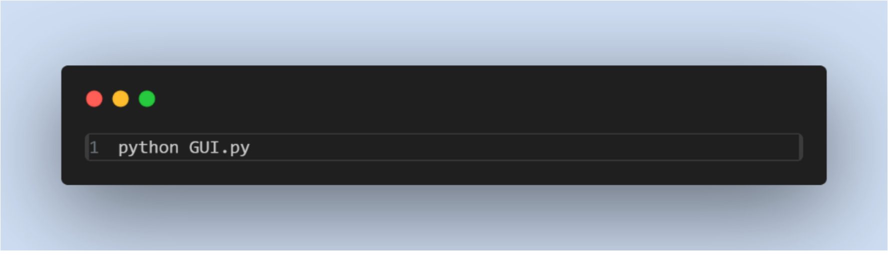
How to Run the Chatbot Pipeline (Local GUI + Cloud Backend)

This guide explains how to run the chatbot pipeline, which consists of a local Gradio-based GUI and a chatbot pipeline backend running either locally or on Kaggle.

Step 1: Open the GUI

1. Open the project folder and locate the file named GUI.py.
2. Open it using any code editor (e.g., **VS Code**, **PyCharm**, or **Sublime Text**).
3. Run the file using one of the following methods:
    - Click the **"Run Python File"** button in your editor.
    - Or open a terminal and run:



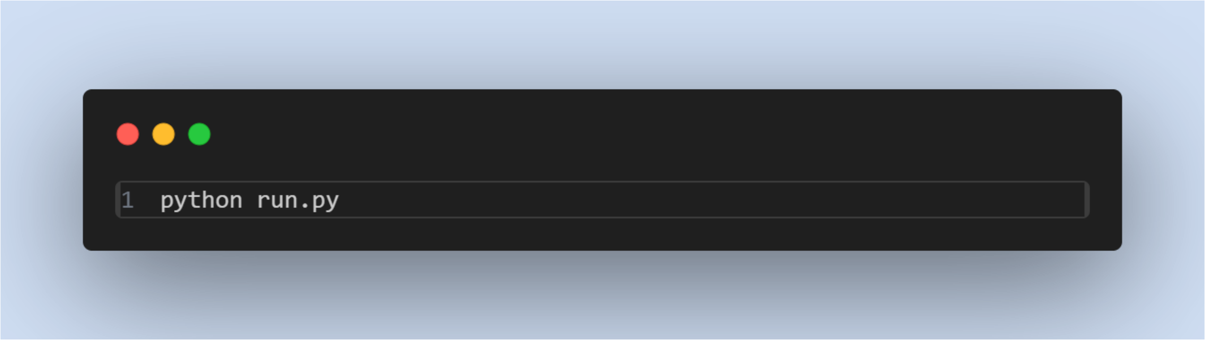
```
python GUI.py
```

4. A **local URL** will appear (e.g., http://localhost:8501).
   Press Ctrl + Left Click on the URL to open the GUI in your browser.

Step 2: Run the Chatbot Pipeline (Backend)

Option 1: Run Locally (if supported)

