MINISTRY OF EDUCATION | وزارة التعليم
IMAM ABDULRAHMAN BIN | جامعة الإمام
FAISAL UNIVERSITY | عبدالرحمن بن فيصل
COLLEGE OF COMPUTER SCIENCE | كلية علوم الحاسب
& INFORMATION TECHNOLOGY | وتقنيةالمعلومات

جامعة الإمام عبدالرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

ARTI 402 – Programming for AI Project

Academic Year (2024-2025) – First Semester

Project Title:

TextBlob Sentiment Analysis

Team Number: 4

Section Number: 7FA1

Team Members:

No.	Name	ID	Role
1	Raghad Adel Aljiban	2220004952	Leader
2	Shoug Abdulaziz Almoaibed	2220004372	Member
3	Haneen Hasan Alomri	2210003424	Member

Table of Contents

Contents

Table of figures

Introduction

In our modern era with rapid development of the Internet, e-commerce has become more common, dominant and reliable. As data collection and processing is faster using applications and techniques, making decisions faster and more accurate outcomes.
Customer reviews represent a critical dataset, as they provide valuable perspectives into a product's quality and performance, aiding prospective buyers in making informed choices. Moreover, these reviews hold substantial significance for business owners, by collecting and analyzing reviews and trying to isolate the negatives and positives of the product, this targeted analysis helps manufacturers to know all aspects of their product and gain a comprehensive understanding.

In this report we will handle a case using sentiment analysis, which is a specialized technique in natural language processing, to extract meaningful insights from customer reviews. Sentiment analysis plays a pivotal role in processing and interpreting the emotional tone of textual data, offering a systematic approach to isolating positive and negative aspects. Using a dedicated Python library for sentiment analysis, we aim to demonstrate its utility in analyzing customer feedback to inform product development and business strategies.

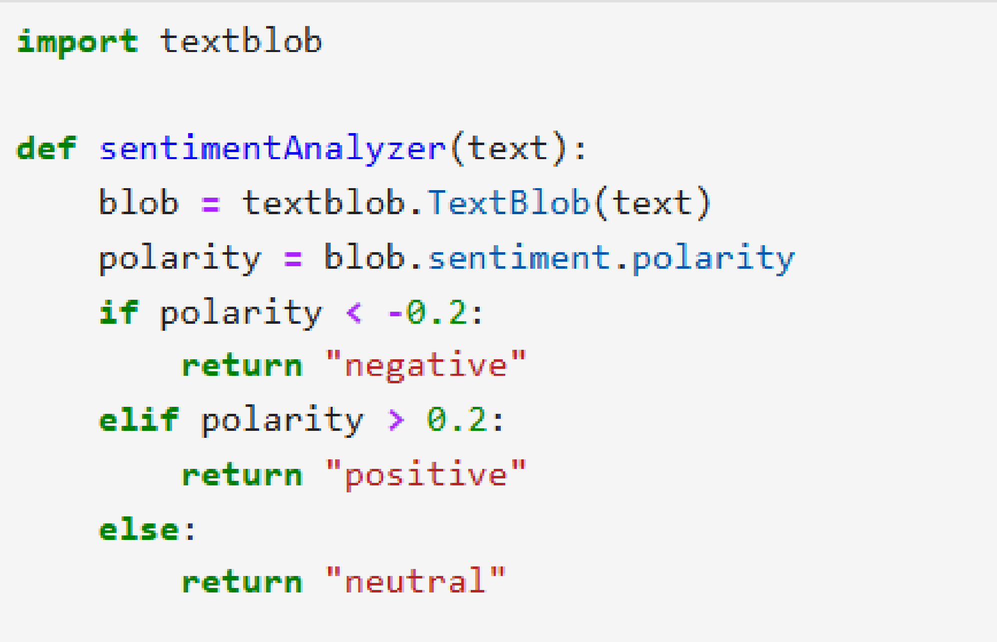
To leverage these insights, we will create a basic sentiment analysis tool using TextBlob. TextBlob, a widely used Python library for natural language processing (NLP), simplifies the complexities of handling textual data.

Problem Statement

Vendors on websites such as Amazon frequently get a large number of reviews. These evaluations offer important insights on product quality, customer happiness, and product enhancement process. Nonetheless, it is extremely difficult to analyze such massive amounts of data in an efficient way without the use of an automated system. Hence, it is crucial to utilize a tool that would assist effectively in managing this substantial and fluctuating volume of data. The sentiment analysis tool will carry out by categorizing reviews and analyzing them to better understand their content.

AI Solution

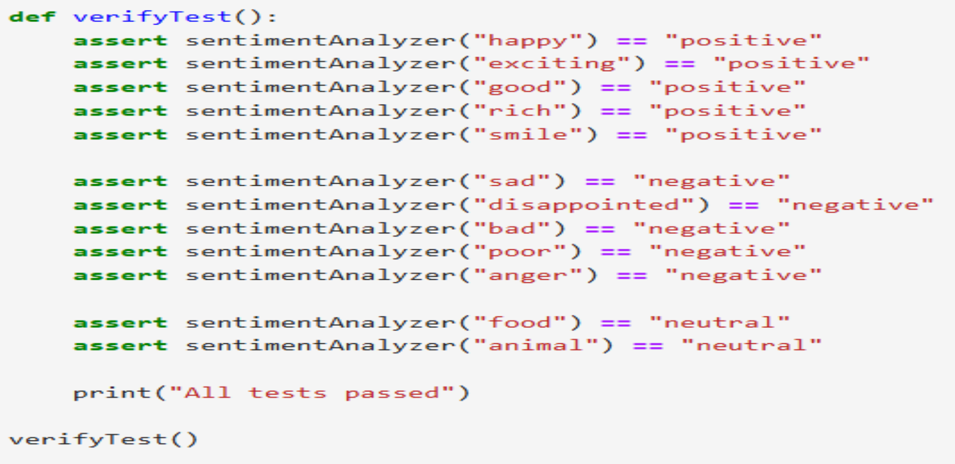
We chose jupyter to illustrate our code and use TextBlob to provide a straightforward AI solution to the issue we had stated. In order to comprehend and make use of TextBlob, we developed a straightforward function that determines polarity and then assigns a certain string according to it, as seen below:



```python
import textblob

def sentimentAnalyzer(text):
    blob = textblob.TextBlob(text)
    polarity = blob.sentiment.polarity
    if polarity < -0.2:
        return "negative"
    elif polarity > 0.2:
        return "positive"
    else:
        return "neutral"
```

Figure 1: Sentiment Analysis Function.

