Student Name: THIVESHAN G

Register Number: 410723104089

Institution: Dhanalakshmi College of Engineering

Department: Computer Science Engineering

Date of Submission: 07-05-2025

Github Repository Link: https://github.com/THIVESHAN-OFFX/NM-THIVESHAN

Project Title: Revolutionizing customer support with an intelligent chatbot for automated assistance

---

1. Problem Statement

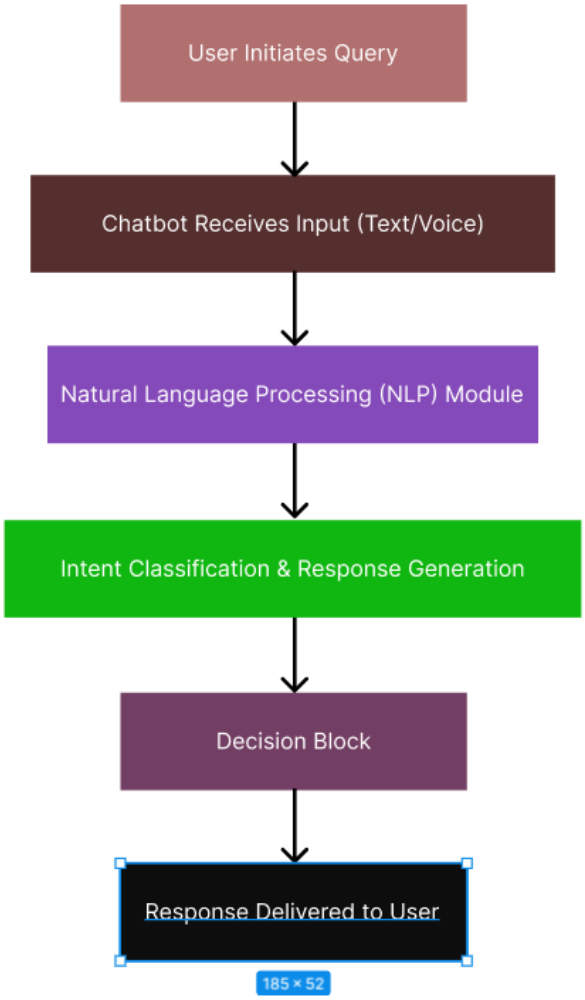
● *Automate customer support using an intelligent chatbot to reduce response times and improve efficiency.*
● *Minimize operational costs by replacing repetitive human interactions with AI-driven assistance.*
● *Ensure consistent and accurate support across all user interactions, regardless of time or volume.*
● *Improve customer satisfaction by providing 24/7 real-time responses to common queries.*
● *Design the chatbot to handle a wide range of support scenarios using natural language understanding.*
● *Evolved goal: Based on early development, the focus was refined to enhance contextual accuracy and integrate continuous learning from user interactions.*

2. Project Objectives

● *Develop an AI-powered chatbot capable of handling customer support queries with minimal human intervention.*

● *Train the chatbot using real-world support data to ensure high accuracy and contextual understanding.*
● *Implement natural language processing (NLP) techniques to enable human-like interactions.*
● *Integrate the chatbot with existing communication channels such as websites, messaging apps, or helpdesks.*
● *Evaluate the chatbot's performance using metrics like response accuracy, resolution rate, and user satisfaction.*

3. Flowchart of the Project Workflow



4. Data Description

***Dataset Name**: Customer Support Interaction Data Set*
***Source**: Internal company data / Customer Service Logs*
***Type of Data**: Structured tabular data, unstructured text data*
***Records and Features**:*

- ***Records**: 10,000+ customer support interactions (queries, responses, feedback)*
- ***Features**: 20+ features, including:*
  - ***Numeric**: Response time, number of escalations, customer satisfaction rating (1-5 scale), average handle time*
  - ***Categorical**: Query type (technical issue, billing inquiry, etc.), support channel (chat, email, social media), agent ID, sentiment (positive, neutral, negative)*
  - ***Textual**: Customer query, agent response, sentiment analysis results*

***Static or Dynamic**: Static dataset (historical data of customer support interactions)*

5. Data Preprocessing

- ***Handle Missing Data**: Impute or remove missing values.*
- ***Text Preprocessing**: Tokenize, remove stop words, lemmatize.*
- ***Encode Features**: One-hot encode categories, apply TF-IDF on text.*
- ***Normalize/Standardize**: Scale numeric features.*
- ***Feature Selection**: Remove redundant features.*
- ***Split Data**: Train-test split and balance classes.*

6. Exploratory Data Analysis (EDA)

- ***Data Summary**: Review dataset size, feature types, and missing values.*

- ***Missing Value Analysis**: Visualize and calculate the percentage of missing data.*
- ***Univariate Analysis**: Visualize distributions of numeric (histograms) and categorical features.*
- ***Bivariate Analysis**: Explore relationships between target variable and features using scatter plots or heatmaps.*
- ***Sentiment Analysis**: Analyze sentiment distribution and its correlation with target variable.*
- ***Outlier Detection**: Identify outliers in numeric features using box plots or Z-scores.*

7. Feature Engineering

- ***Text Preprocessing**: Tokenize, remove stop words, and apply lemmatization/stemming on text.*

- ***TF-IDF**: Convert text data (customer queries, responses) into numerical format.*

- ***Sentiment Analysis**: Extract sentiment from customer queries and responses.*

- ***Categorical Encoding**: One-hot encode categorical features (e.g., query type, support channel).*

- ***Interaction Features**: Create features like average response time per query type.*

8. Model Building

- ***Select Model**: Choose models like Logistic Regression, Random Forest, or Neural Networks.*

- ***Train Model**: Fit the model on the training data.*

- ***Hyperparameter Tuning**: Optimize model parameters using Grid Search or Random Search.*

- ***Cross-Validation**: Apply k-fold cross-validation for better model evaluation.*

- ***Evaluate Performance**: Use metrics like accuracy, precision, recall, and F1-score.*

9. Visualization of Results & Model Insights

- ***Confusion Matrix**: Visualize true vs. predicted classifications for performance evaluation.*

- ***ROC Curve**: Plot the Receiver Operating Characteristic curve for classification models.*

- ***Feature Importance**: Use bar charts to show feature importance in models like Random Forest.*

- ***Precision-Recall Curve**: Visualize trade-off between precision and recall, especially in imbalanced datasets.*

- ***Learning Curves**: Plot training vs. validation accuracy/loss to assess model overfitting/underfitting.*

10. Tools and Technologies Used

- ***Programming Language**: Python*

- ***Libraries**: Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, TensorFlow/Keras*

- ***Data Processing**: Jupyter Notebook, NLTK, SpaCy (for text preprocessing)*

- ***Modeling**: Scikit-learn (Logistic Regression, Random Forest), TensorFlow/Keras (Deep Learning)*

- ***Visualization**: Matplotlib, Seaborn, Plotly*

- ***Version Control**: Git, GitHub for code management and collaboration*

- ***Cloud/Hosting**: AWS, Google Cloud (for model deployment and scaling)*

11. Team Members and Contributions

Team Leader : Pragadeesh A
- *Data cleaning & EDA*

Team Member : Thiveshan G
- *Feature engineering*

Team Member : Sanjai Ram N
- *Model development*

Team Member : Sabarisan D
- *Documentation and reporting*