1. Open a **new terminal window**.
2. Navigate to the app folder inside the project.
3. Run the following command:

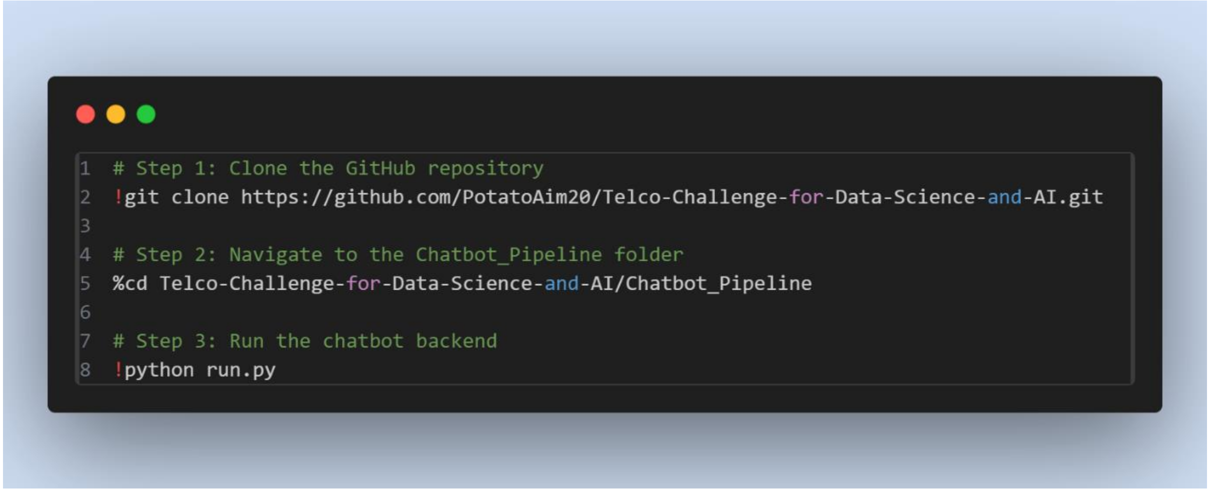


```
python run.py
```

⚠️ Note: If you can't run it locally (e.g., due to hardware constraints), proceed to **Option 2** below to run it on **Kaggle**.

Option 2: Run on Kaggle (Recommended for GPU support)

1. Go to https://www.kaggle.com and open it in your browser.
2. Create a **new notebook**.
3. Import the notebook file named ChatbotPipeline_Kaggle_Deployment.ipynb.
4. Enable the **P100 GPU** from the notebook **Settings** tab.
5. Run the cells in the notebook. One of the key cells will perform the following:

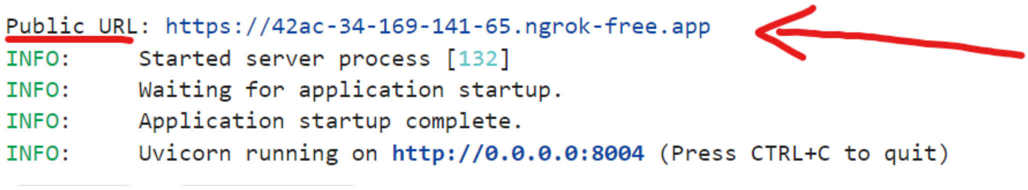


```
# Step 1: Clone the GitHub repository
!git clone https://github.com/PotatoAim20/Telco-Challenge-for-Data-Science-and-AI.git

# Step 2: Navigate to the Chatbot_Pipeline folder
%cd Telco-Challenge-for-Data-Science-and-AI/Chatbot_Pipeline

# Step 3: Run the chatbot backend
!python run.py
```

📌 This is the same project, uploaded to GitHub to make it easy to use in cloud environments like Kaggle.

6. After execution, wait **about 2 minutes** for the environment to be set up and the LLM to be downloaded.

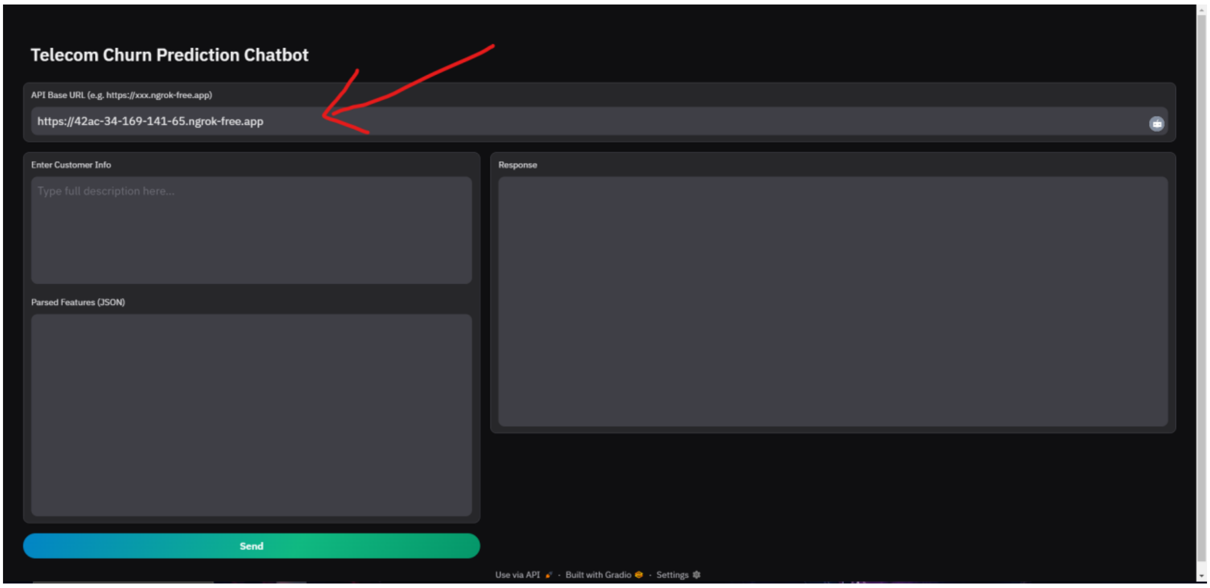
7. A public **URL** will appear once the backend is ready.
   Copy this URL and **leave the cell running** — do not stop or close the notebook.



```
Public URL: https://42ac-34-169-141-65.ngrok-free.app
INFO:     Started server process [132]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8004 (Press CTRL+C to quit)
```

Step 3: Connect the GUI to the Backend

1. Go back to your local GUI (Gradio interface opened in Step 1).
2. Paste the **URL** from the Kaggle notebook into the **API Base URL** field.
3. You're now ready to use the chatbot pipeline.



Telecom Churn Prediction Chatbot

API Base URL (e.g. https://xxx.ngrok-free.app): https://42ac-34-169-141-65.ngrok-free.app

Enter Customer Info
Type full description here...
Parsed Features (JSON)
Response
Send
Use via API · Built with Gradio · Settings

