Sentiment Analysis
Group 2

ALVI MUHAMMAD YOUSAF, AMMAR AHMAD ABDUL KARIM,
APATENKO ANASTASIIA, ARORA SONALI

Task 1- ETL Mastodon

- Connected to Mastodon: Connected with mastodon.social.
- Read search terms: Loaded search terms from a JSON file named `terms.json`.
- Connected to the Data Base: Established a connection to an SQLite database file named `sentiment.db`.
- Created a table
- Cleaned messages

Task 2 – ETL Amazon

Libraries used – sqlite3, Selenium, bs4, and time

Reads terms.txt and loads up chrome using Selenium. Time.sleep set to 20 seconds for the user to login.

Structured SQL DB.

```
driver = webdriver.Chrome()
driver.get('https://www.amazon.com/S
time.sleep(20)
```

```
# Create Tables If Not Exists
cursor.execute('''
CREATE TABLE IF NOT EXISTS reviews (
                    SID INTEGER PRIMARY KEY AUTOINCREMENT,
                    Product TEXT,
                    User TEXT,
                    Date TEXT,
                    Message TEXT,
                    Sentiment TEXT DEFAULT '',
                    Dateconverted DATE
)
```

Task 2 – ETL Amazon

Loops through each product and URL specified in the terms.txt

Time.sleep set to 3 seconds for the page to load.

```python
for product_name, url in product_data:
    print(f"Scraping reviews for: {product_name}")
    while url is not None:
        driver.get(url)
        time.sleep(3)
        html_data = BeautifulSoup(driver.page_source, 'html.parser')
        reviews = html_data.find_all('li', {'data-hook': 'review'})
        for review in reviews:
            user = review.find('span', {'class': 'a-profile-name'}).text.strip()
            raw_date = review.find('span', {'data-hook': 'review-date'}).text.strip()
            message = review.find('span', {'data-hook': 'review-body'}).text.strip()
```

Task 2 – ETL Amazon

Then inserts the data into the SQL DB.

The code handles pagination by detecting the 'Next' button on each review page and dynamically updating the URL until all pages are scraped.

```
            cursor.execute('''
                INSERT INTO reviews (Product, User, Date, Message)
                VALUES (?, ?, ?, ?)
            ''', (product_name, user, date, message))
        url_check = html_data.find('li', {'class': 'a-last'})
        if url_check and 'a-disabled' not in url_check.get('class', []):
            url = 'https://www.amazon.com' + url_check.a['href']
        else:
            url = None
    print(f"Finished scraping reviews for: {product_name}")
conn.commit()
driver.quit()
```

Task 3 SENTIMENT ANALYSIS

1. Preparation

```python
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

analyzer = SentimentIntensityAnalyzer()

sample_phrases = [
"I like it very much. It keeps my health in check."
    ]
for phrase in sample_phrases:
    sentiment_scores = analyzer.polarity_scores(phrase)
    print(f"Phrase: {phrase}")
    print(f"Sentiment Scores: {sentiment_scores}\n")
```

VaderSentiment

```
Phrase: I like it very much. It keeps my health in check.
Sentiment Scores: {'neg': 0.0, 'neu': 0.8, 'pos': 0.2, 'compound': 0.3612}
```

Task 3 SENTIMENT ANALYSIS

2. Creating a sentiment system

```
if compound_score > 0.7 and pos_score > 0.3:
    sentiment = 'Strong Positive'
elif compound_score > 0.2 and pos_score > neg_score:
    sentiment = 'Positive'
elif -0.2 <= compound_score <= 0.2 and neu_score >= 0.6:
    sentiment = 'Neutral'
elif -0.5 < compound_score < -0.2 and neg_score > pos_score:
    sentiment = 'Negative'
else:
    sentiment = 'Strong Negative'
```

	message	sentiment
1	I recently purchased th	Positive
2	My Galaxy (4) watch wa	Positive
3	I got this Samsung Wa	Positive
4	The strap is too big. Sa	Positive
5	I upgraded from an ol	Positive
6	I've been using the Sar	Positive
7	This watch has been fa	Strong Positive
8	This is wonderful perfe	Positive
9	I gave up on Fitbits aft	Positive
10	I have been wearing th	Strong Negative
11	I bought this watch to	Positive
12	My husband got me th	Positive
13	My last smartwatch wa	Positive
14	I bought this for my bı	Positive

