

# **AI-Powered Chatbot Using Python, NLTK, and scikit-learn**

## **1. Introduction**

The project focuses on developing an AI-powered chatbot capable of understanding user intents and extracting relevant entities from natural language queries. The chatbot is designed to simulate human-like conversations, making interactions with users intuitive and effective. Integration with messaging platforms like Telegram enables real-time communication.

## **2. Objectives**

Build a chatbot that recognizes user intents using machine learning.

Extract key entities (e.g., names, dates) from user input using NLP techniques.

Integrate the chatbot with Telegram for seamless user interaction.

Provide customizable responses based on user queries.

## **3. Tools and Technologies**

Python: Core programming language.

NLTK: For tokenization, lemmatization, and text preprocessing.

scikit-learn: For building the intent classification model.

## **Project Report: AI-Powered Chatbot**

spaCy: For Named Entity Recognition (NER).

python-telegram-bot: For Telegram bot integration.

### **4. Methodology**

Data Preparation: Collected training data representing various user intents and sample queries.

Preprocessing: Tokenized and lemmatized user input to standardize text.

Model Training: Used scikit-learn's machine learning algorithms to classify user intents.

Entity Extraction: Utilized spaCy to extract entities such as person names and dates.

Response Generation: Mapped predicted intents to appropriate responses.

Platform Integration: Connected the chatbot with Telegram using its API for live interaction.

### **5. Results**

The chatbot successfully classifies intents with reasonable accuracy.

Extracted entities like person names and dates are correctly identified.

## **Project Report: AI-Powered Chatbot**

Telegram integration allows real-time communication with users.

The system responds contextually based on detected intents and entities.

### **6. Challenges**

Handling ambiguous user inputs.

Limited training data impacted intent classification accuracy.

Entity extraction depends on spaCy's pre-trained model and may miss domain-specific entities.

### **7. Future Work**

Expand training dataset for better intent recognition.

Incorporate dialogue management for multi-turn conversations.

Add support for more messaging platforms like Slack and Web.

Integrate calendar APIs for automated meeting scheduling.

### **8. Conclusion**

The AI chatbot demonstrates effective use of NLP and machine learning techniques to interact

## **Project Report: AI-Powered Chatbot**

naturally with users and perform tasks like booking meetings. Integration with Telegram provides a practical interface for deployment. With further enhancements, the chatbot can be developed into a robust virtual assistant.

### **Created by**

Ayush Parchure