How the Marketing Team Can Use the Chatbot

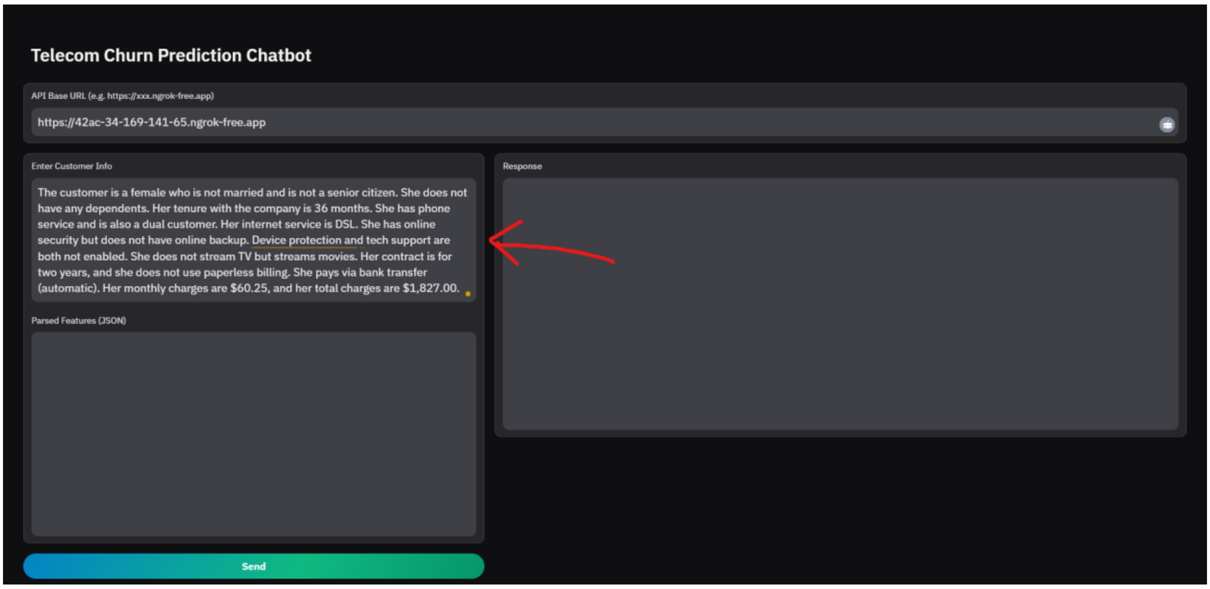
1. Enter Customer Description

The user starts by entering a natural language description of the customer in the **Enter Customer Info** text box.

For example:

The customer is a female who is not married and is not a senior citizen. She does not have any dependents. Her tenure with the company is 36 months. She has phone service and is also a dual customer. Her internet service is DSL. She has online security but does not have online backup. Device protection and tech support are both not enabled. She does not stream TV but streams movies. Her contract is for two years, and she does not use paperless billing. She pays via bank transfer (automatic). Her monthly charges are $60.25, and her total charges are $1,827.00.

After entering the description, press **Send**.



Telecom Churn Prediction Chatbot

API Base URL (e.g. https://xxx.ngrok-free.app)

https://42ac-34-169-141-65.ngrok-free.app

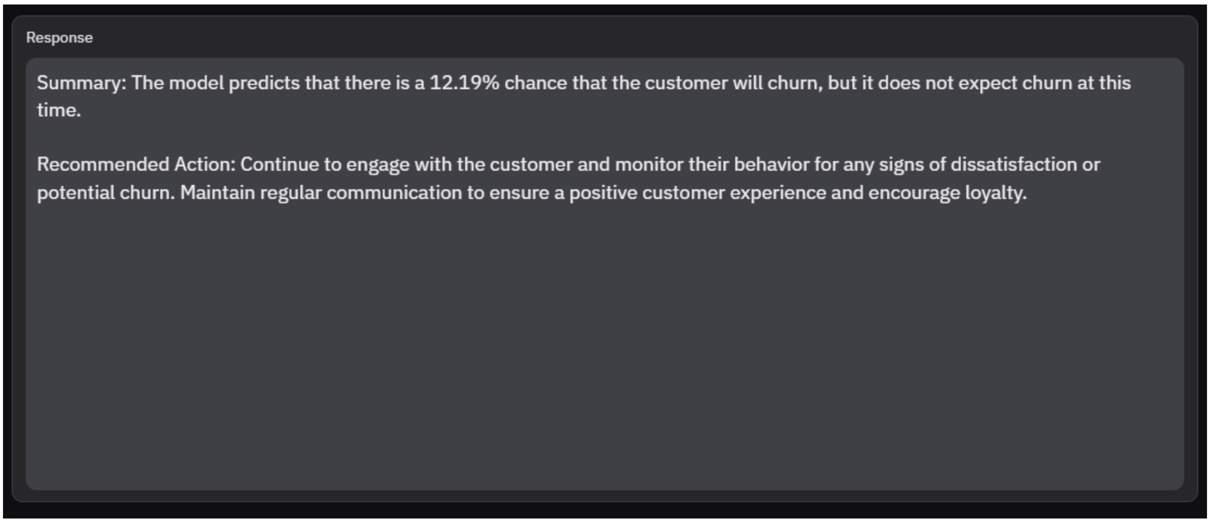
Enter Customer Info

The customer is a female who is not married and is not a senior citizen. She does not have any dependents. Her tenure with the company is 36 months. She has phone service and is also a dual customer. Her internet service is DSL. She has online security but does not have online backup. Device protection and tech support are both not enabled. She does not stream TV but streams movies. Her contract is for two years, and she does not use paperless billing. She pays via bank transfer (automatic). Her monthly charges are $60.25, and her total charges are $1,827.00.