Task 3 SENTIMENT ANALYSIS

3. Converting date column

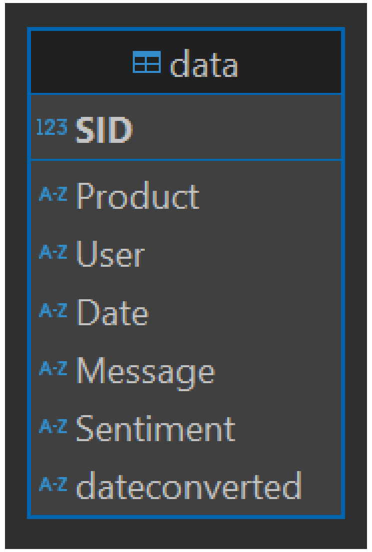
	date	dateconverted
1	December 30, 2024	2024-12-30
2	August 25, 2024	2024-08-25
3	November 19, 2024	2024-11-19
4	December 9, 2024	2024-12-09
5	December 23, 2024	2024-12-23
6	November 6, 2024	2024-11-06
7	December 22, 2024	2024-12-22
8	December 5, 2024	2024-12-05
9	December 20, 2024	2024-12-20
10	December 28, 2024	2024-12-28
11	December 14, 2024	2024-12-14
12	November 24, 2024	2024-11-24
13	December 4, 2024	2024-12-04
14	December 20, 2024	2024-12-20

```sql
CREATE TABLE reviews (
        SID INTEGER PRIMARY KEY AUTOINCREMENT,
        product TEXT,
        user TEXT,
        date TEXT,
        message TEXT,
        sentiment TEXT DEFAULT ''
    , dateconverted DATE);
```

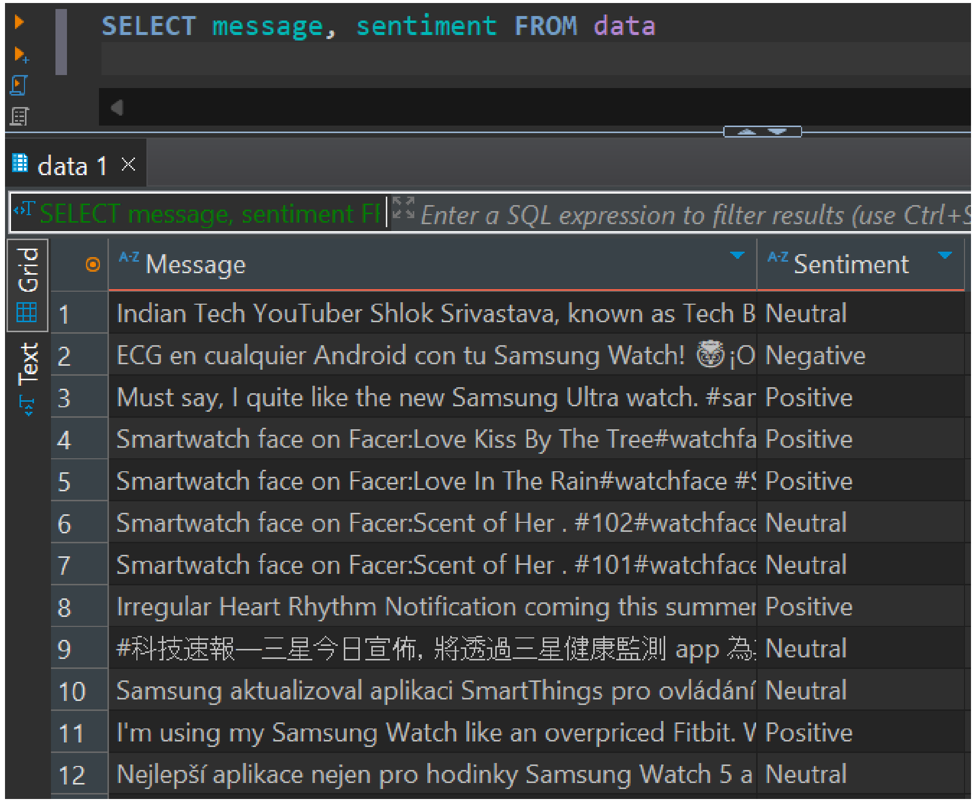
```python
for row in rows:
    SID, Date = row
    if Date:
        parts = Date.split()
        month = parts[0]
        day = parts[1].replace(',', '')
        year = parts[2]
        month_number = {
            "January": "01", "February": "02", "March": "03", "April": "04",
            "May": "05", "June": "06", "July": "07", "August": "08",
            "September": "09", "October": "10", "November": "11", "December": "12"
        }[month]
        new_date = f"{year}-{month_number}-{day.zfill(2)}"
```

