Sentiment Insight Engine for Amazon Reviews

Huang Kaixin (A0295969X)
Li Muyang (A0296021L)
Sun Hetong (A0297150A)
Yan Guoxinyi (A0280447B)
Yu Wei (A0162552J)
Group 8(Github Link), NUS Business School

I. Introduction

In the current digital retail environment, customer feedback holds immense strategic value [1]. Platforms like Amazon generate a constant stream of product reviews, which—if analyzed properly—can provide real-time signals about product satisfaction, delivery efficiency, usability, and more. However, despite the potential of this data, most businesses still rely on outdated mechanisms to process it. Manual review tagging, keyword filters, and rule-based monitoring are common practices, yet they are inherently limited. These methods are not only labor-intensive and slow, but they also fail to capture nuanced emotions, latent patterns, or newly emerging concerns [2].

This gap between data availability and actionable insight inspired the core objective of our project. We aim to build an automated pipeline that processes large volumes of customer reviews using natural language processing (NLP) and machine learning models. Specifically, we employ Support Vector Machine (SVM) [3] for sentiment analysis and BART [4]-based classification for review understanding. These models enable us to move beyond surface-level statistics, offering a deeper, contextual reading of customer opinions.

Our dataset consists of over 4,000 Amazon product reviews across different product categories. Each entry contains the full review text, overall rating, and helpfulness metrics. By training our models on Twitter sentiment data and applying them to the Amazon context, we demonstrate cross-domain adaptability [5], allowing businesses to react swiftly to feedback without relying on manual labor. Ultimately, our solution provides a scalable way for platforms to detect dissatisfaction early, enhance recommendation systems with real sentiment data, and make more responsive business decisions.

II. EDA

All 4,915 reviews include valid ratings, timestamps, vote counts and word-counts, so there is no missing data 1. The mean star rating is 4.59 ($\sigma = 0.996$), with both the median and upper quartile at 5, confirming a heavy positive bias in user feedback. In contrast, review lengths average 50.4 words ($\sigma = 59.1$), ranging from one-word comments to detailed narratives exceeding 1,500 words, which points to a long-tail phenomenon in text verbosity [6].

A closer look at the rating histogram shows that five-star reviews make up 80% of the sample, four-star roughly 11%, and one- to three-star the remaining 9%, underscoring a pronounced class imbalance that must be addressed in any predictive modeling effort 2 [14]. Word-count frequencies reveal that over half of the reviews are extremely brief (below 30 words), about 30% occupy the 30–60 word band, and counts drop off sharply beyond 100 words, although a handful of outliers stretch the distribution's right tail 3. Temporally, the corpus is dominated by entries from 2013 (55%) and 2014 (35%), with minimal contributions before or after that period, suggesting that trend analyses should focus on this two-year window 4.

The word cloud surfaces "card," "phone," "SD" and "SanDisk" as the most salient tokens, reflecting the product's core use-case, while performance-related terms like "fast" and "storage," value indicators such as "price," and defect markers like "problem" highlight the primary drivers of user sentiment 4. Finally, plotting review length against star rating reveals no clear linear relationship—both succinct and lengthy entries span the full rating spectrum—though the longest reviews occur at the sentiment extremes, meriting deeper qualitative inspection 6.

III. Data Preprocessing

Given the unlabelled nature of the Amazon product review dataset, we leveraged a labeled Twitter sentiment dataset to train our sentiment classification models. To ensure consistent input formats and semantic representation between the two domains, we applied an identical preprocessing pipeline to both datasets. This step is crucial in enabling effective transfer learning across domains [5].

A. Twitter Dataset Preprocessing

To prepare the Twitter dataset for model training, we applied a structured set of natural language processing (NLP) steps focused on cleaning, standardizing, and embedding the text data:

- **Text Cleaning:** Hashtags were removed, and user handles (e.g., `@username`) were replaced with generic placeholders. URLs were normalized to ensure consistency across tweets.
- **Named Entity Removal:** Using spaCy's `en_core_web_sm` model, named entities such as person names, organizations, and locations were eliminated to reduce dataset-specific bias [7].

