Introduction:

Python chatbots have exploded in popularity in the technology and commercial sectors during the last several years. These clever bots are skilled at mimicking natural human languages and talking with humans that businesses use across several industrial sectors. From e-commerce companies to healthcare facilities, it appears as though everyone is utilizing this handy tool to generate business benefits.

What is a Chatbot?

A chatbot is a kind of artificial intelligence-based software meant to communicate with humans in their natural languages. These chatbots often interact via audio or textual means and mimic human languages to connect with humans in a human-like manner. A chatbot is, without a doubt, one of the most important uses of natural language processing

A ChatBot is merely software by which humans can interact with each other. Examples include Apple's Siri, Amazon's Alexa, Google Assistant, and Microsoft's Cortana.

A. Interaction of User for asking the name

First, we will ask Username

```
print("BOT: What is your name?")

user_name = input()
```

B. Response of ChatBot

I assigning questions and answers ChatBot must ask us, for example, we have assigned three variables with different answers, most important point to note it can be used in code and can be also updated automatically by just changing in values of variables

```
name = "Bot Number 286"

monsoon = "rainy"

mood = "Smiley"

resp = {

"what's your name?": [
"They call me {0}".format(name),
"I usually go by {0}".format(name),
"My name is the {0}".format(name) ],
"what's today's weather?": [
"The weather is {0}".format(monsoon),
"It's {0} today".format(monsoon)],
"how are you?": [
"I am feeling {0}".format(mood),
"{0}! How about you?".format(mood),
"I am {0}! How about yourself?".format(mood), ],
"": [
"Hey! Are you there?",
"What do you mean by these?",
],
"default": [
"This is a default message"] }
```

C. Another Function

Sometimes the questions added are not related to available questions, and sometimes some letters are forgotten to write in the chat. At that time, the bot will not answer any questions, but another function is forward.

```
def real(xtext):
if "name" in xtext:
ytext = "what's your name?"
elif "monsoon" in xtext:
ytext = "what's today's weather?"
elif "how are" in xtext:
ytext = "how are you?"
else:
```

ytext = ""

return ytext

D. Sending back the message function

def send_message(message):

print((message))

response = res(message)

print((response))

E. Final Step to break the loop

while 1:

my_input = input()

my_input = my_input.lower()

related_text = real(my_input)

send_message(related_text)

if my_input == "exit" or my_input == "stop":

break

Preprocess a dataset for a chatbot using Python.

Creating a chatbot in Python involves several steps, and preprocessing the dataset is a crucial part of it. Below is a simplified guide on how to preprocess a dataset for a chatbot using Python.

1. Choose a Dataset:

Select a dataset that fits your chatbot's purpose. Common chatbot datasets include conversational data, question-answer pairs, or dialogue datasets. Ensure the data is in a structured format like CSV, JSON, or plain text.

2. Import Libraries:

import pandas as pd

```
import numpy as np

import re

from sklearn.model_selection import train_test_split
```

3. Load and Inspect Data:

```
# Assuming your data is in a CSV file
data = pd.read_csv('your_dataset.csv')

# Check the structure of your data
print(data.head())
```

4. Text Cleaning:

Clean the text data to remove noise and irrelevant information. Common cleaning steps include converting to lowercase, removing special characters, and handling contractions.

```
def clean_text(text):
    text = text.lower()
    text = re.sub(r"[^a-zA-Z0-9]", " ", text)
    # Add more cleaning steps as needed
    return text

data['cleaned_text'] = data['text_column'].apply(clean_text)
```

5. Tokenization:

Tokenize the text into individual words or subwords. You can use tokenization libraries like NLTK or Spacy.

```
from nltk.tokenize import word_tokenize

data['tokenized_text'] = data['cleaned_text'].apply(word_tokenize)
```

6. Train-Test Split:

Split the dataset into training and testing sets

```
train_data, test_data = train_test_split(data, test_size=0.2, random_state=42)
```

7. Prepare Input-Output Pairs:

Create input-output pairs for training the chatbot. Depending on your chatbot type, you might have single-turn or multi-turn conversations

```
# For a simple question-answer chatbot
X_train = train_data['tokenized_text'].values
y_train = train_data['response_column'].values

X_test = test_data['tokenized_text'].values
y_test = test_data['response_column'].values
```

8. Vectorization:

Convert the tokenized text into numerical vectors using techniques like TF-IDF or word embeddings.

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf_vectorizer = TfidfVectorizer()
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)
```

9. Build and Train the Model:

Use a machine learning model or a deep learning framework (like TensorFlow or PyTorch) to build and train your chatbot model.

10. Evaluate and Test:

Evaluate the performance of your chatbot on the test set and make adjustments as needed

```
from sklearn.metrics import accuracy_score
y_pred = model.predict(X_test_tfidf)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
```

Program:

```
pip install chatterbot

from chatterbot import ChatBot

from chatterbot.trainersimport ChatterBotCorpusTrainer

from chatterbot.trainers import ListTrainer

chatbot = ChatBot('MyBot')

trainer = ChatterBotCorpusTrainer(chatbot)

trainer.train('chatterbot.corpus.english')

list_trainer = ListTrainer(chatbot)

ist_trainer.train([

'Hi there!',

'Hello!',

'How are you?',

'I am good, thank you!'

,'What is your name?',

'I am a chatbot.'])

response = chatbot.get_response('Hi')

print(response)

python chatbot.py
```

Output:

```
you : Hi there!

chatbot : Hello!

you : How are you?

chatbot : I am good, thanks!

you : What is your name?

chatbot : I am a chatbot
```