Task 3 SENTIMENT ANALYSIS

Creating sentiment analysis for Mastodon database



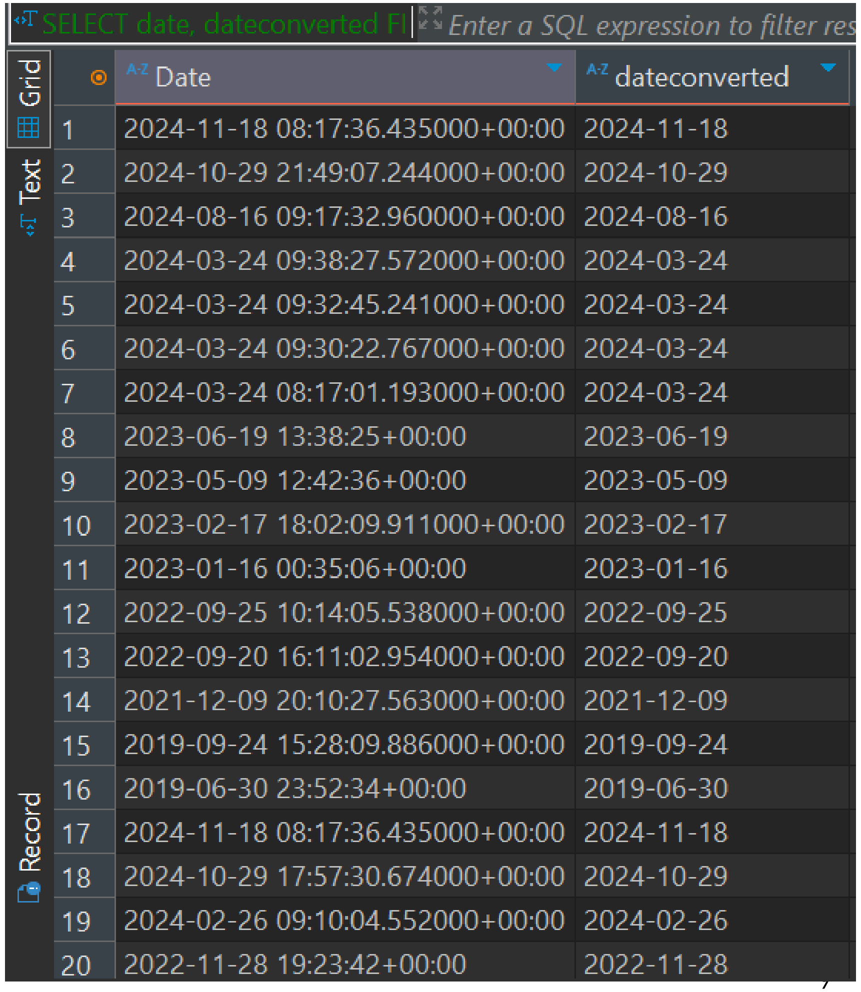
data
SID
Product
User
Date
Message
Sentiment
dateconverted



```sql
SELECT message, sentiment FROM data
```


	Message	Sentiment
1	Indian Tech YouTuber Shlok Srivastava, known as Tech B	Neutral
2	ECG en cualquier Android con tu Samsung Watch! ¡O	Negative
3	Must say, I quite like the new Samsung Ultra watch. #sar	Positive
4	Smartwatch face on Facer:Love Kiss By The Tree#watchfa	Positive
5	Smartwatch face on Facer:Love In The Rain#watchface #!	Positive
6	Smartwatch face on Facer:Scent of Her . #102#watchface	Neutral
7	Smartwatch face on Facer:Scent of Her . #101#watchface	Neutral
8	Irregular Heart Rhythm Notification coming this summer	Positive
9	#科技速報—三星今日宣佈，將透過三星健康監測 app 為	Neutral
10	Samsung aktualizoval aplikaci SmartThings pro ovládání	Neutral
11	I'm using my Samsung Watch like an overpriced Fitbit. V	Positive
12	Nejlepší aplikace nejen pro hodinky Samsung Watch 5 a	Neutral

