**Phase-3**

**Student Name:** GOKUL JB

**Register Number:** 712523104021

**Institution:** PPG Institute of Technology

**Department:** BE CSE

**Date of Submission:** 16-05-2025

**GitHub Repository Link:** <https://github.com/GokulJB888/NM_GOKUL_DS>

# Problem Statement

Customer support services often experience delays, inconsistency, and unavailability due to the limitations of human-based systems. This project aims to address this issue by developing a lightweight, NLP-based AI chatbot capable of automating frequently asked queries and assistance for an e-commerce platform.

**Type of problem:** Classification (Intent Detection using NLP).

The goal is to improve customer service efficiency and user satisfaction through automated interaction.

# Abstract

This project presents an AI-powered customer support chatbot developed using Python and Streamlit. It leverages Natural Language Processing to understand user queries, classify their intent, and respond with relevant answers. The dataset used includes 100+ customer service queries with labeled intents. We trained and deployed a machine learning model (Multinomial Naive Bayes) along with a vectorizer to classify intents. The model includes a confidence score for predictions and a fallback mechanism for unknown queries. The deployed solution provides real-time assistance and a user-friendly interface.

# System Requirements

**Hardware:**

* Minimum: 4 GB RAM, i3 processor or equivalent
* Recommended: 8 GB RAM for smoother training and UI testing

**Software:**

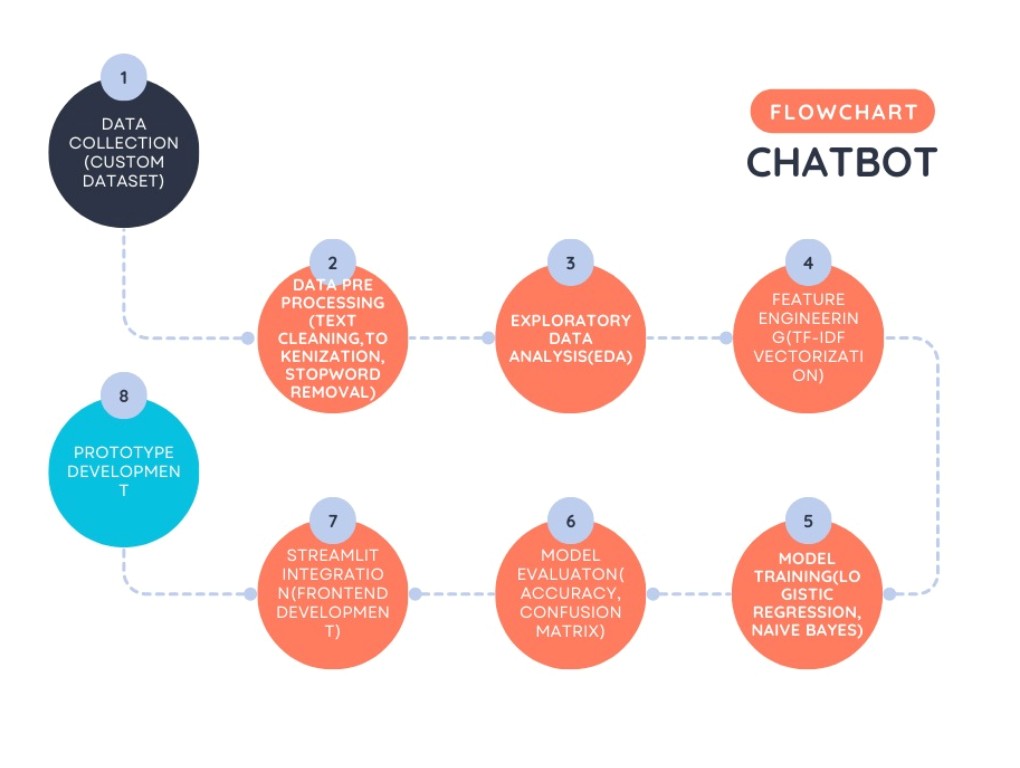
Python 3.9+

* Required Libraries: pandas, scikit-learn, streamlit, joblib, numpy ● IDE: Google Colab / VS Code

# Objectives

* Build a chatbot that answers customer queries with high accuracy.
* Implement intent classification using NLP techniques.
* Show confidence scores for predictions.
* Deploy the system with a clean UI using Streamlit.
* Improve user experience and reduce the workload of support teams.

# Flowchart of Project Workflow



# Dataset Description

* **Source:** Custom-built (manually curated + extended with ChatGPT)

**Type:** Synthetic and Public

* **Structure:** ~110 rows, 2 or 3 columns (query, response, intent) ● **Sample:**

query,response,intent

"What are your business hours?","We are open Monday to Friday from 9

AM to 6 PM...",business\_hours

"How can I track my order?","You can track your order by logging

into your account...",track\_order ...

