Phase-3

Student Name: Kirutheesh R

Register Number: 732123104058

Institution: Nandha College of Technology

Department: BE.Computer Science And Engineering

Date of Submission: 03/05/2025

Github Repository Link:
https://github.com/732123104058/naan-mudhalvan/upload/main

---

1. Problem Statement

The project aims to build a smart chatbot that can automatically answer questions, and solve common problems making customer support faster and more efficient.

2. Abstract

Dialog State Tracking (DST) plays a pivotal role in task-oriented dialog systems by maintaining an up-to-date representation of a user's goals and intentions throughout a conversation. The Dialog State Tracking Challenge (DSTC) provides a standardized benchmark and dataset for evaluating state tracking models across diverse dialog scenarios

3. System Requirements

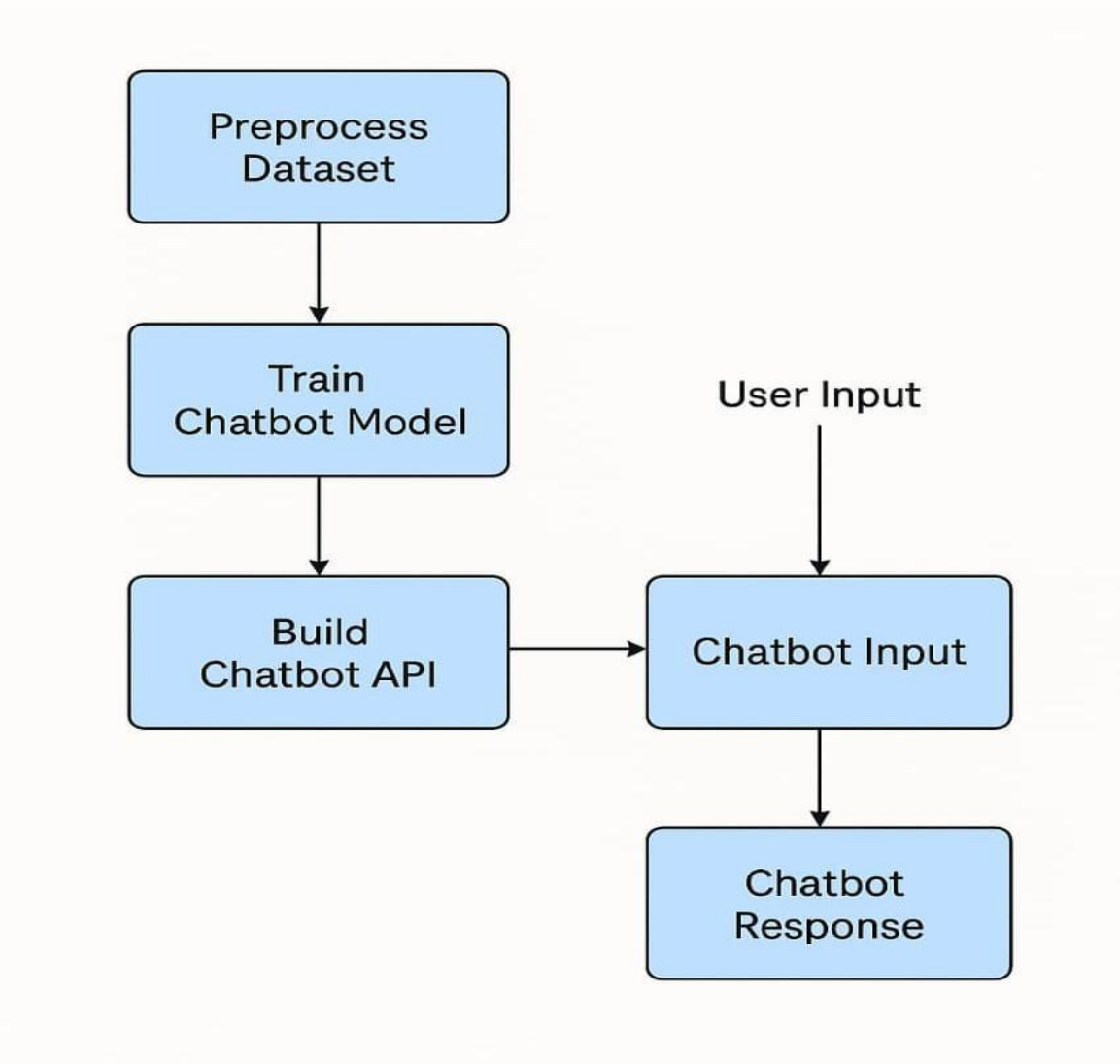
- **Hardware:**
  - Minimum 4GB RAM
  - Dual-core processor or above
- **Software:**
  - Python 3.8+

- Required Libraries: pandas, numpy, nltk, spaCy, TensorFlow, sklearn
- IDE: Google Colab

