Customer Sentiment Analysis - Flipkart

Objective

As a Data Analyst at Flipkart, you have been tasked with gauging customer sentiment towards the iPhone 16 128GB model. The primary goal of this project is to analyze public perception and evaluate customer reactions by performing sentiment analysis on product reviews posted by users. By extracting and processing customer reviews, you will derive insights about the overall sentiment (positive or negative) surrounding the product, which can be useful for decision-making, improving customer experience, and identifying key areas for product improvement.

Tasks

1. Data Collection (Web Scraping)

Tool: Selenium and BeautifulSoup

Task: Scrape at least 300 customer reviews from Flipkart's product page for the iPhone 16 128GB model. Each review should include:

- **Username:** The name of the reviewer.
- **City:** The city of the reviewer.
- **Date:** The date of purchase of the reviewer.
- **Rating:** The rating provided by the user (1 to 5 stars).
- **Review Text:** The content of the customer's review, which may contain valuable information regarding their experience with the product.

Steps:

1. **Set Up Selenium:**
    - Automate browser interactions and navigate to Flipkart’s product page for the iPhone 16 128GB.
2. **Extract Reviews with BeautifulSoup:**
    - Parse the HTML of the reviews using BeautifulSoup and extract the relevant details:
        - **Username**
        - **City**
        - **Date**
        - **Rating**
        - **Review Text**
3. **Handle Pagination:**
    - Ensure the scraper is capable of navigating through multiple pages to retrieve a comprehensive dataset of at least 300 reviews.

In [1]:

```python
# Import the necessary librariess
import requests
import time
import pandas as pd
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
```

In [26]:

```python
# Create empty lists to store the user data such as Name, City, Date of Purchase, Review & Rating
Names = []
Cities = []
Dates = []
Reviews = []
Ratings = []

# Assign the url of the flipkart website and use selenium to scrape data
url = """https://www.flipkart.com/apple-iphone-16-white-128-gb/product-reviews/itm7c0281cd247be?pid=MOBH4DQF849
driver = webdriver.Chrome()
driver.get(url)


while len(Names) < 320:

    time.sleep(2)
    soup = BeautifulSoup(driver.page_source, "html.parser")
    # Scrape names
    temp_names = soup.find_all("p", {"class": "_2NsDsF AwS1CA"})
    for name in temp_names:
        Names.append(name.text)

    # Scrape cities
    temp_cities = soup.find_all("p", {"class": "MztJPv"})
    for city in temp_cities:
        Cities.append(city.text)

    # Scrape dates
    temp_dates = soup.find_all("p", {"class": "_2NsDsF"})
    for date in temp_dates:
        Dates.append(date.text)
    Actual_Dates = Dates[1::2]

    # Scrape reviews
    temp_reviews = soup.find_all("div", {"class": "ZmyHeo"})
    for review in temp_reviews:
        Reviews.append(review.text)

    # Scrape ratings
    temp_ratings = soup.find_all("div", class_ = "XQDdHH Ga3i8K")
    for ratings in temp_ratings:
        Ratings.append(ratings.text)

    # Try to click the "Next" button
    try:
        next_button = driver.find_element(By.XPATH, "//span[text()='Next']")
        next_button.click()
        time.sleep(5)
    except:
        break
```

In [24]:

```
# Combine data into a DataFrame
data = pd.DataFrame({
    "Name": Names[:-1],
    "City": Cities[:-1],
    "Date": Actual_Dates[:-1],
    "Review": Reviews[:-1],
    "Ratings": Ratings[:-1]
})

# Save to a CSV files
data.to_csv("flipkart_reviews.csv", index=False)
```

2. Data Cleaning and Preprocessing

Tool: Pandas

Task: Clean and preprocess the scraped data for analysis.

Steps:

1. **Remove Duplicates:**

   - Eliminate any duplicate reviews to ensure data quality.

2. **Handle Missing Values:**

   - Address missing or incomplete data, such as missing review text or rating, by either:
     - Removing rows with missing values.
     - Filling in missing values if applicable.

3. **Text Preprocessing:**

   - **Convert to Lowercase:** Standardize the text by converting all review text to lowercase.
   - **Remove Irrelevant Characters:** Strip out special characters, punctuation, and extra spaces.

In [27]:

```
# Assign the scraped dataset(csv file) to a dataframe
new_data = pd.read_csv('flipkart_reviews.csv')
new_data
```

Out[27]:

	Name	City	Date	Review	Ratings
0	Aryan Kumar	Certified Buyer, Ranchi	4 months ago	Loved itREAD MORE	5
1	Flipkart Customer	Certified Buyer, Uttara Kannada District	5 months ago	Super and cool photoREAD MORE	5
2	Amit Kumar Chaudhary	Certified Buyer, Patna	2 months ago	SatisfiedREAD MORE	5
3	Ravi Arya	Certified Buyer, Gangoh	5 months ago	I love it ❤❤READ MORE	5
4	Disha Singh	Certified Buyer, Agartala	2 months ago	Doing good as of now!READ MORE	5
...	...	...	...	...	...
64	Rajat Rai	Certified Buyer, Maunath Bhanjan	9 months ago	More smoothly then 15READ MORE	5
65	jayata banerjee	Certified Buyer, Asansol	5 months ago	Amazing device, everything is top notchREAD MORE	5
66	Yograj Meena	Certified Buyer, Mathura	6 months ago	I had previously used OnePlus 8 pro ,Samsung ...	5
67	Shikha Swarup	Certified Buyer, Sonbhadra District	9 months ago	Upgraded from iPhone 14 Pro Max loved the new ...	5
68	Rishab Guleria	Certified Buyer, Ghumarwin	9 months ago	Switched from 12 mini to iPhone 16 - battery 1...	5

