitative and subjective, based on 10 rounds of manual inspection using informal criteria. To ensure reproducibility and statistical rigor, future work should design a lightweight annotation rubric, involve dual annotators, and report inter-rater agreement via Cohen's Kappa.

In terms of cost-efficiency, reliance on a proprietary platform (e.g., Gemini Flash) raises concerns of vendor lock-in and scalability bottlenecks. We recommend benchmarking open-source alternatives such as Mistral-7B-Instruct, LLaMA 3, or Mixtral, and experimenting with parameter-efficient tuning methods like LoRA (Low-Rank Adaptation). These techniques can enable custom fine-tuning for domain-specific feedback (e.g., medical devices, software reviews) at a fraction of the computational cost.

8.4 Domain & Language Generalisability

All experiments use English-language Amazon data; performance may erode on multilingual or domain-specific corpora such as hospitality or fintech reviews. Cross-lingual sentence embeddings and domain-adaptive pre-training—combined with language-aware sentiment lexicons—are essential for broad deployment. A staged rollout with shadow-mode monitoring can quantify drift before full production release.

8.5 Explainability & Ethical Compliance

Lastly, the current system lacks model interpretability mechanisms, which poses risks in decision transparency and regulatory compliance. As NLP models grow in complexity, understanding why a prediction was made becomes crucial—especially in high-impact scenarios like loan approval or medical diagnosis.

We propose the integration of token-level SHAP or LIME visualizations to interpret RoBERTa ensemble decisions and reveal feature attributions. Similarly, representative review exemplars should accompany each discovered topic to aid human interpretability. From a governance standpoint, the pipeline must embed PII redaction, respect user consent, and align with evolving AI regulations such as the EU AI Act or Singapore's PDPA. Fairness audits across demographic subgroups and bias mitigation pipelines should also be implemented.

In summary, by addressing these five areas—data diversity, topic interpretability, LLM cost evaluation, domain adaptability, and explainability—the current pipeline can evolve into a production-grade, responsible NLP system capable of delivering continuous improvements to customer experience across industries and languages.

9. Conclusion

This study introduces a robust, end-to-end natural language processing (NLP) framework designed to convert unstructured customer feedback into actionable business intelligence. The system adopts a three-stage pipeline—sentiment classification, topic clustering, and large language model (LLM)-based synthesis—that collectively transforms raw textual reviews into strategic insights. Leveraging a fine-tuned RoBERTa model, our pipeline achieves high-accuracy sentiment analysis (81.5%) while maintaining strong recall for negative sentiment—an essential attribute for early identification of customer dissatisfaction. Through BERTopic's contextual clustering, the framework uncovers 25 coherent and business-relevant themes from over 15,000 negative reviews, enabling fine-grained categorization of common customer complaints. Finally, by deploying Gemini 2.0 Flash, the system translates clusters into cluster-specific summaries and tailored recommendations, thereby closing the loop from detection to action.

Beyond individual component success, our results highlight the synergistic advantage of combining classical, deep learning, and generative AI models in a unified architecture. Contextual embeddings substantially outperformed traditional feature-based methods, and ensemble classification models delivered balanced performance across sentiment categories—essential for real-world deployment where sentiment distribution is often imbalanced or ambiguous. LLM integration introduced a novel layer of interpretability and decision relevance, bridging data science with operations. This human-like layer of synthesis transforms descriptive analytics into prescriptive insights, enabling downstream teams to act swiftly and confidently on user feedback.

While the framework is not without limitations—including challenges in cross-domain generalization, evaluation standardization, and explainability—it remains highly modular and extensible. The open-sourced codebase supports transparency and reproducibility, and its plug-and-play design allows for adaptation to different industries, languages, and business objectives. As such, this work serves as a foundational blueprint for organizations seeking to operationalize customer feedback analysis at scale. By automating the extraction of pain points and generating prioritized actions, our approach contributes to improving product quality, elevating service standards, and ultimately enhancing customer satisfaction in today's competitive landscape.

References

Jindal, N., & Liu, B. (2008). Opinion spam and analysis. Proceedings of the 2008 International Conference on Web Search and Data Mining, 219–230.