2. View Prediction and Recommendations

The response box will display:

- The **model's prediction** on whether the customer is likely to churn.
- Suggested **recommended actions** based on the prediction.

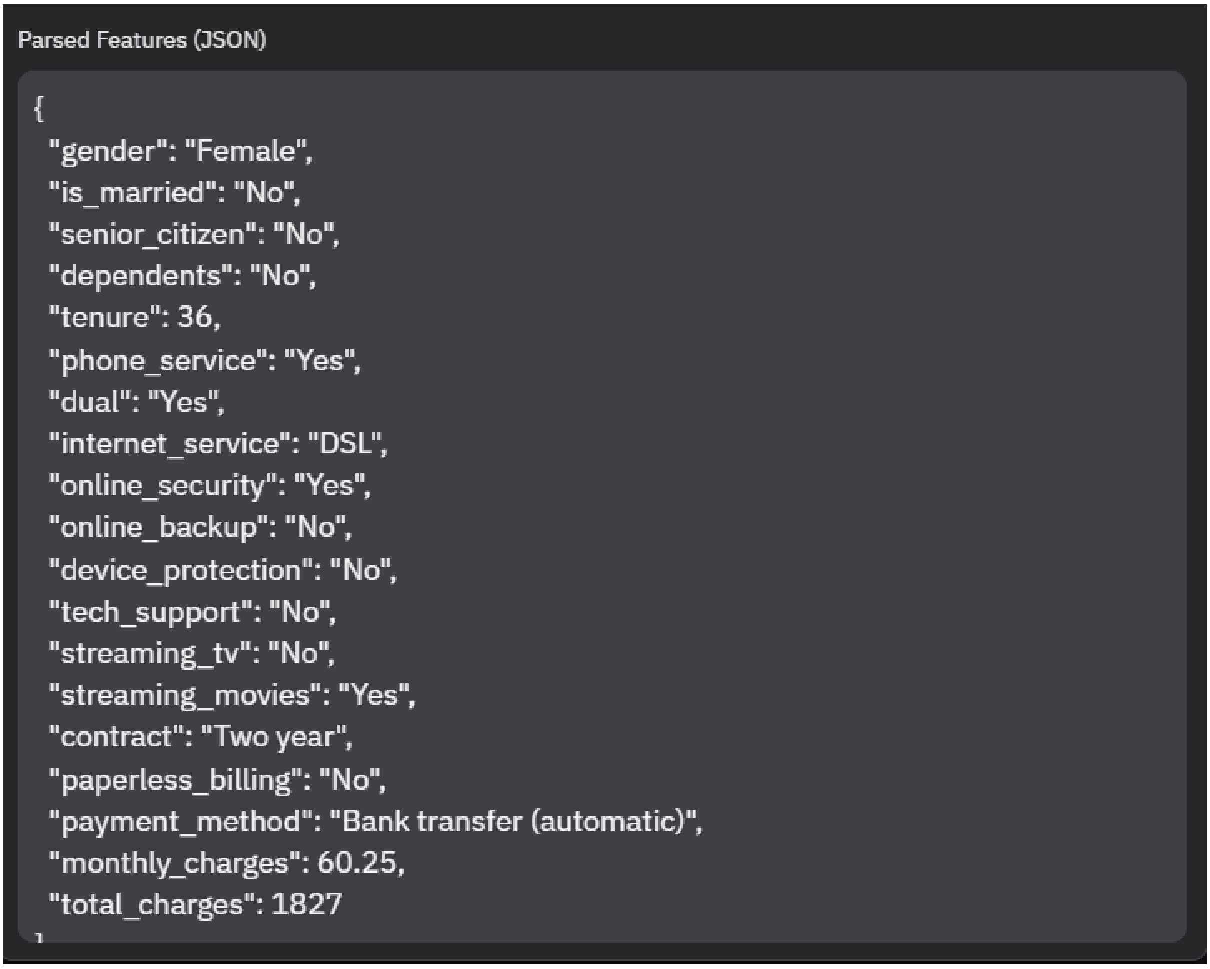


Summary: The model predicts that there is a 12.19% chance that the customer will churn, but it does not expect churn at this time.

