Project Description: Customer Frustration Alerting System

Overview

The Customer Frustration Alerting System is a Flask-based application designed to detect and respond to customer frustration in real-time. Using natural language processing (NLP) techniques and sentiment analysis, the system evaluates customer messages for signs of negative sentiment and triggers alerts when frustration is detected. The alerts are sent to the DevRev platform for immediate action, ensuring timely customer support and improved satisfaction.

Key Features

1. Webhook for Automatic Analysis:
    - Listens for customer messages via a `/webhook` endpoint.
    - Analyzes the sentiment of incoming messages using the TextBlob NLP library.
    - Detects frustration based on a predefined sentiment polarity threshold (`FRUSTRATION_THRESHOLD`).
2. Frustration Alert System:
    - Sends alerts to the DevRev platform when customer frustration is detected.
    - Alerts include detailed information such as the customer ID, message content, and sentiment score.
    - Uses DevRev's API for seamless integration.
3. Slash Command for Manual Analysis:
    - Provides a `/slash-analyze` endpoint for analyzing messages manually.
    - Responds with immediate feedback on the sentiment score and frustration detection.
4. Health Check Endpoint:
    - Offers a `/health` endpoint to check the application's status and ensure it is running properly.

How It Works

1. Sentiment Analysis:
    - The system uses TextBlob to calculate the polarity of customer messages.
    - Polarity values range from -1 (very negative) to +1 (very positive).
    - If the polarity is below the frustration threshold (e.g., -0.3), the message is flagged as frustrated.

2. Alert Triggering:
    - When frustration is detected, the system constructs an alert payload with details such as message content, sentiment score, and customer ID.
    - The alert is sent to the DevRev platform using an authenticated API request.
3. Manual Analysis:
    - Users can submit messages for manual sentiment analysis using the `/slash-analyze` endpoint.
    - The system provides immediate feedback in a user-friendly format.

Technology Stack

- Backend: Flask (Python web framework)
- Sentiment Analysis: TextBlob (NLP library)
- API Integration: DevRev API for alert management
- Deployment: Runs on a local or cloud server accessible via HTTP endpoints

---

Endpoints

1. `/webhook` (POST):
    - Automatically analyzes customer messages and triggers alerts if frustration is detected.
2. `/slash-analyze` (POST):
    - Allows manual sentiment analysis of a message.
3. `/health` (GET):
    - Provides a status report of the application.

Use Cases

- Customer Support: Automatically detects frustrated customers, enabling proactive responses.
- Real-Time Monitoring: Provides a continuous analysis of customer sentiment to ensure a high-quality support experience.
- Actionable Insights: Offers sentiment insights to help support teams prioritize urgent issues.

Potential Enhancements

- Integrate with additional NLP libraries for multilingual sentiment analysis.
- Add support for multiple thresholds to detect different levels of sentiment (e.g., anger, dissatisfaction).
- Include a dashboard for real-time visualization of sentiment trends and alert metrics.

This project provides an intelligent and automated solution for improving customer experience by quickly addressing frustration, thereby fostering better engagement and loyalty.