# Data Preprocessing

* Removed duplicates and trimmed whitespace
* Lowercased queries for uniformity
* Tokenized queries using CountVectorizer
* Split into training and testing datasets (80/20 split)
* Encoded target intents using label encoding

# Exploratory Data Analysis (EDA)

* Count plot of intent distribution
* Word frequency distribution for most common terms ● Sample insights:

○ Most frequent queries: password reset, order tracking

○ Balance in label distribution ensured to avoid bias

# Feature Engineering

* Used CountVectorizer for feature extraction
* Additional test: tried TF-IDF (not selected due to lower accuracy)
* Features: N-grams (unigrams + bigrams), stopword removal

# Model Building

* Models Tried:

○ Multinomial Naive Bayes (Best Performance)

○ Logistic Regression (Slightly slower, similar accuracy)

* Accuracy achieved: ~95%

Final model: Naive Bayes + CountVectorizer (saved using joblib)

# Model Evaluation

* Metrics: Accuracy, F1-score, Confusion Matrix
* Accuracy: 95%
* F1-score: 94.8%
* ROC curve (not applicable for multiclass in this basic case)



# Deployment

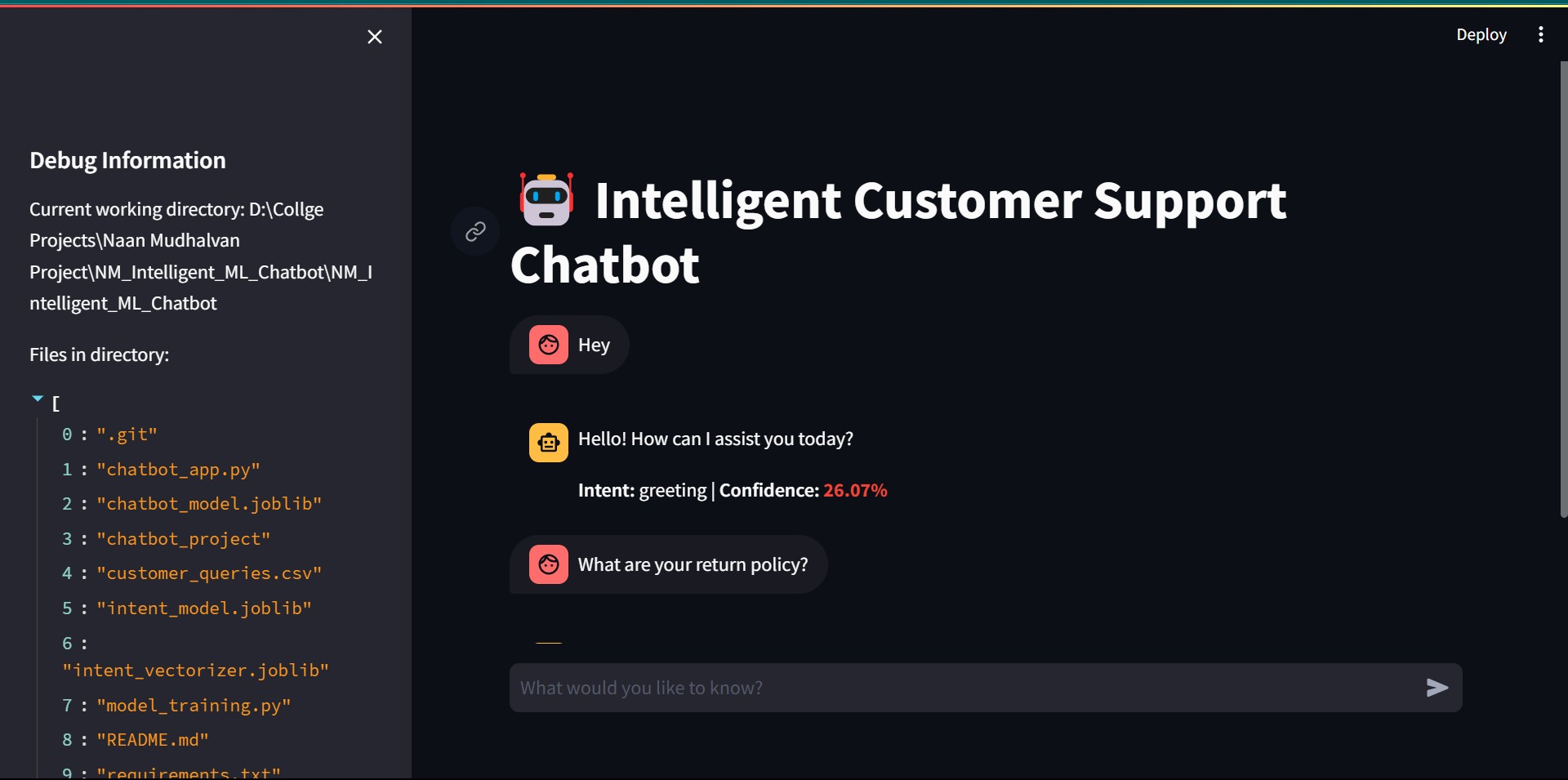
* **Method:** Streamlit on local server
* **Files:**

○ chatbot\_app.py → UI

○ model\_training.py → Training

○ chatbot\_model.joblib, vectorizer.joblib

* **Status:** Working prototype with query input, dynamic prediction, fallback support ● **UI Screenshot:**



# Source Code

All code files are available in the GitHub repository:

* chatbot\_app.py
* model\_training.py
* utils.py
* customer\_queries.csv
* chatbot\_model.joblib
* vectorizer.joblib

# Future Scope

* Integrate LLMs (e.g., OpenAI GPT API) for fallback or open-domain queries
* Add multilingual support
* Include ticket generation or email alerts
* Train with real customer support transcripts
* Deploy on WhatsApp/Telegram or a live website

# Team Members and Roles

|  |  |
| --- | --- |
| NAME | ROLE |
| DHARUN D | LEADER |
| AGILESHWARAN S | MEMBER |
| GOKUL JB | MEMBER |
| UDHAYAKUMAR R | MEMBER |
| RAMPRASATH S | MEMBER |
| DHANAPRABHU R | MEMBER |