Customer Feedback Analysis and Summarization

Hithaishi Raghavendra Reddy
raghavendrareddy.h@northeastern.edu
Northeastern University
Boston, Massachusetts

Rohit Sunilkumar Hooda
hooda.r@northeastern.edu
Northeastern University
Boston, Massachusetts

Tanvi Prashant Magdum
magdum.t@northeastern.edu
Northeastern University
Boston, Massachusetts

ACM Reference Format:
Hithaishi Raghavendra Reddy, Rohit Sunilkumar Hooda, and Tanvi Prashant Magdum. 2024. Customer Feedback Analysis and Summarization. In . ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Customer feedback is an invaluable resource for businesses, offering information that can help improve products, services, and the overall customer experience. In today's fast-paced and competitive market, understanding what customers feel and say about your offerings is crucial. This project, Customer Feedback Analysis and Summarization, aims to harness the power of data to discover trends in consumer reviews. Using PySpark and Natural Language Processing (NLP) techniques, we'll analyze large-scale Amazon product reviews to efficiently handle and process vast amounts of text data, apply NLP tools to clean, analyze, and extract sentiments from customer feedback, identify recurring themes and summarize reviews to reveal actionable insights. By focusing on real-world applications, this project demonstrates how businesses can utilise technology to gain deeper knowledge into customer sentiment and identify common issues or highlights that matter most to consumers.

2 METHODOLOGY

2.1 Data Collection

For this project, we used a dataset of Amazon product reviews, which offers a treasure trove of information about customer opinions. The dataset is publicly available on Kaggle under the title Consumer Reviews of Amazon Products. It includes details such as customer ratings, review text, and product categories—making it ideal for sentiment and trend analysis. Here's how we approached the data: 1. Loading the Dataset: We loaded the dataset into a PySpark DataFrame, which is well-suited for handling large datasets. The structure of the dataset (its schema) was explored to identify key columns like reviewText and rating. 2. Getting to Know the Data: We did a quick exploration to check for missing values and understand how the data is distributed, such as the breakdown of ratings and the length of reviews. This helped us prepare the data for deeper analysis and ensured we're working with quality inputs. By organizing the data and understanding its structure, we extracted sentiments, identified patterns, and summarised key insights that can help businesses make informed decisions.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Data Cleaning

The goal of this step was to clean the data and prepare the raw dataset by addressing inconsistencies and ensuring the data was ready for sentiment analysis. Below are the steps performed:

(1) **Convert Text to Lowercase:**
   (a) Firstly, we made sure to convert all the text to lowercase characters only so that we could maintain consistency across all the reviews and eliminate the case sensitivity during analysis.
   (b) Thereafter, we removed punctuations, numbers, spaces, and special characters. This helped simplify the text and focus on meaningful content.
   (c) Finally, we filtered `NULL` and empty values. This ensured that only meaningful and non-empty reviews were retained.

2.3 Data Preprocessing

After cleaning the dataset, we preprocessed the data to prepare it for analysis and modeling. The following are the preprocessing steps:

(1) We used the `Tokenizer` to split the text into individual words and tokens. Thereafter, we eliminated common words that do not add meaningful information (words like e.g., "the," "is," "and").
(2) We categorized the reviews into 2 ratings:
   • Any ratings above 4 were marked as **Positive**.
   • Reviews below 2 were marked as **Negative**.
   • All the reviews between 2 and 4 were marked as **Neutral**.
   This was mainly done for sentiment-based grouping and insights.
(3) The reviews that resulted in empty tokens after stop word removal were filtered out using `size(filtered_tokens) > 0`. This ensured only useful reviews were retained for analysis.
(4) Finally, filtered tokens were concatenated back into summarized text. This was done to create clean summaries of reviews for reporting and insights.

