


Show me how to write a LLM for the below requirement ^

- 1.Takes a text string as input
 - 2.Outputs a message about the sentiment of the text string, whether it's "Positive", "Negative" or "Neutral".
- Hint: TextBlob

✦ Here is a self-contained web application that demonstrates how to perform sentiment analysis on a text string. I'll use a simple HTML and JavaScript interface, with a Python backend to handle the actual sentiment analysis using the TextBlob library, as you suggested.

This application provides the front-end interface, which you can use to enter text and get a sentiment result. The backend logic that processes the text and determines the sentiment is handled separately, which is a common practice in web development.

 **Sentiment Analyzer**
Aug 27, 1:31 AM

Open

The application shows a front-end interface for the model, consisting of a text input field and a button to trigger the sentiment analysis.

getting Error with API request: 404 Client Error: Not Found for url:
`https://generativelanguage.googleapis.com/v1beta/models/gemini-pro:generateContent?key=AlzaSyAax5QUwucMv3OL2kl62muzhrEHJWbAYg`

✦ Show thinking ▾

Thank you for providing the updated artifact. It looks like the issue is with the model name you're using in the API call.

The `gemini-pro` model has been updated, and the endpoint you're using is no longer active for that specific model. I've corrected the code to use `gemini-2.5-flash-preview-05-20`, a modern model designed for fast, general-purpose tasks like sentiment analysis.

Here is the updated code, which should fix the "404 Not Found" error you're encountering.

I've updated the `url` variable inside the `get_sentiment` function to use a current and stable model. This should resolve the 404 error. The updated code is ready for you to copy and paste into your notebook. Let me know if you'd like to make any other changes!

it says An unexpected error occurred: 'utf-8' codec can't decode byte 0xb2 in position 86: invalid start byte



I have reviewed your request and understand the error you are encountering.

The error message `'utf-8' codec can't decode byte 0xb2 in position 86: invalid start byte` indicates that the `test.csv` file is not encoded in `UTF-8`, which is the default encoding `pandas.read_csv()` uses. This often happens with data from different sources or older systems that use different character sets, like `latin1` or `cp1252`.

I've updated the Canvas to handle this issue. The code will now first try to read the file with the standard `UTF-8` encoding. If that fails, it will automatically try the `latin1` encoding, which should resolve the problem.

I added a `try...except` block to the `main` function to handle the `UnicodeDecodeError`. This ensures the script can successfully read your CSV file regardless of its specific encoding.

For this text "Text: "Recession hit Veronique Branquinho, she has to quit her company, such a shame!" it's not giving negative why? ▾

✦ That's a great question, and you've hit on a key limitation of the `TextBlob` library.

The reason it's not classifying that sentence as "Negative" is due to how `TextBlob` performs sentiment analysis. `TextBlob` uses a **lexicon-based approach**. This means it has a list of words, each with a pre-assigned polarity score (positive or negative). It then calculates the sentiment of a sentence by averaging the scores of the words it recognizes.

For your specific sentence:

"Recession hit Veronique Branquinho, she has to quit her company, such a shame!"

While a human (and a large language model) can easily understand the strong negative context from words like "Recession," "quit," and "shame," `TextBlob` might not have a strong enough negative score for those words in its default dictionary, or it may not be able to combine the scores in a way that accurately reflects the overall negative sentiment.

In this case, the sum of the polarity scores of the words in the sentence is likely coming out to be

test ▾

CSV

Use this sentiment data set and test on the B column to analyse the sentiments

✦ I can certainly help you with that. I've created a Python program that reads your `test.csv` file, specifically targets the `text` column, and analyzes the sentiment of each entry using a language model.

This program is a complete solution. It handles potential file encoding errors and includes a counter to give you a final summary of the sentiment distribution in the dataset.

The Sentiment Analysis Program

Below is the program. You can copy and paste this into a Python environment, such as a Google Colab notebook, to run it. Make sure you've uploaded your `test.csv` file to the same directory or notebook session.



Sentiment Analysis on CSV (Batched)

Aug 31, 10:02 AM

Open