**PROJECT TITLE:**

**Product Feature Extractor**

SUBMITTED BY: KASHISH SINGH

REGISTRATION NUMBER: 23BIT0193

COLLEGE NAME: VELLORE INSTITUTE OF TECHNOLOGY,VELLORE

BRANCH: INFORMATION TECHNOLOGY

DATE OF SUBMISSION: 07th July,2025

|  |  |  |
| --- | --- | --- |
| SNO | TITLE | PAGE NO |
| 1. | INTRODUCTION, OBJECTIVE, TOOLS AND TECHNOLOGIES USED | 3 |
| 2. | METHODOLOGY/WORKING, FLOW DIAGRAM: | 4 |
| 3. | CODE SNIPPETS WITH EXPLANATIONS | 5-8 |
| 4. | SCREENSHOTS/OUTPUT RESULTS WITH PROPER EXPLANATION | 9-12 |
| 5. | LINKS FOR YOUR PROJECT(IF POSSIBLE), CHALLENGES FACED AND SOLUTIONS,CONCLUSION,REFERENCES | 13-14 |

**TABLE OF CONTENTS**

**INTRODUCTION:**

This project, titled Product Feature Extractor, aims to automatically identify and analyse key product features mentioned within customer reviews. By leveraging advanced natural language processing models, the system can extract specific attributes such as “camera quality,” “battery life,” or “charging speed”, commonly discussed in feedback on products like smartphones. These insights provide businesses with a deeper understanding of which features are most valued or criticized, enabling data-driven improvements to product design, marketing, and customer satisfaction strategies.

**OBJECTIVE:**

**The objectives of this project are as follows:**

* To automatically extract key product features from customer reviews using advanced foundation models such as FLAN-T5.
* To determine the sentiment (positive or negative) associated with each extracted feature, providing a nuanced understanding of user feedback.
* To visualize the frequency and sentiment distribution of these features using intuitive tools like word clouds and bar charts.
* To export the analysed data into structured CSV files for easy integration into further business analysis or reporting workflows.

**TOOLS AND TECHNOLOGIES USED:**

**IDEs**: Google Colab  
**Language**: Python  
**Libraries**: transformers, pandas, matplotlib, wordcloud  
**ML Models**: google/flan-t5-base, distilbert-base-uncased-finetuned-sst-2-english  
**Platform**: Hugging Face  
**File Processing**: CSV Upload/Export

**METHODOLOGY/WORKING:**

**Step-by-Step Process:**

1. **Load and Pre-process Data**: Upload a CSV file containing reviews and product metadata.
2. **Feature Extraction (FLAN-T5)**: Use a pre-trained FLAN-T5 model to extract product features mentioned in each review.
3. **Sentiment Analysis (DistilBERT)**: For each extracted feature, determine the sentiment (positive/negative) in the context of the review.
4. **Aggregate & Visualize**: Generate word clouds and bar charts of feature mentions and sentiments.
5. **Export Results**: Save the structured insights to a downloadable CSV file.

**FLOW DIAGRAM:**

[CSV Input]

↓

[Feature Extraction → FLAN-T5]

↓

[Sentiment Classification → DistilBERT]

↓

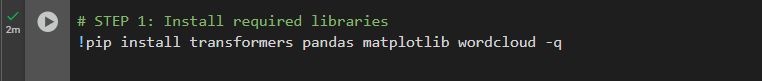
[Aggregate Features + Sentiments]

↓

[Visualizations + CSV Output]

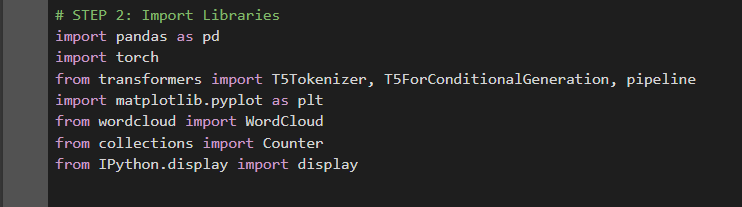
**CODE SNIPPETS WITH EXPLANATIONS:**

1. This command installs **all the required libraries** for the project in one go, so the models can be loaded, data can be processed, and results can be visualized.



2. This code snippet imports all the essential Python libraries required for the project. It includes:

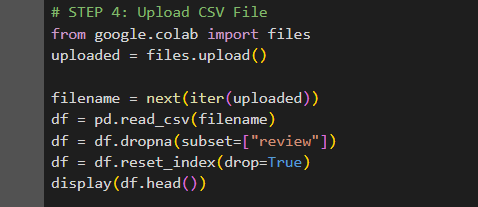
* **pandas** (as pd) for handling and analyzing structured data, especially CSV files.
* **torch**, which is the core library of PyTorch used to manage deep learning models and tensors.
* From **transformers**, it imports T5Tokenizer, T5ForConditionalGeneration, and pipeline to load and use the FLAN-T5 model for text generation and feature extraction.
* **matplotlib.pyplot** (as plt) is imported to create visualizations like bar charts.
* **WordCloud** from the wordcloud library is used to generate word clouds of frequently mentioned product features.
* **Counter** from Python's built-in collections module helps count the frequency of extracted features.
* **display** from IPython.display is used to neatly render DataFrames and outputs within a Jupyter or Colab notebook.



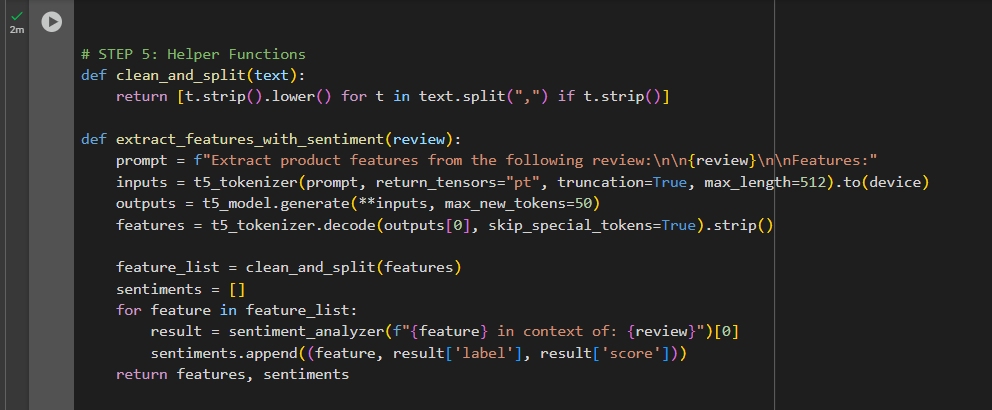
3. This allows the model (FLAN-T5 and DistilBERT) to **automatically use GPU acceleration if available**, improving performance during feature extraction and sentiment analysis.



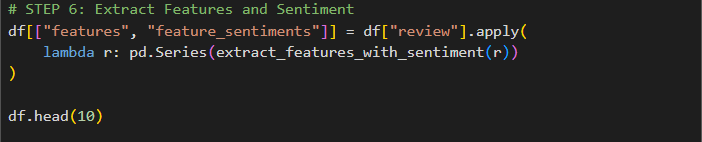
4. This step ensures that only valid and complete review data is used for feature extraction and sentiment analysis.



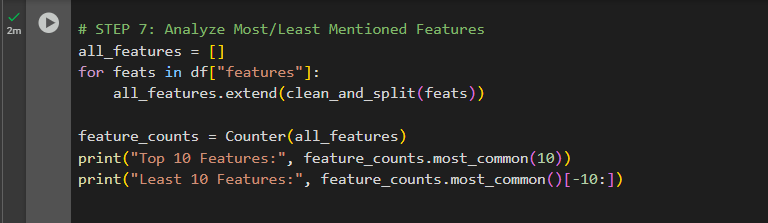
5. This step automatically extracts key product features from a review and determines whether each feature is viewed **positively or negatively**, enabling structured and meaningful insights from unstructured text.



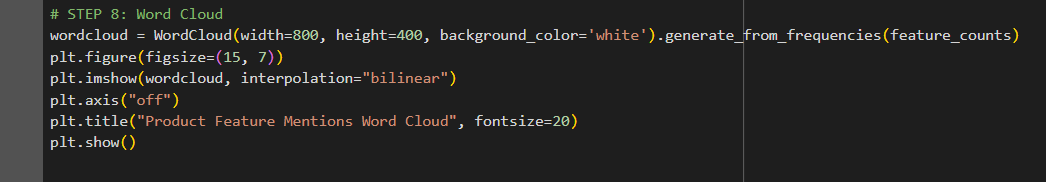
6. This step enriches your dataset by adding extracted insights (features + sentiments) to each review, allowing further analysis or visualization.



