



OGTIP INTERNSHIP

Simple Hospital Chatbot

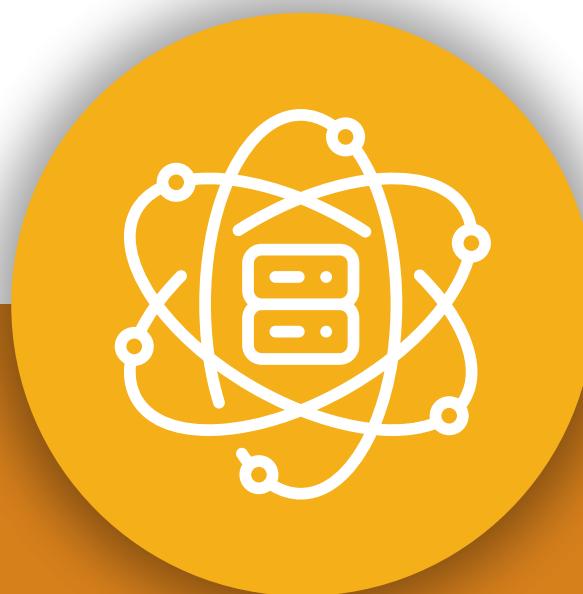
By Carrissa Smith

Problem Statement:

"As a Data Scientist you are required to make a hospital chatbot for a renowned hospital providing first class assistance in peak hrs of the hospital. For your reference you are being given a data set that includes {Queries/response} collection of data. Make necessary changes to the dataset to make an interactive chatbot. Build a machine learning model that can generate response to an individual's query to help the hospital customer care department."



Project Tasks



**Provide
Complete EDA
analysis**



**Provide predictive
machine
learning/Deep
learning models
for the given
dataset.**



**Include charts and
diagrams for
better display of
Analysis.**



EDA

Normalization

Shape of data

Null values

Normalization

Using `df = pd.json_normalize(responses, 'intents', errors='ignore')`
we were able to transform the json file to a dataframe.

```
{"intents": [  
    {"tag": "greeting",  
        "patterns": ["Hi there", "How are you", "Is anyone there?", "Hey", "Hola", "Hello", "Good day"],  
        "responses": ["Hello, thanks for asking", "Good to see you again", "Hi there, how can I help?"],  
        "context": []},  
    {"tag": "goodbye",  
        "patterns": ["Bye", "See you later", "Goodbye", "Nice chatting to you, bye", "Till next time"],  
        "responses": ["See you!", "Have a nice day", "Bye! Come back again soon."],  
        "context": []},  
    {"tag": "thanks",  
        "patterns": ["Thanks", "Thank you", "That's helpful", "Awesome, thanks", "Thanks for helping me"],  
        "responses": ["Happy to help!", "Any time!", "My pleasure"],  
        "context": []},  
    {"tag": "noanswer",  
        "patterns": [],  
        "responses": ["Sorry, can't understand you", "Please give me more info", "Not sure I understand"],  
        "context": []},  
    {"tag": "options",  
        "patterns": ["How you could help me?", "What you can do?", "What help you provide?", "How you can be helpful?", "What support is offered"],  
        "responses": ["I can guide you through Adverse drug reaction list, Blood pressure tracking, Hospitals and Pharmacies", "Offering support for Adverse drug reaction, Blood pressure, Hospitals and Pharmacies"],  
        "context": []},  
    {"tag": "adverse_drug",  
        "patterns": ["How to check Adverse drug reaction?", "Open adverse drugs module", "Give me a list of drugs causing adverse behavior", "List all drugs suitable for patient with adverse reaction", "Which drugs dont have adverse reaction?" ],  
        "responses": ["Navigating to Adverse drug reaction module"],  
        "context": []},  
    ]}
```



Lorem ipsum dolor sit amet, cons
adipiscing elit. In ultricies, justo eu u
egnoscimus in nunc fermentum
faucibus erat urna sed magna. Morbi
arcu vel arcu feugiat, ut mollis sapi

