# SARANATHAN COLLEGE OF ENGINEERING

# CREATING CHATBOT USING PYTHON - Artificial Intelligence

- [Introduction](#introduction)

- [Project Overview](#project-overview)

- [Purpose](#purpose)

- [Literature Survey](#literature-survey)

- [Problem Statement](#problem-statement)

- [Solution architecture](#solution-architecture)

- [Problem solution](#problem-solution)

- [Data Collection](#data-collection)

- [Data Processing](#data-processing)

- [Labelling](#labelling)

- [data splitting](#data-splitting)

- [Feature Extraction](#feature-extraction)

- [Model Selection](#model-selection)

- [Evaluation](#evaluation)

- [User Interface](#user-interface)

- [How to run the project](#how-to-run-the-project)

- [Installation](#installation)

- [project file](#project-file)

- [Conclusion](#conclusion)

## Introduction

### Project Overview

This project involves the development of a chatbot using Python to facilitate human-computer interactions. The chatbot will be designed to understand and respond to user queries in a natural language using natural language processing (NLP) techniques. It will provide assistance, information, and engagement in a user-friendly manner, catering to a diverse range of user needs. The project will leverage Python's rich libraries, such as NLTK and spaCy, to enhance the chatbot's language understanding capabilities. The ultimate goal is to create a functional, intelligent, and versatile chatbot that can be deployed across various platforms and domains to enhance user experiences.

### Purpose

\*1. Automating Customer Support

\*2. Enhancing User Engagement

\*3. Data Analysis and Insights

\*4. Streamlining Information Retrieval

\*5. AI-Powered Conversational Experiences

## Literature Survey

### Existing Problem Statement

1)Natural Language Understanding (NLU): Developing chatbots that can accurately understand and interpret user input, especially in complex or context-dependent conversations, remains a challenge.

2)Context Handling: Maintaining context and continuity in conversations is a significant issue. Chatbots often struggle to remember previous interactions and provide contextually relevant responses.

3)Personalization: Tailoring chatbot responses to individual user preferences and behaviors is challenging, as it requires effective data management and machine learning algorithms.

4)Limited Domain Knowledge: Chatbots may lack domain-specific knowledge and struggle to provide detailed or accurate information in specialized fields like medicine, law, or finance.

5)Emotional Intelligence: Creating chatbots that can understand and respond to user emotions, such as empathy or frustration, is a complex problem in human-computer interaction.

6)Integration and Scalability: Integrating chatbots with various platforms, databases, and APIs, while ensuring scalability and performance, poses technical challenges.

7)Data Privacy and Security: Maintaining user data privacy and security while using chatbots to handle sensitive information is a growing concern.

8)Multimodal Interaction: Developing chatbots that can handle voice, text, and visual inputs and provide coherent responses across different modalities is a research challenge.

9)Ethical and Bias Issues: Ensuring that chatbots are designed and trained ethically and do not perpetuate biases is a critical problem in AI chatbot development.

10)User Acceptance and Trust: Building chatbots that users trust and are comfortable interacting with is a psychological and design challenge.

### Problem Statement

The challenge is to create a chatbot in Python that provides exceptional customer service, answering user queries on a website or application. The objective is to deliver high-quality support to users, ensuring a positive user experience and customer satisfaction.

## Problem solution

### Data Collection

Collect a large dataset of news articles, including both credible and fake news. The dataset should be diverse and representative.

### Data Processing

Preprocess the text data, including lowercasing, vectorisation, and removing stop words and punctuation.

### Labelling

Manually or semi-automatically label the collected data as either credible or fake news.

### data splitting

Split the dataset into Training,Validation and test sets to evaluate the models performance.

### Feature Extraction

Use the preprocessed data to extract features for model training

### Model Selection

Choose the best-performing model based on validation results.

### Evaluation

Evaluate the model's performance on the test dataset to assess its ability to detect fake news accurately.

### Continuous Monitoring

Continuously monitor the system's performance and retrain the model as new data becomes available.

### User Interface

Develop a user-friendly interface for users to submit news articles and receive fake news detection results.

## How to run the project

### Installation

You can find the installation documentation for the

[Jupyter platform, on ReadTheDocs](https://jupyter.readthedocs.io/en/latest/install.html).

The documentation for advanced usage of Jupyter notebook can be found

[here](https://jupyter-notebook.readthedocs.io/en/latest/).

For a local installation, make sure you have

[pip installed](https://pip.readthedocs.io/en/stable/installing/) and run:

```bash

pip install notebook

```

#### project file

\* project file is attacted

\* after installing the jupyter notebook, run the code in it

##Steps

Dataset link: https://www.kaggle.com/datasets/grafstor/simple-dialogs-for-chatbot

\* 1.Start: The process begins here.

\* 2.Input Text: The input to the system is the news article or text that needs to be checked for authenticity.

\* 3.Preprocessing:

\* Text Cleaning: Remove any irrelevant characters, punctuation, and special symbols.

\* Tokenization: Split the text into individual words or tokens.

\* Stopword Removal: Eliminate common words that do not carry much meaning (e.g., "the," "is," "and").

\* 4.Feature Extraction:

\* Extract features from the preprocessed text. These features might include word frequencies, TF-IDF scores, or n-grams.

\* 5.Machine Learning Model:

\* This is a simplified step. In a real-world scenario, this would be a machine learning or deep learning model trained on labeled data.

\* The model takes the extracted features and use that preprocessed data for giving the output to the user

\* 6.Output:

\* Chatbot gives answers for the user queries related to customer service

\* 8.End: The process ends here.

## Conclusion

The creation of a chatbot using Python for the AI project has yielded a versatile and intelligent conversational agent capable of understanding and responding to user queries. The chatbot's robust NLP capabilities, driven by Python libraries, enable it to interpret user intent and context effectively. User feedback mechanisms have been integrated, facilitating continuous learning and improvement over time. Python's scalability ensures the chatbot's performance, even in high-traffic scenarios. The project sets the stage for a promising future, with opportunities for personalization, domain-specific applications, and multimodal interactions.

## Contact details

Email id: harshini3331@gmail.com

I had attached my code in this repository