69 rows × 5 columns

In [5]:

```
# Check the basic info of the dataframe
new_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 69 entries, 0 to 68
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   Name     69 non-null     object
 1   City     69 non-null     object
 2   Date     69 non-null     object
 3   Review   69 non-null     object
 4   Ratings  69 non-null     int64 
dtypes: int64(1), object(4)
memory usage: 2.8+ KB
```

In [6]:

```
# Check valye counts of the Name column
new_data['Name'].value_counts()
```

Out[6]:

```
Name
Flipkart Customer       4
Aryan Kumar             1
Amit Kumar Chaudhary    1
Ravi Arya               1
Disha Singh             1
                       ..
Rajat  Rai              1
jayata banerjee         1
Yograj Meena            1
Shikha Swarup           1
Rishab Guleria          1
Name: count, Length: 66, dtype: int64
```

In [7]:

```
# Drop the duplicates from the dataframe
new_data = new_data.copy()
new_data = new_data.drop_duplicates()
new_data
```

Out[7]:

	Name	City	Date	Review	Ratings
0	Aryan Kumar	Certified Buyer, Ranchi	4 months ago	Loved itREAD MORE	5
1	Flipkart Customer	Certified Buyer, Uttara Kannada District	5 months ago	Super and cool photoREAD MORE	5
2	Amit Kumar Chaudhary	Certified Buyer, Patna	2 months ago	SatisfiedREAD MORE	5
3	Ravi Arya	Certified Buyer, Gangoh	5 months ago	I love it ❤❤READ MORE	5
4	Disha Singh	Certified Buyer, Agartala	2 months ago	Doing good as of now!READ MORE	5
...	...	...	...	...	...
64	Rajat Rai	Certified Buyer, Maunath Bhanjan	9 months ago	More smoothly then 15READ MORE	5
65	jayata banerjee	Certified Buyer, Asansol	5 months ago	Amazing device, everything is top notchREAD MORE	5
66	Yograj Meena	Certified Buyer, Mathura	6 months ago	I had previously used OnePlus 8 pro ,Samsung ...	5
67	Shikha Swarup	Certified Buyer, Sonbhadra District	9 months ago	Upgraded from iPhone 14 Pro Max loved the new ...	5
68	Rishab Guleria	Certified Buyer, Ghumarwin	9 months ago	Switched from 12 mini to iPhone 16 - battery 1...	5

69 rows × 5 columns

In [8]:

```
# Convert the Name column data into Title Case
new_data['Name'] = new_data['Name'].str.title()
new_data.head()
```

Out[8]:

	Name	City	Date	Review	Ratings
0	Aryan Kumar	Certified Buyer, Ranchi	4 months ago	Loved itREAD MORE	5
1	Flipkart Customer	Certified Buyer, Uttara Kannada District	5 months ago	Super and cool photoREAD MORE	5
2	Amit Kumar Chaudhary	Certified Buyer, Patna	2 months ago	SatisfiedREAD MORE	5
3	Ravi Arya	Certified Buyer, Gangoh	5 months ago	I love it ❤❤READ MORE	5
4	Disha Singh	Certified Buyer, Agartala	2 months ago	Doing good as of now!READ MORE	5

In [9]:

```
# Clean data of City column by removing unwanted characters/ part of string
new_data['City'] = new_data['City'].str.replace("Certified Buyer, ", "", regex=False).str.strip()
new_data.head()
```

Out[9]:

	Name	City	Date	Review	Ratings
0	Aryan Kumar	Ranchi	4 months ago	Loved itREAD MORE	5
1	Flipkart Customer	Uttara Kannada District	5 months ago	Super and cool photoREAD MORE	5
2	Amit Kumar Chaudhary	Patna	2 months ago	SatisfiedREAD MORE	5
3	Ravi Arya	Gangoh	5 months ago	I love it ❤❤READ MORE	5
4	Disha Singh	Agartala	2 months ago	Doing good as of now!READ MORE	5

In [10]:

```
 #Clean data of Review column by removing unwanted characters/ part of string and converting to lowercase
new_data['Review'] = new_data['Review'].str.lower().str.replace("read more", "", regex=False)
new_data.head()
```

Out[10]:

	Name	City	Date	Review	Ratings
0	Aryan Kumar	Ranchi	4 months ago	loved it	5
1	Flipkart Customer	Uttara Kannada District	5 months ago	super and cool photo	5
2	Amit Kumar Chaudhary	Patna	2 months ago	satisfied	5
3	Ravi Arya	Gangoh	5 months ago	i love it ❤❤	5
4	Disha Singh	Agartala	2 months ago	doing good as of now!	5

3. Sentiment Analysis

Tool: TextBlob

Task: Analyze the sentiment of each review to classify them as either positive or negative.

Steps:

1. **Perform Sentiment Analysis:**
    - Use `TextBlob` to analyze the sentiment of each review text.
    - Extract the following scores from `TextBlob` :
        - **Polarity:** A score between -1 (negative) and +1 (positive).
        - **Subjectivity:** A measure of how subjective or objective the text is.
2. **Define Sentiment Thresholds:**
    - **Extremely Positive Sentiment:** Polarity score > 0.75
    - **Positive Sentiment:** 0 < Polarity score <= 0.75
    - **Neutral Sentiment:** Polarity score = 0
    - **Negative Sentiment:** -0.75 <= Polarity score < 0
    - **Extremely Negative Sentiment:** Polarity score < -0.75
3. **Store Sentiment Classification:**
    - Add a new column in the dataset to store the sentiment classification for each review as either **Extremely Positive**, **Positive**, **Neutral**, **Negative** or **Extremely Negative**.

In [11]:

```
# Import libraries for Sentimental analysis of review sentences
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import sent_tokenize
from nltk.tokenize import word_tokenize
from textblob import TextBlob
import string
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('punkt_tab')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\vinod\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\vinod\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\vinod\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package punkt_tab to
[nltk_data]     C:\Users\vinod\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
```

Out[11]:

```
True
```

In [12]:

```
# Create a column called Reviews_t that stores tokenized sentences from the Review column using the sent_tokeni
new_data["Reviews_t"] = new_data['Review'].apply(sent_tokenize)
new_data
```

Out[12]:

	Name	City	Date	Review	Ratings	Reviews_t
0	Aryan Kumar	Ranchi	4 months ago	loved it	5	[loved it]
1	Flipkart Customer	Uttara Kannada District	5 months ago	super and cool photo	5	[super and cool photo]
2	Amit Kumar Chaudhary	Patna	2 months ago	satisfied	5	[satisfied]
3	Ravi Arya	Gangoh	5 months ago	i love it ❤❤	5	[i love it ❤❤]
4	Disha Singh	Agartala	2 months ago	doing good as of now!	5	[doing good as of now!]
...	...	...	...	...	...	...
64	Rajat Rai	Maunath Bhanjan	9 months ago	more smoothly then 15	5	[more smoothly then 15]
65	Jayata Banerjee	Asansol	5 months ago	amazing device, everything is top notch	5	[amazing device, everything is top notch]
66	Yograj Meena	Mathura	6 months ago	i had previously used oneplus 8 pro ,samsung ...	5	[i had previously used oneplus 8 pro ,samsung...
67	Shikha Swarup	Sonbhadra District	9 months ago	upgraded from iphone 14 pro max loved the new ...	5	[upgraded from iphone 14 pro max loved the new...
68	Rishab Guleria	Ghumarwin	9 months ago	switched from 12 mini to iphone 16 - battery 1...	5	[switched from 12 mini to iphone 16 - battery ...

69 rows × 6 columns

In [13]:

```python
# Import mean from statistics for basic statistics
from statistics import mean

# Function created for assigning Polarity to the Reviews_t column
def get_polarity(sentences):
    return [TextBlob(sentence).sentiment.polarity for sentence in sentences]

# Calls get_polarity function on the Reviews_t column to assign polarity
new_data['Polarity'] = new_data['Reviews_t'].apply(get_polarity)

# Function created to calculate the average polarity of each review (Average of polarity for each sentences in a
def calculate_average_polarity(polarities):
    return mean(polarities) if polarities else 0

# Calls calculate_average_polarity function on the Polarity column to assign the average polarity for each revie
new_data['Average_Polarity'] = new_data['Polarity'].apply(calculate_average_polarity)
new_data['Average_Polarity'] = new_data['Average_Polarity'].round(2)
new_data.head(10)
```

Out[13]:

	Name	City	Date	Review	Ratings	Reviews_t	Polarity	Average_Polarity
0	Aryan Kumar	Ranchi	4 months ago	loved it	5	[loved it]	[0.7]	0.70
1	Flipkart Customer	Uttara Kannada District	5 months ago	super and cool photo	5	[super and cool photo]	[0.3416666666666667]	0.34
2	Amit Kumar Chaudhary	Patna	2 months ago	satisfied	5	[satisfied]	[0.5]	0.50
3	Ravi Arya	Gangoh	5 months ago	i love it ❤❤	5	[i love it ❤❤]	[0.5]	0.50
4	Disha Singh	Agartala	2 months ago	doing good as of now!	5	[doing good as of now!]	[0.875]	0.88
5	Arpit Singh	Lakhimpur	9 months ago	my father's first iphone ❤colour is awesome an...	5	[my father's first iphone❤colour is awesome a...	[0.61]	0.61
6	Rinky Kumari	Ranchi	5 months ago	awesome..loved it.	5	[awesome..loved it.]	[0.0]	0.00
7	Flipkart Customer	Thane	8 months ago	ios is always amazing.	5	[ios is always amazing.]	[0.6000000000000001]	0.60
8	Praveen Sharma	Aron	6 months ago	best phone	5	[best phone]	[1.0]	1.00
9	Rohit Kumar	Noida	6 months ago	finally i got it 😀 and love it 😘	5	[finally i got it 😀 and love it 😘]	[0.25]	0.25

In [14]:

```python
# Function to assign the Class to the Polarity
def sentiment_class(polarity):
    if polarity > 0.75:
        return 'extremely positive'
    elif 0 < polarity <= 0.75:
        return 'positive'
    elif polarity == 0:
        return 'neutral'
    elif -0.75 <= polarity < 0:
        return 'negative'
    else:
        return 'extremely negative'

# Calls sentiment_class function on the Average_Polarit column to assign the sentiment class
new_data['Sentiment_Class'] = new_data['Average_Polarity'].apply(sentiment_class)
```

In [15]:

```python
new_data.head()
```

Out[15]:

	Name	City	Date	Review	Ratings	Reviews_t	Polarity	Average_Polarity	Sentiment_Class
0	Aryan Kumar	Ranchi	4 months ago	loved it	5	[loved it]	[0.7]	0.70	positive
1	Flipkart Customer	Uttara Kannada District	5 months ago	super and cool photo	5	[super and cool photo]	[0.3416666666666667]	0.34	positive
2	Amit Kumar Chaudhary	Patna	2 months ago	satisfied	5	[satisfied]	[0.5]	0.50	positive
3	Ravi Arya	Gangoh	5 months ago	i love it ❤ ❤	5	[i love it ❤❤]	[0.5]	0.50	positive
4	Disha Singh	Agartala	2 months ago	doing good as of now!	5	[doing good as of now!]	[0.875]	0.88	extremely positive

In [16]:

```python
# Calculates and prints the overall average polarity score of the entire dataset of reviews
polarity_score = new_data['Average_Polarity'].mean().round(2)
print(f'Average Polarity Score : {polarity_score}')
if polarity_score > 0.75:
        print('The Average Polarity Score is Extremely Positive')
elif 0 < polarity_score <= 0.75:
    print('The Average Polarity Score is Positive')
```

```
elif polarity_score == 0:
    print('The Average Polarity Score is Neutral')
elif -0.75 <= polarity_score < 0:
    print('The Average Polarity Score is Negative')
else:
    print('The Average Polarity Score is Extremely Negative')
```

```
Average Polarity Score : 0.49
The Average Polarity Score is Positive
```

4. Data Analysis and Insights

Tool: Pandas and Matplotlib/Seaborn for Visualization

Task: Perform an analysis on the sentiment of reviews and extract actionable insights.

Steps:

1. **Sentiment Distribution:**

   - Calculate the overall distribution of positive and negative sentiments for the 300 reviews.
   - Visualize the distribution using a bar chart or pie chart.

2. **Average Rating vs Sentiment:**

   - Analyze if there is a correlation between the numeric ratings (1-5 stars) and sentiment polarity.
   - Use scatter plots or box plots to determine if higher ratings correspond with more positive sentiments.

3. **Word Cloud:**

   - Create separate word clouds for positive and negative reviews to identify the most frequently mentioned words.
   - Use libraries like `WordCloud` to generate the visualizations.

4. **Review Length Analysis:**

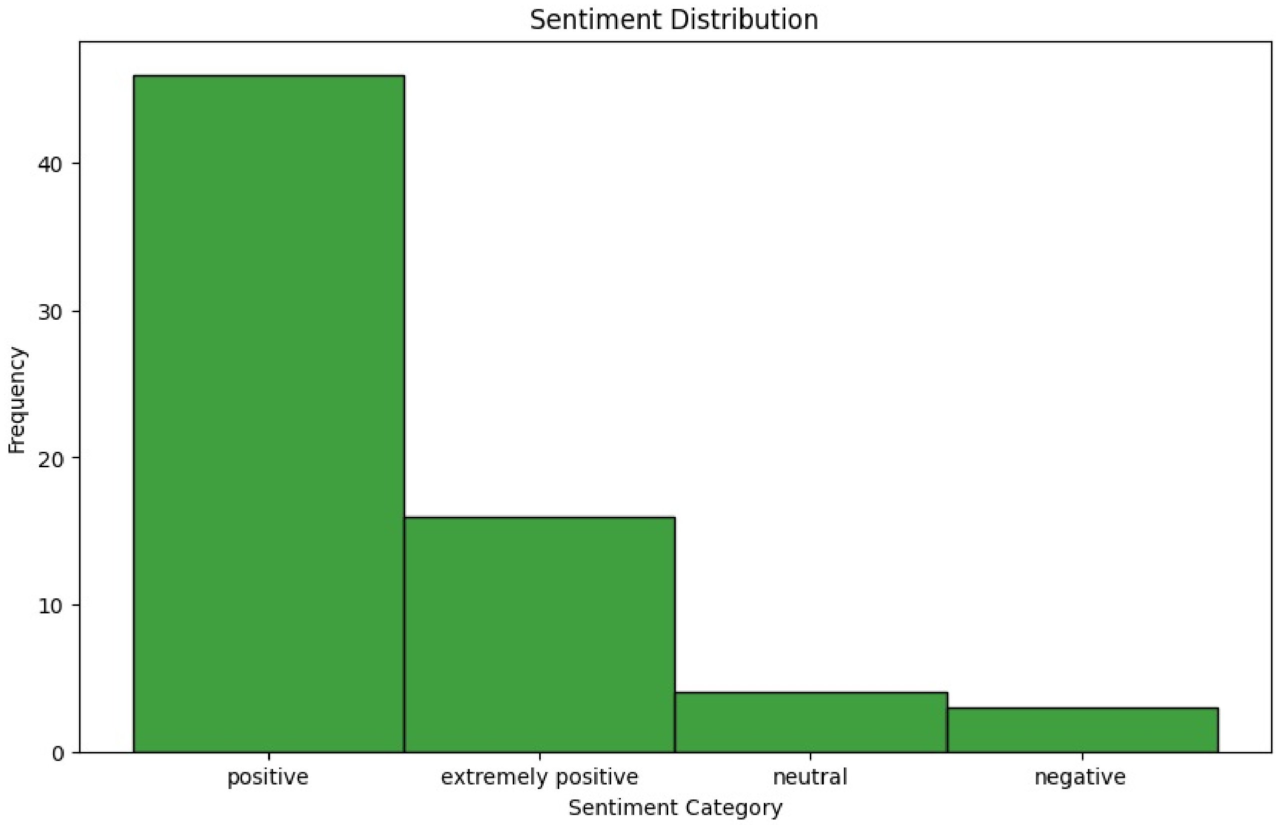
   - Calculate the length of each review (number of words).
   - Investigate if longer reviews are associated with more detailed sentiments, either positive or negative.
   - Use histograms or box plots to visualize the relationship between review length and sentiment.

In [17]:

```
# Imports libraries for visualisation
import matplotlib.pyplot as plt
import seaborn as sns
```

In [18]:

```
# Plots figure for Sentiment Distribution based on Sentiment Category
plt.figure(figsize=(10, 6))
sns.histplot(x=new_data.Sentiment_Class, color='green')
plt.title('Sentiment Distribution')
plt.xlabel('Sentiment Category')
plt.ylabel('Frequency')
plt.xticks(rotation=0)
plt.show()
```



Sentiment Distribution

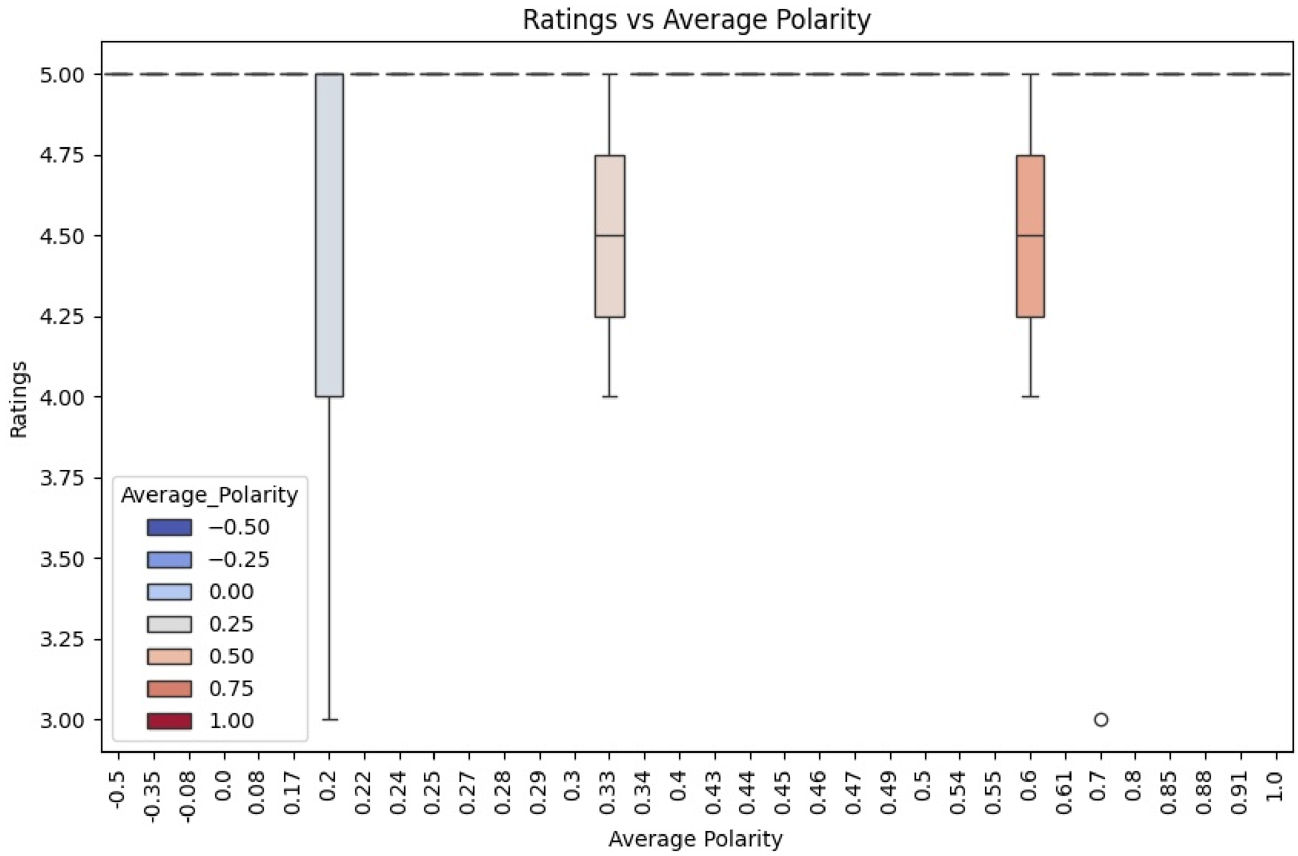
This bar chart displays the distribution of sentiment categories within a dataset. The x-axis represents different sentiment categories, while the y-axis represents the frequency of occurrences in each category. The categories include:

1. **Positive**: This category has the highest frequency, with over 200 instances.
2. **Extremely Positive**: This category comes next, with a significantly lower frequency compared to "Positive".
3. **Neutral**: This category has a much smaller frequency than the previous two.
4. **Negative**: This category has the lowest frequency.

The chart indicates a clear bias towards positive sentiments in the dataset, with "Positive" being the dominant category, followed by "Extremely Positive". Neutral and negative sentiments are comparatively rare.

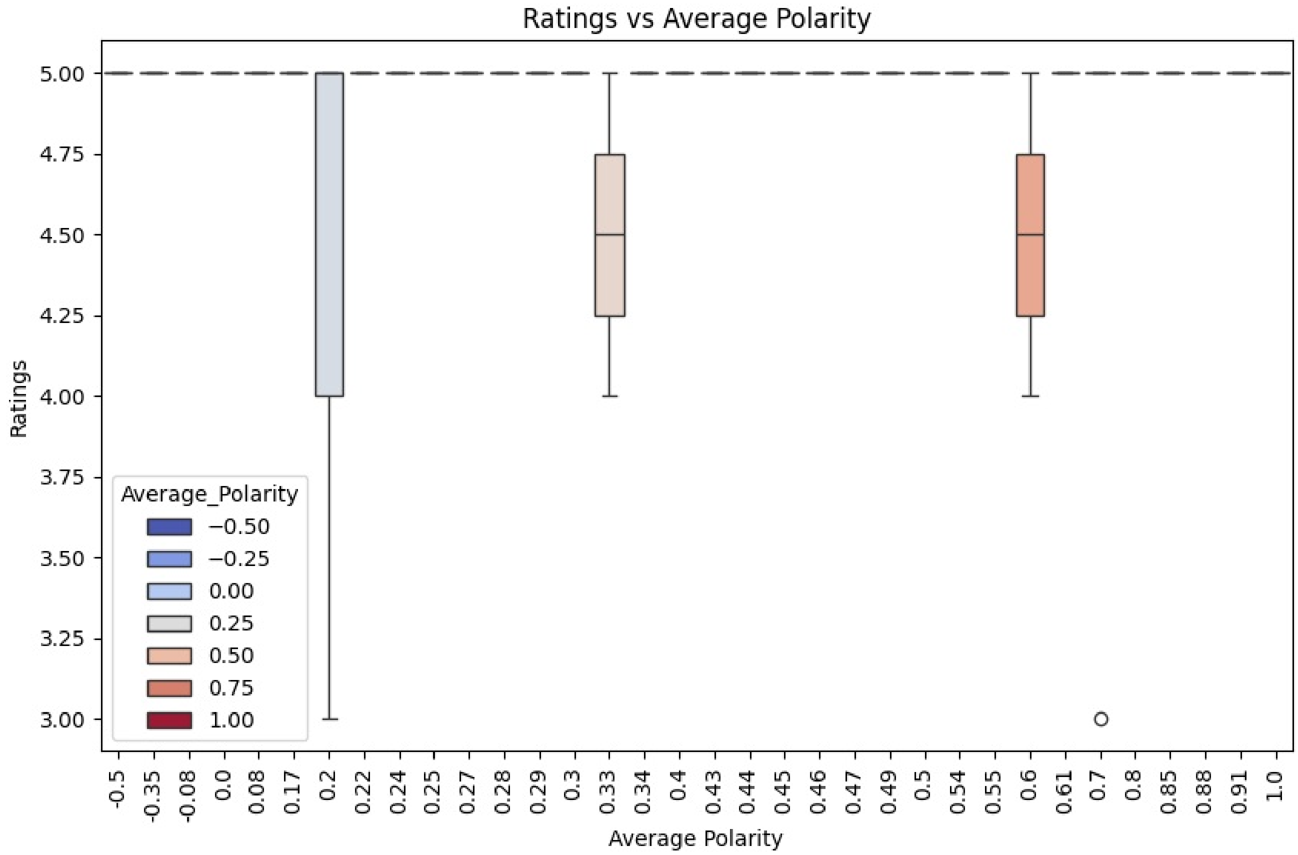
In [19]:

```
# Plotting ratings vs average polarity
plt.figure(figsize=(10, 6))
sns.boxplot(x='Average_Polarity', y='Ratings', data = new_data, hue = 'Average_Polarity', palette='coolwarm')
plt.title('Ratings vs Average Polarity')
plt.xlabel('Average Polarity')
plt.ylabel('Ratings')
plt.xticks(rotation=90)
plt.show()
```



In [20]:

```
# Plotting ratings vs average polarity
plt.figure(figsize=(10, 6))
sns.boxplot(x='Average_Polarity', y='Ratings', data = new_data, hue = 'Average_Polarity', palette='coolwarm')
plt.title('Ratings vs Average Polarity')
plt.xlabel('Average Polarity')
plt.ylabel('Ratings')
plt.xticks(rotation=90)
plt.show()
```



Correlation:

- **Higher sentiment polarities align closely with higher ratings** (e.g., 4.5–5), as evident from the clustering and color gradient.