After receiving a text, this function will examine its polarity. It will then determine if this text is neutral, positive, or negative. We used the following test to confirm the accuracy of this function. Function:



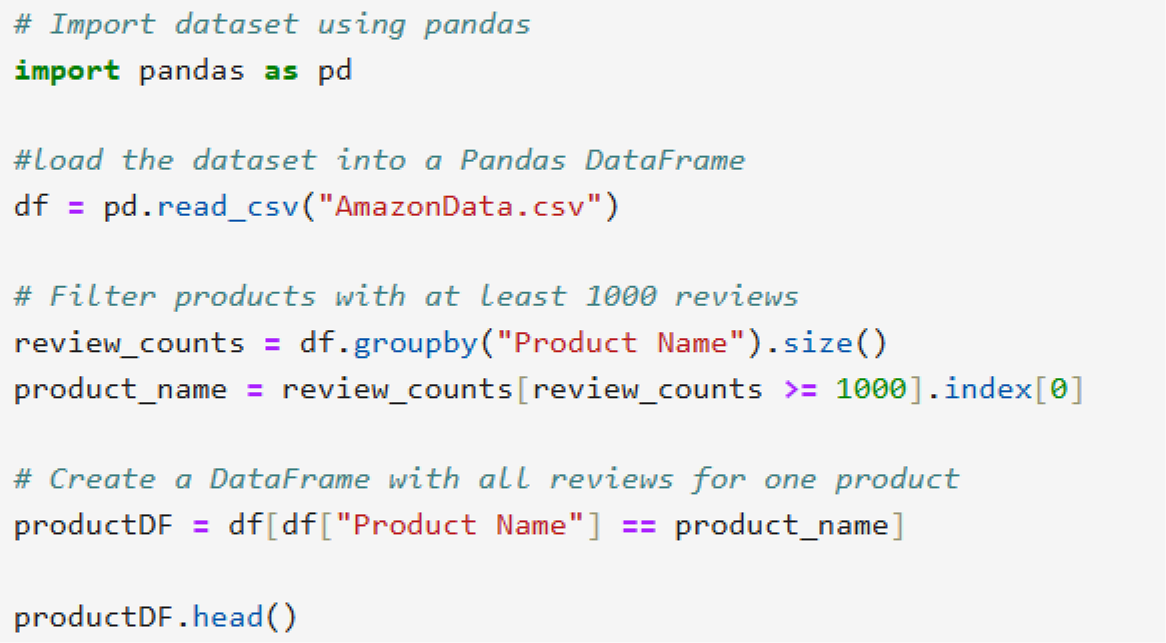
```
def verifyTest():
    assert sentimentAnalyzer("happy") == "positive"
    assert sentimentAnalyzer("exciting") == "positive"
    assert sentimentAnalyzer("good") == "positive"
    assert sentimentAnalyzer("rich") == "positive"
    assert sentimentAnalyzer("smile") == "positive"

    assert sentimentAnalyzer("sad") == "negative"
    assert sentimentAnalyzer("disappointed") == "negative"
    assert sentimentAnalyzer("bad") == "negative"
    assert sentimentAnalyzer("poor") == "negative"
    assert sentimentAnalyzer("anger") == "negative"

    assert sentimentAnalyzer("food") == "neutral"
    assert sentimentAnalyzer("animal") == "neutral"

    print("All tests passed")

verifyTest()
```

Figure 2:Test function to ensure sentiment analysis is accurate.

This guarantees the effective operation of our sentiment analysis. An extremely effective method of examining many Amazon customer reviews is to use TextBlob. In the use case section that follows, we will go into greater depth about this.

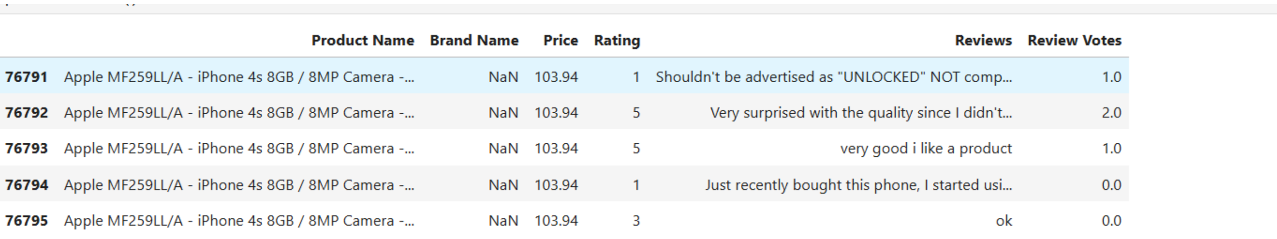
Use Case

We made the decision to separate our application to concentrate on a single product: Apple MF259LL/A – iphone 4s 8GB / 8MP. We did this by using a vast amount of data collection and several customer reviews based on other goods. As can be seen below, we identified the product's allocated ID after examining the provided data and then constructed a data frame to represent just that product:



```python
# Import dataset using pandas
import pandas as pd

#Load the dataset into a Pandas DataFrame
df = pd.read_csv("AmazonData.csv")

# Filter products with at least 1000 reviews
review_counts = df.groupby("Product Name").size()
product_name = review_counts[review_counts >= 1000].index[0]

# Create a DataFrame with all reviews for one product
productDF = df[df["Product Name"] == product_name]

productDF.head()
```