McAuley Lab. (2023). Amazon Reviews 2023. Hugging Face Datasets. https://huggingface.co/datasets/McAuley-Lab/Amazon-Reviews-2023

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,1135–1144.

Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1249.

Appendix

Figure A1. Sentiment Distribution (Raw Sample)

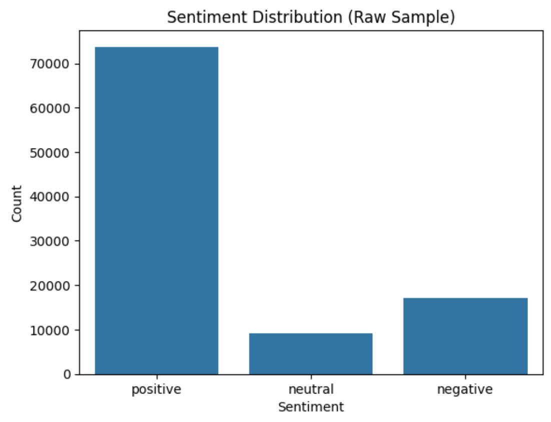


Figure A2. Sentiment Distribution in Balanced Dataset

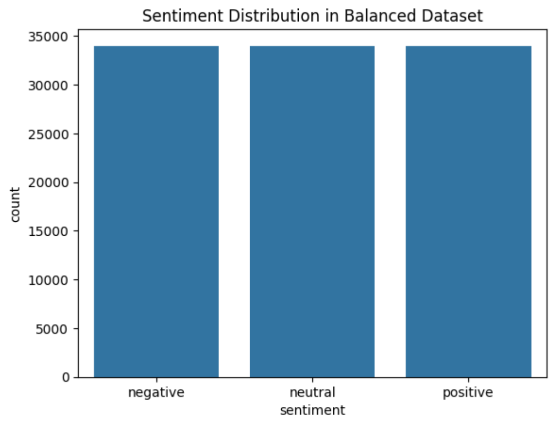


Figure A3. Sentiment Distribution: Raw Sample vs. Balanced Dataset

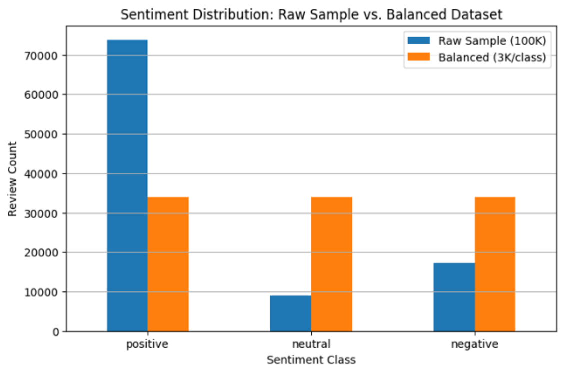


Figure A4. Text Length Distribution (Raw Sample)

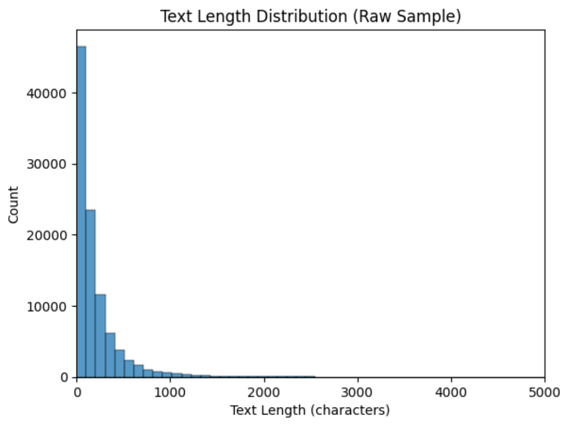


Figure A5. Text Length Distribution (Balanced Dataset)