- **Tokenization & POS Tagging:** Cleaned texts were tokenized, and part-of-speech (POS) tags were assigned to each token to assist in accurate lemmatization [6].
- **Lemmatization & Token Filtering:** Lemmatization was applied based on POS tags. Digits, punctuation, and stopwords were removed to retain the most semantically meaningful components of each text.
- **Word Embedding:** The processed tokens were converted into vector representations using the pre-trained `word2vec-google-news-300` model [8], resulting in dense embeddings suitable for input to machine learning models.

B. Amazon Dataset Preprocessing

As the Amazon dataset serves as the target domain for inference, we ensured preprocessing consistency by applying the exact same pipeline used on the Twitter dataset. Each Amazon review was cleaned, tokenized, POS-tagged, and lemmatized using the same configurations. The resulting text data was then vectorized using the same Word2Vec embedding model. This uniform treatment is key to preserving the learned semantic boundaries from the Twitter domain and enabling robust sentiment prediction on the Amazon dataset.

By aligning the preprocessing across both datasets, we established a reliable foundation for cross-domain generalization—allowing the SVM and BART-based models trained on social media data to effectively interpret customer sentiment in product reviews.

IV. Methodology

A. Classification

To conduct a meaningful analysis of customer reviews, it is essential to categorize the review texts accurately. Categorization enables us to uncover key themes and recurring issues in the texts, providing a solid foundation for deeper insights.

Traditionally, clustering is a common approach used for labeling unsupervised data. Through clustering algorithms, similar data points are grouped into clusters based on feature similarity. Analysts can then assign general labels to each cluster by interpreting their common characteristics. However, this method presents notable limitations [9]. First, there is no guarantee that the clusters will align with the intended classification criteria—such as specific customer concerns or product issues. Second, summarizing diverse texts within a cluster into a single representative label is often subjective and imprecise.

To overcome these limitations, we adopt a more advanced approach using LLM, specifically, Zero-Shot Classification [10]. This method allows us to classify texts without needing labeled training data. Instead, it relies on LLM's ability to generalize from natural language prompts and assign input texts to pre-defined categories. This is especially well-suited for our context, where pre-labeled data is unavailable, and the classification is centered on user experience, such as customer concerns or product features.

We implemented Zero-Shot Classification using three state-of-the-art pre-trained models: **BART-large-MNLI** [11], **RoBERTa-large-MNLI** [12], and **DeBERTa-large-MNLI** [13]. To evaluate the reliability of these models in our classification task, we manually labeled a set of 40 review texts across five common e-commerce categories: *product quality*, *functionality*, *pricing*, *customer service*, and *shipping*. Each review was assigned one or more of these categories based on the issues reflected in the text.

We then applied each model to the same set of reviews, asking them to predict the top two most likely categories. The model predictions were compared with the manual labels to compute an accuracy score. Our evaluation showed that BART-large-MNLI and RoBERTa-large-MNLI performed significantly better, each achieving 34.5 correct matches out of 40. In contrast, DeBERTa-large-MNLI lagged behind, with only 27.5 correct predictions. Considering both accuracy and computational efficiency, BART-large-MNLI was selected as the preferred model for large-scale application.

Based on the initial classification, we further refined our analysis by introducing subcategories within the *functionality* class. The product under study is a memory card, and functional concerns typically involve four key attributes: *storage capacity*, *read/write speed*, *device compatibility*, and *durability*. We designed a second Zero-Shot Classification task to assign these more detailed functional labels to reviews identified as related to functionality.

In this stage, we did not conduct a separate manual evaluation due to time constraints and the previous validation of the model's performance. Instead, we reused the BART-large-MNLI model for its demonstrated effectiveness, reducing the need for additional manual labeling efforts.

This two-phased classification approach, starting with broader issue categories, followed by more detailed segmentation, allows us to generate actionable insights from unstructured review texts in a scalable and cost-effective way.

B. Sentiment Analysis

In our Amazon product review experiment, one of the main challenges was that the raw data did not contain any explicit sentiment labels. To circumvent this limitation, we used a transfer learning approach where we first trained a number of different sentiment analysis models on labeled Twitter data and then utilized the optimal model to predict sentiment labels on our Amazon data and generate sentiment labels.