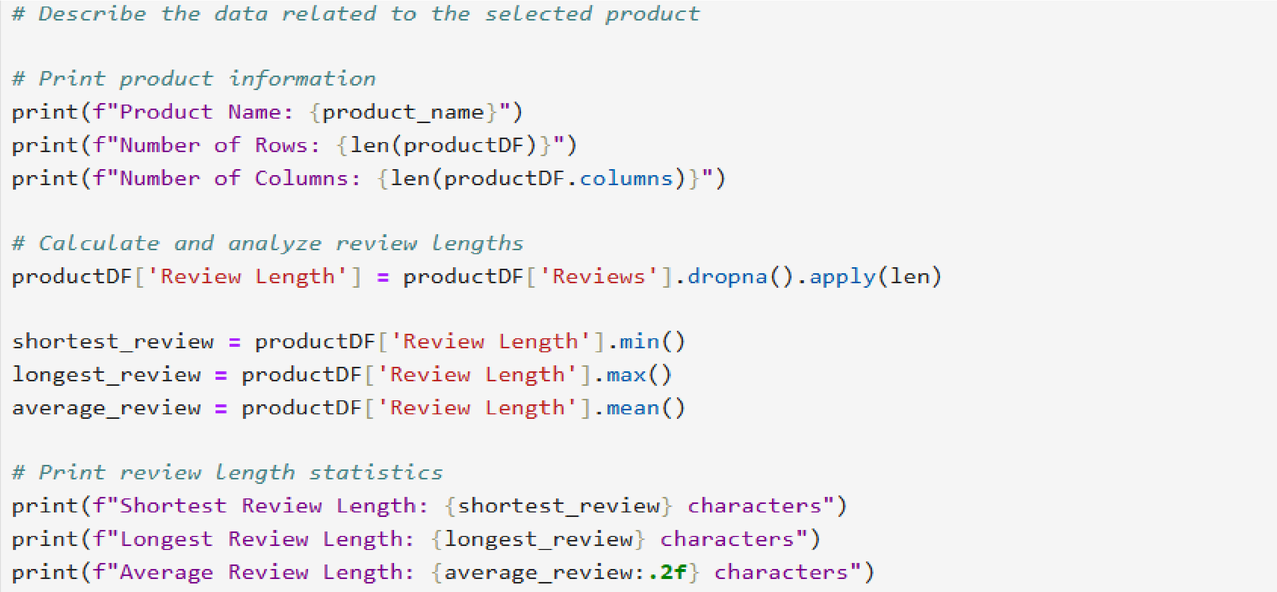
Figure 3:Read the data from the collection then isolate the product we need.



	Product Name	Brand Name	Price	Rating	Reviews	Review Votes
76791	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	1	Shouldn't be advertised as "UNLOCKED" NOT comp...	1.0
76792	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	5	Very surprised with the quality since I didn't...	2.0
76793	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	5	very good i like a product	1.0
76794	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	1	Just recently bought this phone, I started usi...	0.0
76795	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	3	ok	0.0

Figure 4:Table head of result.

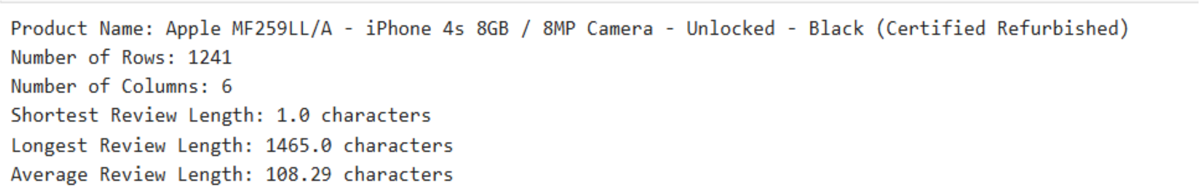
The first thing we did before applying sentiment analysis was describe the current data we have. So we use the following code:



```python
# Describe the data related to the selected product

# Print product information
print(f"Product Name: {product_name}")
print(f"Number of Rows: {len(productDF)}")
print(f"Number of Columns: {len(productDF.columns)}")

# Calculate and analyze review lengths
productDF['Review Length'] = productDF['Reviews'].dropna().apply(len)

shortest_review = productDF['Review Length'].min()
longest_review = productDF['Review Length'].max()
average_review = productDF['Review Length'].mean()

# Print review length statistics
print(f"Shortest Review Length: {shortest_review} characters")
print(f"Longest Review Length: {longest_review} characters")
print(f"Average Review Length: {average_review:.2f} characters")
```

Figure 5:Code to describe the data we have isolated.

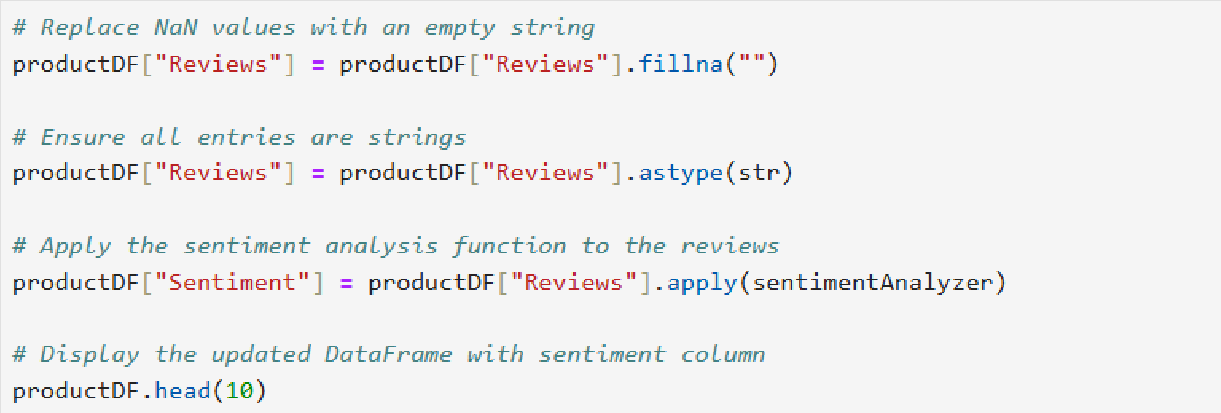
This resulted in the following output:



```
Product Name: Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera - Unlocked - Black (Certified Refurbished)
Number of Rows: 1241
Number of Columns: 6
Shortest Review Length: 1.0 characters
Longest Review Length: 1465.0 characters
Average Review Length: 108.29 characters
```