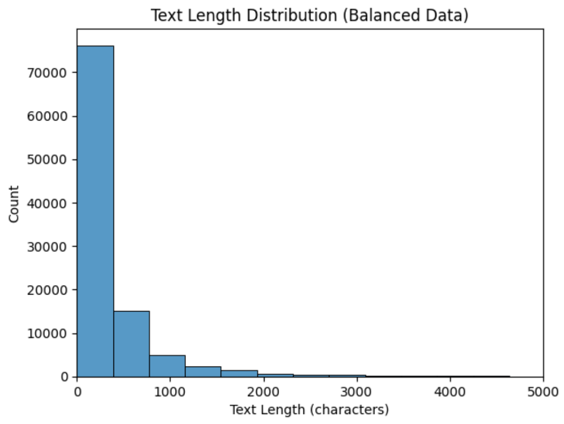


Figure A6. Helpful Votes Distribution by Sentiment (Log Scale, Raw Sample)

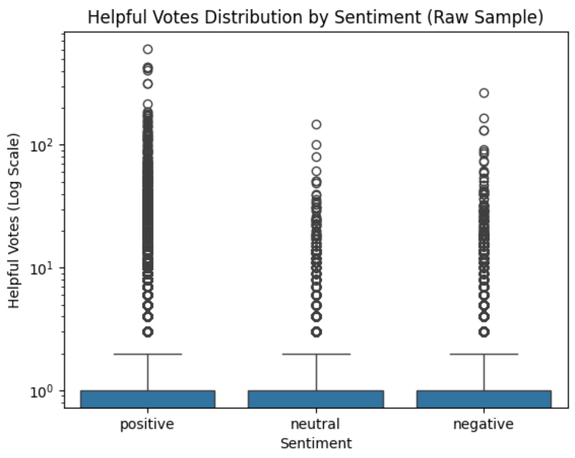


Figure A7. Sentiment Keyword Word Clouds (TF-IDF Top Terms for Positive vs. Negative Reviews

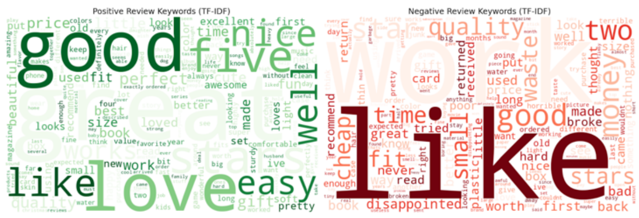
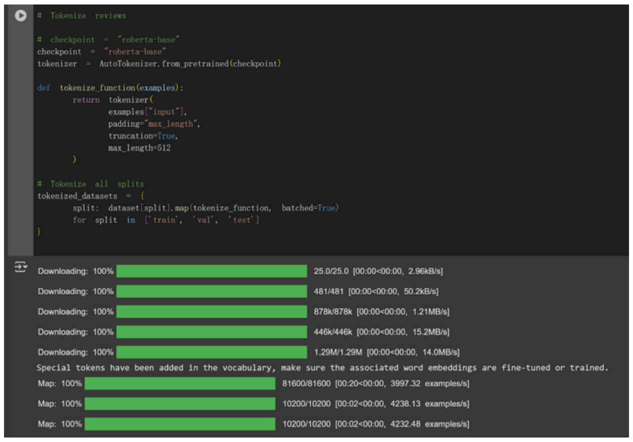


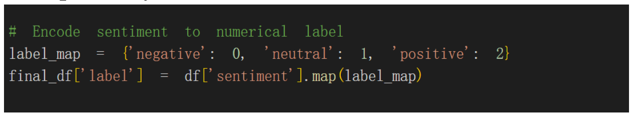
Figure B1. Tokenizer implementation using the Hugging Face AutoTokenizer with truncation to 512 tokens and padding to maximum sequence length across all train/val/test splits.



```
# Tokenize reviews

# checkpoint = "roberta-base"
checkpoint = "roberta-base"
tokenizer = AutoTokenizer.from_pretrained(checkpoint)

def tokenize_function(examples):
        return tokenizer(
                examples["input"],
                padding="max_length",
                truncation=True,
                max_length=512
        )

# Tokenize all splits
tokenized_datasets = {
        split: dataset[split].map(tokenize_function, batched=True)
        for split in ['train', 'val', 'test']
}
```

