**Technical Assessment: Chatbot - Marketplace Support Expert System**

# **Introduction**:

This Python ipynb file implements a simple chatbot capable of answering queries related to the market. The chatbot utilizes natural language processing (NLP) techniques to understand and respond to user input. It leverages the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization method and cosine similarity to identify the most relevant response from a predefined dataset.

## **Dataset**

* **Creation**: The dataset used by the chatbot was created using ChatGPT.
* **Files**: There are two text files used in the implementation:

**1. data.txt:** This file contains the dataset used by the chatbot to retrieve answers to user queries about the market.

**2. FAQ.txt**: This file contains a list of questions that users can ask to test the chatbot's functionality.

## **Libraries** **Used**

* **io**: Provides functions for handling file I/O operations.
* **random**: Used for generating random responses.
* **string**: Offers string processing utilities.
* **warnings**: Used to suppress warnings during execution.
* **numpy**: Provides support for mathematical functions and arrays.
* **sklearn**: Utilized for machine learning tasks, specifically for TF-IDF vectorization and cosine similarity calculations.
* **nltk**: Natural Language Toolkit library, used for NLP tasks like tokenization and lemmatization.

## **Data** **Processing**

* The script reads the dataset from the `data.txt` file.
* It tokenizes the raw text into sentences and words using NLTK's `sent\_tokenize` and `word\_tokenize` functions.
* Lemmatization is applied to normalize the words to their base forms using NLTK's WordNetLemmatizer.

## **Greeting Functionality**

The chatbot includes a greeting functionality where it responds to common greetings such as "hello," "hi," etc. with random responses.

## **Response** **Generation**

The `response` function generates an appropriate response to the user input by computing the TF-IDF vectors for both the user input and the dataset sentences.

Cosine similarity is then calculated between the user input vector and all dataset vectors to determine the most similar response.

The response with the highest cosine similarity score is selected as the output.

## **Main Loop**

The main loop of the chatbot continuously accepts user input until the user types "bye" to exit the conversation.

It handles user input by first checking for greetings and responding accordingly.

If the input is not a greeting, the chatbot generates a response based on the input query using the `response` function.

**Example Usage**

**IDRIS ROBO**: My name is IDRIS ROBO. I will answer your queries about Market. If you want to exit, type Bye!

**User**: Hello

**IDRIS ROBO**: hi there

**User**: How can I place an order?

**IDRIS ROBO**: to place an order, simply browse our website, select the desired items, and proceed to checkout.

**User**: Thank you

**IDRIS ROBO**: You are welcome..

## **Conclusion**

This chatbot prototype provides a basic framework for handling market-related queries using NLP techniques. It demonstrates how TF-IDF vectorization and cosine similarity can be used to generate relevant responses from a dataset. Additionally, the presence of the FAQ.txt file allows users to test the chatbot's functionality by asking predefined questions.