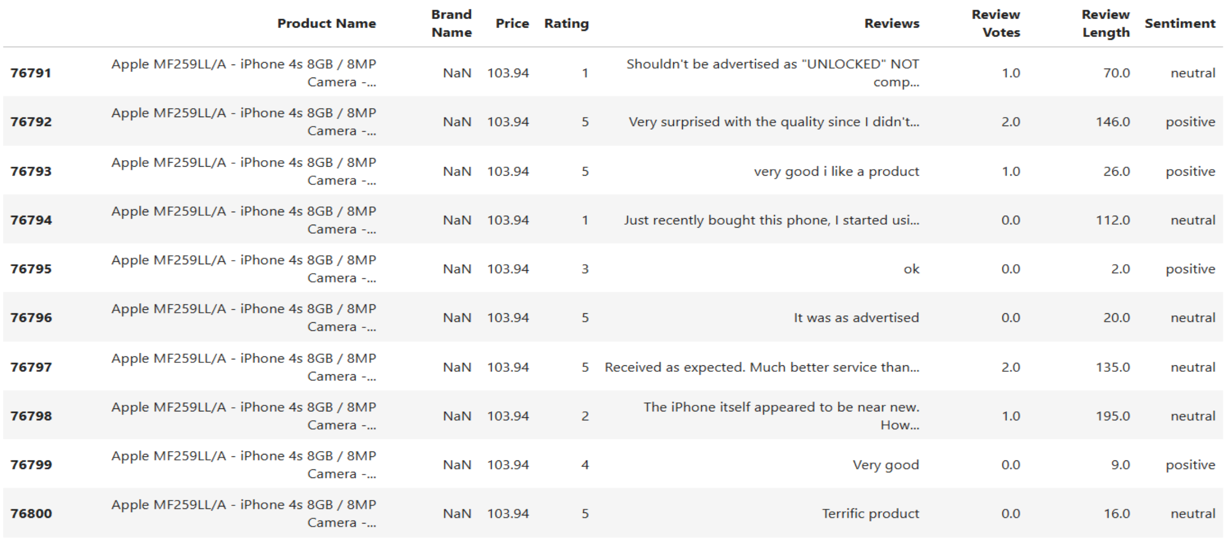
Figure 6:Output describing the isolated data.

We can now use sentiment analysis and a variety of graphing techniques to observe how it affects the app. Initially, sentiment analysis is applied to every review and added as an additional column to the data frame. The code and outcome are displayed below:



```python
# Replace NaN values with an empty string
productDF["Reviews"] = productDF["Reviews"].fillna("")

# Ensure all entries are strings
productDF["Reviews"] = productDF["Reviews"].astype(str)

# Apply the sentiment analysis function to the reviews
productDF["Sentiment"] = productDF["Reviews"].apply(sentimentAnalyzer)

# Display the updated DataFrame with sentiment column
productDF.head(10)
```

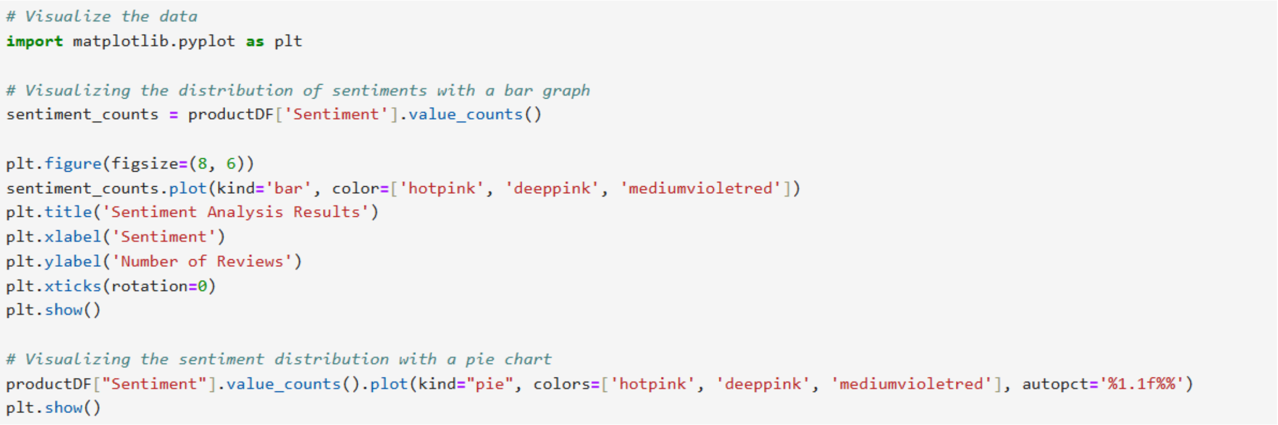
Figure 7:Code to apply sentiment analysis to all reviews and add it to a sentiment column.



	Product Name	Brand Name	Price	Rating	Reviews	Review Votes	Review Length	Sentiment
76791	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	1	Shouldn't be advertised as "UNLOCKED" NOT comp...	1.0	70.0	neutral
76792	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	5	Very surprised with the quality since I didn't...	2.0	146.0	positive
76793	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	5	very good i like a product	1.0	26.0	positive
76794	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	1	Just recently bought this phone, I started usi...	0.0	112.0	neutral
76795	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	3	ok	0.0	2.0	positive
76796	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	5	It was as advertised	0.0	20.0	neutral
76797	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	5	Received as expected. Much better service than...	2.0	135.0	neutral
76798	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	2	The iPhone itself appeared to be near new. How...	1.0	195.0	neutral
76799	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	4	Very good	0.0	9.0	positive
76800	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	5	Terrific product	0.0	16.0	neutral

Figure 8:Resulting DataFrame.