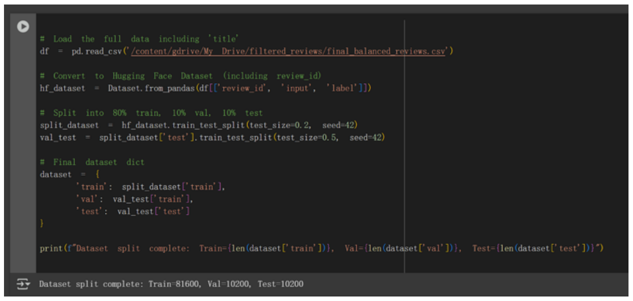
```
Downloading: 100%  25.0/25.0 [00:00<00:00, 2.96kB/s]
Downloading: 100%  481/481 [00:00<00:00, 50.2kB/s]
Downloading: 100%  878k/878k [00:00<00:00, 1.21MB/s]
Downloading: 100%  446k/446k [00:00<00:00, 15.2MB/s]
Downloading: 100%  1.29M/1.29M [00:00<00:00, 14.0MB/s]
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-tuned or trained.
Map: 100%  81600/81600 [00:20<00:00, 3997.32 examples/s]
Map: 100%  10200/10200 [00:02<00:00, 4238.13 examples/s]
Map: 100%  10200/10200 [00:02<00:00, 4232.48 examples/s]
```

Figure B2. Sentiment Label Encoding Mapping for Model Compatibility



```
# Encode sentiment to numerical label
label_map = {'negative': 0, 'neutral': 1, 'positive': 2}
final_df['label'] = df['sentiment'].map(label_map)
```

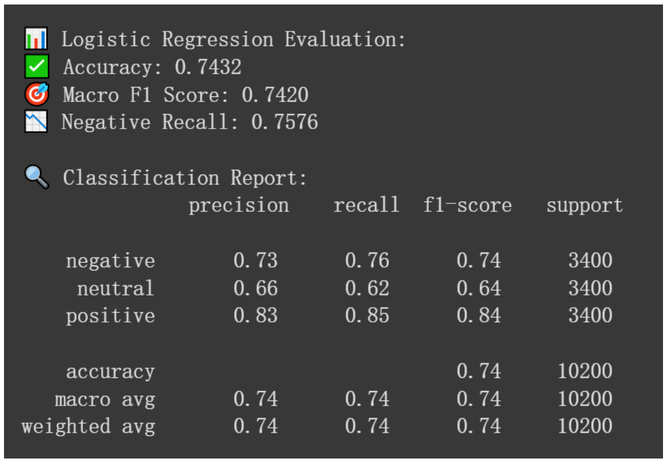
Figure B3. Dataset Split into Train/Validation/Test using Hugging Face Dataset API



```
# Load the full data including 'title'
df = pd.read_csv('/content/gdrive/My Drive/filtered_reviews/final_balanced_reviews.csv')

# Convert to Hugging Face Dataset (including review_id)
hf_dataset = Dataset.from_pandas(df[['review_id', 'input', 'label']])

# Split into 80% train, 10% val, 10% test
split_dataset = hf_dataset.train_test_split(test_size=0.2, seed=42)
val_test = split_dataset['test'].train_test_split(test_size=0.5, seed=42)

# Final dataset dict
dataset = {
        'train': split_dataset['train'],
        'val': val_test['train'],
        'test': val_test['test']
}

print(f"Dataset split complete: Train={len(dataset['train'])}, Val={len(dataset['val'])}, Test={len(dataset['test'])}")
```

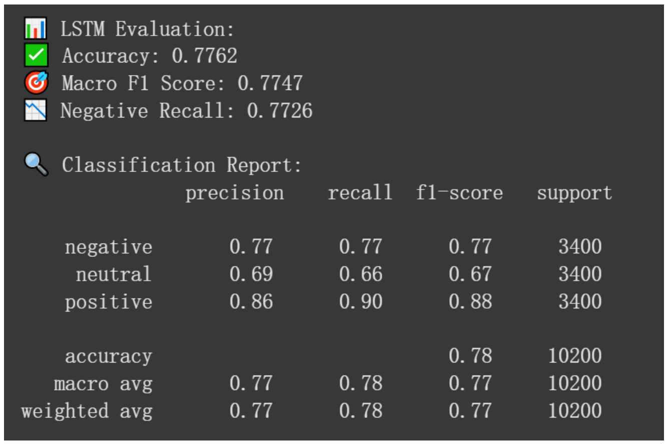
```
Dataset split complete: Train=81600, Val=10200, Test=10200
```

