Leveraging NLP and Machine Learning to Understand Customer Sentiment

A Study of Customer Reviews on Digital Products

Dae Young Kim

College of Engineering
Michigan State University

December 2024

Table of Contents

1. Problem Definition and Motivation

Sentiment analysis, a key application of natural language processing (NLP), has been widely explored across industries due to its ability to provide actionable insights from textual data. In the context of e-commerce, particularly for digital products, understanding customer sentiments expressed in reviews is essential for improving customer satisfaction and driving business profitability. Customer reviews provide a wealth of information about user preferences, pain points, and overall experiences, making them invaluable for businesses seeking to align their offerings with market demands (Rasappan, Premkumar, Sinha, & Chandrasekaran, 2024).

In the competitive online marketplace, where the supply chain, fast-moving consumer goods (FMCG), and e-commerce platforms are intertwined, analyzing user sentiment is crucial (Deliverect, 2024). Customer reviews, along with behavioral data such as search queries, time spent on specific product pages, and click patterns, can collectively inform decision-making. For instance, businesses can use these insights to provide personalized product recommendations tailored to individual customer interests, thereby enhancing the shopping experience and increasing sales.

Effective sentiment analysis also plays a pivotal role in inventory management, particularly given the vast and diverse customer base served by e-commerce platforms. Managing inventory for such a large audience becomes more efficient when businesses can anticipate customer demands based on sentiment trends (Wood, Reiners, & Srivastava, 2013). To achieve this, data science techniques, big data analytics, and parallel computing tools become indispensable.

The main goal of this study was to analyze customer reviews for digital products. I aimed to classify sentiments as positive or negative, giving a clear picture of customer opinions. I also wanted to identify which product features, such as battery life, sound quality, or durability, received the most positive or negative feedback. This would help me find areas that are performing well or need improvement. Finally, I aimed to use the sentiment data to provide actionable recommendations, helping businesses improve their product features, enhance customer experiences, and stay competitive in the market.

2. Data Description

2.1 Dataset Overview

For this study, I utilized the Annotated Customer Review Dataset, a publicly available dataset curated by Minqing Hu and Bing Liu in their research published at the 2004 ACM SIGKDD

International Conference on Knowledge Discovery & Data Mining (KDD-04) and the National Conference on Artificial Intelligence (AAAI-2004) (Liu & Hu, 2004). The dataset, specifically designed for sentiment analysis and opinion mining tasks, includes customer reviews of five distinct products sourced from Amazon.

The dataset was obtained from the web page *Opinion Mining, Sentiment Analysis, and Opinion Spam Detection* hosted by the University of Illinois at Chicago. It is annotated with sentiment and feature markers, offering a rich resource for analyzing customer opinions and the associated product features.

2.2 Product Categories

The dataset includes reviews for the five products listed in the table title “Product categories” below.

Product	Category	Number of Reviews
Canon G3	Digital Camera	46
Nikon Coolpix 4300	Digital Camera	35
Nokia 6610	Cellular Phone	42
Creative Labs Nomad Jukebox Zen Xtra 40GB	MP3 Player	96
Apex AD2600 Progressive-scan DVD Player	DVD Player	100

Table: Product categories

2.3 Annotation Details

The dataset uses a structured annotation system to capture sentiment nuances, feature references, and contextual elements. Key annotation symbols can be viewed in the table title “Annotation symbols” below.

Symbol	Meaning	Description	Sentiment Levels
[t]	Title	Indicates the title of the review	N/A
[+n]	Positive Sentiment	Positive sentiment about a product feature	1 (weakest) to 3 (strongest)
[-n]	Negative Sentiment	Negative sentiment about a product feature	1 (weakest) to 3 (strongest)
##	Sentence Marker	Marks the beginning of each sentence in the review	N/A
[u]	Unmentioned Feature	A feature not explicitly mentioned in the sentence	N/A
[p]	Pronoun Resolution	Feature that requires resolving a pronoun to identify	N/A
[s]	Suggestion/ Recommendation	Denotes a suggestion or recommendation in the text	N/A

[cc]	Cross-Brand Comparison	Comparison with a competing product from a different brand	N/A
[cs]	Same-Brand Comparison	Comparison with a competing product from the same brand	N/A

Table: Annotation symbols

2.4 Dataset Example

The dataset is organized as a text-based file where individual reviews are separated by specific markers and header information. Each product's review set begins with a header block containing metadata like product name and source (amazon.com). Reviews are delimited by distinctive separators, with each review containing multiple annotated sentences. Each sentence is marked with special annotation symbols that provide semantic and sentiment information. The text uses a plain text format with no specific structured data format, making it a raw text corpus that requires parsing. Reviews are separated by lines of asterisks or other distinctive markers, and each review contains multiple sentences annotated with symbols like [t] for title, ## to mark sentence starts, and [+n] or [-n] to indicate sentiment intensity.

These annotations go beyond simple star ratings by tracking specific sentiments about product features, indicating whether comments are positive or negative (and how strongly), noting comparisons with other products, and marking suggestions or unmentioned features. For example, in a review about Nokia 6610 in the figure titled “Nokia 6610 sample raw data” below, the annotation shows "+3" next to "phone" to indicate an extremely positive sentiment, or "-2" next to "customer service" to show a negative experience.

```
*****************************************************************************
* Annotated by: Minqing Hu and Bing Liu, 2004.
*               Department of Computer Sicence
*               University of Illinois at Chicago
*
* Product name: Nokia 6610
* Review Source: amazon.com
*
* See Readme.txt to find the meaning of each symbol.
*****************************************************************************

[t]excellent phone , excellent service .
##i am a business user who heavily depend on mobile service .
phone[+3], work[+2]##there is much which has been said in other reviews about the features of this phone , it is a great phone , mine worked without any problems right out of the box .
##just double check with customer service to ensure the number provided by amazon is for the city / exchange you wanted .
at&t customer service[-2]##after several years of torture in the hands of at&t customer service i am delighted to drop them , and look forward to august 2004 when i will convert our other 3 family-phones from at&t to t-
mobile !
signal quality[+3]##i have had the phone for 1 week , the signal quality has been great in the detroit area ( suburbs ) and in my recent road trip between detroit and northern kentucky ( cincinnati ) i experienced
perfect signal and reception along i-75 , far superior to at &#38; t 's which does not work along several long stretches on that same route .
##i have owned motorola , panasonic and nokia phones over the last 8 years and generally prefer nokia , this phone combines many of the best nokia features , the only feature missing for me is the voice recognition .
speaker phone[+2],radio[+2],infrared[+2]##my favorite features , although there are many , are the speaker phone , the radio and the infrared .
speaker phone[+2]##the speaker phone is very functional and i use it in the car , very audible even with freeway noise .
infrared[+2]##the infrared is a blessing if you have a previous nokia and want to transfer your old phone book to this phone , saved me hours of re-entering my numbers .
##the combination of the nokia 6610 and t-mobile service ( signal , price , service ) is a winner , i highly recommend it .
[t]good phone , so-so service .
sprint[-2]##the day finally arrived when i was sure i 'd leave sprint .
sprint plan[-2],sprint customer service[-3]##after years with that carrier 's expensive plans and horrible customer service , portability seemed heaven-sent .
```

Figure: Nokia 6610 sample raw data

This structured annotation allowed me to systematically extract detailed insights related to product features, customer sentiments, and comparative evaluations across different electronic devices.

