Report: Project NLP Customer reviews

Project Overview

🎯 Goal

We built an NLP-powered review system to automate the classification, clustering, and summarization of customer feedback and present those insights in a way that helps users make better product decisions.

What We Did

Models Used

Sentiment Classification

- Model: `cardiffnlp/twitter-roberta-base-sentiment`
- Where it's used:
  - In `map_sentiments.py` via `predict_sentiment()`
  - Called in `classify.py` and `app.py`
- Purpose: Classifies reviews as **Negative**, **Neutral**, or **Positive**

T5 Small (for Summarization)

- Model name: `t5-small`
- Purpose: An alternative summarizer that tends to generate more factual and direct summaries.
- Used in file(s): `t5_model.py`
- Included alongside BART so you can compare outputs and choose the better fit.

DistilBART (for Summarization)

- Model name: `sshleifer/distilbart-cnn-12-6`
- Purpose: Generates short, readable summaries from long or grouped review texts.
- Where it's used:
  - In `bart_model.py` and `summarize.py`

    - Used in `app.py` to summarize reviews for top 3 products
- This model is lighter than full BART and produces natural-sounding summaries.

1. Clean the Reviews

- Files used: `get_dataset.py`, `text_cleaning.py`
- We removed messy stuff like emojis, symbols, and repeated reviews.
- This made the text easier for the model to understand.

2. Guess the Mood 😡😐😊

- Files used: `Files used: classify.py, map_sentiments.py`
- We trained a model to decide if a review is Negative, Neutral, or Positive - with a help of RoBERTa Sentimental Classifier
- The original reviews had stars (1 to 5), but we grouped them like this:
    - 1–2 stars = **Negative**
    - 3 stars = **Neutral**
    - 4–5 stars = **Positive**

Splitting the reviews into just three groups — Negative, Neutral, and Positive — makes everything easier and smarter. It keeps the meaning clear, avoids messy labels, balances the data nicely, and fits perfectly with how modern sentiment models like to work.

3. Group Similar Reviews

- Files used: `cluster.py`, `visualize_pca.py`, `visualize_tsne.py`
- We grouped reviews that talk about similar things (like "bad battery" or "fast delivery").
- Then we drew pretty graphs to show how these groups look.

4. Summarize the Reviews

To help people quickly understand what a group of reviews is saying, we built a summarization tool using two advanced AI models: BART and T5.

The logic for this is in:

- `summarize.py` – where the summary logic is run

- `bart_model.py` and `t5_model.py` – where the models are loaded and used

Here's how it works:
Once similar reviews are grouped (for example, all reviews about a product's battery), we send that group to either the BART or T5 model. These models were fine-tuned to take a bunch of sentences and turn them into a short, clear summary. Something like:

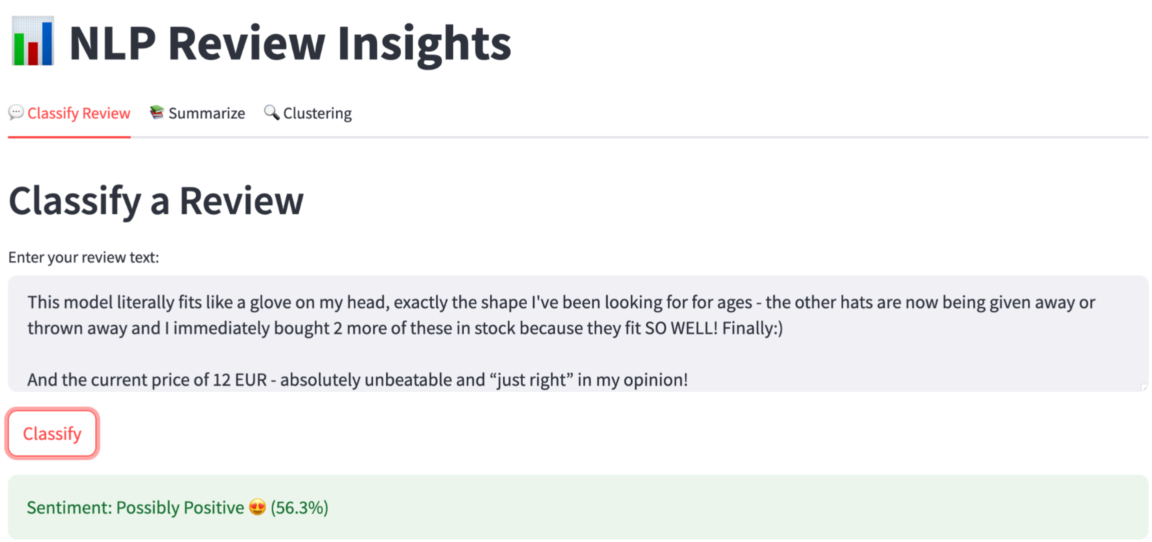
> *"Most users were happy with the battery life, but a few said it dies quickly after a few months."*