Figure C1. Logistic Regression Test Results



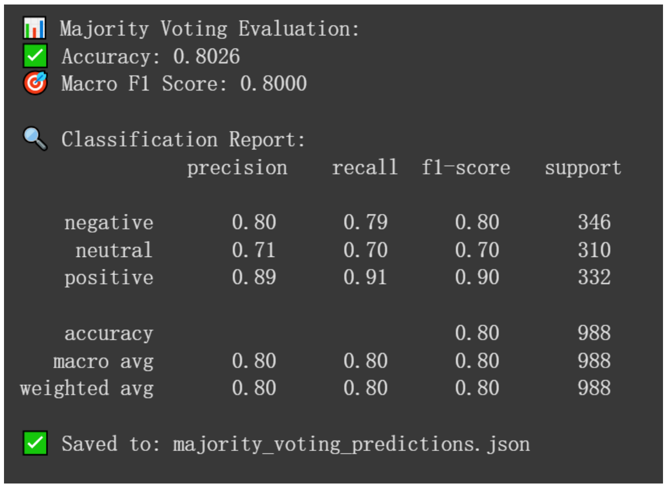
```
📊 Logistic Regression Evaluation:
✅ Accuracy: 0.7432
🎯 Macro F1 Score: 0.7420
📉 Negative Recall: 0.7576

🔍 Classification Report:
              precision    recall  f1-score   support

    negative       0.73      0.76      0.74      3400
     neutral       0.66      0.62      0.64      3400
    positive       0.83      0.85      0.84      3400

    accuracy                           0.74     10200
   macro avg       0.74      0.74      0.74     10200
weighted avg       0.74      0.74      0.74     10200
```

Figure C2. Bi-LSTM Test Results



```
📊 LSTM Evaluation:
✅ Accuracy: 0.7762
🎯 Macro F1 Score: 0.7747
📉 Negative Recall: 0.7726

🔍 Classification Report:
              precision    recall  f1-score   support

    negative       0.77      0.77      0.77      3400
     neutral       0.69      0.66      0.67      3400
    positive       0.86      0.90      0.88      3400

    accuracy                           0.78     10200
   macro avg       0.77      0.78      0.77     10200
weighted avg       0.77      0.78      0.77     10200
```

Figure C3. Majority Voting Ensemble Test Results



```
📊 Majority Voting Evaluation:
✅ Accuracy: 0.8026
🎯 Macro F1 Score: 0.8000

🔍 Classification Report:
              precision    recall  f1-score   support

    negative       0.80      0.79      0.80       346
     neutral       0.71      0.70      0.70       310
    positive       0.89      0.91      0.90       332

    accuracy                           0.80       988
   macro avg       0.80      0.80      0.80       988
weighted avg       0.80      0.80      0.80       988

✅ Saved to: majority_voting_predictions.json
```

Figure D1. Elbow Method and Silhouette Score (TF-IDF + Title + Text)

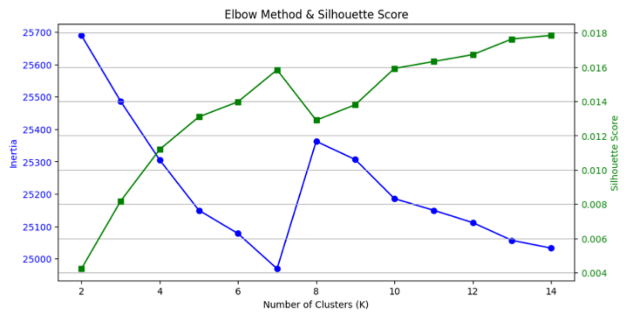


Figure D2. Elbow Method and Silhouette Score using Sentence-BERT (MPNet)