Converting date format for better further visualisation



	Date	dateconverted
1	2024-11-18 08:17:36.435000+00:00	2024-11-18
2	2024-10-29 21:49:07.244000+00:00	2024-10-29
3	2024-08-16 09:17:32.960000+00:00	2024-08-16
4	2024-03-24 09:38:27.572000+00:00	2024-03-24
5	2024-03-24 09:32:45.241000+00:00	2024-03-24
6	2024-03-24 09:30:22.767000+00:00	2024-03-24
7	2024-03-24 08:17:01.193000+00:00	2024-03-24
8	2023-06-19 13:38:25+00:00	2023-06-19
9	2023-05-09 12:42:36+00:00	2023-05-09
10	2023-02-17 18:02:09.911000+00:00	2023-02-17
11	2023-01-16 00:35:06+00:00	2023-01-16
12	2022-09-25 10:14:05.538000+00:00	2022-09-25
13	2022-09-20 16:11:02.954000+00:00	2022-09-20
14	2021-12-09 20:10:27.563000+00:00	2021-12-09
15	2019-09-24 15:28:09.886000+00:00	2019-09-24
16	2019-06-30 23:52:34+00:00	2019-06-30
17	2024-11-18 08:17:36.435000+00:00	2024-11-18
18	2024-10-29 17:57:30.674000+00:00	2024-10-29
19	2024-02-26 09:10:04.552000+00:00	2024-02-26
20	2022-11-28 19:23:42+00:00	2022-11-28

Task 4 VISUALIZATION (Amazon Data)

1. Loading data

```
'''Amazon Data'''

# Step 1: Load Data
# Connect to the SQLite database and fetch relevant columns
conn = sqlite3.connect("sentiment.db")
query = """
SELECT SID, Product, Dateconverted AS date, Sentiment
FROM reviews
WHERE date IS NOT NULL AND Sentiment != ''
"""
df = pd.read_sql_query(query, conn)  # Load the query result into a pandas DataFrame
conn.close()  # Close the database connection
```

2. Sentiment Mapping

```
#Step 2: Transform Data
# Map sentiment labels (e.g., "Positive", "Negative") to numerical sentiment scores
sentiment_map = {'Strong Positive': 2, 'Positive': 1, 'Neutral': 0, 'Negative': -1, 'Strong Negative': -2}
df['SentimentScore'] = df['sentiment'].map(sentiment_map)
```

3. Sentiment Trend Calculation:

```
#Calculate sentiment trend: Group by product and date, and compute the average sentiment score
grouped_trend = df.groupby(['product', 'date'])['SentimentScore'].mean().reset_index()
```

4. Sentiment Distribution Calculation:

```
#Calculate sentiment distribution: Count the occurrences of each sentiment per product
distribution = df.groupby(['product', 'sentiment']).size().reset_index(name='Counts')

#Calculate the total count of sentiments per product
total_counts = distribution.groupby('product')['Counts'].sum().reset_index(name='Total')

#Merge the total counts with the distribution data to calculate percentage contribution
distribution = distribution.merge(total_counts, on='product')
distribution['Percentage'] = (distribution['Counts'] / distribution['Total']) * 100
```

5. Visualizing data using Line charts

```
#Add line plots (trend) and pie charts (distribution) for each product
for i, product in enumerate(unique_products):
    #Line Chart: Plot sentiment trend over time for the current product
    product_data = grouped_trend[grouped_trend['product'] == product]
    ax = axes[i * 2]  # Select the appropriate subplot for the line chart
    sns.lineplot(data=product_data, x='date', y='SentimentScore', ax=ax)
    ax.set_title(f"Sentiment Trend for {product}")
    ax.set_xlabel("Date")
    ax.set_ylabel("Average Sentiment Score")

    #Format x-axis dates for better readability
    ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))  #Format dates as YYYY-MM-DD
    ax.xaxis.set_major_locator(mdates.MonthLocator())  #Show major ticks at the start of each month
    ax.tick_params(axis='x', rotation=45)  #Rotate x-axis labels for better visibility
```