3 RESULTS

3.1 Sentiment Analysis Overview

In this analysis, we used customer reviews from various product categories to analyze sentiment and key phrases. The sentiment analysis results were derived from the classification of feedback as positive, neutral, or negative, and we employed N-Grams (bigrams)

and TF-IDF for key phrase extraction. The final model, which utilized Logistic Regression, achieved an impressive accuracy of 91%, indicating that it is highly reliable in classifying sentiment across the reviews.

3.2 Sentiment Distribution by Category

We observed notable differences in sentiment across product categories. For instance, Electronics showed a predominance of positive sentiment (93.2%), with a smaller fraction of negative sentiment (2.71%) and neutral sentiment (4.09%). Similarly, Health & Beauty garnered strong positive feedback, with 86.09% of reviews classified as positive, 9.5% negative, and 4.42% neutral. These high percentages of positive sentiment suggest that, for the most part, customers are satisfied with these product categories.

In contrast, categories such as Toys & Games and Office Supplies exhibited more diverse sentiments. While Toys & Games saw a dominant positive sentiment (91.23%), Office Supplies experienced a higher portion of negative and neutral feedback (respectively, 1.54% and 5.53%), despite the overall high positive sentiment (92.93%).

3.3 Key Phrase Extraction and Summarized Feedback

Key phrase extraction using N-Grams (bigrams) revealed recurring themes in the customer feedback. Positive reviews tended to focus on aspects such as value for money, product quality, and satisfaction with performance, while negative feedback was centered around issues such as product failure, lack of effectiveness, and poor quality.

For example, positive feedback often contained summarized phrases such as "good quality" or "great value," while negative feedback included phrases like "poor quality" or "didn't work as expected." This highlights that the key determinants of customer satisfaction are tied to the perceived performance and quality of the products, while dissatisfaction arises from expectations not being met.

3.4 Feedback Trends and Actionable Insights

3.4.1 Overall Positive Sentiment. The data revealed that most customers are satisfied with the products they purchased. Positive sentiment was dominant across a variety of categories, suggesting that product quality and performance are generally meeting customer expectations.

3.4.2 Neutral Feedback. The occurrence of neutral sentiment in several categories indicates that some customers felt indifferent or experienced mixed feelings about the products. Neutral feedback often expressed a lack of strong opinion or indicated that the product "works as expected" but didn't exceed expectations. This suggests there is room for improvement in terms of making products more compelling to customers.

3.4.3 Negative Sentiment. Negative feedback, while less frequent than positive feedback, still highlighted areas for improvement. Common themes in negative reviews included dissatisfaction with product durability, effectiveness, and overall quality. For companies, addressing these issues could significantly improve customer retention and satisfaction.

3.4.4 Customer Expectations vs. Product Delivery. In many cases, negative sentiment seems to arise from unmet expectations. Reviews often noted that the product "didn't live up to expectations" or "wasn't as good as expected." These reviews point to a gap between customer expectations and the actual performance or quality of the product.

primaryCategories	sentiment	count
Electronics	negative	434
Computers,Electro...	neutral	2
Electronics,Media	neutral	3
Health & Beauty	negative	1146
Computers,Electro...	positive	2
Fire Tablets,Comp...	positive	1
Office Supplies,E...	neutral	36
Office Supplies,E...	positive	605
Electronics,Media	negative	5
Electronics,Furni...	positive	2
Animals & Pet Sup...	neutral	1
Electronics	neutral	654
Health & Beauty	positive	10390
Electronics,Media	positive	201
Toys & Games,Elec...	positive	1529
Health & Beauty	neutral	533
Fire Tablets,Comp...	negative	81
Computers,Electro...	negative	13
Home & Garden	positive	2
Animals & Pet Sup...	positive	5