3. Data Preprocessing

3.1 Review and Sentence Segmentation

I prepared the dataset through several steps per product category to ensure it was clean and suitable for analysis. First, I segmented the reviews of each product category into manageable parts. To do this, I used the tags provided in the dataset. The [t] tags helped me identify the beginning of each review, while the ## tags were used to split the reviews into individual sentences. This step allowed me to focus on smaller text units, which made it easier to process sentiment annotations tied to specific product features.

```
# Split the data into individual reviews based on '[t]' tags
reviews = [review for review in data.split('[t]') if review.strip()]
print(f'{product}: {len(reviews)}')

# List to store each processed review
processed_reviews = []

# For each review, split sentences and extract important information
for review in reviews:
    sentences = review.split('##') # each sentence starts with '##'
```

Figure: Review and sentence segmentation code

3.2 Extracting Feature-Sentiment Pairs

I extracted feature-sentiment pairs from the reviews. I used regular expressions to identify product features mentioned in the text, such as "battery" or "sound quality," and linked them to the sentiment expressed by the customer. Each sentiment was categorized as positive or negative and given a strength score from 1 to 3, with 1 being the weakest and 3 being the strongest. For instance, if a sentence has "example[+1], test[-2], code[+3]", then output would be [('example', '+', '1'), ('test', '-', '2'), ('code', '+', '3')].

```
# Extract feature-sentiment pairs
# Keeping feature information in JSON format for now
features_in_sentence = re.findall(r'(\w+)\[([+-])(\d)\]', sentence) # find feature, polarity, and strength using patterns within square brackets
    # returns all non-overlapping matches of the pattern in the string as a list of tuples; each tuple contains the captured groups
    # so, from a sentence with matches like word[+1], get a list of tuples with the word, the sign, and the digit
    # - (\w+): matches one or more word characters (letters, digits, and underscores) and captures them in a group
    # - \[: matches the literal '[' character
    # - ([+-]): matches a single '+' or '-' character and captures it in a group
    # - (\d): matches a single digit and captures it in a group
    # - \]`: atches the literal ']' character
    # if sentence = "example[+1], test[-2], code[+3]" --> output: [('example', '+', '1'), ('test', '-', '2'), ('code', '+', '3')]
for feature, polarity, strength in features_in_sentence:
    features.append({
        "feature": feature,
        "polarity": 1 if polarity == '+' else -1,
        "strength": int(strength)
    })
```

Figure: Feature-sentiment paris extraction code

3.3 Text Cleaning

I also cleaned the text to remove unnecessary information. All text was converted to lowercase to ensure consistency. Special characters, punctuation, and irrelevant tags such as [s], [cc], and [cs] were removed because they did not contribute to sentiment analysis. However, I kept the [u] and [p] tags because they were important for identifying features through pronoun resolution. After cleaning, I tokenized the text into individual words and removed stop words, such as "the," "is," and "and," using the NLTK library. These steps allowed me to focus on the meaningful parts of the text.

3.4 Lemmatization

To standardize the text further, I applied lemmatization using NLTK. This process transformed words into their base forms, such as converting "running" and "runs" into "run." By doing this, I reduced redundancy in the dataset and ensured that variations of the same word were treated as a single term. This step improved the quality of the features extracted from the text, which I expected to enhance the performance of the sentiment analysis models.

```python
# Clean and normalize the text and removed stop words
# leaving only words useful for sentiment analysis
def preprocess_text(text):
    # convert text to lowercase
    text = text.lower()
    # remove unnecessary characters (punctuation and special characters)
    text = re.sub(r'[^\w\s]', '', text) #
    # tokenize text and remove stop words
    tokens = nltk.word_tokenize(text)
    # lemmatize e.g. rocks -> rock, better -> good
    tokens = [lemmatizer.lemmatize(word) for word in tokens if word not in stop_words]
    return ' '.join(tokens)
```

```python
# Keep only the text and clean it
# remove all [u] and [p] tags from sentences
# sentence = re.sub(r'\[u\]|\[p\]', '', sentence)
# remove tags
sentence_text = re.sub(r'\[.*?\]', '', sentence).strip()
    # removing everything within and including the square brackets
    # - \[: matches the literal '[' character
    # - .*?: matches any character (.) zero or more times (*),
    #        but as few times as possible (?), in a non-greedy manner
    # - \]`: matches the literal ']' character
    # if sentence = "example[+1], test[-2], code[+3]" --> output: example, test, code
review_text += sentence_text + " "
```

Figure: Text cleaning and lemmatization codes (left and right)

3.5 Target Labeling

Finally, I created target labels for each review. Initially, I labeled reviews as positive, negative, or neutral based on the sentiment scores of the product features. However, the model struggled with the neutral class because of limited data and imbalance issues. To address this, I decided to use only two classes: positive and negative. As displayed in the figure titled “Target labeling logic”, reviews were labeled as positive if the total positive sentiment score was higher than the negative score, and vice versa. In cases where the scores were equal, I resolved ties by comparing the strongest individual sentiment scores. If the strongest positive score was equal to or greater than the strongest negative score, the review was labeled as positive. Otherwise, it was labeled as negative.

```python
def label_class(features):
    # Extract strengths for positive and negative sentiments
    positive_strengths = [f['strength'] for f in features if f['polarity'] == 1]
    negative_strengths = [f['strength'] for f in features if f['polarity'] == -1]

    # Calculate total scores for positive and negative sentiments
    positive_score = sum(positive_strengths)
    negative_score = sum(negative_strengths)

    # Compare total scores to determine overall sentiment
    if positive_score > negative_score:
        return 'positive'
    elif negative_score > positive_score:
        return 'negative'
    else:
        # Resolve neutrality by comparing the maximum individual strengths
        # If both strengths are equal, default to 'positive'
        if max(positive_strengths, default=0) >= max(negative_strengths, default=0):
            return 'positive'
        else:
            return 'negative'
```

Figure: Target labeling logic

After the preprocessing, I had a dataset of 314 reviews in total, well-structured and ready for analysis. By focusing on meaningful information and addressing class imbalances, I targeted to improve the accuracy and reliability of the sentiment classification models.