6. Visualizing data using Pie charts (Product and Sentiment)

```python
#Pie Chart: Plot sentiment distribution for the current product
product_distribution = distribution[distribution['product'] == product]
pastel_colors = sns.color_palette("pastel")  #Use pastel colors for better visuals
pie_colors = pastel_colors[:len(product_distribution['sentiment'])]  #Assign colors to pie sections
ax = axes[i * 2 + 1]  #Select the subplot for the pie chart
wedges, texts, autotexts = ax.pie(
    product_distribution['Percentage'],
    labels=None,  #Exclude labels directly on the pie chart
    autopct='%1.1f%%',  #Show percentages with 1 decimal place
    startangle=140,  #Rotate the pie chart for consistent positioning
    colors=pie_colors,
    shadow=True  #Add shadow for better depth effect
)
```

7. Displaying plots

```python
#Adjust layout to prevent overlapping elements
plt.tight_layout()

#Display all plots
plt.show()
```

8. Adding Python Script to Power BI desktop

9. Data analysis and visualization in Power BI

Amazon Visualizations

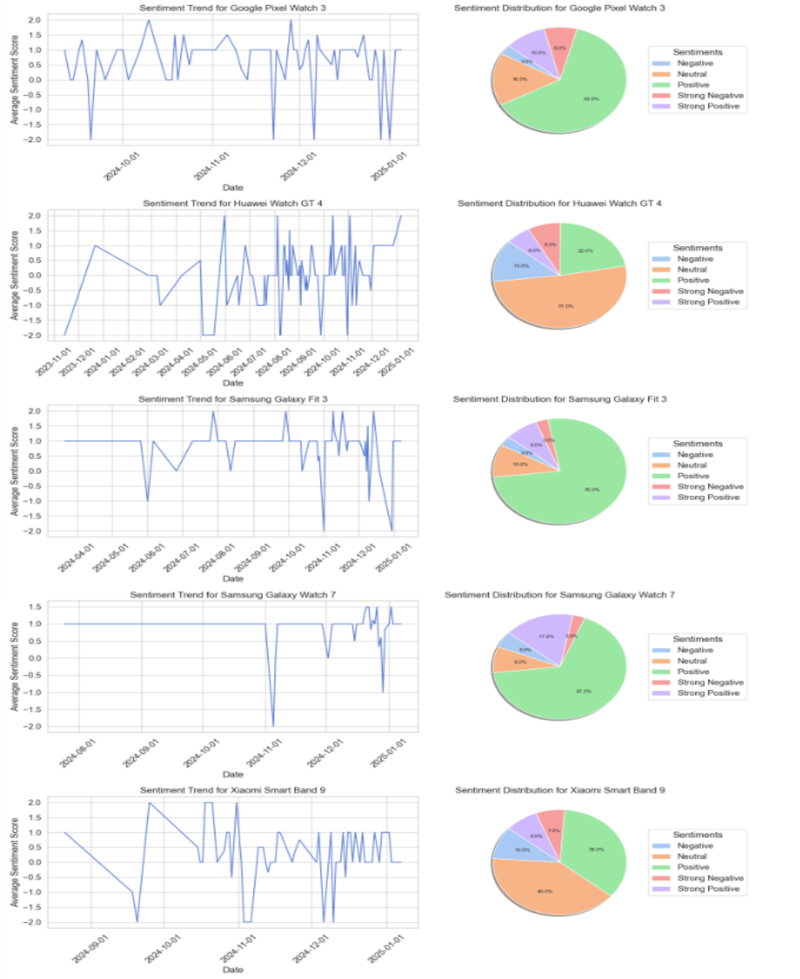
The sentiment fluctuates frequently, likely due to software updates, feature releases, or varying user experiences with battery life and performance.

A downward trend is visible in some periods, which could be attributed to software glitches, inaccurate health tracking, or delays in firmware updates.

The unstable sentiment trend might be influenced by firmware updates, mixed user experiences with fitness tracking accuracy, or app synchronization issues.

Stable sentiment trend with minor fluctuations suggests consistent performance. Occasional dips may be due to reports of overheating, battery drain, or feature expectations not being met.