1) Training Data and Approach: The sentiment analysis pipeline was trained on a labeled Twitter dataset that had three sentiment classes: negative (0), neutral (1), and positive (2). This training regime allowed us to pick up on subtle emotional expressions inherent in short-form social media texts before transferring this knowledge to e-commerce reviews, which have a tendency to convey similar linguistic patterns despite being from different domains.

We tried five various classification models to see which would be the most appropriate for this sentiment classification task:

1) Support Vector Machine (SVM) [3] with optimized parameters
2) Random Forest with reduced complexity
3) XGBoost classifier
4) Simple Neural Network implemented in PyTorch
5) Deep Neural Network with batch normalization and dropout

Each model was trained with the appropriate hyperparameter conditions and then extensively tested on a test set from the same Twitter corpus in order to analyze transferability.

2) Model Performance Comparison: Our evaluation metrics focused on classification accuracy, precision, recall, and F1-scores [15] across the three sentiment classes. Additionally, we considered training time as a secondary efficiency metric. Table I summarizes the performance comparison of all implemented models.

TABLE I: Performance Comparison of Sentiment Analysis Models

Model	Accuracy	Macro Avg F1	Weighted Avg F1	Training Time (s)
SVM	0.6607	0.66	0.66	130.72
Random Forest	0.6179	0.62	0.62	11.81
XGBoost	0.6388	0.64	0.64	2.73
Neural Network	0.6536	0.66	0.65	4.81
Deep Neural Network	0.6570	0.66	0.66	1.27

SVM model exhibited the highest overall accuracy of 66.07%, closely followed by Deep Neural Network (65.70%) and Neural Network (65.36%) implementations. The worst performer was Random Forest with an accuracy of 61.79%, with XGBoost performing moderately at 63.88%.

3) Performance Analysis by Sentiment Class: A closer examination of the class-specific metrics revealed important patterns in model performance:

a) SVM Performance:

- Strong precision (0.75) for positive sentiment classification
- High recall (0.73) for neutral sentiment detection
- Balanced performance across classes with F1-scores ranging from 0.63–0.70

b) Deep Neural Network Performance:

- More consistent recall values across all three classes (0.66, 0.61, 0.70)
- Strong positive sentiment classification (F1-score: 0.71)
- Comparable weighted average F1-score (0.66) to SVM

The confusion matrices indicated that the SVM model was best at separating out the extreme sentiment polarities (positive vs. negative), whereas the neural network methods were more balanced in their performance across all three classes.

4) Model Selection Rationale: Even though we achieved the same accuracy with SVM as with the neural network models, we chose the SVM classifier as our final model due to the following reasons:

1) **Robustness to Overfitting**: SVM's margin-based approach makes it generalize better when transferring to the Amazon domain, which has different linguistic patterns than the Twitter training data.
2) **Interpretability**: In contrast to neural networks that act as "black boxes," SVM is easier to interpret for the importance of features, giving insights into what linguistic features are influential in sentiment classification.
3) **Class Separation**: The SVM performed better at discriminating between the positive and negative sentiments (precision of 0.75 and 0.69 respectively), which was also important for our further analysis of product reception.
4) **Consistent Performance**: The SVM showed well-balanced precision-recall tradeoffs across all three sentiment classes, with macro and weighted F1-score averages of 0.66.

In order to balance bias and variance, the radial basis function (RBF) kernel was optimized with optimized `C` and `gamma` values. This was done through a reduced parameter grid search to optimize the SVM hyperparameters.

These drawbacks draw attention to the difficulties that come with cross-domain sentiment analysis and point out areas that could use further development.

5) Application to Amazon Dataset: After choosing a model, we used the best SVM classifier to analyze our dataset of Amazon reviews, producing sentiment labels that we then integrated into our larger analysis framework. We were able to measure sentiment distributions across product categories, monitor sentiment trends over time, and correlate sentiment with other review metrics like helpfulness votes and ratings thanks to this method.

We successfully bridged the gap between labeled Twitter data and unlabeled Amazon reviews by utilizing this transfer learning technique with the SVM classifier, allowing for a more thorough comprehension of customer sentiment patterns in e-commerce contexts.