4. Analysis and Insights

Understanding customer sentiment and its relationship with product features is critical for identifying strengths and areas for improvement in digital products. In this section, I present key analyses performed on the dataset, along with the insights derived from them. The exploration included various visualizations and statistical evaluations to uncover patterns and trends in customer reviews.

4.1 Performed Analyses

No	Analysis	Description
1	Distribution of Reviews by Product	Overview of review spread across different products
2	Sentiment Polarity Distribution by Sentiment	Analysis of sentiment polarities
3	Sentiment Distribution Across Products	How sentiments vary between different products
4	Top 10 Most Mentioned Features by Sentiment	Key features discussed in reviews, categorized by sentiment
5	Most Common Words in Positive Reviews	Wordcloud visualization of positive review vocabulary
6	Most Common Words in Negative Reviews	Wordcloud visualization of negative review vocabulary
7	Top 10 Features in Positive Reviews	Most frequently mentioned positive features
8	Top 10 Features in Negative Reviews	Most frequently mentioned negative features
9	Length of Reviews by Sentiment	Analysis of review lengths by sentiment
10	Length of Reviews by Product	Analysis of review lengths by product

11	Correlation Between Feature-Based Sentiment Metrics	Analysis of correlations among sentiment metrics derived from features
12	Correlation Between Reviews and Sentiment Strength	Analysis of correlations among sentiment metrics derived from reviews

Table: List of performed exploratory analysis

4.2 Visuals and Observation

Distribution of Reviews by Product and Sentiment Polarity

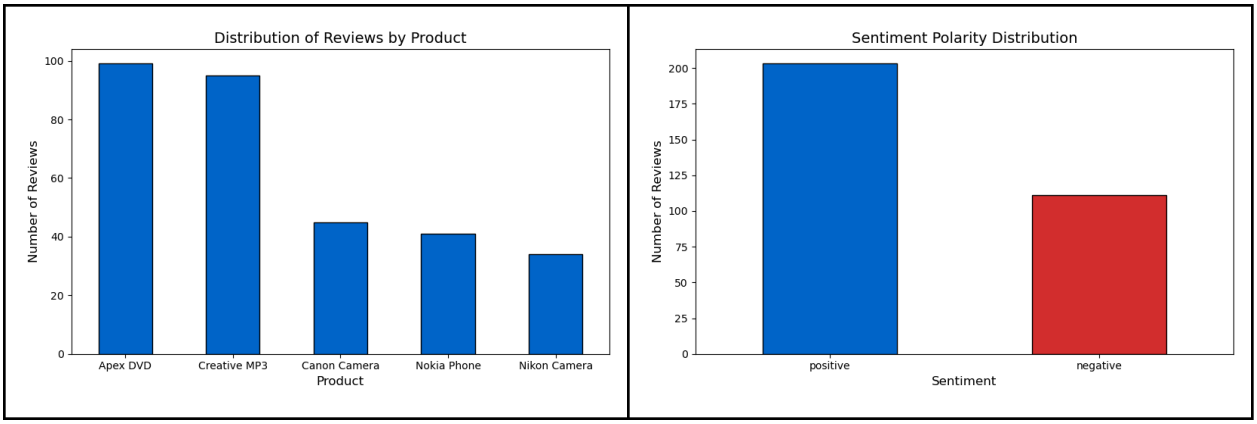

Figure: Distribution of reviews by product (left) and sentiment (right)

I analyzed the distribution of reviews across products and the polarity of sentiments expressed. The results revealed that the DVD player and MP3 player garnered the most customer attention. Across the dataset, there were more positive reviews than negative ones, indicating an overall positive sentiment among the customers.

Sentiment Distribution Across Products and Features

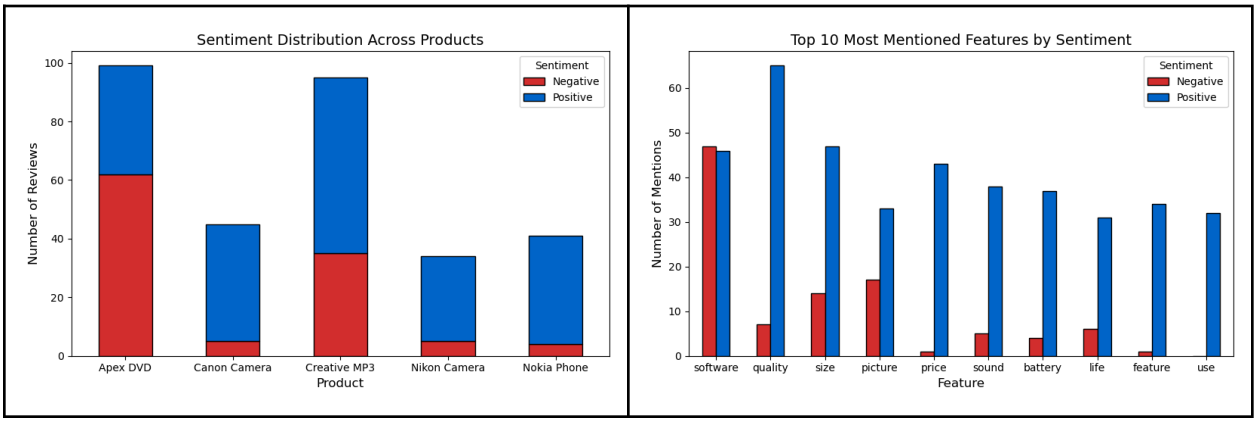

Figure: Sentiment distribution across products (left) and features (right)

Examining sentiment distribution by product showed that most products, such as those from Canon, Nikon, and Nokia, received predominantly positive reviews. However, the DVD player stood out as an exception, receiving more negative reviews than positive ones. I also found that customers are generally satisfied with features like quality, size, price, sound, battery, and usability. Significant negative reviews for software, size, and picture features highlight the need to address issues specific to these key areas.