Examining the head that everyone feels positively, but that is insufficient. To determine why certain reviews are negative, we must use sentiment analysis to separate out the unfavorable ones. First, we need to use some plots to represent all the neutral, negative, and positive sentiments.



```python
# Visualize the data
import matplotlib.pyplot as plt

# Visualizing the distribution of sentiments with a bar graph
sentiment_counts = productDF['Sentiment'].value_counts()

plt.figure(figsize=(8, 6))
sentiment_counts.plot(kind='bar', color=['hotpink', 'deeppink', 'mediumvioletred'])
plt.title('Sentiment Analysis Results')
plt.xlabel('Sentiment')
plt.ylabel('Number of Reviews')
plt.xticks(rotation=0)
plt.show()

# Visualizing the sentiment distribution with a pie chart
productDF["Sentiment"].value_counts().plot(kind="pie", colors=['hotpink', 'deeppink', 'mediumvioletred'], autopct='%1.1f%%')
plt.show()
```

Figure 9:Visual results code.

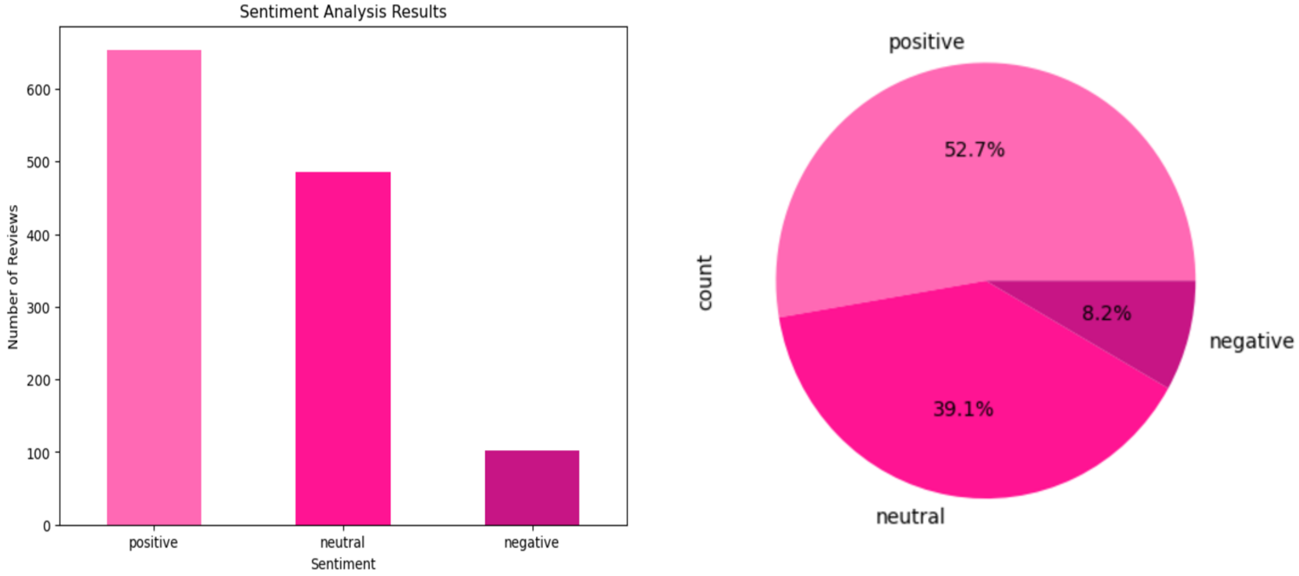
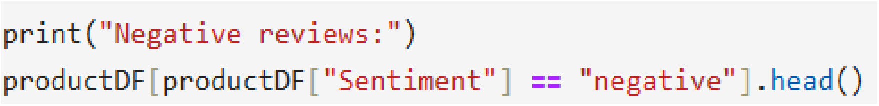


Figure 10:Graphs show positive, negative, and neutral sentiment.

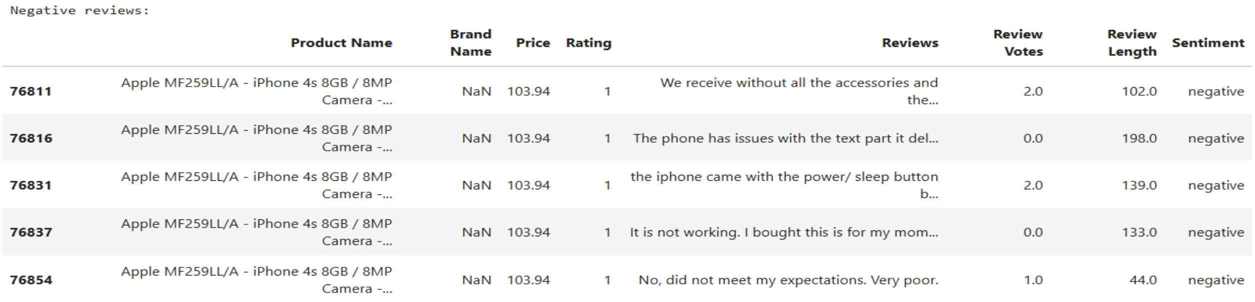
Majority reviews have been part of positive sentiments. This shows good quality and less negative reviews of the people.

These graphs demonstrate that most reviews are good, with a few negative exceptions. To analyze this further, we need to isolate the negative reviews and learn why their sentiment is negative. The code below will print the first 5 of each.



```
print("Negative reviews:")
productDF[productDF["Sentiment"] == "negative"].head()
```

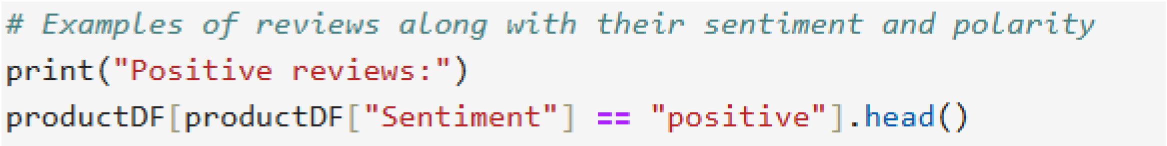
Figure 11:Code to isolate negative reviews.



```
Negative reviews:
```