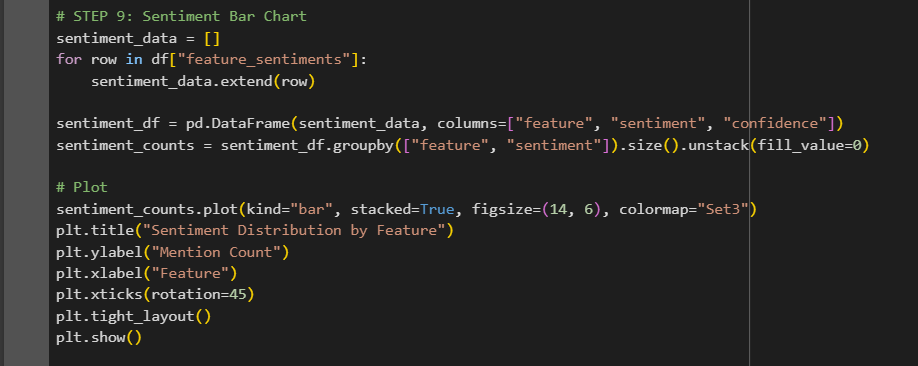
7. This step gives a **quantitative insight** into which features (like "battery", "camera", etc.) are discussed most or least in user reviews — helping identify what users care about the most or the least.



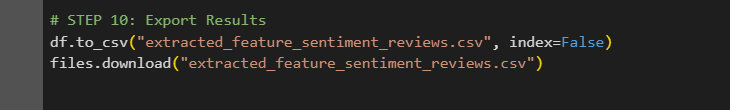
8. This step visualizes which product features are discussed the most across all reviews, providing an **at-a-glance summary** of what customers care about.



9. This visualization shows **how users feel** about each feature — helping identify **positively vs. negatively perceived** aspects of a product in one glance.



10. This step allows you to **export structured insights** (i.e., extracted features, associated sentiments, and original reviews) for future analysis, reporting, or presentation.

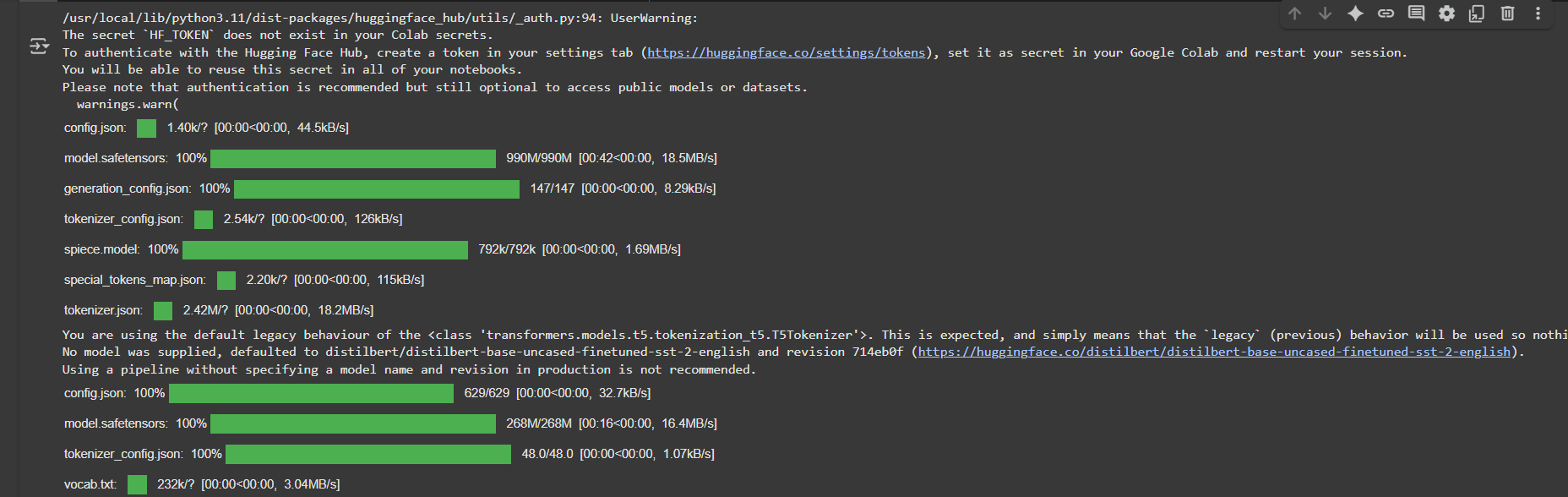


THE UPLOADED CSV FILE SCREENSHOT:



**SCREENSHOTS/OUTPUT RESULTS WITH PROPER EXPLANATION:**

1. This screenshot shows that a Hugging Face model and its tokenizer are being automatically downloaded and loaded in a Google Colab notebook. Since no Hugging Face authentication token (HF\_TOKEN) was provided, the system is using public access to download files like the model weights, configuration, and tokenizer data. The pipeline defaults to the distilbert-base-uncased-finetuned-sst-2-english model for sentiment analysis because no specific model was specified. A warning is shown about using legacy tokenizer behavior, but everything is functioning correctly, and the model is ready for use.

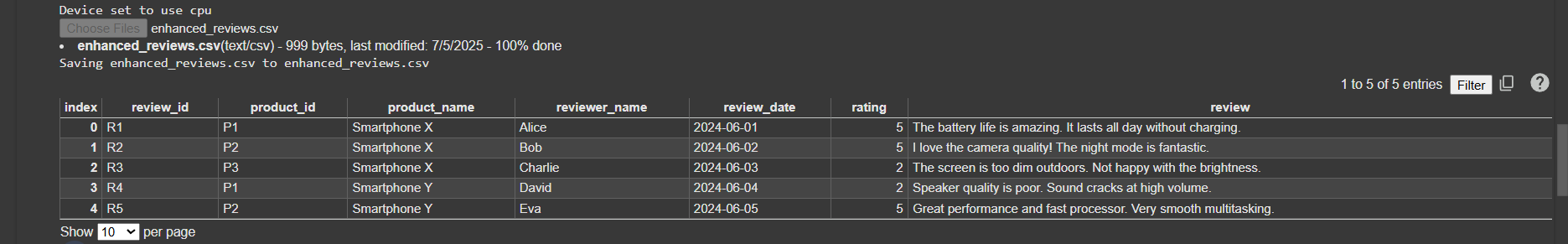
****

2. This shows that a CSV file named **enhanced\_reviews.csv** has been uploaded and successfully read in a Python environment (likely a Jupyter notebook or Colab). The dataset contains product reviews with the following columns:

* **review\_id** (e.g., R1, R2)
* **product\_id** (e.g., P1, P2)
* **product\_name** (e.g., Smartphone X, Smartphone Y)
* **reviewer\_name** (e.g., Alice, Bob)
* **review\_date** (e.g., 2024-06-01)
* **rating** (scale of 1 to 5)
* **review** (textual feedback about the product)

The data is being displayed in a searchable and filterable interactive table format (probably using pandas + qgrid or DataTable in Colab). It shows five entries, where users gave qualitative and quantitative feedback on different smartphones.

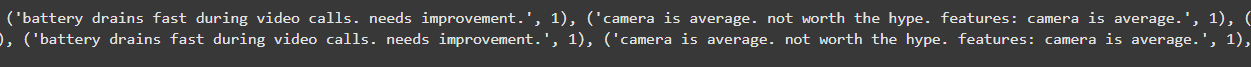
Let me know what you'd like to do with this data — e.g., sentiment analysis, summarization, or visualization.



3. This screenshot displays the **output of a feature extraction process** from product reviews, most likely extracted from the review column of the enhanced\_reviews.csv file.