V. Business Insights

In this analysis, by combined classification results (e.g., topics or aspects identified by BART) with the predicted sentiment scores, we enable detailed monitoring of each issue's sentiment over time, providing a nuanced view of customer feedback. This section outlines key insights derived from the data and proposes next steps to enhance product strategy and customer satisfaction.

A. Key Insights

1) **Aspect-Based Sentiment Analysis**: The integration of classification (stored in `function_details`) with sentiment scores (`predicted_sentiment`) allows us to assess customer opinions on specific product aspects or issues. For example, if `function_details` identifies topics like "space" or "speed" we can evaluate their associated sentiment scores to determine which features are strengths or weaknesses. The mean sentiment of 1.233(on a 0-2 scale) suggests a generally neutral-to-positive reception, but granularity by aspect reveals targeted areas for improvement7 .
2) **Sentiment Trends Over Time**: The addition of a `year_month` column, derived from `reviewTime`,

facilitates temporal analysis of sentiment. With reviews spanning from January 2012 to December 2014 (mean `day_diff` of 437 days)7, we can track sentiment trends for each classified aspect. A declining sentiment for a specific feature over time could signal emerging issues, such as product wear or unmet expectations post-update.8

3) **Prioritization of Customer Issues**: By combining the frequency of aspect mentions (via classification counts) with their sentiment scores, we can prioritize issues that warrant immediate attention. Aspects frequently cited with negative sentiment (e.g., `predicted_sentiment` = 0) indicate critical pain points, while those with high positive sentiment highlight competitive advantages.
4) **Leveraging Qualitative Feedback**: Reviews with extreme sentiments—negative (0) or positive (2)—offer rich qualitative insights. Negative reviews, comprising a smaller portion given the mean sentiment of 1.233 7, may pinpoint specific flaws (e.g., "poor durability"), while positive reviews (75% quartile at 2) can reveal appreciated features11.
5) **Validation Through Ratings**: The dataset's `overall` column, with a mean rating of 4.587 out of 5, aligns with the positive-leaning sentiment (75% of `predicted_sentiment` values at 1 or 2). Correlating sentiment with star ratings validates the sentiment model's accuracy, enhancing confidence in its insights. Discrepancies, if any, could indicate nuanced opinions not captured by ratings alone.
6) **Influence of Helpful Reviews**: The `helpful_yes` and `total_vote` columns allow us to assess review influence. With a mean `helpful_yes` of 1.31 but a maximum of 1952, highly helpful reviews (e.g., those with a high helpfulness ratio: `helpful_yes / total_vote`) carry significant weight. Analyzing their sentiment and content reveals customer priorities—negative helpful reviews may spotlight widespread issues, while positive ones underscore valued features.
7) **Product Lifecycle Insights**: The `day_diff` column, ranging from 1 to 1064 days (approximately 3 years), enables analysis of sentiment across the product lifecycle. A potential correlation between `day_diff` and sentiment could indicate whether satisfaction wanes as the product ages or improves with updates, guiding long-term development strategies.

B. Summary of Findings

The dataset reveals a predominantly positive customer sentiment, with a mean `overall` rating of 4.587 and a `predicted_sentiment` mean of 1.233, where 0, 1, and 2 represent negative, neutral, and positive sentiments, respectively. The sentiment distribution (25% at 1, 50% at 1, 75% at 2) confirms that most reviews are neutral or positive, aligning with high star ratings. The classification in `function_details` categorizes review topics, enabling aspect-specific sentiment analysis when paired with `predicted_sentiment`. Temporal data from `reviewTime` and `day_diff` supports trend monitoring, while `helpful_yes` highlights influential opinions. This combined approach—merging classification and sentiment—provides a robust framework for tracking issue-specific sentiment, identifying strengths, and addressing weaknesses.