Normalization

Using `df = pd.json_normalize(responses, 'intents', errors='ignore')`
we were able to transform the json file to a dataframe.



tag	patterns	responses	context
greeting	[Hi there, How are you, Is anyone there?, Hey,...]	[Hello, thanks for asking, Good to see you aga...]	
goodbye	[Bye, See you later, Goodbye, Nice chatting to...]	[See you!, Have a nice day, Bye! Come back aga...]	
thanks	[Thanks, Thank you, That's helpful, Awesome, t...]	[Happy to help!, Any time!, My pleasure]	
noanswer		[Sorry, can't understand you, Please give me m...]	
options	[How you could help me?, What you can do?, Wha...]	[I can guide you through Adverse drug reaction...]	
adverse_drug	[How to check Adverse drug reaction?, Open adv...]	[Navigating to Adverse drug reaction module]	
blood_pressure	[Open blood pressure module, Task related to b...]	[Navigating to Blood Pressure module]	
blood_pressure_search	[I want to search for blood pressure result hi...]	[Please provide Patient ID, Patient ID?] [search_blood_pressure_by_patient_id]	
search_blood_pressure_by_patient_id		[Loading Blood pressure result for Patient]	
pharmacy_search	[Find me a pharmacy, Find pharmacy, List of ph...]	[Please provide pharmacy name] [search_pharmacy_by_name]	
search_pharmacy_by_name		[Loading pharmacy details]	
hospital_search	[Lookup for hospital, Searching for hospital t...]	[Please provide hospital name or location] [search_hospital_by_params]	
search_hospital_by_params		[Please provide hospital type] [search_hospital_by_type]	
search_hospital_by_type		[Loading hospital details]	

Shape of Data and Null Values

The context column has 10 empty values and the patterns column has 5 empty lists. The shape of the data is (14, 4).

	tag	patterns	responses	context
0	greeting	[Hi there, How are you, Is anyone there?, Hey,...	[Hello, thanks for asking, Good to see you aga...	
1	goodbye	[Bye, See you later, Goodbye, Nice chatting to...	[See you!, Have a nice day, Bye! Come back aga...	
2	thanks	[Thanks, Thank you, That's helpful, Awesome, t...	[Happy to help!, Any time!, My pleasure]	
3	noanswer		[Sorry, can't understand you, Please give me m...	
4	options	[How you could help me?, What you can do?, Wha...	[I can guide you through Adverse drug reaction...	
5	adverse_drug	[How to check Adverse drug reaction?, Open adv...	[Navigating to Adverse drug reaction module]	
6	blood_pressure	[Open blood pressure module, Task related to b...	[Navigating to Blood Pressure module]	
7	blood_pressure_search	[I want to search for blood pressure result hi...	[Please provide Patient ID, Patient ID?] [search_blood_pressure_by_patient_id]	
8	search_blood_pressure_by_patient_id		[Loading Blood pressure result for Patient]	
9	pharmacy_search	[Find me a pharmacy, Find pharmacy, List of ph...	[Please provide pharmacy name]	[search_pharmacy_by_name]
10	search_pharmacy_by_name		[Loading pharmacy details]	
11	hospital_search	[Lookup for hospital, Searching for hospital t...	[Please provide hospital name or location]	[search_hospital_by_params]
12	search_hospital_by_params		[Please provide hospital type]	[search_hospital_by_type]
13	search_hospital_by_type		[Loading hospital details]	

Predictive Deep Learning Model



Preprocessing Training Data



```
#preprocessing training data
#converts text to lowercase
training_data['patterns']= training_data['patterns'].str.lower()
"""creates an instance of tfidvectorizer class and specifies that the vectorizer should consider
unigrams (individual words) and bigrams (pairs of consecutive words)"""
vectorizer= TfidfVectorizer(ngram_range=(1,2), stop_words="english")
training_data_tfidf= vectorizer.fit_transform(training_data['patterns']).toarray()
```

```
#preprocessing target variable(tags)
le= LabelEncoder()
training_data_tags_le= pd.DataFrame({"tags": le.fit_transform(training_data["tags"])})
#This line one-hot encodes the categorical labels in the 'tags' column.
training_data_tags_dummy_encoded= pd.get_dummies(training_data_tags_le["tags"]).to_numpy()
```