Recommended Action: Continue to engage with the customer and monitor their behavior for any signs of dissatisfaction or potential churn. Maintain regular communication to ensure a positive customer experience and encourage loyalty.

3. Check Parsed Features

In the **Parsed Features (JSON)** box, you can review the structured features extracted from the text. This allows you to verify that all customer data was correctly understood and parsed by the model.



Parsed Features (JSON)

```
{
  "gender": "Female",
  "is_married": "No",
  "senior_citizen": "No",
  "dependents": "No",
  "tenure": 36,
  "phone_service": "Yes",
  "dual": "Yes",
  "internet_service": "DSL",
  "online_security": "Yes",
  "online_backup": "No",
  "device_protection": "No",
  "tech_support": "No",
  "streaming_tv": "No",
  "streaming_movies": "Yes",
  "contract": "Two year",
  "paperless_billing": "No",
  "payment_method": "Bank transfer (automatic)",
  "monthly_charges": 60.25,
  "total_charges": 1827
}
```

4. Handling Missing Data

If the entered customer description is missing some features, the response box will notify you with the missing fields. For example, if the description is:

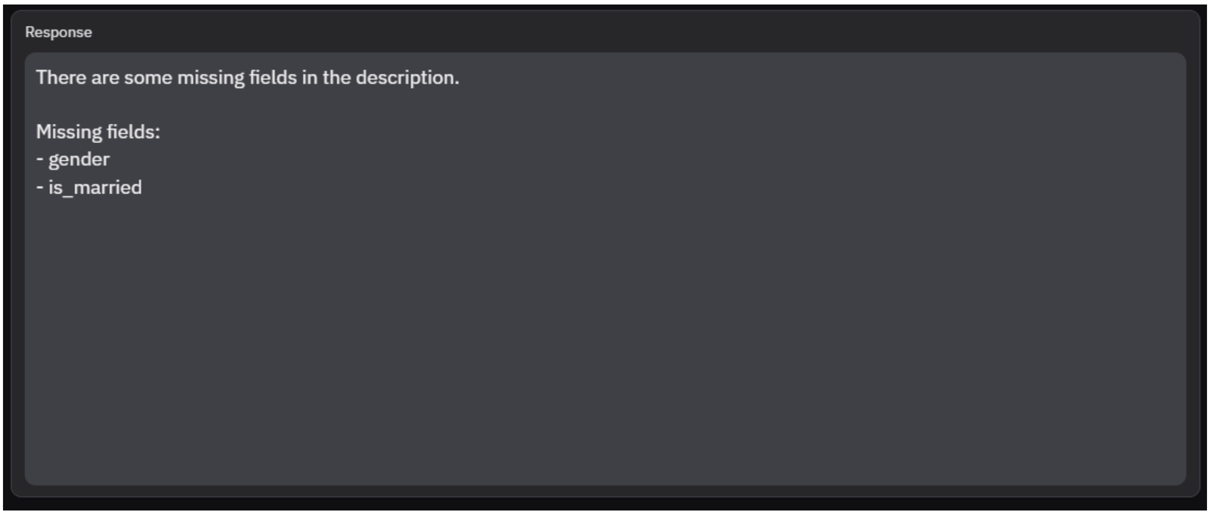
The customer is not a senior citizen and has dependents. Their tenure is 18 months. They have phone service and dual service is set to No phone service. The internet service is Fiber optic with no online security or online backup. Device protection and tech support are both Yes. They stream TV and movies, and their contract is month-to-month with paperless billing enabled. The payment method is electronic check. Monthly charges are 75.50 dollars, and total charges are 1349.75 dollars.