Wordcloud Analysis of Positive and Negative Reviews

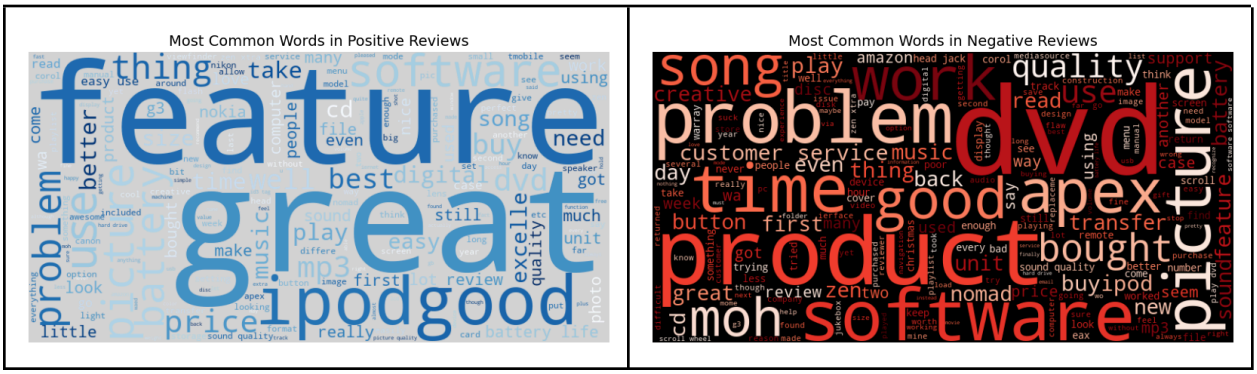

Figure: Most common words in positive (left) and negative (right) reviews

Wordcloud visualizations provided an overview of the most common words in positive and negative reviews. Positive reviews frequently included terms such as "feature," "software," "use," "picture," and "price," reflecting customers' appreciation for these attributes. Negative reviews, however, often mentioned "dvd," "software," "problem," "product," and "time," indicating specific areas of dissatisfaction.

Top Features by Sentiment

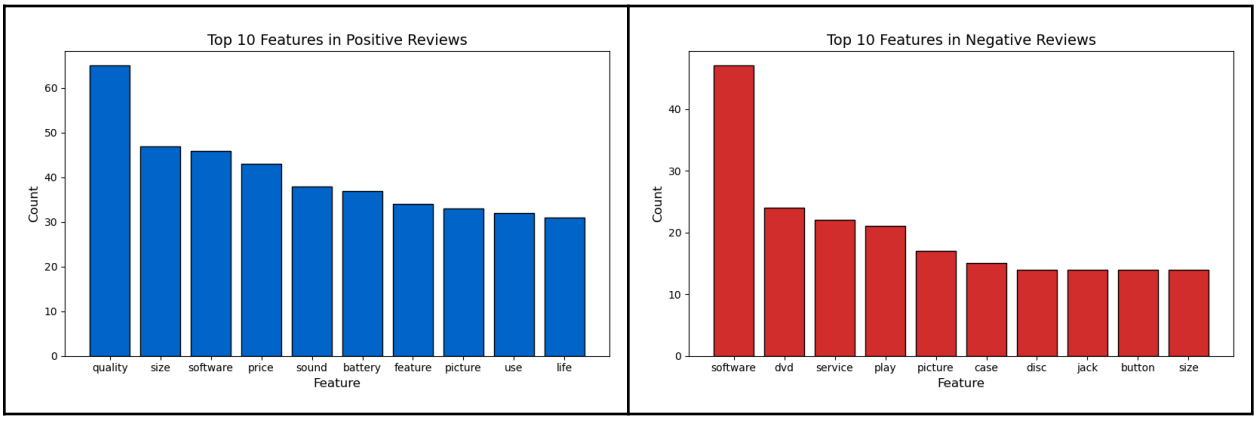

Figure: Top features by positive reviews (left) and negative reviews (right)

I identified the top 10 features mentioned in reviews, categorized by sentiment. Customers expressed high satisfaction with features such as quality, size, price, sound, battery, and usability. On the other hand, features like software, size, and picture were frequently associated

with negative sentiments. These negative mentions suggest potential pain points that manufacturers could focus on to improve customer satisfaction.

Length of Reviews by Sentiment and Product

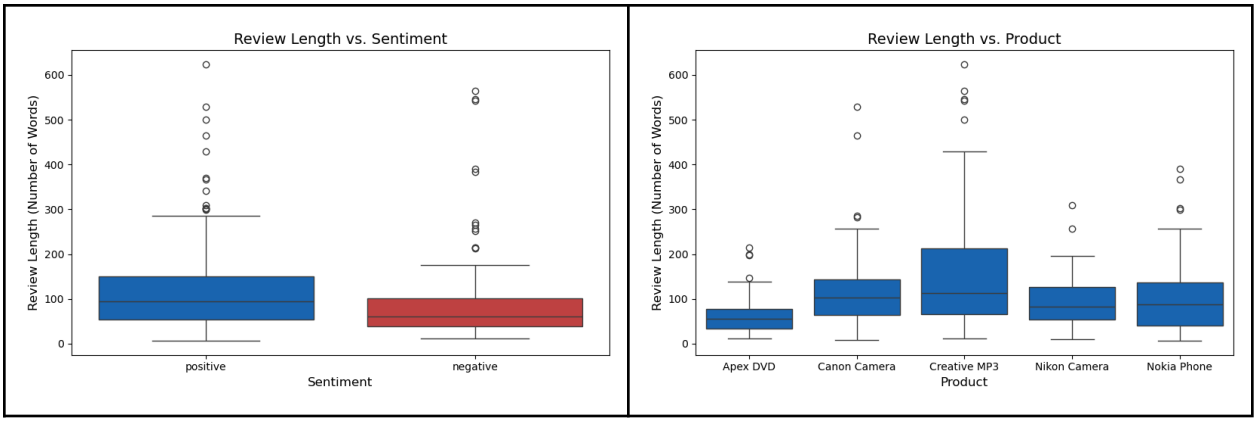

Figure: Review length by sentiment (left) and by product (right)

A boxplot analysis of review lengths revealed that positive reviews were generally longer than negative ones. This suggests that customers are more likely to elaborate when expressing positive experiences, while negative feedback tends to be more concise.