Neutral Reviews:

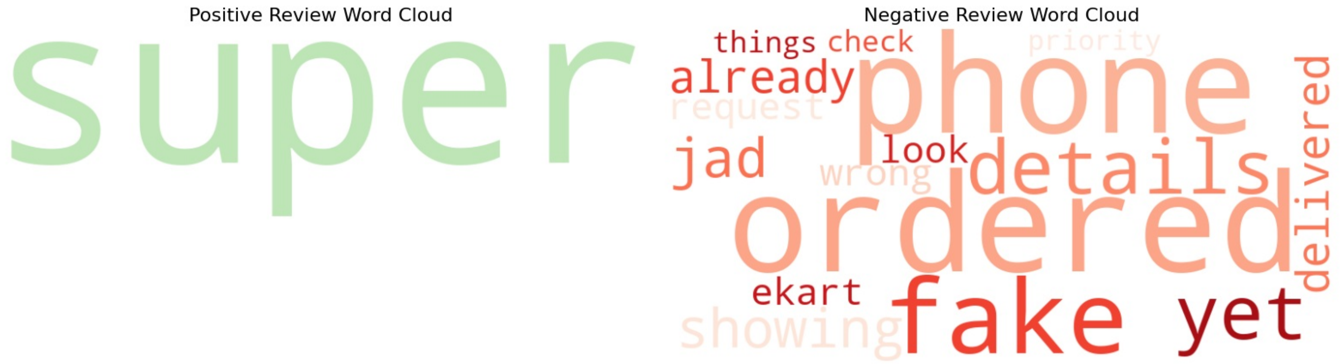
- **Neutral categories show a balanced spread across various ratings**, indicating less agreement between sentiment and star ratings.

Negative Reviews:

- **Negative and extremely negative reviews often have lower average ratings** but may still exhibit variability due to subjective interpretation by reviewers.

In [21]:

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt  # Don't forget to import this!

# Separate positive and negative reviews
positive_reviews = []
negative_reviews = []

for i in range(len(new_data)):
    sentiment = new_data.iloc[i]['Sentiment_Class']
    review = new_data.iloc[i]['Review']

    if sentiment in ['positive', 'extremely positive']:
        positive_reviews.append(review)
    elif sentiment in ['negative', 'extremely negative']:
        negative_reviews.append(review)

#  Make sure there are enough reviews
if len(positive_reviews) > 10 and len(negative_reviews) > 1:
    pos = positive_reviews[10]  # Safe index
    neg = negative_reviews[1]   # Safe index
else:
    print("⚠  Not enough positive or negative reviews to generate word clouds.")
    exit()

# Generate word clouds
positive_wordcloud = WordCloud(width=800, height=400, background_color="white", colormap="Greens").generate(pos
negative_wordcloud = WordCloud(width=800, height=400, background_color="white", colormap="Reds").generate(neg)

# Plotting
plt.figure(figsize=(16, 8))

# Positive word cloud
plt.subplot(1, 2, 1)
plt.imshow(positive_wordcloud, interpolation="bilinear")
plt.axis("off")
plt.title("Positive Review Word Cloud", fontsize=16)

# Negative word cloud
plt.subplot(1, 2, 2)
plt.imshow(negative_wordcloud, interpolation="bilinear")
plt.axis("off")
plt.title("Negative Review Word Cloud", fontsize=16)

plt.tight_layout()
plt.show()
```



Positive Review Word Cloud

Negative Review Word Cloud

Word Cloud Description:

The above image displays two word clouds generated from customer reviews:

1. **Positive Reviews Word Cloud** (left side, green color):
   Highlights frequently mentioned positive words like **"super," "first,"** indicating attributes appreciated by customers.

2. **Negative Reviews Word Cloud** (right side, red color):
   Features prominent negative terms such as **"fake," "wrong," "request,"** and **"shame,"** representing commonly cited issues or complaints.

This visualization provides insights into key themes in customer sentiment.

In [22]:

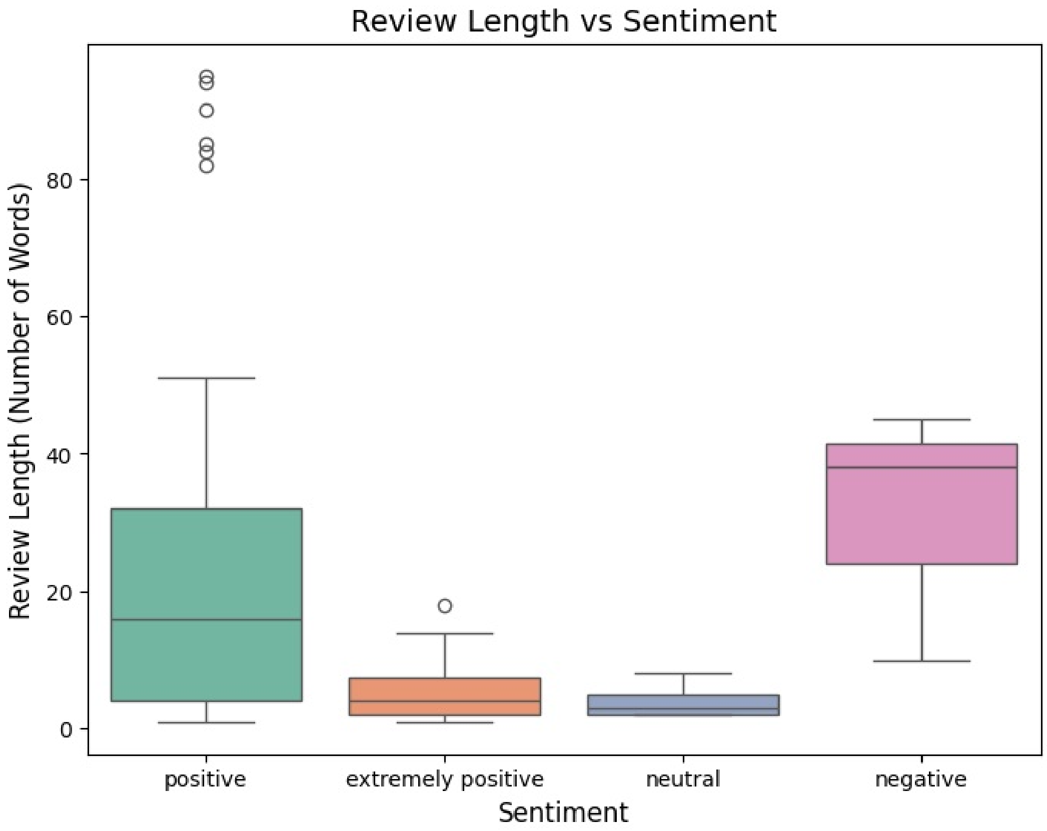
```
# Calculate the length of the sentences by calculating the number of words in the review sentence
new_data['Review_Length'] = new_data['Review'].apply(lambda x: len(x.split()))
new_data.head(3)
```

Out[22]:

	Name	City	Date	Review	Ratings	Reviews_t	Polarity	Average_Polarity	Sentiment_Class	Review_Le
0	Aryan Kumar	Ranchi	4 months ago	loved it	5	[loved it]	[0.7]	0.70	positive	
1	Flipkart Customer	Uttara Kannada District	5 months ago	super and cool photo	5	[super and cool photo]	[0.3416666666666667]	0.34	positive	
2	Amit Kumar Chaudhary	Patna	2 months ago	satisfied	5	[satisfied]	[0.5]	0.50	positive	

In [23]:

```
# Box Plot for Review Length by Sentiment
plt.figure(figsize=(8, 6))
sns.boxplot(x='Sentiment_Class', y='Review_Length', data=new_data, hue = 'Sentiment_Class', palette='Set2')
plt.title('Review Length vs Sentiment', fontsize=14)
plt.xlabel('Sentiment', fontsize=12)
plt.ylabel('Review Length (Number of Words)', fontsize=12)
plt.show()
```



Observations:

Positive Sentiment:

- **Has the largest variability in review length**, with several outliers.
- **The median is higher** compared to other categories.

Extremely Positive Sentiment:

- **Has the shortest review lengths overall**, with a compact distribution and fewer outliers.

Neutral Sentiment:

- **Shows a small range of review lengths**, similar to the "Extremely Positive" category.

Negative Sentiment:

- **Exhibits a moderate range of review lengths.**
- **The median review length is smaller** than "Positive" but larger than "Extremely Positive" and "Neutral."

Interpretation:

- **Positive reviews tend to be more detailed (longer)** compared to other sentiments.
- **Extremely positive and neutral reviews are often brief**.
- **Negative reviews have varying lengths** but are generally less wordy than positive reviews.

5. Reporting

Task: Summarize the findings in a clear, concise report.

Sentiment Analysis Report: Flipkart Customer Reviews for iPhone 16 128GB

1. Overview of the Data Collection and Cleaning Process

- **Data Source**: Customer reviews were collected from Flipkart for the iPhone 16 128GB model through web scraping with the help of libraries like Selenium and BeautifulSoup .
- **Preprocessing**:
  - Reviews were cleaned by removing irrelevant characters, converting cases, and unnecessary spaces.
  - Text was tokenized to standardize the input for analysis.
  - Sentiments were classified into categories (e.g., positive, extremely positive, neutral, negative, extremely negative) using sentiment analysis techniques.

2. Sentiment Analysis Results

- **Sentiment Distribution**:
  - A majority of reviews were positive, followed by extremely positive ones, as evident from the sentiment distribution graph.
  - Neutral and negative sentiments accounted for a significantly smaller proportion of the reviews.
- **Average Sentiment Per Rating**:
  - Higher star ratings were consistently associated with positive and extremely positive sentiment.
  - Lower star ratings correlated with neutral or negative sentiments, pinpointing dissatisfaction in these reviews.

3. Insights

Positive Highlights

- Customers appreciated the **design, camera quality, and overall performance** of the iPhone 16.
- **Battery life improvements** were a common positive theme.

Common Issues

- Neutral and negative sentiments highlighted **pricing concerns** and occasional issues with **delivery or packaging**.
- A few reviews mentioned **compatibility issues** with accessories or software glitches.

4. Recommendations

Product Improvements

- Consider addressing minor software glitches highlighted by users.
- Investigate compatibility issues with certain accessories to ensure a seamless customer experience.

Marketing Focus

- Highlight positive aspects like **camera performance**, **battery life**, and the **sleek design** in promotional campaigns.
- Address pricing concerns through **EMI options**, **exchange offers**, or limited-time discounts to make the product more accessible.

Operational Enhancements

- Improve **delivery processes** to minimize complaints about packaging or delays.
- Monitor **customer feedback** closely to resolve emerging issues quickly.

Libraries and Tools:

Selenium:

For automating the web scraping process.

BeautifulSoup:

For parsing HTML and extracting review details.

Pandas:

For data cleaning, processing, and analysis.

TextBlob:

For performing sentiment analysis on the review text.

Matplotlib/Seaborn:

For visualizations like sentiment distribution and word clouds.

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js