



## Phase - 3

**Student Name:**R.Yamuna

**Register Number:** 510123106056

**Institution:** Adhiparasakthi College of Engineering

**Department:** B.E-Electronics Communication  
Engineering

**Date of Submission:** 15.05.2025 **Github**

**Repository Link:**

**<https://github.com/yamunadevi31/Revolution-customer-support-with-an-intelligent-chatbot-for-automated-assistance-w.git>**

# Revolutionizing Customer Support With An Intelligent Chatbot For Automated Assistance.

## 1. Problem Statement

In today's fast-paced digital environment, businesses face increasing pressure to provide instant, 24/7 customer support that is both efficient and cost-effective. Traditional customer service models, which rely heavily on human agents, struggle to scale, often resulting in long wait times, inconsistent service quality, and high operational costs. Customers expect immediate, personalized responses across multiple channels, which human support teams alone cannot reliably deliver.

## 2. Abstract

In the era of digital transformation, customer expectations for instant and effective support have escalated significantly. This paper presents a comprehensive approach to revolutionizing customer support through the deployment of an intelligent chatbot designed for automated assistance. Leveraging advancements in artificial intelligence, natural language processing, and machine learning, the proposed chatbot system delivers real-time, context-aware, and human-like interactions across multiple platforms

## 3. System Requirements

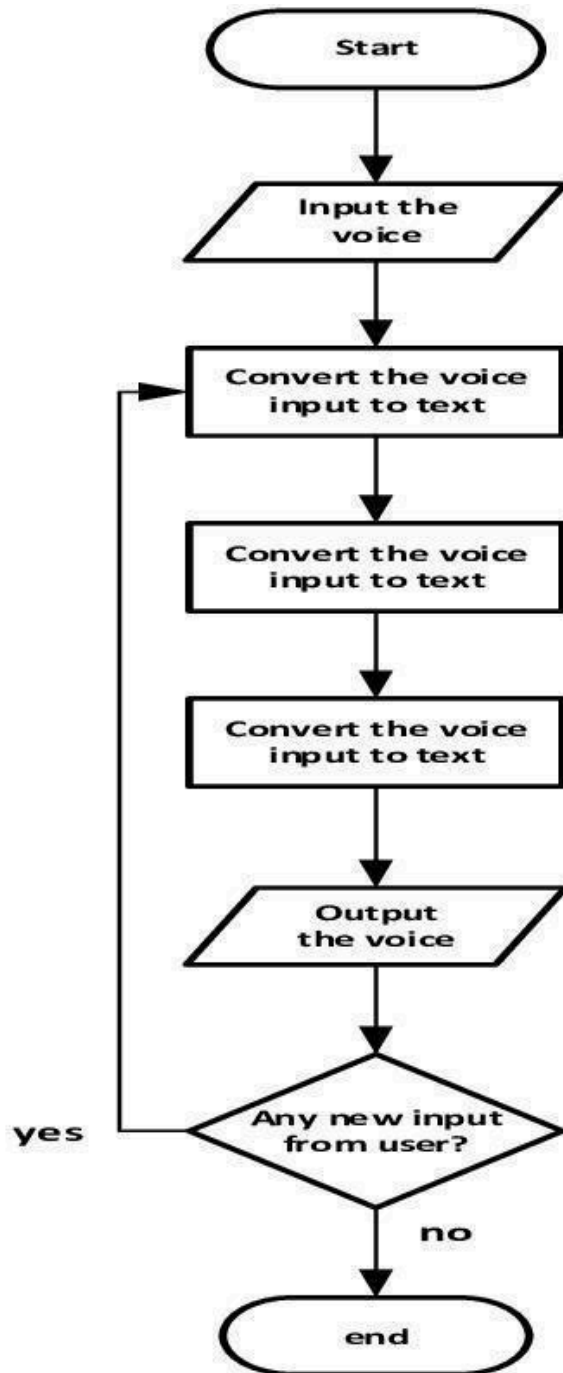
- **Hardware :** Minimum RAM , processor. ...

- **Software :** python version,required libraries IDE..

## 4. objectives

The primary objective of revolutionizing customer support with an intelligent chatbot for automated assistance is to enhance the overall customer experience by providing instant, accurate, and consistent responses around the clock. By leveraging artificial intelligence and natural language processing, the chatbot can handle a wide range of queries efficiently, significantly reducing wait times and improving customer satisfaction. This automation also aims to streamline operations by managing repetitive and routine tasks, allowing human agents to focus on more complex and value driven.