I also looked at how long reviews are for different products. The results show that MP3 reviews are usually the longest, while DVD reviews are shorter. This might mean that people write longer reviews for products that need more explanation or are more engaging, while shorter reviews could mean the product is simpler or less interesting to talk about.

Correlation Between Sentiment Metrics

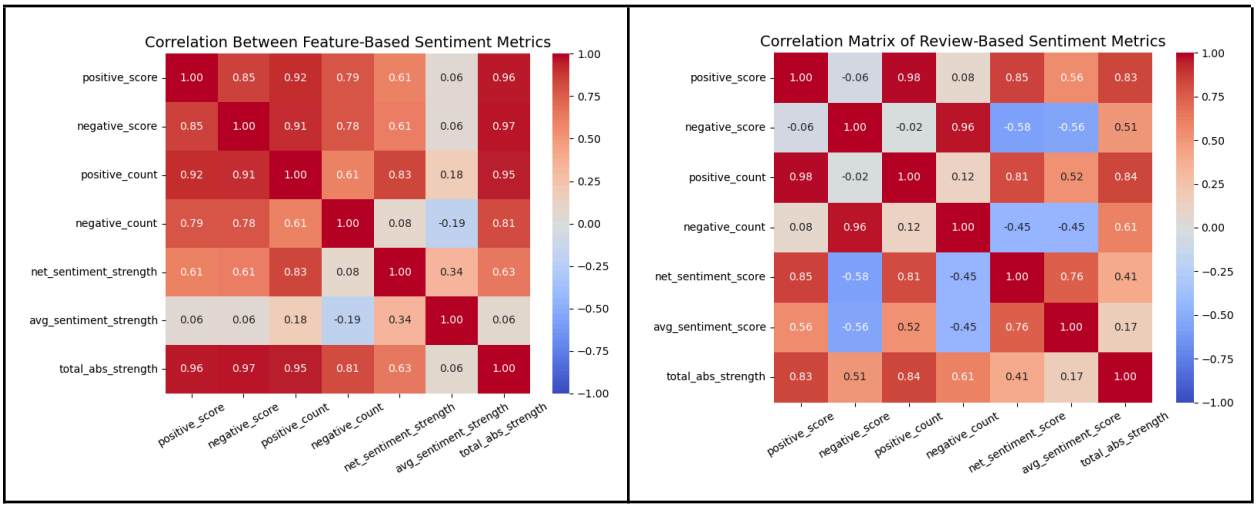

Figure: Correlation between feature-based (left) and review-based (right) sentiment metrics

Lastly, I conducted analyses of correlations among sentiment metrics derived from features and from reviews. I observed that most sentiment metrics showed a strong correlation, ranging from 0.6 to 0.9. This suggests that metrics such as feature counts and sentiment scores tend to move together, reflecting consistent patterns in how customers express their sentiments. However, the average sentiment strength displayed a much weaker correlation, between 0.06 and 0.18. This weak correlation indicates that the intensity of individual sentiments operates somewhat independently of aggregated sentiment scores or frequencies, suggesting that highly emotional responses might not always align with the overall sentiment trends in the dataset.

Interestingly, positive and negative sentiment scores, despite being opposites, demonstrated a strong correlation. This finding suggests that polarized opinions often occur in response to emotionally charged topics, with customers expressing both strong positive and negative sentiments about the same features. For example, while some users might appreciate a product's design, others may find it lacking, leading to a significant emotional divide. Furthermore, I found that positive feature counts correlated strongly with total absolute sentiment strength (0.84). This indicates that positive reviews tend to dominate in emotional intensity compared to negative feedback, reinforcing the idea that customers are more expressive when satisfied.

The contrast between feature-level and review-level correlations provided further insights. Feature-level correlations, such as the strong relationship between total absolute strength and negative sentiment scores (0.97), highlighted areas that need improvement. By addressing these pain points, companies can effectively resolve customer concerns and enhance product quality. On the other hand, review-level insights revealed that positive sentiments often display higher emotional intensity in holistic reviews. This highlights the importance of fostering positive customer experiences, as they contribute to building loyalty, improving satisfaction, and enhancing the overall brand reputation. These insights underscore the dual value of addressing negative feedback and amplifying positive experiences for long-term success.

4.3 Key Insights

The analysis revealed that feature-level insights provide actionable information for improving product quality. For example, addressing common issues with "software" or "picture" could significantly reduce negative feedback. At the review level, I observed that positive reviews tend to dominate in emotional intensity, underscoring the importance of fostering positive experiences to enhance customer satisfaction and loyalty.

5. Modeling and Evaluation

In this section, I compared the performance of different approaches and models to assess the most effective approach for sentiment classification using the given dataset.

The dataset was first split into training (80%) and testing (20%) subsets, considering the small size, stratified by the target labels to maintain the class distribution. For model evaluation, I mainly focused on accuracy and macro F1 score since these metrics provide a balanced view of the model's performance across both positive and negative classes. Accuracy provides an overall measure of model correctness by calculating the proportion of correctly classified samples, while the Macro F1-Score offers a more distinct evaluation by computing the average F1-scores across all classes, ensuring balanced performance assessment that accounts for both precision and recall, regardless of class size.

Most importantly, I explored several techniques for feature extraction (vectorization), class handling, and classifier selection. Below, I detail the methods I utilized and the performance of each model.

5.1 Approach: TF-IDF

Method 1: TF-IDF with Neutral Class

In the first model, I used Term Frequency-Inverse Document Frequency (TF-IDF) as the feature extraction method, with a maximum of 500 features. This model included the neutral class, which resulted in an accuracy of 0.81. However, the macro F1-score was much lower at 0.54. This discrepancy occurred because the model struggled to predict the neutral class effectively. The neutral class was underrepresented in the dataset, which led to class imbalance and hindered the model's ability to correctly classify neutral reviews. The imbalance in the dataset caused the model to perform poorly in predicting reviews that were neither strongly positive nor negative, which affected the overall performance metrics.