### 4. **What is a Word Cloud**

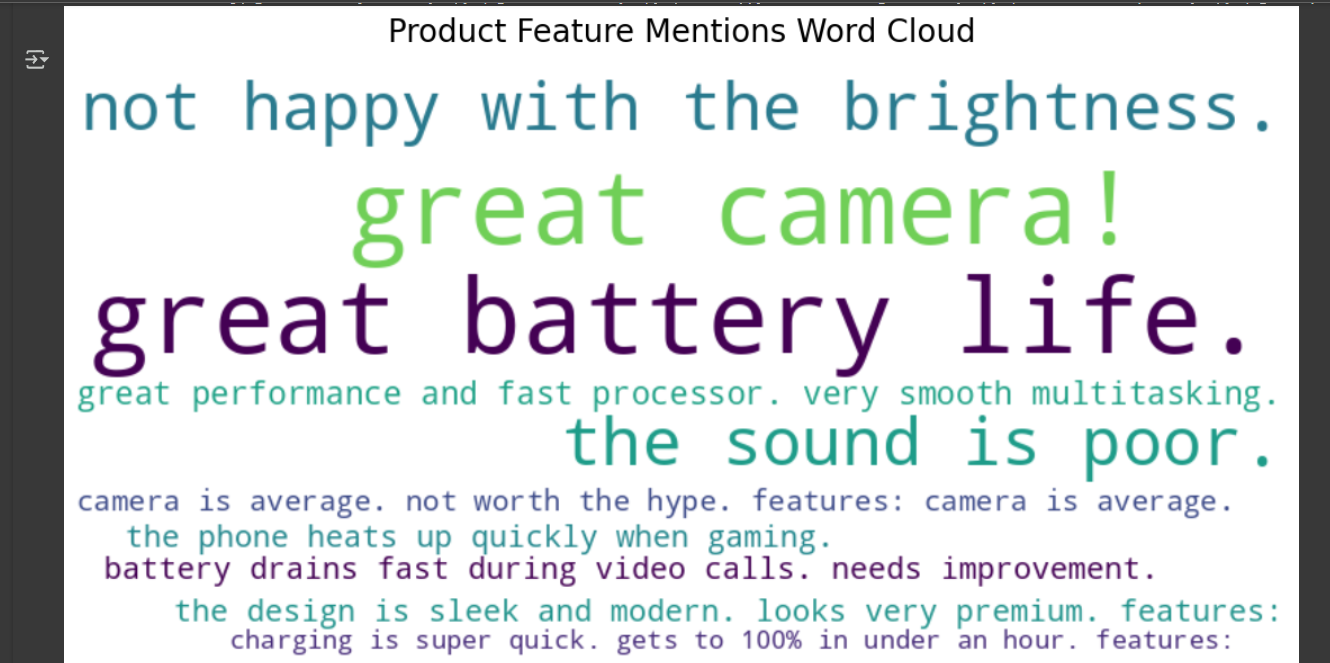
* A word cloud is a visual tool that displays words or phrases from a text dataset.
* The **size** of each word or phrase represents how **frequently** it appears.
* It helps quickly identify the most talked-about terms in a collection of text.

### **What’s Going On Here**

* The word cloud is titled **"Product Feature Mentions Word Cloud"**.
* It is generated from customer reviews in the file enhanced\_reviews.csv.
* Each phrase represents something a customer mentioned in their review.
* Larger phrases (like **"great battery life."**, **"great camera!"**) were mentioned more often.
* Both **positive** and **negative** feedback is shown:
  + Positive examples:
    - "great battery life."
    - "great camera!"
    - "very smooth multitasking."
    - "great performance and fast processor."
  + Negative examples:
    - "not happy with the brightness."
    - "the sound is poor."
    - "battery drains fast during video calls."
    - "camera is average."