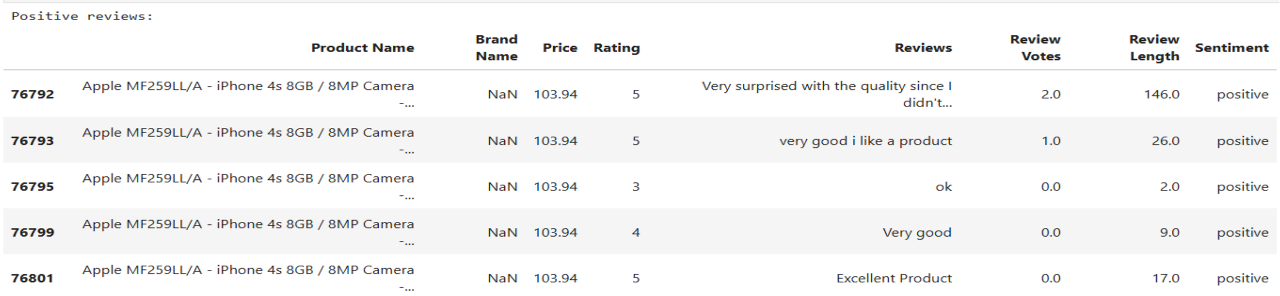
	Product Name	Brand Name	Price	Rating	Reviews	Review Votes	Review Length	Sentiment
76811	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	1	We receive without all the accessories and the...	2.0	102.0	negative
76816	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	1	The phone has issues with the text part it del...	0.0	198.0	negative
76831	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	1	the iphone came with the power/ sleep button b...	2.0	139.0	negative
76837	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	1	It is not working. I bought this is for my mom...	0.0	133.0	negative
76854	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	1	No, did not meet my expectations. Very poor.	1.0	44.0	negative

Figure 12:Output of code to isolate negative reviews.



```python
# Examples of reviews along with their sentiment and polarity
print("Positive reviews:")
productDF[productDF["Sentiment"] == "positive"].head()
```

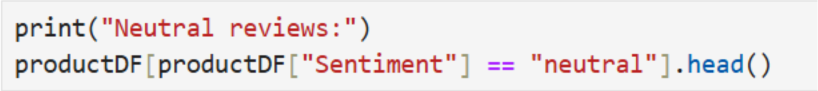
Figure 13:Code to isolate positive reviews



```
Positive reviews:
```

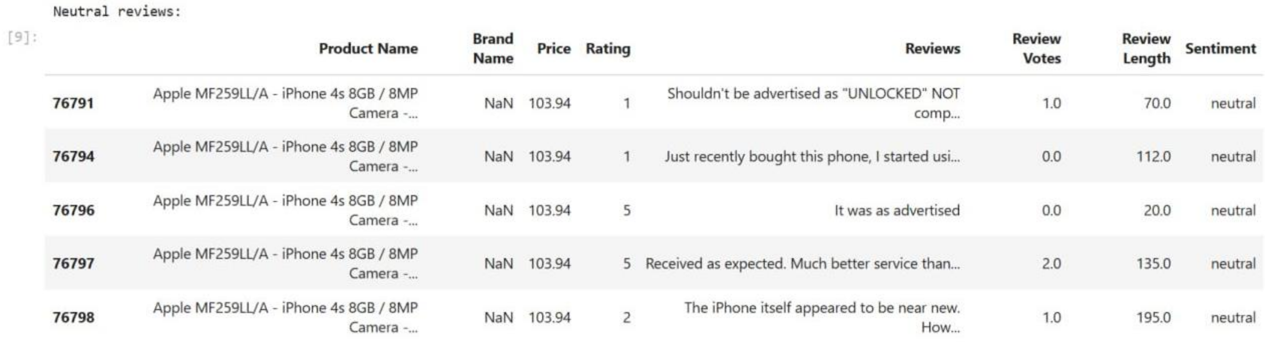
	Product Name	Brand Name	Price	Rating	Reviews	Review Votes	Review Length	Sentiment
76792	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	5	Very surprised with the quality since I didn't...	2.0	146.0	positive
76793	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	5	very good i like a product	1.0	26.0	positive
76795	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	3	ok	0.0	2.0	positive
76799	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	4	Very good	0.0	9.0	positive
76801	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	5	Excellent Product	0.0	17.0	positive

Figure 14:Output of code to isolate positive reviews.



```python
print("Neutral reviews:")
productDF[productDF["Sentiment"] == "neutral"].head()
```

Figure 15:Code to isolate neutral reviews.

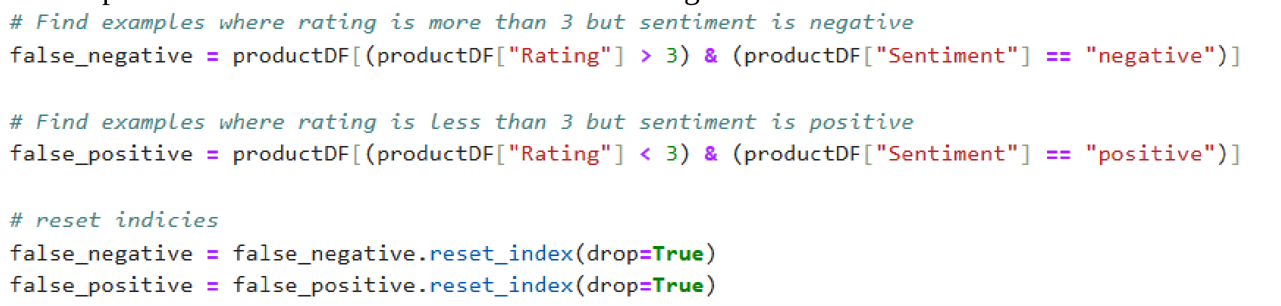


```
Neutral reviews:
[9]:
```

	Product Name	Brand Name	Price	Rating	Reviews	Review Votes	Review Length	Sentiment
76791	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	1	Shouldn't be advertised as "UNLOCKED" NOT comp...	1.0	70.0	neutral
76794	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	1	Just recently bought this phone, I started usi...	0.0	112.0	neutral
76796	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	5	It was as advertised	0.0	20.0	neutral
76797	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	5	Received as expected. Much better service than...	2.0	135.0	neutral
76798	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	2	The iPhone itself appeared to be near new. How...	1.0	195.0	neutral

Figure 16:Output of code to isolate neutral reviews.