Sharp declines and peaks indicate varying user satisfaction, possibly influenced by firmware updates, Bluetooth connectivity issues, or accuracy concerns in fitness tracking.



A significant portion of reviews are positive, indicating overall user satisfaction. However, negative feedback might be due to issues like software bugs, compatibility problems, or high pricing.

The balanced sentiment distribution suggests that while users appreciate design and battery life, some face challenges with app support and integration with non-Huawei devices.

Mostly positive sentiment indicates good reception, likely due to affordability and basic fitness tracking features. Negative reviews may stem from limited smartwatch functionality or display quality.

With over 60% positive reviews, users likely appreciate the premium design and features. Negative feedback might be related to price, battery performance, or software-related issues.

While the majority of reviews are positive, a noticeable portion of negative feedback suggests recurring issues like app stability, step tracking accuracy, or strap durability.

Mastodon Data

1. Data Processing

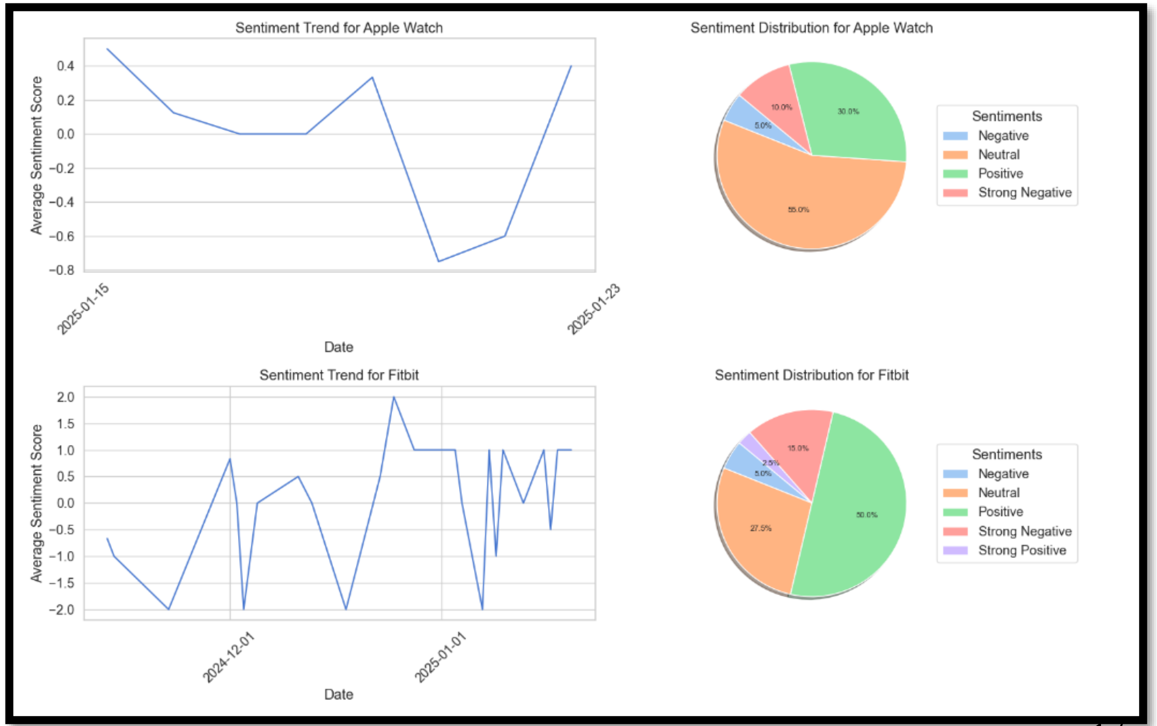
- Extracted relevant columns: Product, Date, Sentiment.
- Converted date to datetime format.
- Mapped sentiment labels to numerical scores:

2. Sentiment Analysis

- Calculated average sentiment score per product and date.
- Computed sentiment distribution percentage for each product.

3. Visualizations

- **Line Chart:** Shows sentiment trends over time for each product.
- **Pie Chart:** Displays sentiment distribution as percentages.