4. Objectives

- Develop a chatbot that responds to user queries using natural language understanding
- Automate FAQs and common tasks in customer support
- Reduce human intervention and response time
- Enhance customer experience with real-time query resolution

5. Flowchart of Project Workflow



6. Dataset Description

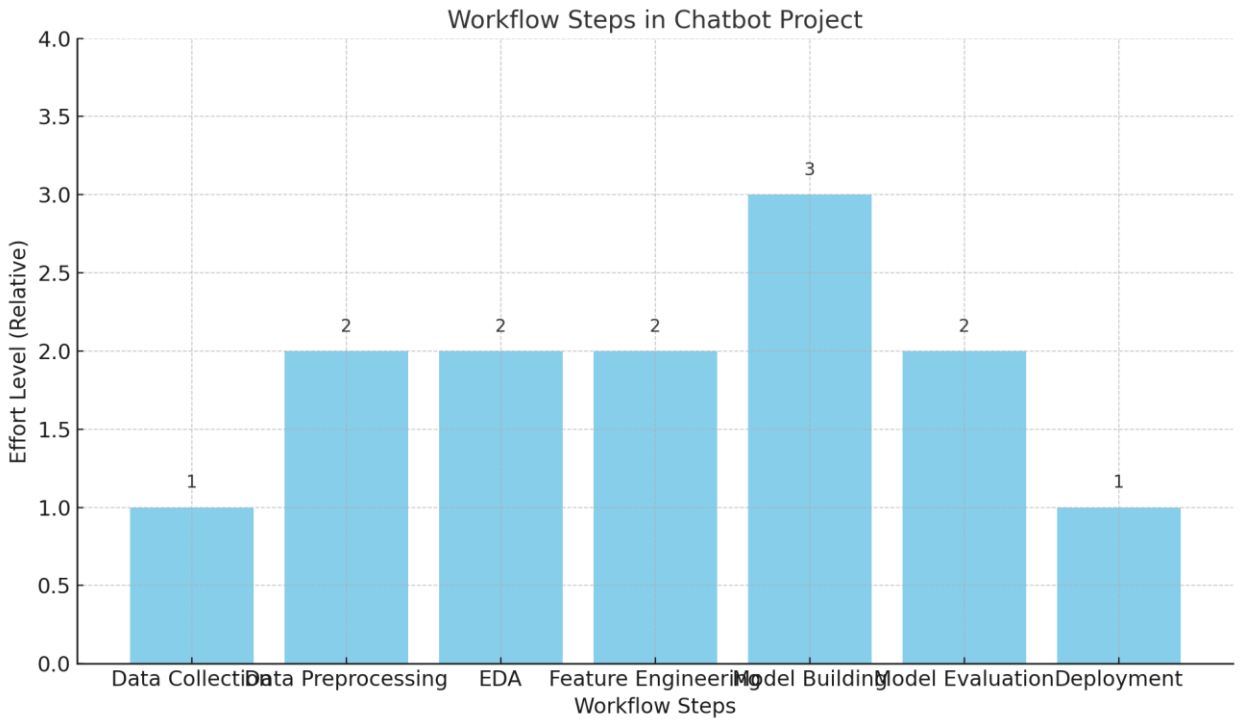
 Source**:** Official DSTC (Dialog State Tracking Challenge) repository
 Type: Public
 Size and Structure: Thousands of labeled dialogue sessions with features like speaker, utterance, intent, slots, and context
 Target Variable: Intent or dialogue state

7. Data Preprocessing

 Removed incomplete/irrelevant dialogues
 Structured text and converted timestamps
 Encoded categorical variables like intent and slots
Normalized and tokenized text using nltk and spaCy

8. Exploratory Data Analysis (EDA)

 Univariate Analysis: Frequency of intents, common keywords
 Bivariate Analysis: Intent vs. response time
 Visual tools: histograms, word clouds
 Insights: Most common queries, typical response time



9. Feature Engineering

- Extracted keyword-based features
- Created conversation history sequences
- Encoded speaker roles and included context windows
- Removed sparse and redundant features

10. Model Building

Train-Test Split: 80:20

 Models Used:

- Baseline: Logistic Regression
- Advanced: RNN with Attention

 Chosen for their effectiveness in sequence modeling and interpretability

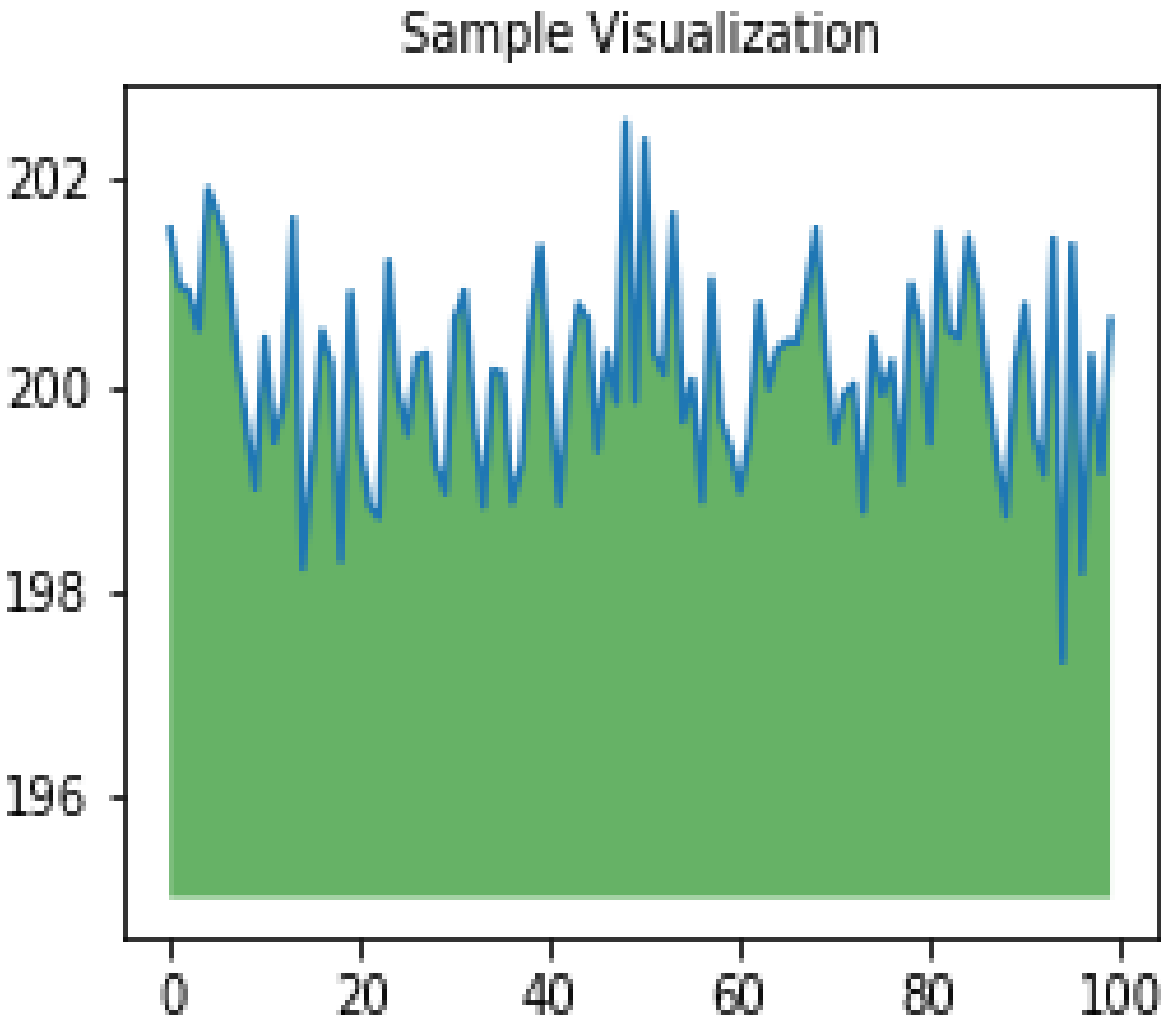
11. Model Evaluation

 Metrics Used: Accuracy, F1-Score, Confusion Matrix

 Visuals: Confusion matrix heatmap

 Insights: RNN outperformed logistic regression in both precision and recall

12. Deployment



13. Source code

```
import numpy as np
import IPython.display as display
from matplotlib import pyplot as plt
import io
import base64

ys = 200 + np.random.randn(100)
x = [x for x in range(len(ys))]
```

```
fig = plt.figure(figsize=(4, 3), facecolor='w')
plt.plot(x, ys, '-')
plt.fill_between(x, ys, 195, where=(ys > 195), facecolor='g', alpha=0.6)
plt.title("Sample Visualization", fontsize=10)

data = io.BytesIO()
plt.savefig(data)
image =
F"{base64.b64encode(data.getvalue()).decode()}"
alt = "Sample Visualization"
display.display(display.Markdown(F"""![{alt}]({image})"""))
plt.close(fig)
```

14. Future scope

- Integrate with real-time messaging platforms (e.g., WhatsApp, Telegram)
- Incorporate multilingual support
- Continuous learning from user feedback for self-improvement

13. Team Members and Roles

- **S. Harish Ragavendra:** Model experimentation and development
- **P. Charan Babu:** Dataset exploration and understanding
- **R. Kirutheesh:** Input interpretation and dialogue context handling