Day67_LaundryShop_RuleBased_Chatbot.ipynb

August 19, 2025

1 Laundry Shop Chatbot (Rule-Based using NLTK)

1.1 Project Description

This is a **rule-based chatbot** built for an imaginary laundry shop.
It can handle **50+ FAQs** related to laundry services like washing, ironing, delivery, pricing, offers, timings, and more.

1.2 Features

- Answer customer FAQs using **pattern matching rules**.

- Provide details about services, prices, timings, offers, and contact info.

- If asked an **out-of-box question**, the bot will politely say:
  "Please call our customer care at +91-9876543210 or drop us an email at support@laundryshop.com"

1.3 Why Rule-Based?

- Easy to implement with predefined rules.

- Good for **small businesses like shops, salons, or laundry services**.

- Predictable and reliable answers for FAQs.

2 NLTK Rule-Based Chatbot with 50+ rules

```python
#  Laundry Shop Chatbot (Rule-Based)
# Import
from nltk.chat.util import Chat, reflections

# Define pairs (50+ Q&A rules)
pairs = [

    # Greetings
    [r"(hi|hello|hey|hii|good morning|good evening)",
```

```python
    ["Hello! Welcome to Fresh & Clean Laundry Shop  . How can I help you today?
↪"]],

    # Services
    [r"(.*)service(.*)",
     ["Our services include washing, ironing, dry cleaning, stain removal,
↪express service, and home delivery."]],
    [r"(i want to know your services|what are your services|tell me your
↪services)",
     ["Sure! We offer washing, ironing, dry cleaning, stain removal, express
↪service, and free home delivery on orders above  300."]],

    # Pricing
    [r"(.*)price(.*)", ["Our pricing depends on the item. Shirts  30, Pants  40,
↪Bedsheet  60, Dry Cleaning starts from  100."]],
    [r"(.*)cost(.*shirt.*)", ["Washing a shirt costs  30."]],
    [r"(.*)cost(.*pant.*)", ["Washing pants costs  40."]],
    [r"(.*)cost(.*bedsheet.*)", ["Washing a bedsheet costs  60 per piece."]],
    [r"(.*)dry clean(.*)", ["Dry cleaning starts from  100 depending on the
↪item."]],

    # Delivery
    [r"(.*)delivery(.*)", ["Regular delivery in 48 hours. Express delivery in
↪24 hours."]],
    [r"(how long|time).*delivery(.*)", ["Delivery usually takes 48 hours.
↪Express delivery is available in 24 hours."]],
    [r"(.*)pickup(.*)", ["We provide free pickup and delivery for orders above
↪ 300."]],

    # Discounts / Offers
    [r"(.*)offer(.*)", ["We currently have 10% off on orders above  500!"]],
    [r"(.*)discount(.*)", ["Yes! 10% discount for orders above  500, and 15%
↪discount for students."]],

    # Payment
    [r"(.*)payment(.*)", ["We accept cash, UPI, debit/credit cards, and online
↪wallets."]],
    [r"(.*)upi(.*)", ["Yes, we accept all UPI payments (PhonePe, Google Pay,
↪Paytm)."]],

    # Membership
    [r"(.*)membership(.*)", ["We have monthly and yearly membership plans with
↪free pickup and priority delivery."]],

    # Contact
```

```python
    [r"(.*)contact(.*)", ["You can call us at +91-9876543210 or email us at␣
 ↪support@laundryshop.com"]],
    [r"(.*)customer care(.*)", ["Please call our customer care at␣
 ↪+91-9876543210 or email support@laundryshop.com"]],

    # Location
    [r"(.*)location(.*)", ["We are located at MG Road, Pune. You can also book␣
 ↪online for pickup."]],

    # Working hours
    [r"(.*)open(.*)", ["We are open from 9 AM to 9 PM, all days of the week."]],
    [r"(.*)close(.*)", ["We close at 9 PM every day."]],

    # Special care
    [r"(.*)stain(.*)", ["Yes, we provide stain removal for coffee, ink, oil,␣
 ↪and more."]],
    [r"(.*)wool(.*)", ["We take special care of woolen clothes with gentle wash.
 ↪"]],
    [r"(.*)silk(.*)", ["Silk clothes are handled with premium dry cleaning."]],

    # Express Service
    [r"(.*)express(.*)", ["Express delivery is available within 24 hours for an␣
 ↪extra  50 per item."]],

    # General Questions
    [r"(.*)who are you(.*)", ["I am Fresh & Clean Laundry Shop's assistant bot␣
 ↪"]],
    [r"(.*)what can you do(.*)", ["I can help you with services, pricing,␣
 ↪delivery, offers, and more."]],
    [r"(.*)how are you(.*)", ["I'm always fresh and clean!  How can I help you␣
 ↪today?"]],

    # Goodbye
    [r"(bye|goodbye|see you|exit|quit|ok bye)",
     ["Thank you for visiting Fresh & Clean Laundry Shop  . Goodbye!"]],

    # Fallback (MUST BE LAST)
    [r"(.*)",
     ["Sorry, I can't answer that. Please call our customer care at␣
 ↪+91-9876543210 or drop an email at support@laundryshop.com"]],
]

# Start Chat
print(" Fresh & Clean Laundry Shop Chatbot ")
chat = Chat(pairs, reflections)
chat.converse()
```

```
Fresh & Clean Laundry Shop Chatbot
```

> Hi

Hello! Welcome to Fresh & Clean Laundry Shop . How can I help you today?

> What are your services?

Our services include washing, ironing, dry cleaning, stain removal, express
service, and home delivery.

> How much does it cost to wash a shirt?

Washing a shirt costs 30.

> What are your dry cleaning charges?

Dry cleaning starts from 100 depending on the item.

> Do you provide pickup?

We provide free pickup and delivery for orders above 300.

> ok bye

Thank you for visiting Fresh & Clean Laundry Shop . Goodbye!

---

3   Laundry Shop Chatbot (Streamlit Version)

We are upgrading our **rule-based chatbot** into a **Streamlit Web Application**.

Features:

- Interactive Chatbox UI
- Professional branding (Laundry Shop)
- Same rule-based engine (50+ Q&A)
- Works directly in the browser

Steps to Run

1. Install Streamlit

    ```
    pip install streamlit
    ```

2. Save the chatbot code into a Python file (example: laundry_chatbot_app.py).

3. Run the app using:

    ```
    streamlit run laundry_chatbot_app.py
    ```

4. Your chatbot will open in the browser at http://localhost:8501.

4 Note:

- This is still rule-based, so it works on predefined Q&A.
- If user input doesn't match, it shows fallback: > "Sorry, I can't answer that. Please call our customer care at +91-9876543210 or drop an email at support@laundryshop.com"