Data visualization and analysis in Power BI

Mastodon Visualizations

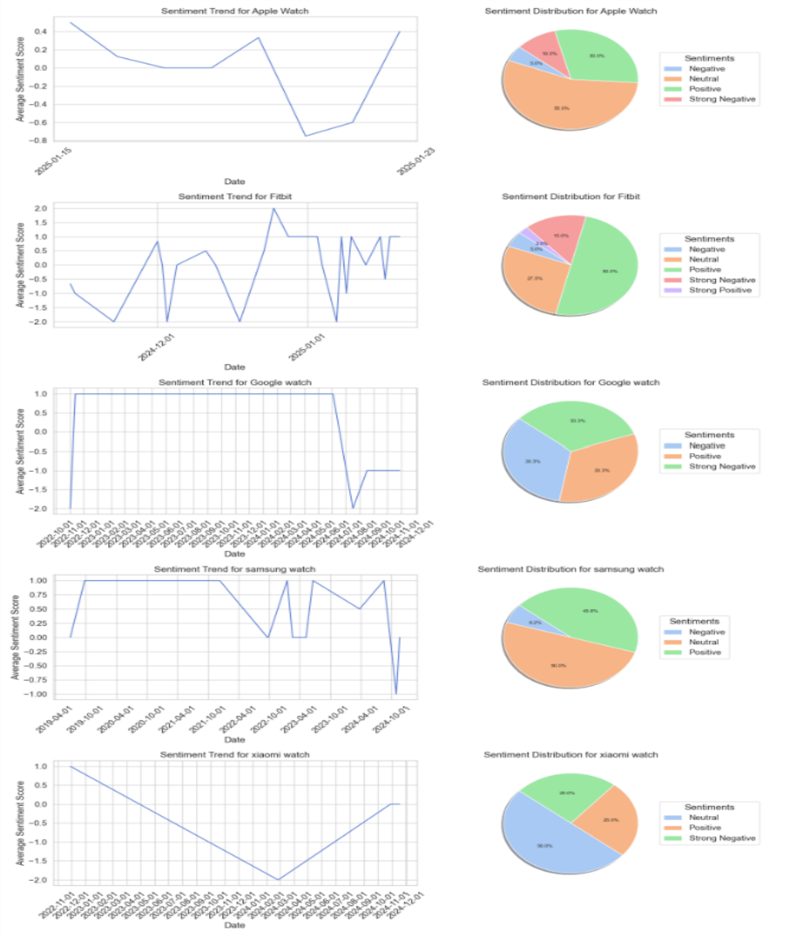
The sentiment trend shows fluctuations, likely influenced by software updates, battery life concerns, and user experiences with new features.

The trend indicates some negative spikes, which could be due to concerns over Fitbit's accuracy in health tracking, syncing issues, or recent policy changes.

The relatively stable sentiment trend might indicate limited discussion or strong but consistent opinions on the device, possibly due to a lack of major updates or issues.

There are some sharp sentiment drops, possibly due to software glitches, connectivity problems, or battery drain issues.

A steady upward trend suggests growing user satisfaction, potentially driven by affordability, value-for-money features, and recent software improvements.



The sentiment is mostly positive, suggesting user satisfaction with design and functionality. The negative portion may stem from pricing concerns, ecosystem lock-in, or hardware durability issues.

A fairly mixed sentiment distribution suggests that while users appreciate fitness tracking features, complaints may revolve around app integration, battery life, and device reliability.

The sentiment is balanced, with both positive and negative feedback. Users likely appreciate the software ecosystem, but challenges with battery performance and app optimization could be influencing negative reviews.

Mostly positive reviews indicate satisfaction with features and design. The negative share might be due to software updates causing unexpected issues or concerns over long-term durability.

A mix of positive and neutral sentiment suggests that while the device meets expectations in terms of affordability, some users face concerns over accuracy in tracking and build quality.

Conclusion

This project effectively analyzed user sentiments on **smart and fitness watches** using data from **Mastodon** and **Amazon**. By implementing an **ETL pipeline**, sentiment classification, and visualization, we tracked **trends over time** and examined **sentiment distribution per product**.
The results provide valuable insights into **consumer perceptions**, helping to identify shifts in sentiment and overall user satisfaction. The **Power BI visualizations**—including **time-series charts and pie charts**—offer a clear representation of sentiment trends.
This project lays the foundation for **further enhancements**, such as **expanding sentiment categories, incorporating machine learning models, or enabling real-time analysis**, making it a powerful tool for **market research and customer feedback analysis**.