Sentiment Analysis of Product Reviews (COMP3125 Individual Project)

Joseph Lorizio
Wentworth Computer Science

***Abstract*—This project analyzes customer sentiment in Amazon product reviews. I used basic text processing and a simple machine learning model to classify reviews as positive or negative. The analysis showed most reviews (76%) were positive, with different words appearing in good and bad reviews. A basic model was able to predict sentiment with 85% accuracy. This shows how companies can use review analysis to understand customer opinions.**

Keywords*—*sentiment, product, reviews, classification

I. Introduction

Understanding what customers think about products is important for businesses. Online reviews contain valuable information, but it's hard to read thousands of them manually. Sentiment analysis helps by automatically identifying whether reviews are positive or negative.

In this project, I analyzed Amazon food reviews to find patterns in customer opinions. I wanted to see what makes customers happy or unhappy with products. By looking at both the ratings and the words used in reviews, I could identify what customers care about most. This information can help companies improve their products and make customers happier.

II. Datasets

A. Source of dataset

I used the Amazon Fine Food Reviews dataset from Kaggle. This dataset contains customer reviews of food products sold on Amazon from 2002 to 2012. It was collected by researchers at Stanford University for a study about online reviews. The dataset is good for sentiment analysis because it includes both star ratings and written reviews.

B. Character of the datasets

The Amazon Fine Food Reviews dataset is in CSV format and contains 568,454 reviews. It has 10 columns with information about each review.

Feature	**Dataset Attributes**		
	Description	***Type***	***Example***
Id	Review Id	Integer	1
ProductId	Product code	String	B001E42354
UserId	User code	String	A3SG43Z324
ProfileName	User name	String	JoeCool
Score	Rating (1-5)	Integer	5
Summary	Short review	String	“Great taste!”
Text	Full review	String	“I think this product...”

For my analysis, I mainly used the Score and Text columns. I also created a new column called Sentiment based on the Score: reviews with 4-5 stars were labeled as Positive, 3 stars as Neutral, and 1-2 stars as Negative. I cleaned the Text column by converting everything to lowercase, removing special characters, and removing common words like "the" and "and" that don't tell us about sentiment.

III. Methodology

A. Data Cleaning

Before analyzing the reviews, I needed to clean the text. I did this in several steps:

1. Converted all text to lowercase
2. Removed special characters and numbers
3. Split text into individual words
4. Removed common words like "the" and "and"

This cleaning process was necessary because computers can't understand raw text. By cleaning the text, I could focus on the important words that express opinions.

I used the NLTK library in Python to help with this process. NLTK provides tools for working with text data, including a list of common words to remove.

B. Basic Machine Learning with Naive Bayes

To automatically classify reviews as positive or negative, I used a simple machine learning algorithm called Naive Bayes. This algorithm is good for text classification because it's easy to understand and works well even with simple features.

Naive Bayes works by calculating how likely different words are to appear in positive and negative reviews. For example, if the word "delicious" appears mostly in positive reviews during training, the algorithm will predict that new reviews containing "delicious" are probably positive.

I used the scikit-learn library to implement Naive Bayes. The model was trained on 80% of the data and tested on the remaining 20%. I used the CountVectorizer to convert text into vectors that the model could understand, counting how many times each word appears in each review.

C. Word Clouds for Visualization

To see which words were most common in positive and negative reviews, I created word clouds. A word cloud is a picture where the size of each word shows how often it appears in the text. Bigger words appear more frequently.

I created separate word clouds for positive and negative reviews using the WordCloud library in Python. This gave me a visual way to compare the language used in different types of reviews.

IV. Results

A. Count of Reviews: Positive vs. Negative

My first question was about the overall sentiment distribution in the reviews. I found that most reviews were positive:

- Positive reviews (4-5 stars): 76%
- Neutral reviews (3 stars): 10%
- Negative reviews (1-2 stars): 14%

This shows that customers were generally satisfied with the food products they purchased. The high percentage of positive reviews is common in online shopping platforms.

B. Common Words: Positive vs. Negative

My second question looked at the words used in positive and negative reviews. The word clouds showed clear differences: In positive reviews, common words included:

- good
- great
- love
- delicious
- flavor
- taste

In negative reviews, common words included:

- disappointed
- bad
- expected
- waste
- money
- return

These differences show what customers like and dislike. Positive reviews focus on taste and flavor, while negative reviews mention expectations not being met and wasted money.

C. Simple Model Review Prediction

My final question was whether a basic model could predict sentiment from review text. The Naive Bayes model achieved 89% accuracy, which means it correctly identified the sentiment in 89 out of 100 reviews.

The model was better at identifying positive reviews (93% accuracy) than negative reviews (60% accuracy). This difference is probably because there were more positive reviews in the training data.

These results show that even simple machine learning models can effectively analyze sentiment in product reviews.

V. Discussion

While my project showed good results, there are some limitations. First, classifying reviews as just positive or negative is basic. Many reviews contain both positive and negative comments about different aspects of a product.

The way I represented text (by counting words) ignores word order and context, such as phrases like "not good" might be misinterpreted because "not" and "good" are counted separately.

The dataset had many more positive than negative reviews, which might make the model biased toward positive predictions.

In the future, this project could be improved by:

1. Using more advanced text analysis methods that consider word context
2. Looking at specific aspects of products like taste, packaging, price separately from each other
3. Analyzing how reviews change over time
4. Balancing the dataset to have equal numbers of positive and negative reviews

VI. Conclusion

This project showed how sentiment analysis can help understand customer opinions in product reviews. I found that most Amazon food reviews were positive (76%), suggesting customers were generally satisfied. The words used in positive reviews focused on taste and quality, while negative reviews mentioned disappointment and value for money. Negative reviews were also longer on average, showing that unhappy customers write more.

My simple machine learning model was able to predict sentiment with 89% accuracy, which is good for a basic approach. This shows that companies can use automated tools to analyze customer feedback, even without advanced techniques.

These findings can help businesses understand what makes customers happy or unhappy. By analyzing reviews, companies can identify areas for improvement, focus on features customers care about, and address common complaints. This can lead to better products and more satisfied customers.

Acknowledgment

I would like to thank Professor Pang for helping me learn skills used in the project. I also appreciate the Stanford researchers who created the Amazon Fine Food Reviews dataset.

References

[1] [1] J. McAuley and J. Leskovec, "From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews," in Proceedings of the 22nd international conference on World Wide Web, 2013.

[2] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python. O'Reilly Media, 2009.

[3] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, 2011.

[4] Amazon Fine Food Reviews Dataset: https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews