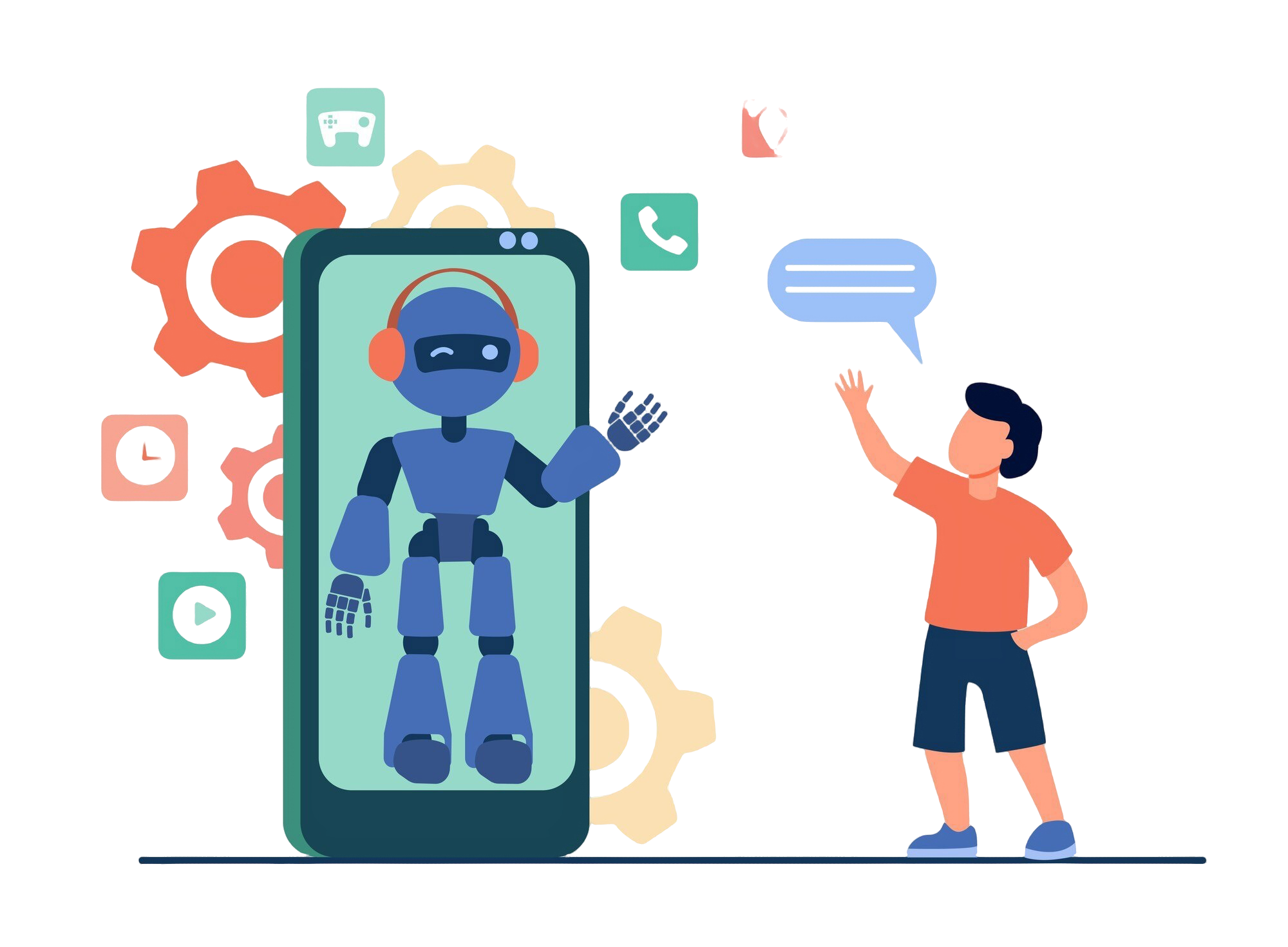
**Implementation of Chatbot using NLP**



**Problem Statement**

Develop a chatbot capable of understanding and responding to user inputs by identifying intents and extracting entities. The chatbot should provide relevant responses based on trained intents, assisting in creating seamless user interactions.

**Aim**

The aim of this project is to develop an intents-based chatbot using Natural Language Processing (NLP) techniques and Logistic Regression. The chatbot will provide a user-friendly interface to classify user inputs and deliver meaningful responses, enhancing the interactive experience.

**Learning Objectives**

The objectives of this project are to:

* Learn how chatbots process user input, recognize intents, and generate responses.
* Use tokenization and TF-IDF vectorization to preprocess and analyse textual data.
* Train and evaluate a Logistic Regression model for intent classification.
* Create and deploy an interactive chatbot interface using the Streamlit framework.

**About Project**

This project focuses on developing a functional chatbot capable of understanding user inputs by identifying intents and extracting entities. By leveraging NLP techniques and a Logistic Regression model, the chatbot interprets text inputs and provides appropriate responses. A Streamlit-based interface ensures user-friendly interaction, enabling seamless communication with the chatbot. This project serves as a foundational step toward creating advanced conversational agents, with scope for improvement through deeper datasets and sophisticated NLP techniques.

**Data Source Link**

The dataset used in this project consists of a labelled collection of intents, patterns, and responses stored in a JSON file.

Link: [DATASET](https://edunetfoundationorg-my.sharepoint.com/:u:/g/personal/namra_edunetfoundation_org/ETWNH3Y0di5GrTtGApRaqdoB0ZeuBuvwS8K3s-1mwIZO5A?e=HCJNZd)

**Tools Used**

1. **Python**:

* Used as the core programming language for implementing the chatbot, preprocessing data, training the model, and deploying the application.
  + NLTK: Used for tokenization and preprocessing user input to make it suitable for training and predictions.
  + Scikit-learn:
    - TF-IDF Vectorizer: For transforming textual data into numerical representations suitable for machine learning.
    - Logistic Regression: Used as the machine learning algorithm for classifying intents based on user input.
  + Streamlit: Leveraged for developing an interactive chatbot interface where users can input text and view responses in real time.

2. **Jupyter** **Notebooks**:

* Employed Jupyter Notebooks for an interactive and collaborative coding environment. Jupyter Notebooks provided a seamless platform for code execution, visualization, and documentation.

**Findings and Insights**

**Data Preparation:**

* Intent Parsing: The intents dataset, stored in a JSON file, was loaded and parsed to extract patterns (user input examples) and their corresponding tags (intents).
* Preprocessing: Patterns were converted into numerical representations using the TF-IDF vectorizer to prepare them for machine learning.

**Model Training:**

* Logistic Regression Classifier: The model was trained on the TF-IDF-transformed patterns to predict user intents with high accuracy.
* Hyperparameter Optimization: Adjusted parameters such as max\_iter and random\_state to ensure model stability and efficiency.

**Chatbot Functionality:**

* Prediction: The trained model classifies user input into one of the predefined intents by transforming input text into a vectorized format.
* Dynamic Response Generation: Based on the predicted intent, the chatbot dynamically selects a random response from the corresponding response list, ensuring varied interactions.

**Deployment Insights:**

* Interactive User Interface: Streamlit was used to create a user-friendly interface where users could input text and view chatbot responses in real time.
* Session Management: Incorporated counters to manage unique user interactions and ensure proper session handling.

**Performance and Limitations:**

* The model efficiently handled predefined intents and responses but might require further enhancement to support more complex conversational flows or unseen intents.
* The reliance on predefined tags and responses limits adaptability to dynamic user input outside the dataset's scope.

This implementation highlights the foundational steps to building a chatbot, demonstrating effective integration of NLP techniques, machine learning, and web deployment using minimal resources.

**Conclusion**

The intents-based chatbot project successfully created a conversational agent capable of classifying user intents and providing contextually relevant responses. Using NLP techniques, Logistic Regression, and Streamlit, the chatbot demonstrates the potential for enhanced user interaction. Future work could focus on expanding the dataset, incorporating deep learning models for better accuracy, and extending the chatbot's functionality to handle more complex conversations.