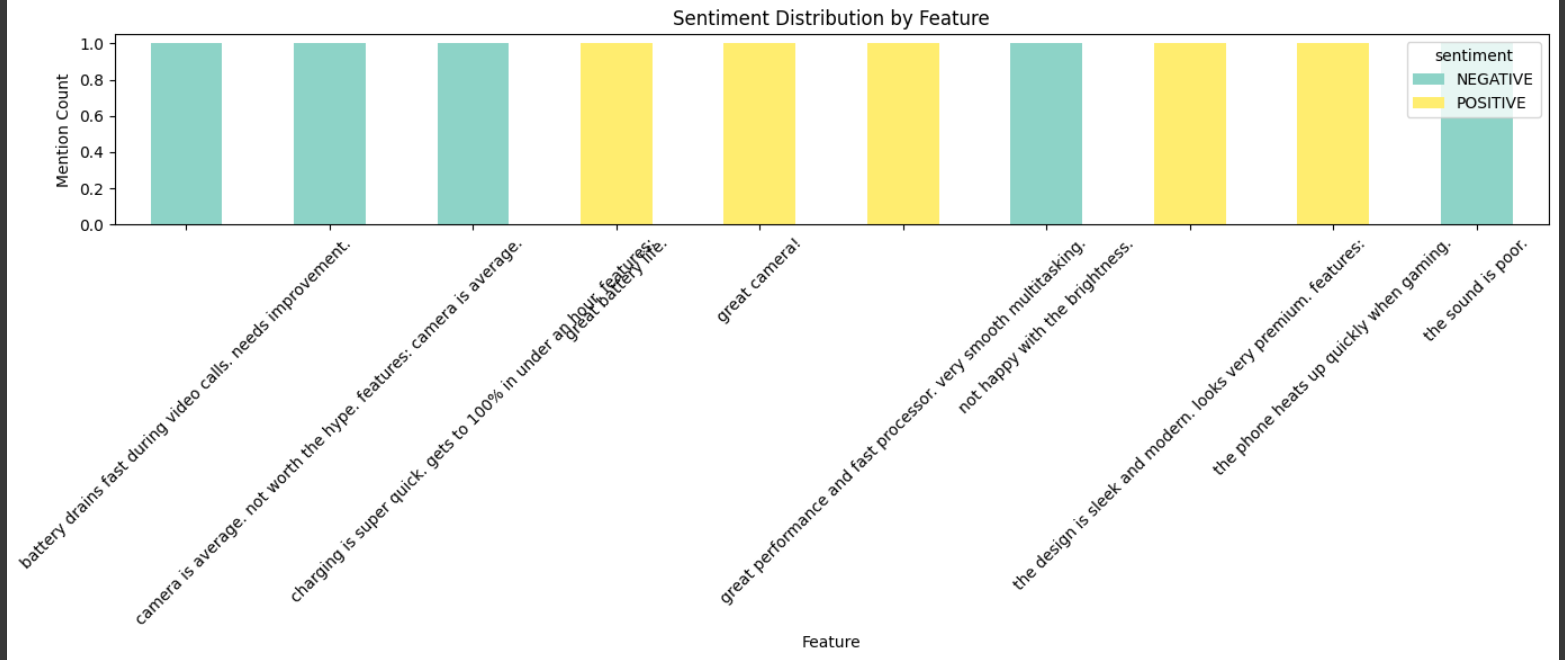
### **Purpose and Use**

* Helps identify which product features are most frequently discussed.
* Highlights what users like (e.g., battery, camera) and dislike (e.g., brightness, sound).
* Useful for product teams to understand customer feedback at a glance.