C. Next Steps

Based on these insights, we propose the following actionable steps to leverage the analysis for business growth:

- **Targeted Product Improvements**: Use aspect-based sentiment analysis to prioritize enhancements. For instance, if "compatibility" consistently shows negative sentiment, invest in R&D to improve compatibility with other devices or durability. This addresses customer pain points directly, boosting satisfaction.
- **Refined Marketing Campaigns**: Highlight features with strong positive sentiment (e.g., "product quality" if identified in `function_details`) in advertising. Positive feedback from reviews can be quoted or emphasized to build trust and attract new customers.
- **Enhanced Customer Support**: Address recurring complaints identified in negative sentiment reviews by updating support resources, such as FAQs or tutorials. Proactive resolution of common issues can reduce dissatisfaction and improve retention.
- **Continuous Sentiment Monitoring**: Implement a real-time dashboard aggregating `year_month`, `function_details`, and `predicted_sentiment` to track trends. This enables rapid detection of emerging issues or validation of product updates' impact, fostering agility in decision-making.
- **Engage with Influential Reviews**: Respond to highly helpful reviews (high `helpful_yes` ratios) to demonstrate responsiveness. For negative ones, offer solutions; for positive ones, express gratitude. This enhances brand reputation and customer loyalty.
- **Lifecycle Optimization**: Analyze sentiment versus `day_diff` to assess product aging effects. If sentiment declines over time, plan timely updates or replacements. Conversely, sustained positive sentiment could justify extending the product's market life.
- **Benchmarking Opportunities**: If the dataset expands to include multiple products or competitors (not currently evident), compare aspect sentiments to identify market positioning. This could reveal areas where the product outperforms or lags, informing strategic adjustments.

D. Conclusion

By combining classification with sentiment analysis, this study provides a powerful tool for monitoring issue-specific customer sentiment, as demonstrated through the integration of `function_details` and `predicted_sentiment`. The insights—ranging from aspect-specific feedback to lifecycle trends—equip businesses to refine products, enhance customer experiences, and strengthen market presence. Implementing the proposed next steps ensures data-driven decisions that align with customer needs, driving both immediate improvements and long-term growth. Future analyses could expand this framework by incorporating product identifiers or competitor data, further enriching strategic insights.

VI. Conclusion

This project affirms the growing importance of combining machine learning with natural language processing to tackle real-world data challenges. Through the construction of an automated review analysis pipeline, we have demonstrated how a hybrid modeling approach—grounded in SVM and BART—can yield meaningful results in both sentiment classification and review interpretation.

The sentiment model, fine-tuned with external datasets and validated on Amazon product reviews, showed strong accuracy in distinguishing between positive, negative, and neutral tones. This enabled us to track shifts in customer satisfaction over time, uncover underlying patterns of approval or dissatisfaction, and generate a more nuanced rating system that reflects true consumer sentiment, beyond simple star ratings.

Our work also proves that insights derived from review text are not limited to qualitative understanding. They can be systematically quantified and operationalized. For instance, the sentiment-derived scores generated in our analysis could be directly fed into product ranking and recommendation systems to optimize user engagement and improve conversion rates.

More broadly, the modularity and transferability of our framework mean that it can be adapted to other platforms, industries, or languages where customer reviews serve as a vital feedback channel. This positions our project as not just a proof-of-concept, but a scalable template for intelligent review analysis in any data-rich environment.

VII. Limitations & Future Improvements

While our project demonstrates the feasibility and value of automated review analysis using machine learning, several limitations were observed that highlight opportunities for future enhancement.

One key limitation is the lack of issue-type classification in our final implementation. Although initially proposed, time constraints prevented us from developing a categorization pipeline that could differentiate between concerns such as delivery problems, product defects, or customer service complaints. Without this thematic layer, the insights remain general, lacking the specificity needed for targeted operational responses. In the future, implementing aspect-based sentiment analysis or multi-label classification could greatly enrich the system's diagnostic capabilities.

Secondly, the ambiguity and complexity of natural language continue to pose challenges. Sarcasm, idiomatic expressions, and mixed sentiment in reviews often lead to misclassifications, especially in cases labeled as neutral. Incorporating additional language models fine-tuned on e-commerce-specific corpora, or adding attention mechanisms that focus on opinion-bearing words, may help improve performance in these edge cases.

Another limitation lies in the data structure itself. Many products in the dataset had only one or two reviews, making it difficult to aggregate sentiment at the item level. To address this, future work could incorporate real-time crawling or batching to collect a minimum threshold of reviews per item before assigning scores, ensuring statistical reliability.