Creating the DNN



This was the creation of our deep neural network. We created a sequential model which is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor.

```
hospbot= Sequential()
#These lines add layers to the sequential model.
hospbot.add(Dense(10, input_shape=(len(training_data_tfidf[0]),)))
hospbot.add(Dense(8))
hospbot.add(Dense(8))
hospbot.add(Dense(6))
#This line adds the output layer.
hospbot.add(Dense(len(training_data_tags_dummy_encoded[0])), activation="softmax"))
hospbot.compile(optimizer="rmsprop", loss="categorical_crossentropy", metrics=["accuracy"])
```

TfidfVectorizer



Now we are using the TfidfVectorizer to learn the vocabulary and idf (inverse document frequency) from the training data. It analyzes the patterns in the text data and builds a vocabulary based on the specified n-gram range.

```
# fitting TfIdfVectorizer with training data to preprocess inputs
#The purpose of lowercasing is often to ensure consistency and improve the performance of text-based
training_data["patterns"] = training_data["patterns"].str.lower()
vectorizer = TfidfVectorizer(ngram_range=(1, 2), stop_words="english")
vectorizer.fit(training_data["patterns"])

TfidfVectorizer
TfidfVectorizer(ngram_range=(1, 2), stop_words='english')
```

Predicting Tag



```
# fitting LabelEncoder with target variable(tags) for inverse transformation of predictions
le = LabelEncoder()
le.fit(training_data["tags"])
#Now we're using the TfidfVectorizer to learn the vocabulary and
#idf (inverse document frequency) from the training data. It analyzes
#the pdff
LabelEncoder()

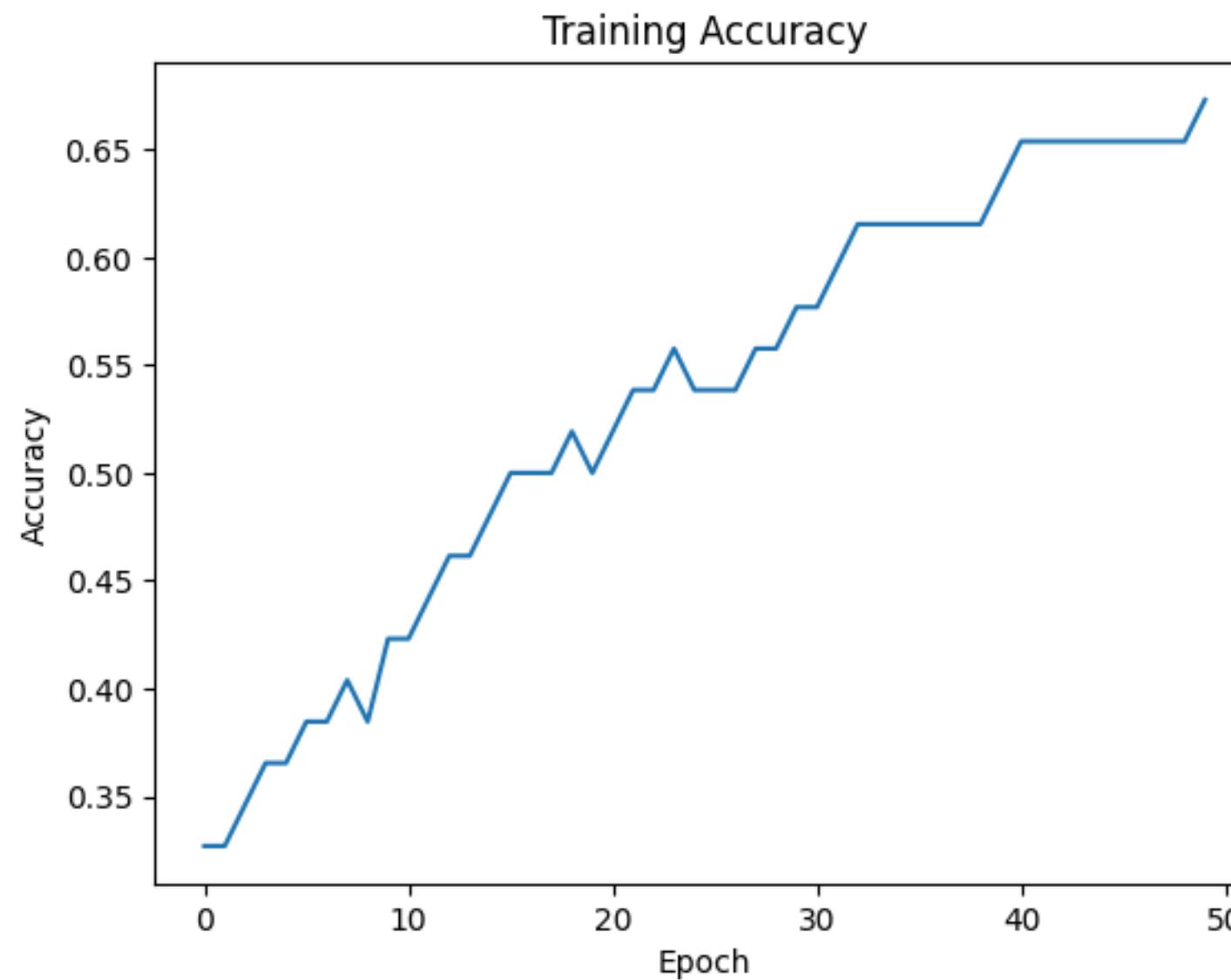
# transforming input and predicting intent
def predict_tag(inp_str):
    inp_data_tfidf = vectorizer.transform([inp_str.lower()]).toarray()
    #Prediction using the Neural Network Model
    predicted_proba = hospbot.predict(inp_data_tfidf)
    #Decoding Predictions
    encoded_label = [np.argmax(predicted_proba)]
    predicted_tag = le.inverse_transform(encoded_label)[0]
    #Stripping Whitespace
    predicted_tag = predicted_tag.strip()
    return predicted_tag
```

