Customer Sentimental Analysis - Iphone 15 128gb

Objective:

To analyze customer sentiment towards the iPhone 15 128GB model by performing sentiment analysis on user reviews. This project aims to extract insights on public perception, identify key areas for improvement, and support decision-making at Flipkart to enhance customer experience and optimize product offerings based on real user feedback.

Libraries and Tools:

- **Selenium**: Web scraping automation.
- **BeautifulSoup**: HTML parsing.
- **Pandas**: Data cleaning and analysis.
- **TextBlob**: Sentiment analysis.
- **Matplotlib/Seaborn**: Data visualization.

1. Data Collection (Web Scraping)

- **Tools**: BeautifulSoup, Selenium
- **Steps**:
  - Use Selenium to scrape at least 160 reviews from Flipkart's iPhone 15 128GB product page.
  - Extract **Username**, **Rating**, and **Review Text**.
  - Handle pagination to collect reviews from multiple pages.

In [2]:
```
#importing the libraries used in webscraping

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 [3]:
```
# 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-15-blue-128-gb/product-reviews/itmbf14ef54f645d?pid=MOBGTAGPAQNVFZZY&lid=LSTMOBGTAGPAQNVFZZYQRLPCQ&marketplace=FLIPKART"""
driver = webdriver.Chrome()
driver.get(url)


while len(Names) < 300:

    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 [8]:
```
# Combine data into a DataFrame
min_length = min(len(Names), len(Cities), len(Actual_Dates), len(Reviews), len(Ratings))
Names, Cities, Actual_Dates, Reviews, Ratings = (
    Names[:min_length], Cities[:min_length], Actual_Dates[:min_length], Reviews[:min_length], Ratings[:min_length]
)

# Create DataFrame
df = pd.DataFrame({
    "Name": Names,
    "City": Cities,
    "Date": Actual_Dates,
    "Review": Reviews,
    "Ratings": Ratings
})
```

```
# Save to a csv files
df.to_csv("iphone_15_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 [12]:
```
# Assign the scraped dataset(csv file) to a dataframe
data = pd.read_csv('iphone_15_reviews.csv')
data
```

Out[12]:

	Name	City	Date	Review	Ratings
0	Ajin V	Certified Buyer, Balaghat	Oct, 2023	High quality camera😍READ MORE	5
1	bijaya mohanty	Certified Buyer, Baleshwar	8 months ago	Just go for it.Amazing one.Beautiful camera wi...	5
2	Mousam Guha Roy	Certified Buyer, Matialihat	Oct, 2023	Very niceREAD MORE	4
3	Prithivi Boruah	Certified Buyer, Bokajan	Oct, 2023	Camera Quality Is Improved Loving ItREAD MORE	5
4	Nikhil Kumar	Certified Buyer, Meerut Division	Jan, 2024	Switch from OnePlus to iPhone I am stunned wit...	5
...	...	...	...	...	...
155	Rishi Singh	Certified Buyer, Noida	Jan, 2024	Superb performanceREAD MORE	5
156	Rohit Kumar Mishra	Certified Buyer, Jodhpur District	Jan, 2024	Nice iPhoneREAD MORE	5
157	Praveenkumar Yedulla	Certified Buyer, Hyderabad	Jan, 2024	Good experience I am using first time apple m...	4
158	Sabir Khan	Certified Buyer, Saharanpur District	Jan, 2024	Very nice phone im happy very good 👍READ MORE	5
159	Sourav Patra	Certified Buyer, Medinipur	Jan, 2024	I absolutely adore it. It brings me immense jo...	5

160 rows × 5 columns

In [13]:
```
# Check the basic info of the dataframe
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 160 entries, 0 to 159
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   Name     160 non-null    object
 1   City     160 non-null    object
 2   Date     160 non-null    object
 3   Review   160 non-null    object
 4   Ratings  160 non-null    int64 
dtypes: int64(1), object(4)
memory usage: 6.4+ KB
```

In [20]:
```
# Check value counts of the Name column
data['Name'].value_counts()
```

Out[20]:
```
Name
Flipkart Customer       13
Nikhil Kumar             2
Shubhanker  Singh        2
Ankit  Verma             2
Monish B                 2
                        ..
Rishi Singh              1
Rohit Kumar Mishra       1
Praveenkumar Yedulla     1
Sabir Khan               1
Sourav Patra             1
Name: count, Length: 136, dtype: int64
```

In [21]:
```
# Drop the duplicates from the dataframe
data = data.copy()
data = data.drop_duplicates()
data
```

Out[21]:

	Name	City	Date	Review	Ratings
0	Ajin V	Certified Buyer, Balaghat	Oct, 2023	High quality camera😍READ MORE	5
1	bijaya mohanty	Certified Buyer, Baleshwar	8 months ago	Just go for it.Amazing one.Beautiful camera wi...	5
2	Mousam Guha Roy	Certified Buyer, Matialihat	Oct, 2023	Very niceREAD MORE	4
3	Prithivi Boruah	Certified Buyer, Bokajan	Oct, 2023	Camera Quality Is Improved Loving ItREAD MORE	5
4	Nikhil Kumar	Certified Buyer, Meerut Division	Jan, 2024	Switch from OnePlus to iPhone I am stunned wit...	5
...	...	...	...	...	...
155	Rishi Singh	Certified Buyer, Noida	Jan, 2024	Superb performanceREAD MORE	5
156	Rohit Kumar Mishra	Certified Buyer, Jodhpur District	Jan, 2024	Nice iPhoneREAD MORE	5
157	Praveenkumar Yedulla	Certified Buyer, Hyderabad	Jan, 2024	Good experience I am using first time apple m...	4
158	Sabir Khan	Certified Buyer, Saharanpur District	Jan, 2024	Very nice phone im happy very good 👍READ MORE	5
159	Sourav Patra	Certified Buyer, Medinipur	Jan, 2024	I absolutely adore it. It brings me immense jo...	5

153 rows × 5 columns

In [22]:
```
# Convert the Name column data into Title Case
data['Name'] = data['Name'].str.title()
data.head()
```

Out[22]:

	Name	City	Date	Review	Ratings
0	Ajin V	Certified Buyer, Balaghat	Oct, 2023	High quality camera😍READ MORE	5
1	Bijaya Mohanty	Certified Buyer, Baleshwar	8 months ago	Just go for it.Amazing one.Beautiful camera wi...	5
2	Mousam Guha Roy	Certified Buyer, Matialihat	Oct, 2023	Very niceREAD MORE	4
3	Prithivi Boruah	Certified Buyer, Bokajan	Oct, 2023	Camera Quality Is Improved Loving ItREAD MORE	5
4	Nikhil Kumar	Certified Buyer, Meerut Division	Jan, 2024	Switch from OnePlus to iPhone I am stunned wit...	5

In [23]:
```
# Clean data of City column by removing unwanted characters/ part of string
data['City'] = data['City'].str.replace("Certified Buyer, ", "", regex=False).str.strip()
data.head()
```

Out[23]:

	Name	City	Date	Review	Ratings
0	Ajin V	Balaghat	Oct, 2023	High quality camera😍READ MORE	5
1	Bijaya Mohanty	Baleshwar	8 months ago	Just go for it.Amazing one.Beautiful camera wi...	5
2	Mousam Guha Roy	Matialihat	Oct, 2023	Very niceREAD MORE	4
3	Prithivi Boruah	Bokajan	Oct, 2023	Camera Quality Is Improved Loving ItREAD MORE	5
4	Nikhil Kumar	Meerut Division	Jan, 2024	Switch from OnePlus to iPhone I am stunned wit...	5

In [24]:
```
# Clean data of Review column by removing unwanted characters/ part of string and converting to lowercase
data['Review'] = data['Review'].str.lower().str.replace("read more", "", regex=False)
data.head()
```

Out[24]:

	Name	City	Date	Review	Ratings
0	Ajin V	Balaghat	Oct, 2023	high quality camera😍	5
1	Bijaya Mohanty	Baleshwar	8 months ago	just go for it.amazing one.beautiful camera wi...	5
2	Mousam Guha Roy	Matialihat	Oct, 2023	very nice	4
3	Prithivi Boruah	Bokajan	Oct, 2023	camera quality is improved loving it	5
4	Nikhil Kumar	Meerut Division	Jan, 2024	switch from oneplus to iphone i am stunned wit...	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 [35]:
```
# 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_data] Downloading package stopwords to
[nltk_data]     C:\Users\actfo\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\actfo\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\actfo\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Out[35]: True

In [40]:
```
# Create a column called Reviews_t that stores tokenized sentences from the Review column using the sent_tokenize function.
data["Reviews_t"] = data['Review'].apply(sent_tokenize)
data
```

Out[40]:

	Name	City	Date	Review	Ratings	Reviews_t
0	Ajin V	Balaghat	Oct, 2023	high quality camera 😍	5	[high quality camera 😍]
1	Bijaya Mohanty	Baleshwar	8 months ago	just go for it.amazing one.beautiful camera wi...	5	[just go for it.amazing one.beautiful camera w...
2	Mousam Guha Roy	Matialihat	Oct, 2023	very nice	4	[very nice]
3	Prithivi Boruah	Bokajan	Oct, 2023	camera quality is improved loving it	5	[camera quality is improved loving it]
4	Nikhil Kumar	Meerut Division	Jan, 2024	switch from oneplus to iphone i am stunned wit...	5	[switch from oneplus to iphone i am stunned wi...
...	...	...	...	...	...	...
155	Rishi Singh	Noida	Jan, 2024	superb performance	5	[superb performance]
156	Rohit Kumar Mishra	Jodhpur District	Jan, 2024	nice iphone	5	[nice iphone]
157	Praveenkumar Yedulla	Hyderabad	Jan, 2024	good experience i am using first time apple m...	4	[good experience i am using first time apple ...
158	Sabir Khan	Saharanpur District	Jan, 2024	very nice phone im happy very good 👍	5	[very nice phone im happy very good 👍]
159	Sourav Patra	Medinipur	Jan, 2024	i absolutely adore it. it brings me immense jo...	5	[i absolutely adore it., it brings me immense ...

153 rows × 6 columns

In [41]:
```
# 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
data['Polarity'] = data['Reviews_t'].apply(get_polarity)

# Function created to calculate the average polarity of each review (Average of polarity for each sentences in a review)
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 review
data['Average_Polarity'] = data['Polarity'].apply(calculate_average_polarity)
data['Average_Polarity'] = data['Average_Polarity'].round(2)
data.head(10)
```

Out[41]:

	Name	City	Date	Review	Ratings	Reviews_t	Polarity	Average_Polarity
0	Ajin V	Balaghat	Oct, 2023	high quality camera 😍	5	[high quality camera 😍]	[0.16]	0.16
1	Bijaya Mohanty	Baleshwar	8 months ago	just go for it.amazing one.beautiful camera wi...	5	[just go for it.amazing one.beautiful camera w...	[0.26666666666666666]	0.27
2	Mousam Guha Roy	Matialihat	Oct, 2023	very nice	4	[very nice]	[0.78]	0.78
3	Prithivi Boruah	Bokajan	Oct, 2023	camera quality is improved loving it	5	[camera quality is improved loving it]	[0.6]	0.60
4	Nikhil Kumar	Meerut Division	Jan, 2024	switch from oneplus to iphone i am stunned wit...	5	[switch from oneplus to iphone i am stunned wi...	[0.0, 1.0]	0.50
5	Flipkart Customer	Aizawl	Jan, 2024	awesome photography experience. battery backup...	5	[awesome photography experience., battery back...	[1.0, 0.7, 0.5]	0.73
6	Sheetla Prasad Maurya	Sultanpur	Oct, 2023	best mobile phonecamera quality is very nice b...	4	[best mobile phonecamera quality is very nice ...	[0.738]	0.74
7	Akshay Meena	Jaipur	Nov, 2023	so beautiful, so elegant, just a vowww😍❤️	5	[so beautiful, so elegant, just a vowww😍❤️]	[0.675]	0.68
8	Raj Singh	Kolkata	Dec, 2023	for me its 10 out of 10✨	5	[for me its 10 out of 10✨]	[0.0]	0.00
9	Mohit Yadav	Mumbai	Nov, 2023	nice ❤️	5	[nice ❤️]	[0.6]	0.60