First, we needed to isolate the false positives and the false negatives. The polarity score from these represents the overview of sentiments and strength distribution:



```
# Find examples where rating is more than 3 but sentiment is negative
false_negative = productDF[(productDF["Rating"] > 3) & (productDF["Sentiment"] == "negative")]

# Find examples where rating is less than 3 but sentiment is positive
false_positive = productDF[(productDF["Rating"] < 3) & (productDF["Sentiment"] == "positive")]

# reset indicies
false_negative = false_negative.reset_index(drop=True)
false_positive = false_positive.reset_index(drop=True)
```

Figure 17:Code to isolate false positives and false negatives.

To analyze client strategies for minimizing negative sentiment and improving reputation and product quality using a dataset, consider the following insights that can be derived from the data:

1. **Sentiment Analysis**:

Sort consumer reviews into three categories using sentiment analysis tools: neutral, negative, and positive. To find times when negative sentiment has increased, look at trends across time. Then examine unfavorable remarks to find recurring themes or problems (such as product flaws or concerns about customer service).

2. **Customer Feedback Metrics**:

To determine customer satisfaction, monitor CSAT scores after a purchase or service encounter.

3. **Product Performance Data**:

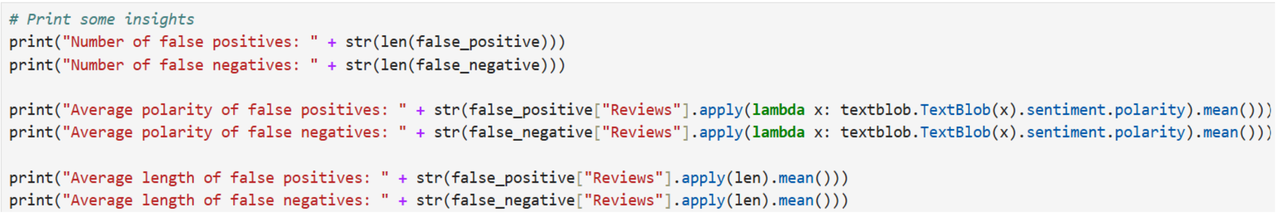
Examine product return and refund to determine which products are performing poorly and might be contributing to negative sentiment.

4. **Competitive Analysis**:

Look for areas where the brand might be falling behind by comparing sentiment and consumer feedback with those of competitors.

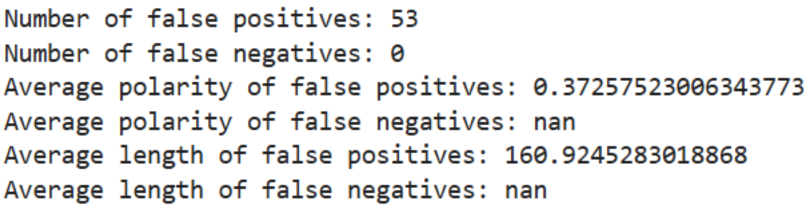
5. **Product Development Insights**:

Look for frequently asked questions or needed features that are absent from the present offers by analyzing consumer feedback.



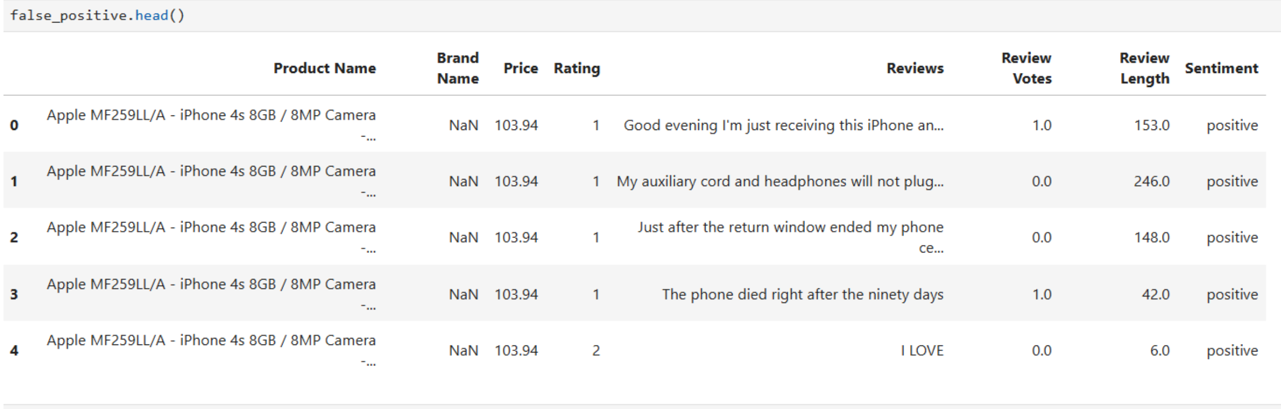
```
# Print some insights
print("Number of false positives: " + str(len(false_positive)))
print("Number of false negatives: " + str(len(false_negative)))

print("Average polarity of false positives: " + str(false_positive["Reviews"].apply(lambda x: textblob.TextBlob(x).sentiment.polarity).mean()))
print("Average polarity of false negatives: " + str(false_negative["Reviews"].apply(lambda x: textblob.TextBlob(x).sentiment.polarity).mean()))

print("Average length of false positives: " + str(false_positive["Reviews"].apply(len).mean()))
print("Average length of false negatives: " + str(false_negative["Reviews"].apply(len).mean()))
```

Figure 18:Analyzing false positives and false negatives.



```
Number of false positives: 53
Number of false negatives: 0
Average polarity of false positives: 0.37257523006343773
Average polarity of false negatives: nan
Average length of false positives: 160.9245283018868
Average length of false negatives: nan
```

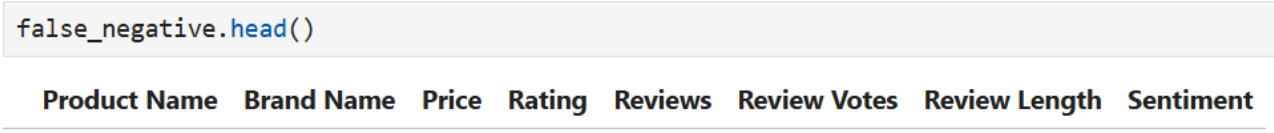
Figure 19:Output for the false flags analysis.



```
false_positive.head()
```