Classification Report:

	precision	recall	f1-score	support
negative	0.76	0.73	0.74	22
neutral	1.00	0.00	0.00	3
positive	0.83	0.92	0.88	38
accuracy			0.81	63
macro avg	0.87	0.55	0.54	63
weighted avg	0.82	0.81	0.79	63

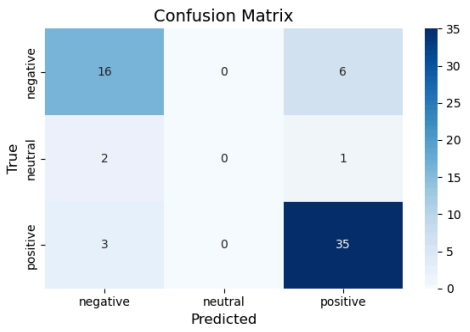

Figure: Performance of TF-IDF (with neutral class) Logistic Regression

Method 2: TF-IDF without Neutral Class

	Model	Accuracy	Macro F1-Score
0	Logistic Regression	0.857143	0.827817
1	MultinomialNB	0.841270	0.816860
2	RandomForest	0.809524	0.758312
3	SVM	0.825397	0.795756
4	XGBoost	0.777778	0.736244

Figure: Comparison of model performance to select the best model

For the second model, I again used TF-IDF with a maximum of 500 features, but this time, I excluded the neutral class and focused on binary sentiment classification (positive and negative). I ran several classifiers, including Logistic Regression, Multinomial Naive Bayes (MultinomialNB), Random Forest Classifier, Support Vector Machine (SVM), and XGBoost in their default forms.

All models were evaluated using accuracy and macro F1-score, and the results were compiled into a DataFrame for comparison (see Figure). Logistic Regression outperformed all other models, achieving an accuracy of 0.86 and a macro F1-score of 0.83 (see Figure). Although GridSearch was performed, the default Logistic Regression model delivered better performance than its tuned counterparts. The success of Logistic Regression can be attributed to its ability to handle sparse, high-dimensional data effectively, making it a suitable choice for TF-IDF features. Based on its strong performance across multiple metrics, I selected Logistic Regression as the baseline model for further experiments.

```
Classification Report:
              precision    recall  f1-score   support

    negative       0.93      0.64      0.76        22
    positive       0.83      0.98      0.90        41

    accuracy                           0.86        63
   macro avg       0.88      0.81      0.83        63
weighted avg       0.87      0.86      0.85        63
```

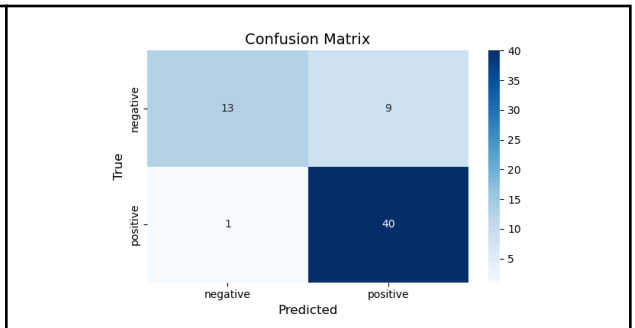

Figure: Performance of TF-IDF (without neutral class) Logistic Regression

Method 3: TF-IDF with N-Grams (Unigrams to Trigrams)

In the third model, I enhanced the feature extraction process by applying TF-IDF with unigrams to trigrams (n-gram range: 1 to 3) to capture more nuanced and important phrases in the reviews. This approach was designed to better represent the context of sentiment by including phrases and combinations of words. The extracted features were then fed into a Logistic

Regression classifier for binary sentiment classification. Despite the addition of n-grams, this model achieved an accuracy of 0.84 and a macro F1-score of 0.81 (see Figure), which was slightly lower than the performance of the baseline model.

Classification Report:

	precision	recall	f1-score	support
negative	0.93	0.59	0.72	22
positive	0.82	0.98	0.89	41
accuracy			0.84	63
macro avg	0.87	0.78	0.81	63
weighted avg	0.86	0.84	0.83	63

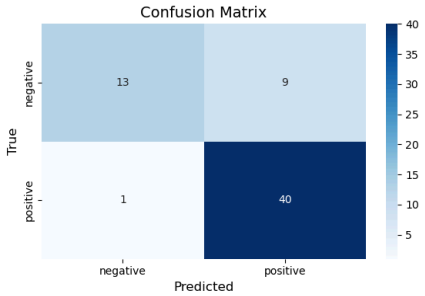

Figure: Performance of TF-IDF (n-grams) Logistic Regression

Several factors might explain the reduced performance: overfitting, noise introduction, and feature sparsity. Including bigrams and trigrams likely introduced too many features, causing the model to overfit the training data and perform poorly on unseen data. Additionally, higher-order n-grams might have introduced noise into the model, especially since the dataset size might not have been large enough to support this additional complexity, leading to decreased performance. The inclusion of bigrams and trigrams also might have resulted in a sparse feature matrix, making it harder for the model to learn meaningful patterns.

Method 4: TF-IDF with Imbalance Handling

In the fourth model, I used TF-IDF with a maximum of 500 features but incorporated techniques to handle class imbalance, such as Random Oversampling (RandomOverSampler) and Synthetic Minority Over-sampling Technique (SMOTE). The goal was to balance the positive and negative class distributions in the dataset. After addressing the class imbalance, I used a Logistic Regression classifier for binary sentiment classification. Both oversampling and SMOTE produced the same results, with an accuracy of 0.78 and a macro F1-score of 0.76 (see Figure).

Classification Report:

	precision	recall	f1-score	support
negative	0.68	0.68	0.68	22
positive	0.83	0.83	0.83	41
accuracy			0.78	63
macro avg	0.76	0.76	0.76	63
weighted avg	0.78	0.78	0.78	63

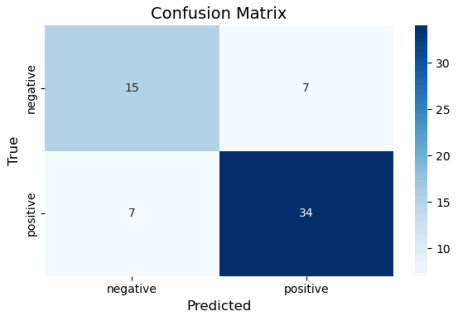

Figure: Performance of TF-IDF (imbalance handled) Logistic Regression

Although these methods helped mitigate the class imbalance, the overall performance actually decreased compared to the baseline. This decline in performance might have been due to the nature of the class imbalance handling methods. For instance, The specific methods used to address class imbalance, such as SMOTE and RandomOverSampler, might not have been optimal for the dataset. These methods might have introduced noise or failed to capture the true distribution of the data, which negatively impacted the model's ability to generalize. Additionally, the feature representation of the text data, using TF-IDF, may not have sufficiently captured the sentiment nuances, leading to less accurate sentiment predictions.