In [43]:
```
# 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
data['Sentiment_Class'] = data['Average_Polarity'].apply(sentiment_class)
data.head()
```

Out[43]:

	Name	City	Date	Review	Ratings	Reviews_t	Polarity	Average_Polarity	Sentiment_Class
0	Ajin V	Balaghat	Oct, 2023	high quality camera 😍	5	[high quality camera 😍]	[0.16]	0.16	positive
1	Bijaya Mohanty	Baleshwar	8 months ago	just go for it.amazing one.beautiful camera wi...	5	[just go for it.amazing one.beautiful camera w...	[0.26666666666666666]	0.27	positive
2	Mousam Guha Roy	Matialihat	Oct, 2023	very nice	4	[very nice]	[0.78]	0.78	extremely positive
3	Prithivi Boruah	Bokajan	Oct, 2023	camera quality is improved loving it	5	[camera quality is improved loving it]	[0.6]	0.60	positive
4	Nikhil Kumar	Meerut Division	Jan, 2024	switch from oneplus to iphone i am stunned wit...	5	[switch from oneplus to iphone i am stunned wi...	[0.0, 1.0]	0.50	positive

In [45]:
```
# Calculates and prints the overall average polarity score of the entire dataset of reviews
polarity_score = 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 : 0.48
The Average Polarity Score is Positive
```

4. Data Analysis and Insights

Tool: Pandas and Matplotlib/Seaborn for Visualization

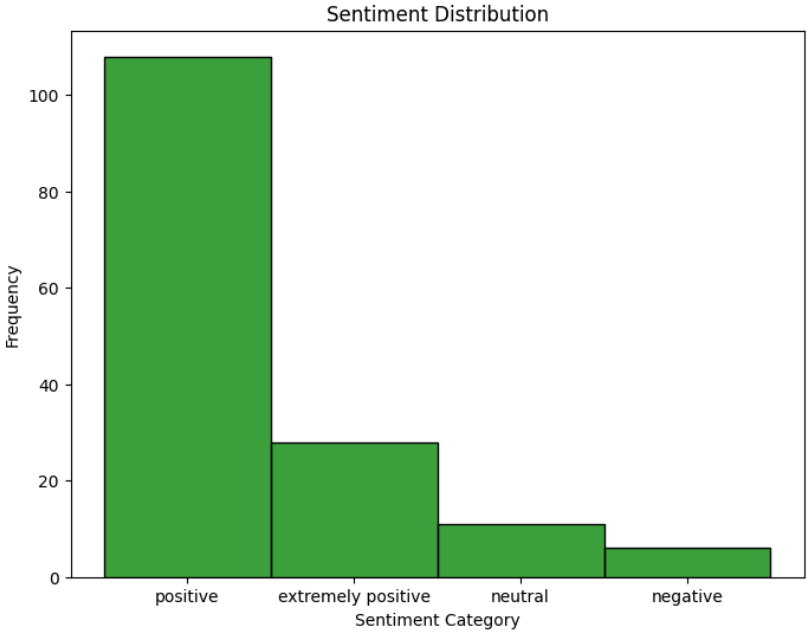
Task: Perform an analysis on the sentiment of reviews and extract actionable insights.