The chatbot will respond:

There are some missing fields in the description.

Missing fields:

- *gender*
- *is_married*



There are some missing fields in the description.

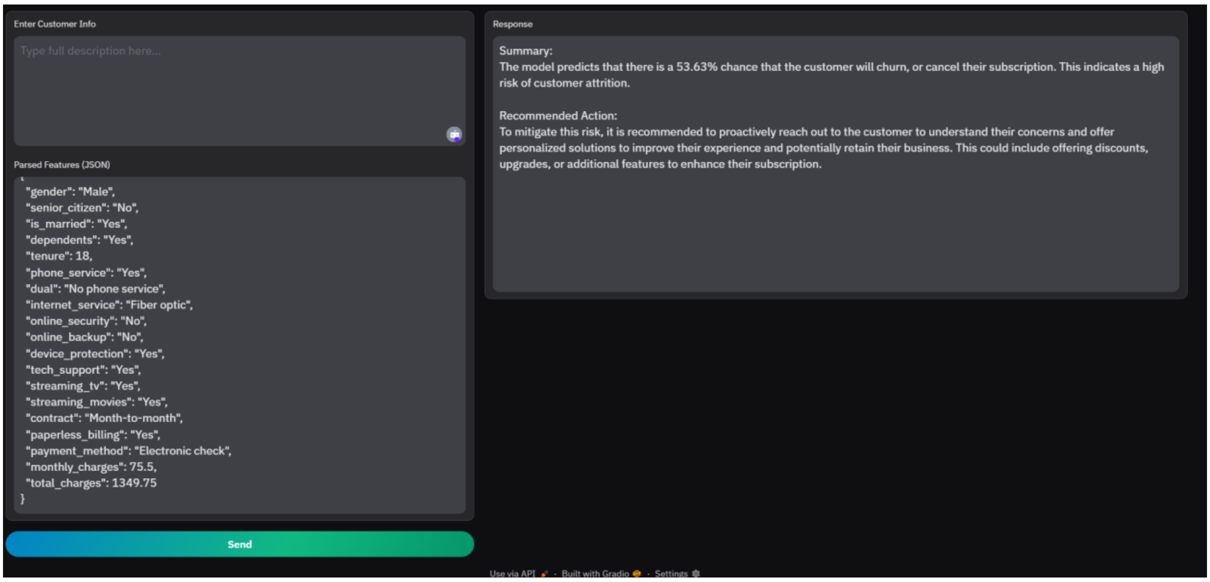
Missing fields:
- gender
- is_married

5. Provide Missing Information

To complete the missing data, enter a new description with the missing info, e.g.:

He is a male and married

Press **Send** again, and the chatbot will update the features accordingly and provide the updated prediction.


Enter Customer Info
Type full description here...
Parsed Features (JSON)
Send
Response
Use via API · Built with Gradio · Settings

```
{
  "gender": "Male",
  "senior_citizen": "No",
  "is_married": "Yes",
  "dependents": "Yes",
  "tenure": 18,
  "phone_service": "Yes",
  "dual": "No phone service",
  "internet_service": "Fiber optic",
  "online_security": "No",
  "online_backup": "No",
  "device_protection": "Yes",
  "tech_support": "Yes",
  "streaming_tv": "Yes",
  "streaming_movies": "Yes",
  "contract": "Month-to-month",
  "paperless_billing": "Yes",
  "payment_method": "Electronic check",
  "monthly_charges": 75.5,
  "total_charges": 1349.75
}
```

Summary:
The model predicts that there is a 53.63% chance that the customer will churn, or cancel their subscription. This indicates a high risk of customer attrition.

Recommended Action:
To mitigate this risk, it is recommended to proactively reach out to the customer to understand their concerns and offer personalized solutions to improve their experience and potentially retain their business. This could include offering discounts, upgrades, or additional features to enhance their subscription.