	Product Name	Brand Name	Price	Rating	Reviews	Review Votes	Review Length	Sentiment
0	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	1	Good evening I'm just receiving this iPhone an...	1.0	153.0	positive
1	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	1	My auxiliary cord and headphones will not plug...	0.0	246.0	positive
2	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	1	Just after the return window ended my phone ce...	0.0	148.0	positive
3	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	1	The phone died right after the ninety days	1.0	42.0	positive
4	Apple MF259LL/A - iPhone 4s 8GB / 8MP Camera -...	NaN	103.94	2	I LOVE	0.0	6.0	positive

Figure 20:Output of false positive.

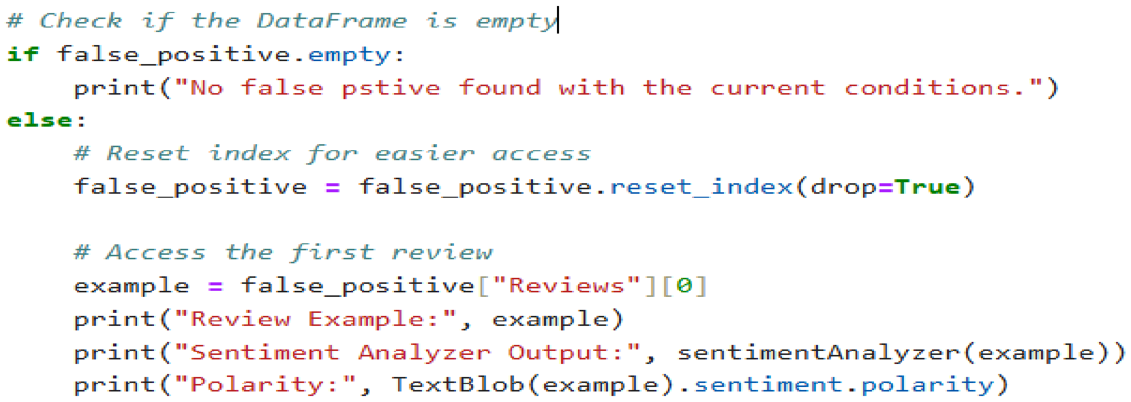


```
false_negative.head()
```

Product Name	Brand Name	Price	Rating	Reviews	Review Votes	Review Length	Sentiment

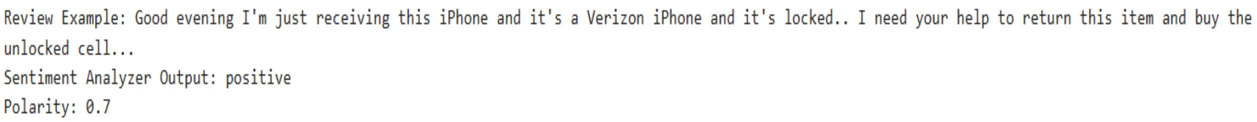
Figure 21:Output of false negative.

The false flags, as you can see, are a vast minority. But we have to discover the reason, so we choose to examine one such instance. We used the code below to accomplish this:



```
# Check if the DataFrame is empty
if false_positive.empty:
    print("No false pstive found with the current conditions.")
else:
    # Reset index for easier access
    false_positive = false_positive.reset_index(drop=True)

    # Access the first review
    example = false_positive["Reviews"][0]
    print("Review Example:", example)
    print("Sentiment Analyzer Output:", sentimentAnalyzer(example))
    print("Polarity:", TextBlob(example).sentiment.polarity)
```

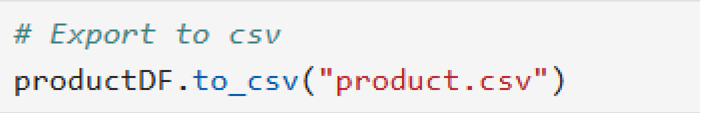
Figure 22: Code to analyze one false positive review.



```
Review Example: Good evening I'm just receiving this iPhone and it's a Verizon iPhone and it's locked.. I need your help to return this item and buy the
unlocked cell...
Sentiment Analyzer Output: positive
Polarity: 0.7
```

Figure 23: Output of the code.

Without knowing the whole context, sentiment analysis frequently relies on words and phrases. The initial greeting "Good evening" can create an impression of friendliness, which might lead the algorithm to interpret the overall sentiment more positively. But request for help which stems from a bad experience (receiving a locked phone). This complexity may not be handled well by it. Consequently, marking it as a positive sentiment when it should be negative.



```
# Export to csv
productDF.to_csv("product.csv")
```

Figure 24: Export to csv

Alignment Between AI Solution and Client’s Needs

The sentiment analysis tool created with the TextBlob library provides an accessible and effective way for analyzing customer reviews. This tool guarantees reliability and precision in sentiment classification, offering clients clear and actionable insights into customer feedback. By comprehending customer preferences and sentiments, decision-makers can pinpoint critical areas for product enhancement, effectively handle issues, and elevate overall customer pleasure. Moreover, utilizing these insights fosters client trust and loyalty while enhancing product development and efficiency. The solution effectively meets the client's requirement to assess reviews rapidly and make data-driven decisions to enhance their offers and reputation.

Conclusion

Briefly, building a sentiment analysis tool delivers enormous value and better way to express customer's reviews. The technology categorizes reviews as positive, negative, or neutral, as it derives clear insights into customer opinions and helps vendors in recognizing their strengths and locations for enhancement and growth. This study enhances decision-making by getting full knowledge in all aspects of their products that customers need, making organizations facilitate addressing issues and refine their goods.

By automating the review analysis process, vendors save time and resources while acquiring vital insights that enhance customer happiness and bolster brand reputation. This technology directly meets the client's requirements, providing a pragmatic approach to managing and comprehending consumer feedback for informed enhancements and fostering trust with their audience.