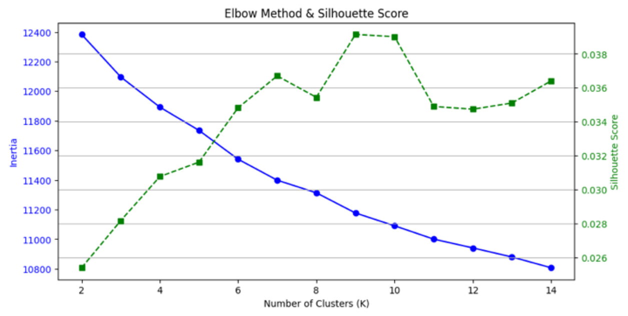
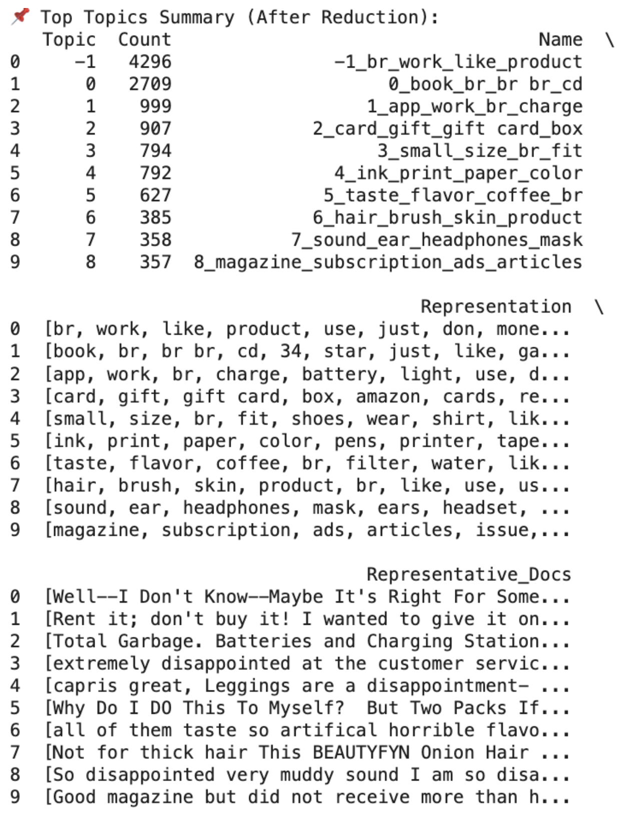


Figure D3. Top 10 Topics Extracted by BERTopic (After Reduction)



```
📌 Top Topics Summary (After Reduction):
   Topic  Count                                  Name  \
0     -1   4296              -1_br_work_like_product
1      0   2709                    0_book_br_br br_cd
2      1    999                  1_app_work_br_charge
3      2    907             2_card_gift_gift card_box
4      3    794                    3_small_size_br_fit
5      4    792               4_ink_print_paper_color
6      5    627              5_taste_flavor_coffee_br
7      6    385             6_hair_brush_skin_product
8      7    358           7_sound_ear_headphones_mask
9      8    357  8_magazine_subscription_ads_articles

                                     Representation  \
0  [br, work, like, product, use, just, don, mone...
1  [book, br, br br, cd, 34, star, just, like, ga...
2  [app, work, br, charge, battery, light, use, d...
3  [card, gift, gift card, box, amazon, cards, re...
4  [small, size, br, fit, shoes, wear, shirt, lik...
5  [ink, print, paper, color, pens, printer, tape...
6  [taste, flavor, coffee, br, filter, water, lik...
7  [hair, brush, skin, product, br, like, use, us...
8  [sound, ear, headphones, mask, ears, headset, ...
9  [magazine, subscription, ads, articles, issue,...

                                Representative_Docs
0  [Well--I Don't Know--Maybe It's Right For Some...
1  [Rent it; don't buy it! I wanted to give it on...
2  [Total Garbage. Batteries and Charging Station...
3  [extremely disappointed at the customer servic...
4  [capris great, Leggings are a disappointment- ...
5  [Why Do I DO This To Myself?  But Two Packs If...
6  [all of them taste so artifical horrible flavo...
7  [Not for thick hair This BEAUTYFYN Onion Hair ...
8  [So disappointed very muddy sound I am so disa...
9  [Good magazine but did not receive more than h...
```