Sentiment Analysis API with LLM Integration

Name: Mausam Sharma

Date: September 29, 2024

Github: link: https://github.com/9785218943/sentiment-analysis-api

Introduction: This project aims to develop a Python-based API that performs sentiment analysis on customer reviews using a Large Language Model (LLM) like the Groq API. The API accepts files containing customer reviews and returns structured sentiment analysis results.

Approach to Solving the Problem: The API was developed using the Flask framework. It accepts either CSV or XLSX files containing customer reviews. The reviews are extracted from the file and sent to the Groq API for sentiment analysis. The API then returns a structured response in JSON format.

Structured Response Implementation: The API returns the sentiment analysis result in the following format:

Json

Copy code

```
{ "positive": score,
 "negative": score,
 "neutral": score
}
```

Examples of API Usage:

Sample Input File (CSV or XLSX)

| Review | |----------------------------------|

| "Great product, very happy!" |

| "The service was poor." |

Sample API Request and Response **API Request**:

POST /upload-file:

API Response:

```json

{

"positive": 0.75, "negative": 0.25, "neutral": 0.00

}

**Error Handling:** The API implements basic error handling to ensure valid file formats (CSV/XLSX). If an invalid file type is uploaded, the API returns an appropriate error message. Additional error handling is included for external API failures and invalid data.

**Analysis of Results:**

### Limitations: - Dependency on the external Groq API. - Limited to 50 reviews for optimal performance. ### Potential Improvements: - Implement retry mechanisms in case the external API fails. - Add support for larger datasets using batch processing.nalysis of Results:

**Conclusion:** The Sentiment Analysis API successfully processes customer reviews and provides sentiment analysis. While the project performs well for small datasets, there is room for improvement in handling larger datasets and improving error resilience.
```