Method 5: TF-IDF with Ensemble Models

I finally applied an ensemble model using default voting with TF-IDF features. In ensemble models, each individual model captures different patterns in the data, and greater diversity among the models often leads to better overall performance. Combining tree-based models with linear models can enhance results. In my case, I used a Voting Classifier with five different models: Logistic Regression, Support Vector Machine (SVM), Random Forest, XGBoost, and K-Nearest Neighbors (KNN). The ensemble model achieved an accuracy of 0.86 and a macro F1-score of 0.83, matching the performance of the baseline model (see Figure).

Classification Report - TF-IDF and Ensemble

	precision	recall	f1-score	support
negative	0.933	0.636	0.757	22.0
positive	0.833	0.976	0.899	41.0
accuracy	0.857	0.857	0.857	0.857
macro avg	0.883	0.806	0.828	63.0
weighted avg	0.868	0.857	0.849	63.0

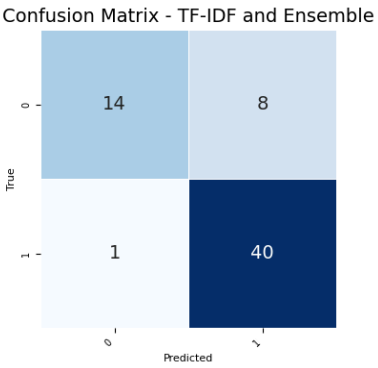

Figure: Performance of TF-IDF ensemble model

5.2 Approach: BERT

BERT (Bidirectional Encoder Representations from Transformers) is a powerful language model that understands the context of words in a sentence by looking at both the words before and after them. Unlike older models like Word2vec and GloVe, which only consider the words that come before, BERT can process information from both directions, making it much better at understanding complex language.

BERT uses a special type of neural network called a Transformer, which is designed to handle sequences of data like text. It has multiple layers that process the input text, paying attention to different parts of the sentence at each layer. This allows it to capture the relationships between words and understand the overall meaning of the text.

Additionally, BERT is trained on a massive amount of text data, which helps it learn the patterns and nuances of human language. This training process enables the model to generate more accurate and contextually relevant results compared to previous models.

I used the BERT model for feature extraction to transform the reviews text data into meaningful numerical representations. By applying BERT's pre-trained model (BERT-base, uncased version), I extracted high-dimensional embeddings from the reviews text, which captured contextual relationships between words, to understand the nuances of language.

Method 1: BERT

Initially, I experimented with a Logistic Regression model using BERT embeddings (with 256 max length). However, the performance was lower than that of the TF-IDF-based model. The BERT-based model achieved an accuracy of 0.83 and a macro F1-score of 0.80 (see Figure).

Classification Report - BERT and Logistic Regression

	precision	recall	f1-score	support
negative	0.789	0.682	0.732	22.0
positive	0.841	0.902	0.871	41.0
accuracy	0.825	0.825	0.825	0.825
macro avg	0.815	0.792	0.801	63.0
weighted avg	0.823	0.825	0.822	63.0

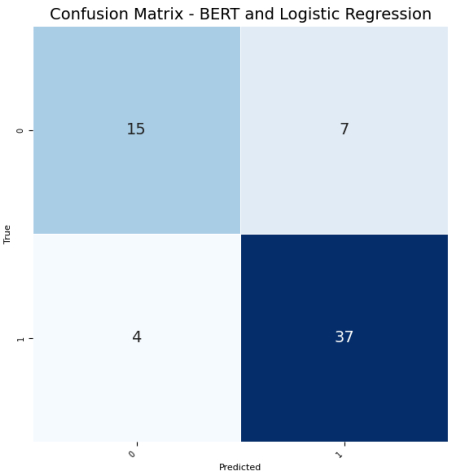

Figure: Performance of BERT Logistic Regression

Method 2: BERT with Ensemble Models

I then included the BERT embeddings as input features in the ensemble models, where they provided rich, context-aware information that helped improve classification performance. Ensemble modeling combines multiple machine learning models to make better predictions by leveraging the strengths of each model. Again, I used a Voting Classifier with five different models: Logistic Regression, Support Vector Machine (SVM), Random Forest, XGBoost, and K-Nearest Neighbors (KNN).

Specifically, I applied soft voting, which differs from hard voting (default) in a key way. While hard voting relies on a majority vote of predicted classes, soft voting calculates the weighted average of predicted class probabilities. For soft voting, I prioritized more reliable models in the ensemble and assigned custom weights to each model: [1, 2, 3, 3, 2], giving higher weights to Random Forest and XGBoost due to their strong performance with BERT in the dataset.

I tested both soft and hard voting methods, and the results demonstrated that soft voting with weighted models and hard voting performed similarly. Most importantly, both ensemble models outperformed the baseline TF-IDF model. The weighted soft voting approach achieved slightly better performance than the hard voting model, with an accuracy of 0.87 and a macro F1 score of 0.86 (see Figure). This represents approximately a +1% improvement in accuracy and a +3% increase in macro F1 score over the baseline TF-IDF model. In comparison, the hard voting approach achieved an accuracy of 0.87 and a macro F1 score of 0.86.

Classification Report - BERT and Ensemble (Soft)

	precision	recall	f1-score	support
negative	0.85	0.773	0.81	22.0
positive	0.884	0.927	0.905	41.0
accuracy	0.873	0.873	0.873	0.873
macro avg	0.867	0.85	0.857	63.0
weighted avg	0.872	0.873	0.872	63.0

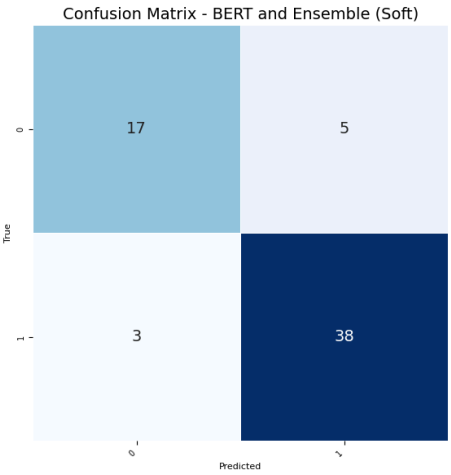

Figure: Performance of BERT soft voting ensemble model

5.6 Comparison and Results

Embedding	Methods	Classifier	Accuracy	Macro F1	Rank
BERT	- Pre-trained bert-base-uncased - Max 256 length	Ensemble (Weighted soft voting)	0.87	0.86	1
BERT	- Pre-trained bert-base-uncased - Max 256 length	Ensemble (Hard voting)	0.87	0.85	2
TF-IDF	- Max 500 features	Ensemble (Hard voting)	0.86	0.83	3
TF-IDF	- Max 500 features	Logistic Regression	0.86	0.83	3
TF-IDF	- Max 500 features - With unigrams to trigrams	Logistic Regression	0.84	0.81	5
BERT	- Pre-trained bert-base-uncased - Max 256 length	Logistic Regression	0.83	0.80	6
TF-IDF	- Max 500 features - Oversampled	Logistic Regression	0.78	0.76	7
TF-IDF	- With neutral sentiment class - Max 500 features	Logistic Regression	0.81	0.54	8

Table: Performance comparison

I compared multiple models using different feature extraction techniques, classifiers, and ensemble methods to identify the best approach for sentiment classification. Each model showed strengths and limitations, which I summarize below.

