**Project Documentation**

**Project Title: Sentiment-Based Chatbot using NLTK**

**Overview**

This is a simple rule-based chatbot that uses **Natural Language Processing (NLP)** to detect the sentiment of user input and respond accordingly. It classifies messages as **positive**, **negative**, or **neutral** using the **VADER (Valence Aware Dictionary and sEntiment Reasoner)** sentiment analysis tool from the nltk library.

**Objectives**

* To understand basic sentiment analysis in Python using NLTK.
* To create a simple interactive chatbot.
* To classify text input based on emotional tone and provide a matching response.

**How it Works**

1. The chatbot reads a message from the user.
2. It analyzes the sentiment score using **VADER**.
3. Based on the **compound score**, it classifies the sentiment as:
   * **Positive** (score ≥ 0.05)
   * **Negative** (score ≤ -0.05)
   * **Neutral** (between -0.05 and 0.05)
4. A random predefined response is returned based on the sentiment.

**Tools and Libraries**

* **Python**
* **nltk (Natural Language Toolkit)**

**Required Setup**

Before running the script, you need to **download the VADER lexicon** which is a pre-trained model used by SentimentIntensityAnalyzer.

python

CopyEdit

# Download required NLTK data (only needs to run once)

nltk.download('vader\_lexicon')

**What is this for?**

* vader\_lexicon is a dictionary of lexical features (words) mapped to emotion intensities.
* It’s used by SentimentIntensityAnalyzer to calculate sentiment scores for input text.
* Downloading it makes sure the analyzer has the data it needs.

### Notes

* This chatbot can be expanded with more specific or personalized responses.
* It's a base model for building advanced AI chatbots with NLP.
* Ideal for learning purposes or small-scale user testing.