Lastly, while the sentiment model trained on Twitter performed reasonably well on Amazon reviews, differences in tone and vocabulary between domains mean that some domain-specific fine-tuning is still necessary. Future improvements could include transfer learning strategies such as domain adversarial training to better bridge linguistic gaps across platforms.

Despite these limitations, our system lays the groundwork for a robust, scalable, and real-time feedback loop. With further refinement in model granularity, interpretability, and deployment infrastructure, the current pipeline could evolve into a full-fledged analytics tool capable of driving customer experience insights across a wide range of platforms and industries.

REFERENCES

[1] J. A. Chevalier and D. Mayzlin, "The effect of word of mouth on sales: Online book reviews," *Journal of Marketing Research*, vol. 43, no. 3, pp. 345–354, 2006.

[2] B. Liu, *Sentiment Analysis and Opinion Mining*. Morgan & Claypool, 2012.

[3] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. of NAACL*, 2019.

[5] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.

[6] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 2nd ed. Prentice Hall, 2009.

[7] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017.

[8] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. of NIPS*, 2013, pp. 3111–3119.

[9] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[10] T. B. Brown *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[11] M. Lewis *et al.*, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. of ACL*, 2019.

[12] Y. Liu *et al.*, "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[13] P. He *et al.*, "DeBERTa: Decoding-enhanced BERT with disentangled attention," in *Proc. of ICLR*, 2021.

[14] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.

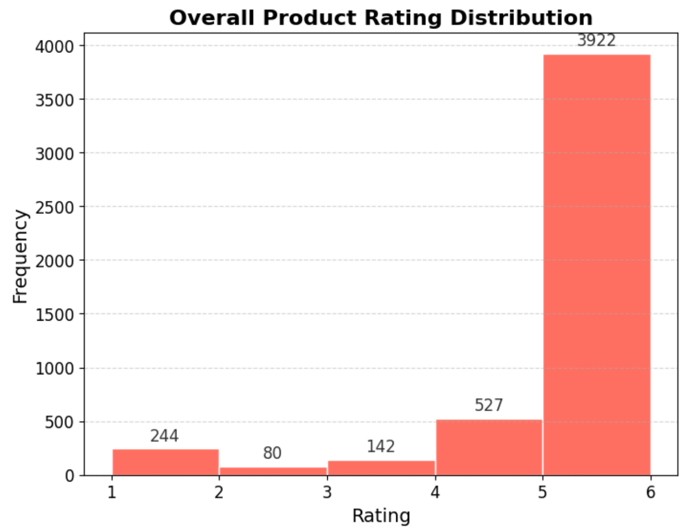
[15] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.

APPENDIX

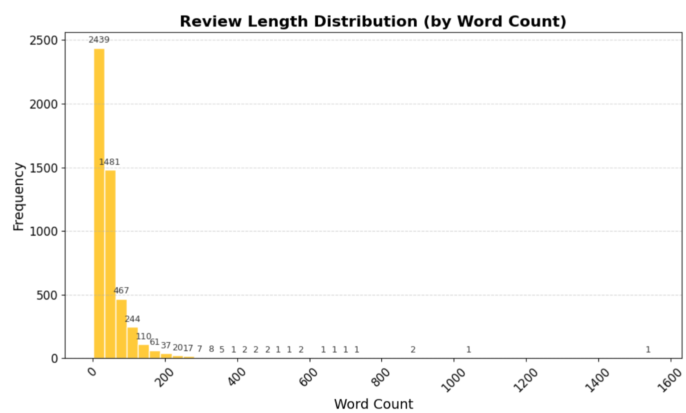
6.1 Summary Statistics of Numeric Variables

	Overall Rating	Review Timestamp (Unix)	Days Since Review	Helpful Votes	Total Votes	Review Length (Words)
Count	4915.000000	4.915000e+03	4915.000000	4915.000000	4915.000000	4915.000000
Mean	4.587589	1.379465e+09	437.367040	1.311089	1.521465	50.442523
Std Deviation	0.996845	1.581857e+07	209.439871	41.619161	44.123095	59.114688
Min	1.000000	1.339200e+09	1.000000	0.000000	0.000000	1.000000
25th Percentile	5.000000	1.365898e+09	281.000000	0.000000	0.000000	23.000000
Median (50th %)	5.000000	1.381277e+09	431.000000	0.000000	0.000000	33.000000
75th Percentile	5.000000	1.392163e+09	601.000000	0.000000	0.000000	55.000000
Max	5.000000	1.406074e+09	1064.000000	1952.000000	2020.000000	1554.000000