Creating a Chat Function

```
#defining chat function
def chat():
    print("_____" )
    print(" | Hospital Chatbot |")
    print(" | _____ |")
    print(" | |")
    print(" | How may I be of assistance? |")
    print(" | Type EXIT to quit |")
    print(" | _____ |")
while True:
    inp = input("Ask anything... : ")
    if inp == "EXIT":
        break
    else:
        if inp:
            tag = predict_tag(inp)
            response = random.choice(responses_by_tag[tag])
            print("Response... : ", response)
        else:
            pass
# calling chat function to start chatting
chat()
```

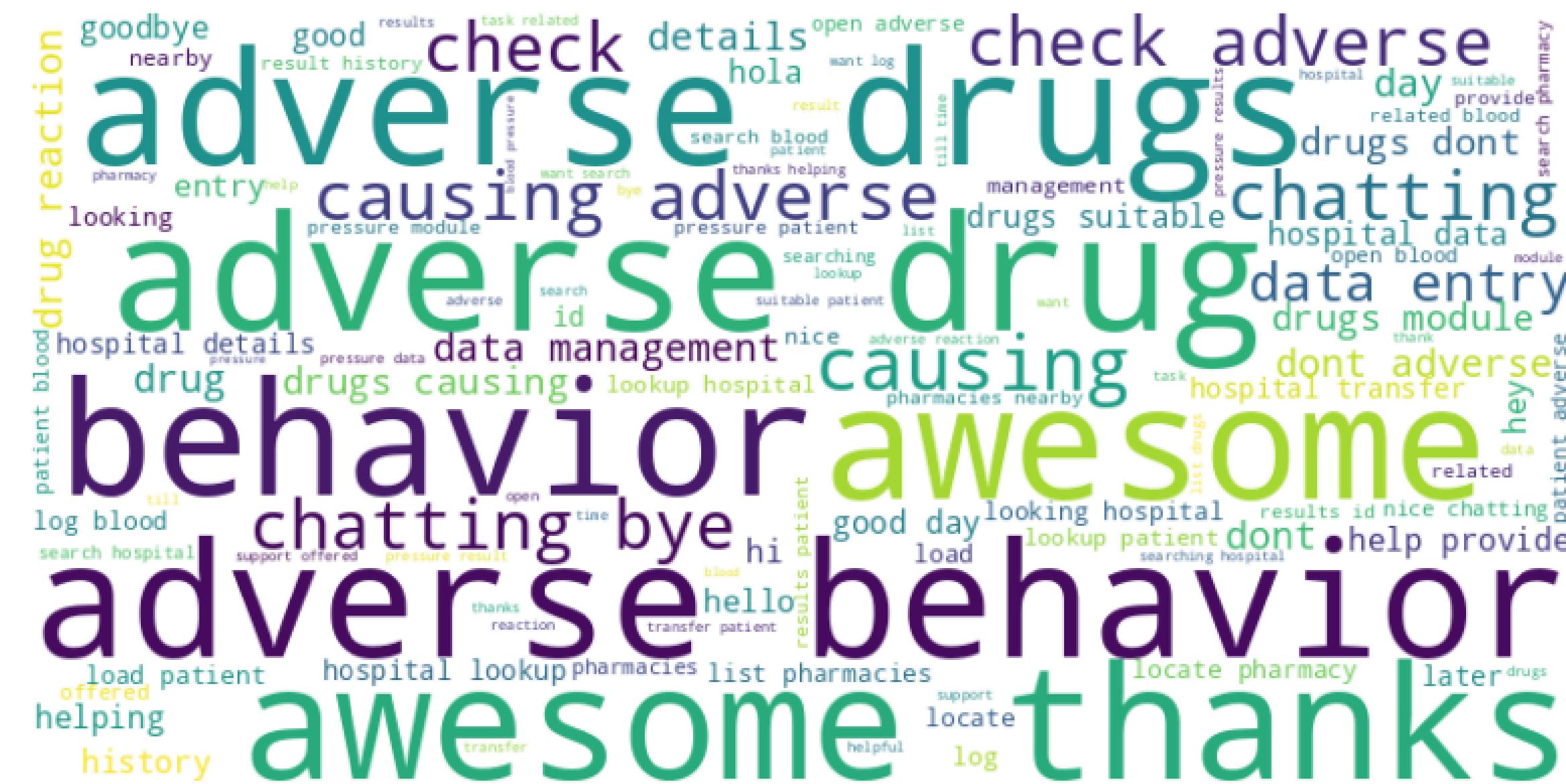
This code defines a simple chat function for a hospital chatbot. The function runs in a loop, prompting the user to input a question or statement. If the user types "EXIT," the loop breaks, and the chat ends. Otherwise, it calls the predict_tag function to predict the tag for the user input, retrieves a random response associated with that tag, and prints the response. The chat continues until the user decides to exit by typing "EXIT."

Training Accuracy



This is a line graph of the training accuracy of the Hospbot deep learning model. We can see that the accuracy of the model grew with the number of iterations from the epoch.

Word Cloud of Vectorizer



Hospital Chatbot

```
| Hospital Chatbot
| 
| How may I be of assistance?
| Type EXIT to quit
|
Ask anything... : Hi there
1/1 [=====] - 0s 108ms/step
Response... : Hi there, how can I help?
Ask anything... : How you could help me?
1/1 [=====] - 0s 90ms/step
Response... : I can guide you through Adverse drug reaction list, Blood pressure tracking, Hospitals and Pharmacies
Ask anything... : How to check Adverse drug reaction?
1/1 [=====] - 0s 21ms/step
Response... : Please provide pharmacy name
Ask anything... : Fontana Pharmacy
1/1 [=====] - 0s 22ms/step
Response... : Please provide pharmacy name
Ask anything... : EXIT
```



Thank You!