only showing top 20 rows

Figure 1: Sentiment Insights

primaryCategories	sentiment	count	total_count	percentage
Electronics,Furni...	positive	2	2	100.0
Office Supplies,E...	negative	10	651	1.54
Office Supplies,E...	positive	605	651	92.93
Office Supplies,E...	neutral	36	651	5.53
Electronics	positive	14905	15993	93.2
Electronics	neutral	654	15993	4.09
Electronics	negative	434	15993	2.71
Home & Garden	positive	2	2	100.0
Electronics,Media	positive	201	209	96.17
Electronics,Media	negative	5	209	2.39
Electronics,Media	neutral	3	209	1.44
Toys & Games,Elec...	negative	54	1676	3.22
Toys & Games,Elec...	neutral	93	1676	5.55
Toys & Games,Elec...	positive	1529	1676	91.23
Office Supplies	positive	9	9	100.0
Health & Beauty	neutral	533	12069	4.42
Health & Beauty	positive	10390	12069	86.09
Health & Beauty	negative	1146	12069	9.5
Animals & Pet Sup...	positive	5	6	83.33
Animals & Pet Sup...	neutral	1	6	16.67

only showing top 20 rows

Figure 2: Sentiment Statistics

Figure 3: Feedback Summary Positive and Negative

3.5 Actionable Recommendations

3.5.1 Enhance Product Consistency. For products receiving mixed reviews, particularly those in the neutral category, ensuring consistent performance can help shift customer perceptions towards more positive sentiment. This could be achieved through better quality control and more accurate product descriptions.

3.5.2 Monitor Negative Feedback. The negative feedback, although less frequent, points to specific areas where customer expectations were not met. It is important for companies to actively address recurring issues, whether it is related to product durability, quality, or performance. Understanding these pain points and resolving them could help convert dissatisfied customers into satisfied ones.

3.5.3 Focus on Product Innovation. While many products received positive feedback, a significant portion of neutral feedback suggests that customers are looking for products that stand out more. Investing in product innovation to offer additional features or benefits could shift neutral feedback towards positive sentiment.

3.5.4 Targeted Customer Communication. Neutral feedback often represents customers who are neither satisfied nor dissatisfied. Communicating more effectively with these customers, offering product demonstrations, or providing better support can help improve their overall perception.

4 DISCUSSION

4.1 Limitation

Despite the comprehensive analysis conducted, there are several limitations that should be considered when interpreting the results. The dataset may not be fully representative of the entire customer base, as it primarily includes feedback from online sources, potentially excluding in-store or other offline customer experiences. This introduces a sampling bias that might not capture the full spectrum of customer sentiments.

The sentiment analysis model, while accurate, is not flawless, and misclassifications can occur, especially when dealing with ambiguous or sarcastic language. The model may also struggle with context-dependent phrases or mixed sentiments, which are common in customer reviews.

The process of key phrase extraction using N-Grams was limited to bigrams, which might not capture all relevant expressions, as longer phrases or context-specific keywords could provide deeper insights. Another limitation is the inherent subjectivity in customer feedback itself; individual experiences and expectations can vary widely, making it challenging to generalize conclusions across all customers.

Lastly, the lack of temporal analysis in this study means that shifts in customer sentiment over time were not explored, which could provide valuable insights for tracking product performance and customer satisfaction.

4.2 Future Research

There are several potential directions for expanding this analysis in the future like incorporating more advanced models for sentiment analysis, such as fine-tuning transformer-based models like BERT, which have demonstrated superior performance in understanding context and handling nuances in text. Future studies could also include a more diverse set of data sources, including customer feedback from social media, customer service interactions, and in-store surveys, to create a more representative sample.

An exploration of temporal trends would also be beneficial, as customer sentiment can fluctuate over time, especially after product updates or changes in customer service. This would enable businesses to better track the long-term performance of their products. Applying more granular feature extraction techniques, such as sentiment analysis on individual phrases or sentence-level evaluation, could yield deeper insights into specific aspects of customer experience. Integrating demographic data, such as customer location or age, could provide further segmentation and allow for more personalized insights and targeted strategies for customer satisfaction.