Steps:

1. **Sentiment Distribution**:
   - Calculate the overall distribution of positive and negative sentiments for the 160 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 [46]:
```
# Imports libraries for visualisation
import matplotlib.pyplot as plt
import seaborn as sns
```

In [53]:
```
# Plots figure for Sentiment Distribution based on Sentiment Category
plt.figure(figsize=(8,6))
sns.histplot(x=data.Sentiment_Class, color='green')
plt.title('Sentiment Distribution')
plt.xlabel('Sentiment Category')
plt.ylabel('Frequency')
plt.xticks(rotation=0)
plt.show()
```



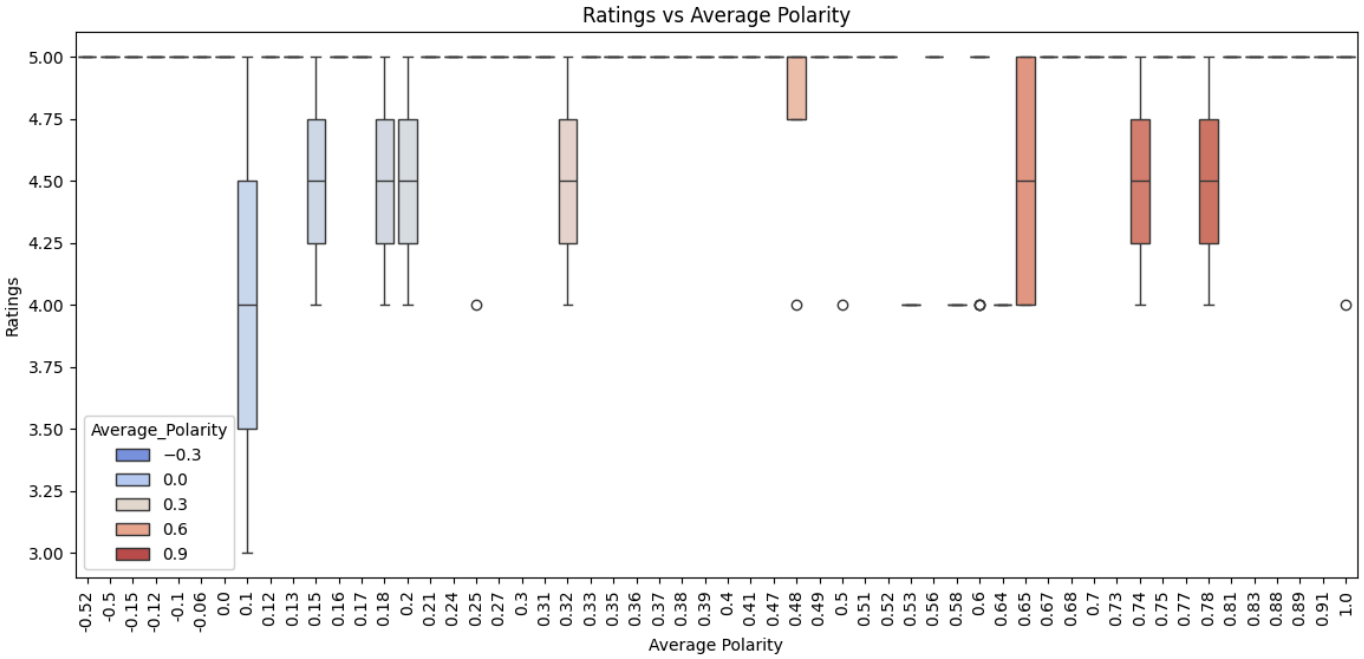
Sentiment Distribution

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 100 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 [55]:
```
# Plotting ratings vs average polarity
plt.figure(figsize=(14, 6))
sns.boxplot(x='Average_Polarity', y='Ratings', data = 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.

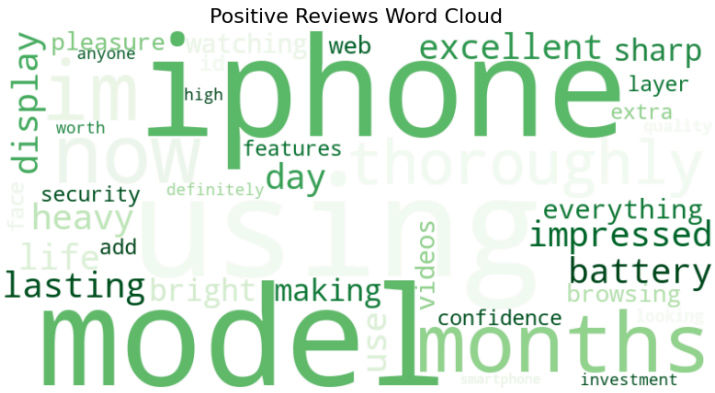
Neutral Reviews:

- **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 [104…
```
from wordcloud import WordCloud


positive_reviews = []
negative_reviews = []