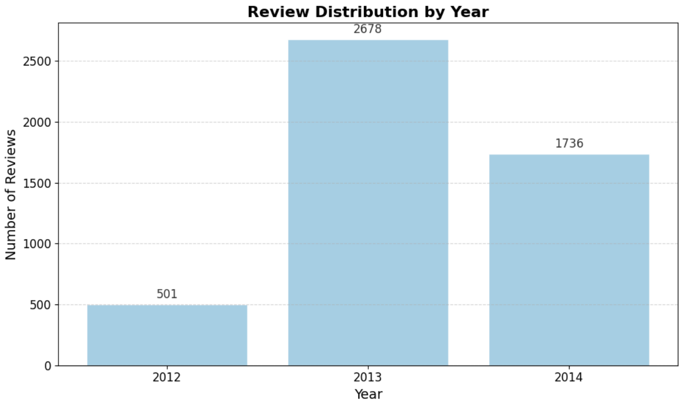
Appendix: Descriptive Statistics of Numeric Variables



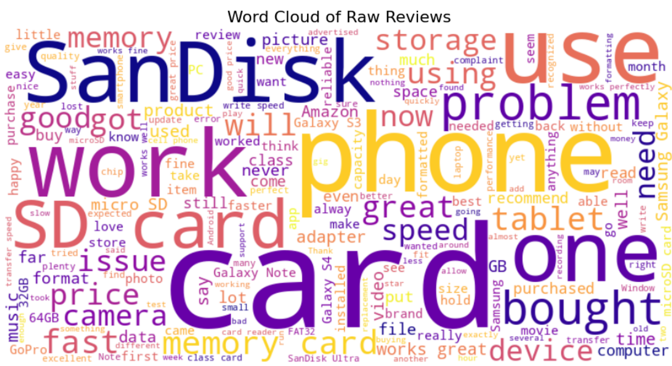
Appendix: Histogram of Overall Star Rating Distribution



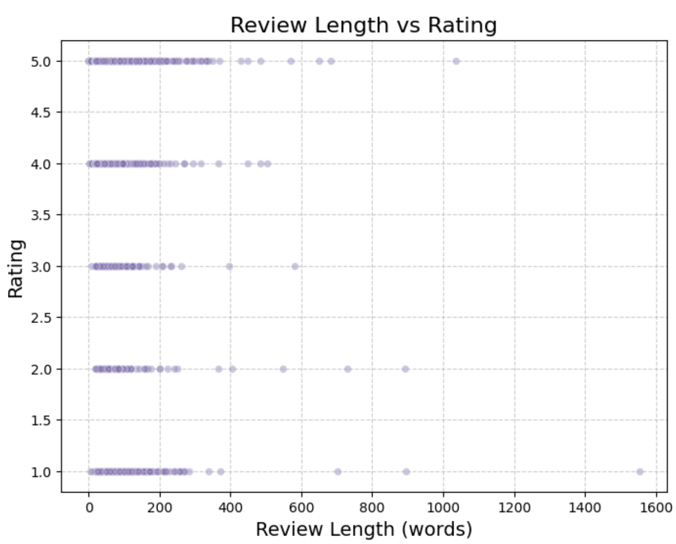
Appendix: Histogram of Review Length (Word Count) Distribution