5 CONCLUSIONS

This analysis provides valuable insights into customer feedback trends and sentiment surrounding various products. By leveraging sentiment analysis and key phrase extraction techniques, we were able to identify patterns in customer opinions, including positive, neutral, and negative sentiments. The model demonstrated strong accuracy in classifying feedback, enabling the identification of key themes and common feedback drivers across product categories. However, the limitations highlighted, such as the exclusion of offline feedback and potential misclassifications by the sentiment model, suggest that there is room for refinement in future studies.

The results indicate that customers express a range of sentiments, with clear distinctions between positive and negative reviews, offering actionable insights for product and service improvement. Businesses can use these findings to make data-driven decisions, refine marketing strategies, and enhance customer satisfaction by addressing key pain points identified in the feedback. Moving forward, there are opportunities to further enhance the analysis by incorporating more advanced models, expanding the data sources, and exploring customer sentiment over time. Such improvements will allow for a more comprehensive understanding of customer experiences and provide a stronger foundation for strategic decision-making.

6 APPENDIX A CODE

```
import kagglehub
path = kagglehub.dataset_download("datafiniti/consumer-
    reviews-of-amazon-products")
print("Path to dataset files:", path)

from pyspark.sql import SparkSession
```

```
spark = SparkSession.builder.appName("AmazonReviewsAnalysis").config("spark.driver.memory", "16g").getOrCreate()
df = spark.read.csv(path, sep=",", header=True, inferSchema=True)
df.printSchema()
df.show(5)

# Data Preprocessing

from pyspark.sql.functions import col, lower, regexp_replace, length, length, trim, when, size
from pyspark.ml.feature import Tokenizer, StopWordsRemover

# Convert text to lowercase and remove punctuation
df_cleaned = df.withColumn("`reviews.text`", lower(col("`reviews.text`")))
df_cleaned = df_cleaned.withColumn("`reviews.text`", regexp_replace(col("`reviews.text`"), "[^a-zA-Z0-9\\s]", ""))

# Tokenization
tokenizer = Tokenizer(inputCol="`reviews.text`", outputCol="tokens")
df_tokenized = tokenizer.transform(df_cleaned)

# Remove stop words
remover = StopWordsRemover(inputCol="tokens", outputCol="filtered_tokens")
df_filtered = remover.transform(df_tokenized)

# Filter Reviews by Ratings
df_positive = df_filtered.filter(col("`reviews.rating`") >= 4)
df_negative = df_filtered.filter(col("`reviews.rating`") <= 2)

# Filter out NULL and empty values in review_body
df_filtered = df_filtered.filter(col("`reviews.text`").isNotNull())
df_filtered = df_filtered.filter(length(trim(col("`reviews.text`"))) > 0)

# Add Sentiment Column
df_filtered = df_filtered.withColumn(
    "sentiment",
    when(col("`reviews.rating`") >= 4, "positive")
    .when(col("`reviews.rating`") <= 2, "negative")
    .otherwise("neutral")
)

# Remove rows with empty filtered_tokens
df_filtered = df_filtered.filter(size(col("filtered_tokens")) > 0)

from pyspark.sql.functions import col, lower, regexp_replace, when, size, concat_ws, round
from pyspark.ml.feature import Tokenizer, StopWordsRemover, CountVectorizer, HashingTF, IDF, NGram
from pyspark.ml.classification import LogisticRegression, RandomForestClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator


# Data Loading

# Feature Extraction

# The accuracy using CountVectorizer as Feature Extraction gives us 88% accuracy,
# in order to improve the accuracy we used Term-Frequency Feature Extraction

# cv = CountVectorizer(inputCol="filtered_tokens", outputCol="key_phrases", vocabSize=50)
# cv_model = cv.fit(train_data)

# # Transform both training and testing data
# train_transformed = cv_model.transform(train_data)
# test_transformed = cv_model.transform(test_data)

# # Step 3: Train a Classifier
# # Add numeric labels for sentiment
# train_transformed = train_transformed.withColumn(
#     "label",
#     when(col("sentiment") == "positive", 1.0)
#     .when(col("sentiment") == "negative", 0.0)
#     .otherwise(2.0)  # Neutral
# )

# test_transformed = test_transformed.withColumn(
#     "label",
#     when(col("sentiment") == "positive", 1.0)
#     .when(col("sentiment") == "negative", 0.0)
#     .otherwise(2.0)  # Neutral
# )

# Feature Extraction with N-Grams
ngram = NGram(n=2, inputCol="filtered_tokens", outputCol="key_phrases")
df_with_ngrams = ngram.transform(df_filtered)

# TF Feature Extraction
hashing_tf = HashingTF(inputCol="key_phrases", outputCol="tf_features")
tf_data = hashing_tf.transform(df_with_ngrams)

# IDF Feature Extraction
idf = IDF(inputCol="tf_features", outputCol="tfidf_features")
idf_model = idf.fit(tf_data)  # Fit on train data
df_with_features = idf_model.transform(tf_data)

train_data_tfidf, test_data_tfidf = df_with_features.randomSplit([0.8, 0.2], seed=42)

# Add numeric labels for sentiment
train_data_tfidf = train_data_tfidf.withColumn(
    "label",
    when(col("sentiment") == "positive", 1.0)
    .when(col("sentiment") == "negative", 0.0)
    .otherwise(2.0)  # Neutral
)

test_data_tfidf = test_data_tfidf.withColumn(
    "label",
    when(col("sentiment") == "positive", 1.0)
    .when(col("sentiment") == "negative", 0.0)
    .otherwise(2.0)  # Neutral
)

# Train a Logistic Regression Model
lr = LogisticRegression(featuresCol="tfidf_features", labelCol="label")
lr_model = lr.fit(train_data_tfidf)
```

```

# Step 4: Make Predictions
predictions = lr_model.transform(test_data_tfidf)

# Step 5: Evaluate the Model
evaluator = MulticlassClassificationEvaluator(
    labelCol="label", predictionCol="prediction",
    metricName="accuracy"
)
accuracy = evaluator.evaluate(predictions)

# Print Predictions and Accuracy
predictions.select("primaryCategories", "sentiment", "
    prediction").show(10, truncate=False)
print(f"Model Accuracy: {accuracy:.2f}")

from pyspark.sql.functions import concat_ws

# Combine tokens back to sentences
df_filtered = df_filtered.withColumn("summary", concat_ws
    (" ", col("filtered_tokens")))

# Summaries for each sentiment
df_positive_summary = df_filtered.filter(col("sentiment")
     == "positive").select("summary").show(10, truncate=
    False)
df_negative_summary = df_filtered.filter(col("sentiment")
     == "negative").select("summary").show(10, truncate=
    False)

insights = df_filtered.groupBy("primaryCategories", "
    sentiment").count()
insights.show()

from pyspark.sql.functions import col, round

# Group by product_category and calculate total reviews
total_reviews = df_filtered.groupBy("primaryCategories").
    count().withColumnRenamed("count", "total_count")

# Join insights with total_reviews on product_category
insights_with_totals = insights.join(total_reviews, on="
    primaryCategories", how="inner")

# Add the percentage column
sentiment_stats = insights_with_totals.withColumn(
    "percentage",
    round((col("count") / col("total_count")) * 100, 2)
)

# Show the result
sentiment_stats.show()

# Summarized Dataset

from pyspark.sql.functions import concat_ws
from google.colab import files

df_summarized = df_with_ngrams.withColumn("
    summarized_feedback", concat_ws(" ", col("
    filtered_tokens")))
df_summarized = df_summarized.withColumn("key_phrases",
    concat_ws(", ", col("key_phrases")))


# Select required columns: key_phrases,
    summarized_feedback, and sentiment
df_final = df_summarized.select("key_phrases", "
    summarized_feedback", "sentiment")

# Show a sample of the resulting dataset
df_final.show(10, truncate=False)
```