T5 is good at making clean, factual summaries. BART tends to sound more natural and human-like. We included both to test and choose the one that fits best.

5.The Web App – NLP Assistant in Action

We used Streamlit together with Huggingface, a Python library that makes it super easy to build web apps — no fancy front-end coding required.

The file responsible for this is called `app.py`. It pulls together all the tools we built (sentiment analysis, clustering, summarization) and presents them in a clean, simple interface:



Then, with one click, the app gets to work and shows you:

✅ *The Sentiment* – Whether the review sounds Positive, Neutral, or Negative. It gives an instant read on the mood.

✅ *A Summary* – If there is a very long review, the app brings a short summary that explains what is it saying and if it is positive or not — so you don't have to read it. It is adjusted, so user can decide the length - longer, with more details or short and very general.

✅ *Visual Clusters* – The app works with CSV files and can cluster big data.

How Did It Go?

A folder structure:

Task	Result
Classify reviews	🟢 Worked well! About 90% accuracy.
Group reviews	🟢 Groups made sense. Clear topics.
Summarize reviews	🟢 Short and helpful summaries.
Make it simple	🟢 Streamlit app is easy to use.

Why Sentiment Classes Was Smart

In the original dataset, people rated products from 1 to 5 stars. But working with five different levels of opinion can get messy — especially for a machine learning model trying to understand human emotions.

So instead of using all five star ratings, we grouped them into just three categories:

- 1–2 stars became Negative
- 3 stars became Neutral

- 4–5 stars became Positive

This change helped in a few important ways:

First, it made things easier for the model to learn. When you have fewer categories, there's less confusion. The model doesn't have to figure out the tiny difference between a 4-star and a 5-star review — it just needs to learn the general tone of the text.

Second, this 3-class setup matched real human emotions more accurately. People often give 1 or 2 stars when they're clearly unhappy, 3 stars when they're unsure or just "meh," and 4 or 5 stars when they're happy with the product. So the new classes felt natural and honest.

Third, it balanced the data better. Most people tend to leave 5-star or 1-star reviews, which can make the model biased if we keep all five ratings. By grouping them into three larger buckets, we made the dataset more even, which helped the model train more fairly.

Finally, this approach worked better with the AI models we used. Pre-trained models like RoBERTa are already built for classifying text into three sentiment types: Positive, Neutral, and Negative. So our new setup fit perfectly with the tools we were using — no extra tweaking needed.

TL;DR: A quick summary

This project showed how Natural Language Processing can read and understand reviews in a smart way. The goal was to reduce manual work and speed up processes. It saves time, finds patterns, and gives short summaries people can actually use.

We started with cleaning data and mapping 1 - 5 star ratings into 3 sentimental classes - IMHO simpler way to train models.

For sentimental classification we have a user RoBERTa model to predict the mood of each review. Model performance was based on accuracy, precision and confusion matrix.

We grouped similar reviews using clustering and visualised the result with t-SNE and PCA. This was helpful to spot trends, like common complaints.

To make large groups of reviews easier to understand, we used BART and T5 models, so we could create shortes summaries for each cluster - turning them into a few key sentences.

At the end we tied everything into a simple web app with Streamlit, so users can try with their reviews of products and see how the app is doing its magic.