5 Full Streamlit Code (`laundry_chatbot_app.py`)

```python
import streamlit as st
from nltk.chat.util import Chat, reflections
import base64


# -------------------------------
#  Chatbot rules
# -------------------------------
pairs = [
    [r"(hi|hello|hey|hii|good morning|good evening)",
     ["Hello! Welcome to Fresh & Clean Laundry Shop  . How can I help you today?"]],
    [r"(.*)service(.*)",
     ["Our services include washing, ironing, dry cleaning, stain removal, express service, and
    [r"(.*)price(.*)",
     ["Our pricing depends on the item. Shirts  30, Pants  40, Bedsheet  60, Dry Cleaning starts
    [r"(.*)delivery(.*)",
     ["Regular delivery in 48 hours. Express delivery in 24 hours."]],
    [r"(.*)offer(.*)",
     ["We currently have 10% off on orders above  500!"]],
    [r"(.*)payment(.*)",
     ["We accept cash, UPI, debit/credit cards, and online wallets."]],
    [r"(.*)contact(.*)",
     ["You can call us at +91-9876543210 or email us at support@laundryshop.com"]],
    [r"(bye|goodbye|see you|exit|quit|ok bye)",
     ["Thank you for visiting Fresh & Clean Laundry Shop  . Goodbye!"]],
    [r"(.*)",
     ["Sorry, I can't answer that. Please call our customer care at +91-9876543210 or drop an e
]

chat = Chat(pairs, reflections)


# -------------------------------
#  Background image
# -------------------------------
def add_bg_from_local(image_file):
    with open(image_file, "rb") as file:
        encoded = base64.b64encode(file.read()).decode()
    st.markdown(
        f"""
        <style>
```

```python
        .stApp {{
            background-image: url("data:image/jpg;base64,{encoded}");
            background-size: cover;
            background-position: center;
            background-attachment: fixed;
        }}
        </style>
        """,
        unsafe_allow_html=True
    )


add_bg_from_local("pexels-myatezhny39-2927523.jpg")  # replace with your file


# ------------------------------
#  UI Title
# ------------------------------
st.markdown("<h1 style='text-align:center; color:white;'> Fresh & Clean Laundry Shop Chatbot</


# ------------------------------
#  Input
# ------------------------------
user_input = st.text_input("  Ask me something:")

if user_input:
    response = chat.respond(user_input)

    #  Instead of appending, overwrite old chat
    st.session_state.history = [("You", user_input), ("Bot", response)]


# ------------------------------
#  Show only the latest Q&A
# ------------------------------
if "history" in st.session_state:
    for speaker, msg in st.session_state.history:
        if speaker == "You":
            st.markdown(
                f"""
                <div style='text-align:right; margin:10px;'>
                    <span style='background:#4CAF50; color:white; padding:10px 15px; border-rad
                        {msg}
                    </span>
                </div>
                """, unsafe_allow_html=True
            )
        else:  # Bot
            st.markdown(
                f"""
                <div style='text-align:left; margin:10px;'>
```
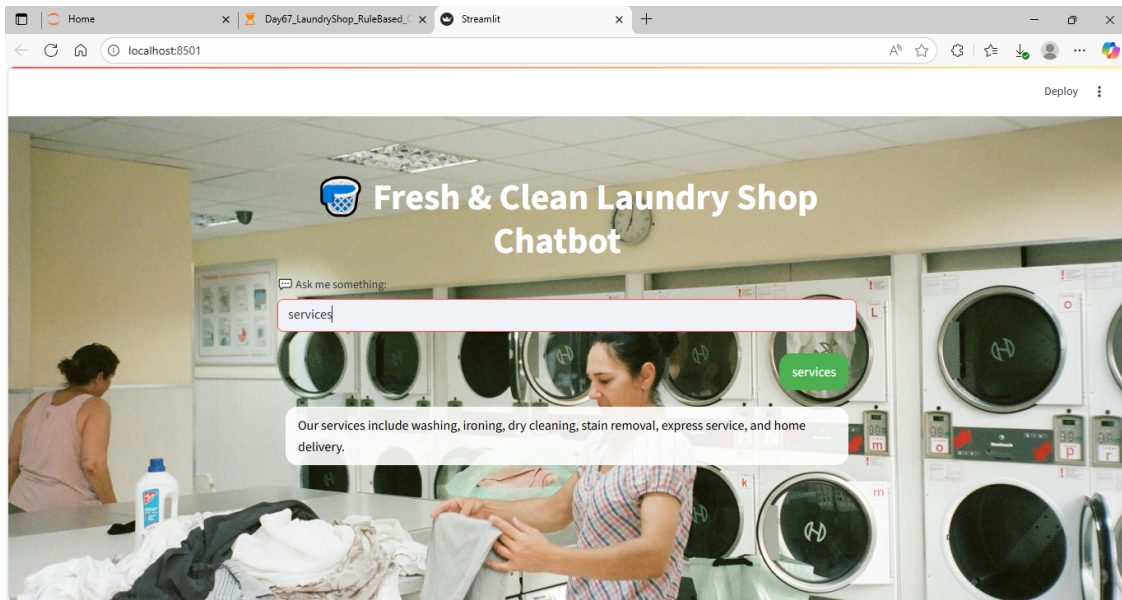
```
        <span style='background:rgba(255,255,255,0.9); color:black; padding:10px 15
            {msg}
        </span>
    </div>
    """, unsafe_allow_html=True
)
```

6   Fresh & Clean Laundry Shop Chatbot (Final Look)

Here is how our chatbot app looks when deployed with Streamlit


```
[ ]: Thank You !
```