The baseline model used TF-IDF features with Logistic Regression and performed well among the simpler models. This model achieved a high accuracy of 0.86 and macro F1-score of 0.83, demonstrating that TF-IDF is a strong representation for text data in binary sentiment classification. However, adding n-grams (unigrams to trigrams) to TF-IDF increased the feature space, leading to slightly lower performance (0.81) due to overfitting and feature sparsity. Handling class imbalance through oversampling methods like SMOTE also did not improve performance (0.76) and possibly introduced noise, highlighting the challenges of balancing classes in smaller datasets.

The highest-performing models were built using BERT features. The pre-trained BERT model captured rich contextual information, which significantly improved classifier performance. Logistic Regression with BERT features (0.80) outperformed simpler models, and ensemble methods such as weighted and hard voting further boosted the results. The ensemble with weighted soft voting achieved the highest accuracy of 0.87 and macro F1-score of 0.86, making it the best model for this study. This success demonstrated the value of combining robust feature extraction with ensemble techniques to handle complex sentiment patterns effectively.

6. Conclusions

The study developed and evaluated various embedding methods and classifiers for sentiment classification to understand their impact on model performance. By focusing on customer reviews for digital products, the primary goals were to classify sentiments into positive or negative categories, identify key product features that received varying sentiment responses, and leverage these insights to provide actionable recommendations. Through these findings, I would like to help businesses refine their product offerings, improve customer satisfaction, and maintain competitiveness. Below, I summarize the key findings of the study.

6.1 Key Insights from Exploratory Analysis

The exploratory analysis provided valuable insights into customer sentiments and product feedback. First, I examined the sentiment distribution across various products. Most products, such as those from Canon, Nikon, and Nokia, received predominantly positive reviews, reflecting overall customer satisfaction. However, the DVD player was an outlier, with more negative reviews than positive ones, indicating a need for improvement. Customers expressed satisfaction with features such as quality, size, price, sound, battery, and usability. In contrast,

features like software, size, and picture were frequently associated with negative sentiments, highlighting potential pain points for manufacturers to address.

Additionally, I analyzed the length of reviews and sentiment correlations to uncover behavioral patterns. Positive reviews were typically longer, suggesting customers elaborated more when sharing positive experiences, while negative feedback was often brief. Correlation analysis of sentiment metrics revealed that most metrics were strongly related, reflecting consistent patterns in customer expression. However, the weak correlation of average sentiment strength suggested that highly emotional responses often do not align with overall trends. These findings emphasized the importance of focusing on both positive and negative feedback to improve product features and foster better customer satisfaction.

6.2 Model Performance and Implications

Category	Model	Description	Accuracy	Macro F1
Top Performers	BERT with Weighted Soft Voting	Best overall performance with advanced contextual understanding.	87%	86%
	BERT with Hard Voting	Similar results to weighted voting but slightly lower macro F1-score.	87%	85%
	TF-IDF with Logistic Regression	Competitive baseline model, simple yet efficient.	86%	83%

Table: Comparison of key models

The study provided valuable insights into how different embedding methods and classifiers affect the performance of sentiment analysis models. Advanced models like BERT with Weighted Soft Voting Ensemble performed the best, achieving 87% accuracy and 86% macro F1-score. However, traditional methods like TF-IDF with Logistic Regression also performed well. Despite being simpler, TF-IDF showed strong feature extraction capabilities, achieving almost the same accuracy as more complex models while being more efficient in terms of computation.

Although I used the dataset from 2004, both old and new machine learning techniques came out to be effective. BERT's rich, contextual embeddings performed very well even with older data, showing that the model can adapt to different time periods. By using ensemble methods that combine multiple classifiers, I was able to take advantage of the strengths of different models, improving the overall predictive performance. This approach highlights the power of ensemble techniques and shows how combining models can overcome the limits of individual classifiers, making sentiment analysis more effective across various types of data.

7. Challenges and Future Work

I faced significant challenges in sentiment analysis. The small dataset made it difficult for the models to capture the diverse ways people express their feelings. Additionally, I struggled with the computational power required, especially when using complex models like BERT.

To improve model performance in the future, I have several potential ideas. I plan to collect more reviews from various sources such as social media and online forums. This could help the models understand sentiments more accurately. I am also interested in exploring new text analysis methods, such as using SpaCy's newer transformer pipeline to first process the text and then passing that information to more advanced classifiers like XGBoost. These steps could help me develop a more powerful and flexible approach to understanding sentiments in text.

References

Rasappan, P., Premkumar, M., Sinha, G., & Chandrasekaran, K. (2024). Transforming sentiment analysis for e-commerce product reviews: Hybrid deep learning model with an innovative term weighting and feature selection. *Information Processing & Management*, 61(3), 103654. https://doi.org/10.1016/j.ipm.2024.103654

Deliverect. (2024). *Retail revolution: A deep dive into FMCG, Q-commerce, and first-party delivery*. Retrieved from https://www.deliverect.com/en-us/blog/fmcg-and-grocery/retail-revolution-a-deep-dive-into-fmcg-q-commerce-and-first-party-delivery

Wood, L. C., Reiners, T., & Srivastava, H. S. (2013). Expanding sales and operations planning using sentiment analysis: Demand and sales clarity from social media. In *Proceedings of ANZAM 2013* (Hobart, Tasmania, Australia). https://doi.org/10.13140/2.1.4165.6320

Liu, B., & Hu, M. (2004). *Opinion Mining, Sentiment Analysis, and Opinion Spam Detection: Feature-Based Opinion Mining and Summarization (or Aspect-Based Sentiment Analysis and Summarization)*. Retrieved from https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html