# Classify the positive & negative reviews separately
for i in range(len(data)):
    if data.iloc[i]['Sentiment_Class'] == 'positive' or data.iloc[i]['Sentiment_Class'] == 'extremely positive':
        positive_reviews.append(data.iloc[i]['Review'])
    elif data.iloc[i]['Sentiment_Class'] == 'negative' or data.iloc[i]['Sentiment_Class'] == 'extremely negative':
        negative_reviews.append(data.iloc[i]['Review'])

# Assign random positive and negative reviews to create a cloud map
pos = positive_reviews[98]
neg = negative_reviews[2]

# Generate word clouds for positive and negative reviews
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)

# Plot the word clouds
plt.figure(figsize=(16, 8))

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

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

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



Word Cloud Description:

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

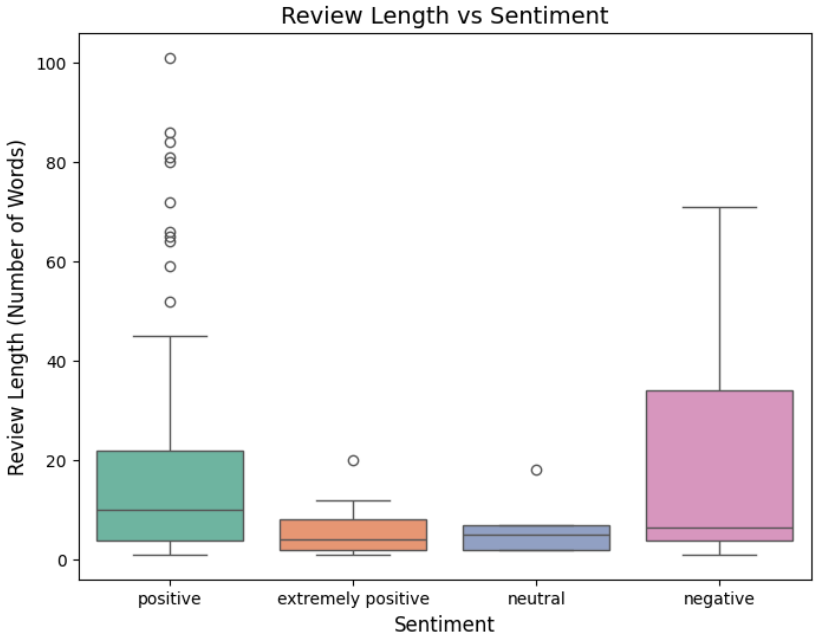
1. **Positive Reviews Word Cloud** (left side, green color):
   Highlights frequently mentioned positive words like **"iphone"**, **"excellent"**, **"pleasure"** and **"model"**, indicating attributes appreciated by customers.

2. **Negative Reviews Word Cloud** (right side, red color):
   Features prominent negative terms such as **"green"**, **"worst"**, **"heatings"** and **"bad"**, representing commonly cited issues or complaints.

This visualization provides insights into key themes in customer sentiment.

In [105…
```
# Calculate the length of the sentences by calculating the number of words in the review sentence
data['Review_Length'] = data['Review'].apply(lambda x: len(x.split()))
```

In [106…
```
# Box Plot for Review Length by Sentiment
plt.figure(figsize=(8, 6))
sns.boxplot(x='Sentiment_Class', y='Review_Length', data= 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 15 128GB

1. Overview of the Data Collection and Cleaning Process:

- **Data Source**: Customer reviews were collected from Flipkart for the iPhone 15 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 15.
  - **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.