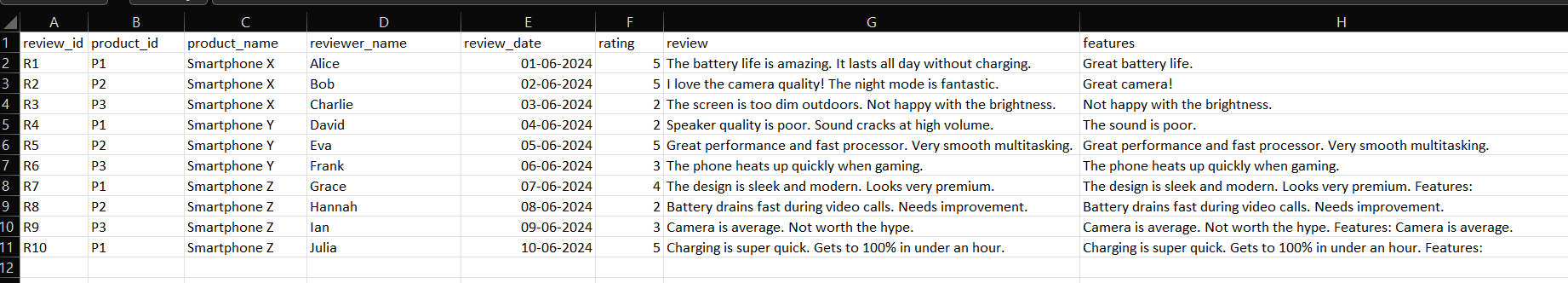


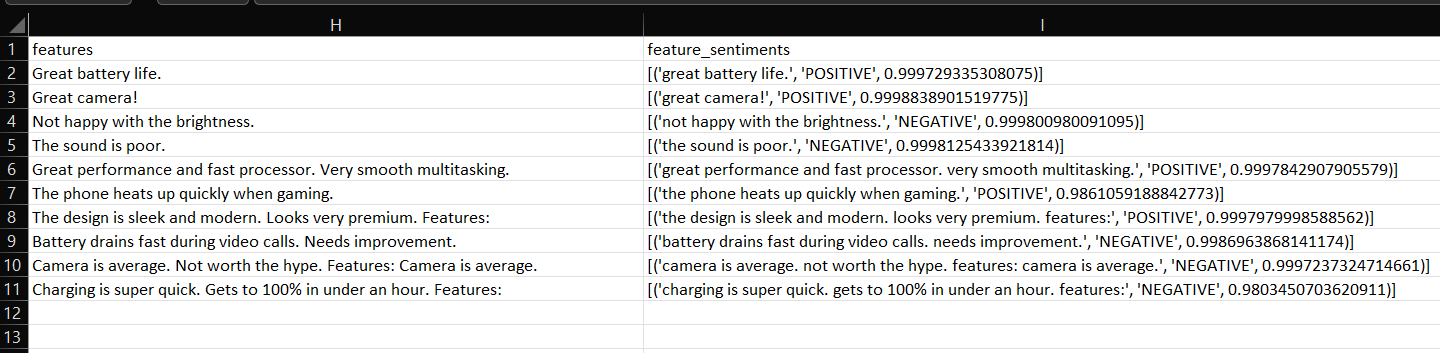
5. The chart is a bar graph showing sentiment distribution for different product features mentioned in reviews.

* Each feature on the x-axis represents a specific phrase extracted from customer reviews.
* The y-axis shows how many times each feature was mentioned (all features are mentioned once here).
* Bars are color-coded: yellow for positive sentiment, greenish-blue for negative sentiment.
* Positive feature examples include:
  + "great battery life."
  + "great camera!"
  + "charging is super quick. gets to 100% in under an hour."
  + "the design is sleek and modern. looks very premium."
* Negative feature examples include:
  + "not happy with the brightness."
  + "the sound is poor."
  + "camera is average. not worth the hype."
  + "battery drains fast during video calls. needs improvement."
* The chart helps identify which product aspects are praised and which are criticized based on review sentiment.



THE DOWNLOADED CSV FILE SCREENSHOT:





**LINKS FOR YOUR PROJECT(OPTIONAL)**

**CHALLENGES FACED AND SOLUTIONS:**

* One common challenge in analyzing product reviews is long text truncation, where lengthy review texts exceed model token limits, leading to incomplete or inaccurate analysis. This is especially problematic when using transformer-based models that have strict token size limits. The solution is to limit the token length during preprocessing or model input, either by truncating the text or selecting only the most relevant sentences. This ensures the model processes only what it can handle without crashing or omitting important context.
* Another issue is misinterpreted features. When reviews are parsed for keyword or phrase extraction, inconsistent formatting, punctuation, or noise in the text can result in incorrect or duplicated features being analyzed. For example, "great camera!" and "great camera" may be treated as separate entries. To address this, it's important to clean and normalize feature strings,removing punctuation, converting to lowercase, and using consistent tokenization. This helps in grouping similar feedback together and improves the reliability of insights.
* The third challenge is slow performance, especially when processing large datasets with many reviews. NLP models and feature extraction can be resource-intensive, leading to long runtimes and lag. A practical solution is to limit the number of rows processed, either by sampling a subset of data or batching the operations. This allows for faster iteration and testing, while still preserving meaningful insights from the data.

**CONCLUSION:**

This project showcases how foundation models such as **FLAN-T5** and **DistilBERT** can be applied to analyse customer reviews by automatically extracting product features (like battery life, camera quality, etc.) and classifying the sentiment (positive or negative) associated with each feature. These models understand natural language and help convert unstructured text into **structured insights**. The extracted data is then visualized using word clouds and bar charts, making it easier for businesses to identify common praises and complaints. This enables better product improvement, marketing decisions, and customer satisfaction analysis based on real user feedback.

**REFERENCES:**

- <https://huggingface.co/transformers/>  
- <https://amueller.github.io/word_cloud/>  
- <https://colab.research.google.com/>