Appendix: Bar Chart of Review Counts by Year



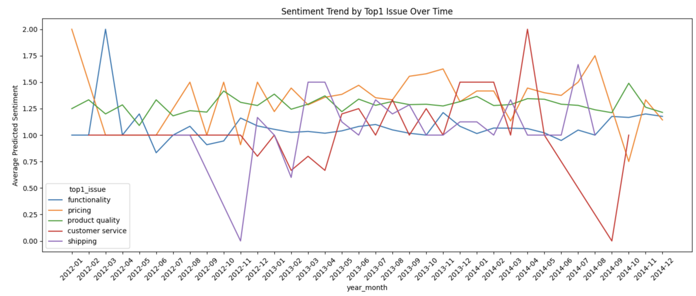
Appendix: Word Cloud of Raw Review Text



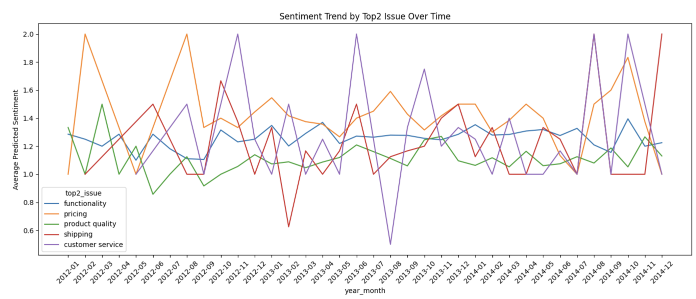
Appendix: Scatter Plot of Review Length versus Star Rating

	overall	unixReviewTime	reviewTime	day_diff	helpful_yes	total_vote	predicted_sentiment
count	4918.000000	4.918000e+03	4918	4918.000000	4918.000000	4918.000000	4918.000000
mean	4.587434	1.379472e+09	2013-09-26 17:15:03.294021888	437.281212	1.310289	1.520333	1.233428
min	1.000000	1.339200e+09	2012-01-09 00:00:00	1.000000	0.000000	0.000000	0.000000
25%	5.000000	1.365898e+09	2013-04-16 00:00:00	281.000000	0.000000	0.000000	1.000000
50%	5.000000	1.381277e+09	2013-10-03 00:00:00	431.000000	0.000000	0.000000	1.000000
75%	5.000000	1.392163e+09	2014-03-02 00:00:00	601.000000	0.000000	0.000000	2.000000
max	5.000000	1.406074e+09	2014-12-07 00:00:00	1064.000000	1952.000000	2020.000000	2.000000
std	0.996629	1.582394e+07	NaN	209.484481	41.606475	44.109644	0.566509

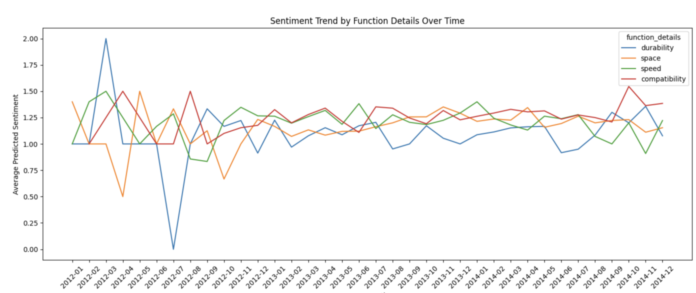
Appendix: Amazon Reviews with Predicted Sentiment



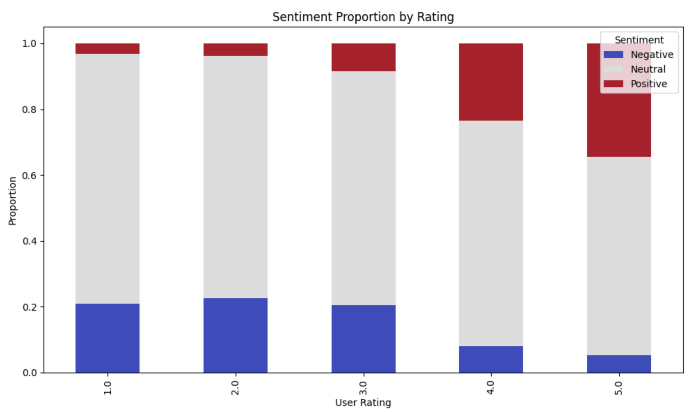
Appendix: Top 1 Issue Sentiment Trends Over Time



Appendix: Top 2 Issue Sentiment Trends Over Time



Appendix: Functionality Details Issue Sentiment Trends Over Time



Appendix: Amazon Reviews Sentiment Proportion by Rating