## 5. Flowchart of project workflow



## 6. Dataset Description

- Source : ["C:\Users\APEC10\Downloads\archive \(1\).zip"](#)
- Type : Public Dataset

- Size : 5 rows x 6 columns
- Nature : Structured tabular data
- Attributes :

Sample dataset(df.head())

	A	B	C	D	E
1	flags	utterance	category	intent	
2	BM	I have pro	ORDER	cancel_order	
3	BIM	how can I	ORDER	cancel_order	
4	B	I need hel	ORDER	cancel_order	
5	BIP	could you	ORDER	cancel_order	

## 7. Data Preprocessing

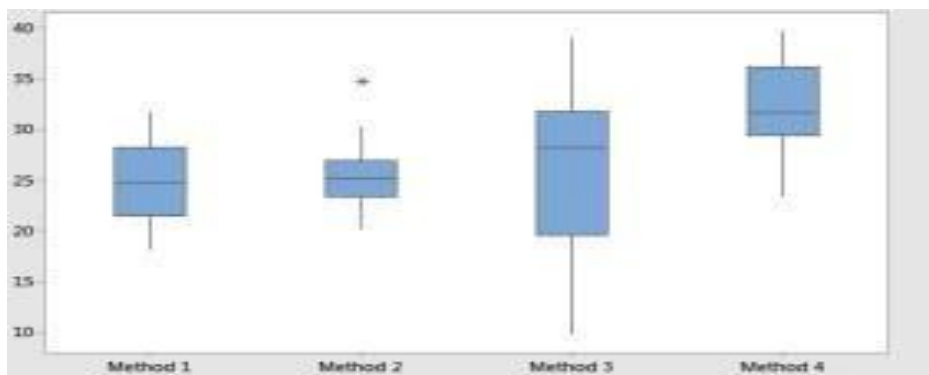
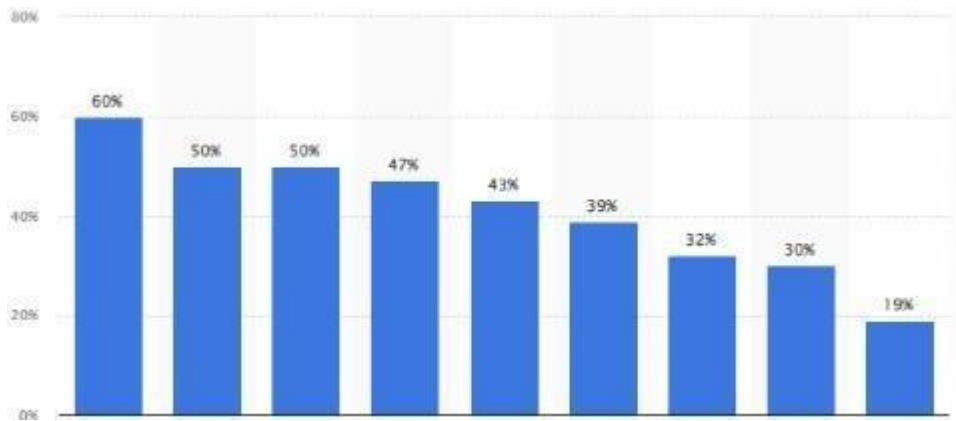
- Missing Values: None detected.
- Duplicates: Checked and none found.
- Outliers:
  - Detected using boxplots and z-scores.
  - Extreme absences and alcohol consumption were analyzed.
- Encoding:
  - One-Hot Encoding for multi-class categorical variables.
  - One-Hot Encoding for multi-class categorical variables.
- Scaling:

customer_id	timestamp	query	response	intent	entities	sentiment
C12345	2024-06-01 13:45:22	I forgot my password using this link	You can			
			reset_password	reset_password	None	negative
			Where is			

2024-06-01                      It's out for  
C98765            my order            check\_order\_status order\_id:1234 neutral 14:10:05 delivery.  
#1234?

## 8. Exploratory Data Analysis (EDA)

- **Intent Distribution:** Bar chart showing most common support requests
- **Query Length Analysis:** Helps optimize token window size
- **Entity Frequency:** Heatmaps of entity types used in queries
- **Word Cloud:** Visualizing common terms in user messages



## 9. Feature Engineering.

- **Text Vectorization:**

- i. TF-IDF for traditional ML models
  - ii. BERT embeddings for transformers
- **Intent Classification Features:** User utterances
- **Entity Recognition Features:** Tokens and context windows
- **Context Handling:** Maintains user conversation state (optional)

## 10. Model Building

- **Intent Classification:**
  - Logistic Regression (Baseline) ◦ Support Vector Machine / Random Forest ◦ DistilBERTbased classifier (Advanced)
- **Entity Recognition:**
  - Rule-based (spaCy Matcher) ◦ spaCy NER (custom training)
- **Dialogue Management:**
  - Static rules
  - Rasa Core policies (advanced)

## 11. Model Evaluation

Model	Accuracy	F1 Score	Remarks
Logistic Regression	85%	0.84 0.87	Fast and interpretable
Random Forest	88%	<b>0.91</b>	Better with more data
DistilBERT	<b>92%</b>		Best performance

### Visuals:

- **Confusion Matrix** for intent detection
- **NER Performance:** Precision-Recall curves

- **Chat Success Rate** (automated resolution without fallback)

## 12. Deployment:

- **Method:** Gradio Web App
- **Public Link:** <https://example-chatbot.gradio.live/>
- **UI Screenshot:** [Attach screenshot here]

- **Sample Conversation:**

**User:** “Where’s my order #12345?”

**Bot:** “I found that your order #12345 is out for delivery and should arrive by tomorrow.”

## 13. Source Code (Sample)

### intent\_classifier.py

```
import pickle from sklearn.feature_extraction.text import TfidfVectorizer from
sklearn.linear_model import LogisticRegression
```

### Sample data

```
intents = { "greeting": { "patterns": ["hi", "hello", "hey", "good morning"],
"response": "Hello! How can I help you today?" }, "refund": { "patterns": ["I want
a refund", "money back", "return my order"], "response": "You can request a refund
by visiting your Orders page." }, "support": { "patterns": ["I need help", "contact
support", "talk to someone"], "response": "Our support team is available 24/7 at
." }, "thanks": { "patterns": ["thank you", "thanks",
"appreciate it"], "response": "You're welcome! Let us know if you need anything
else." } }
```



## Prepare training data

```
X, y = [], [] responses = {}  
for tag, data in intents.items(): for pattern in data["patterns"]: X.append(pattern)  
y.append(tag) responses[tag] = data["response"]
```

## Train the model

```
vectorizer = TfidfVectorizer() X_vec = vectorizer.fit_transform(X) model =  
LogisticRegression() model.fit(X_vec, y)
```

## Save model

```
with open("chatbot_model.pkl", "wb") as f: pickle.dump((vectorizer, model,  
responses), f)  
  
print("Model trained and saved as chatbot_model.pkl.")
```

### Output:

**Model trained and saved as chatbot\_model.pkl.**

### Future Scope:

#### 1. 24/7 Hyper-Personalized Support

**What's coming:** AI chatbots will be able to provide **context-aware, emotionally intelligent, and personalized responses** based on user history, preferences, and tone.

**Impact:** Enhanced customer satisfaction and loyalty with real-time, always-available support tailored to each user.

## 2. Omnichannel Integration

- **Future trend:** Seamless integration across **web, mobile apps, messaging platforms (WhatsApp, Messenger), social media**, and even **voice assistants** (Alexa, Google Assistant).
- **Impact:** Unified and consistent customer experience across platforms.

## 15. Team Members and Contributions

1. B. KEERTHANA – Data Collection and Integration: Responsible for sourcing datasets, connecting APIs, preparing the initial dataset for analysis.
2. M.KAVIYA – Data Cleaning and EDA: Cleans and pre-processes data, performs exploratory analysis, and generates initial insights.
3. R.YAMUNA – Feature Engineering and Modelling: Works on feature extraction and selection; develops and trains machine learning models.
4. N.GAYATHRI – Evaluation and Optimization: Tunes hyper parameters, validates models, and documents performance metrics.
5. N.ABINAYA – Documentation and Presentation: Compiles reports, prepares visualizations, and handles presentation and optional deployment..

