NAAN MUDHALVAN

PROJECT TITLE : CREATE A CHATBOT IN PYTHON

PHASE 3

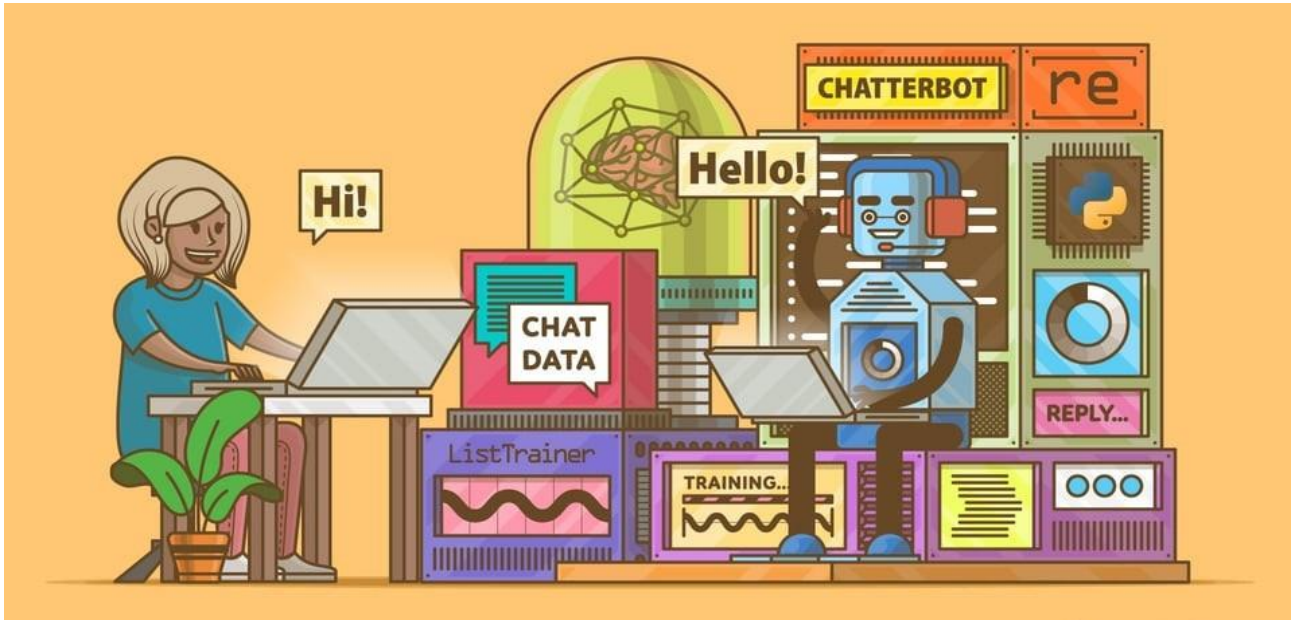
SUBMMISION DOCUMENTS

NAME : MUTHAIAHTHARAN R

REG NO : 712221104008

YEAR&DEPT : 3RD & CSE

COLLEGE : PARK COLLEGE OF ENG & TECH



CREATE A CHATBOT IN PYTHON

Introduction:

Creating a chatbot in Python can be a fun and educational project. A chatbot is a program that simulates conversation with users, providing responses to their inputs. Here's a basic introduction to get you started:

❖ Choose a Chatbot Type:

Decide what type of chatbot you want to create. Is it a rule-based chatbot that follows predefined rules, or an AI-powered chatbot that uses natural language processing (NLP) and machine learning to understand and respond to user inputs? For this introduction, we'll focus on a rule-based chatbot.

❖ Python Environment:

You'll need Python installed on your system. If you don't have it, download and install it from Python's official website.

❖ Select a Library:

For a rule-based chatbot you can use libraries like NLTK or spaCy Jifor text processing. For more advanced AI-powered chatbots, libraries like Rasa or Dialogflow are good choices.

❖ Data Preparation:

You'll need a dataset of predefined questions and responses. You can create this manually or use existing datasets. For instance, you can create a dictionary of questions and corresponding an

❖ Deployment:

You can deploy your chatbot on a website, integrate it with messaging platforms, or create a standalone application depending on your project's goals.

GIVEN DATASET:

	input	target
0	hi, how are you doing?	i'm fine. how about yourself?
1	i'm fine. how about yourself?	i'm pretty good. thanks for asking.
2	i'm pretty good. thanks for asking.	no problem. so how have you been?
3	no problem. so how have you been?	i've been great. what about you?
4	i've been great. what about you?	i've been good. i'm in school right now.

___Necessary step to follow:

1.Import libraries:

Step by importing the necessary libraries

Program:

```
import pandas as pd
import numpy as np
import string
from string import digits
import matplotlib.pyplot as plt
import seaborn as sns
import re
from sklearn.model_selection impor
t train_test_split
import tensorflow as tf
from keras.layers import Input, LS
TM, Embedding, Dense, Bidirectiona
l, Concatenate, Dot, Activation, T
imeDistributed
from keras.models import Model
from keras.utils import plot_model
```

2.Load the dataset:

Load your dataset into a pandas Dataframe.You Can typically find create a Chatbot in python dataset in CSV format ,but you can adapt code to other format as needed.

Program:

```
data_path = "/kaggle/input/simple-
dialogs-for-chatbot/dialogs.txt"
```

```
with open(data_path, 'r', encoding
='utf-8') as f:
    lines = f.read().split('\n')
inputs = []
targets = []
num_samples = 10000  # Number of s
amples to train on.
for line in lines[: min(num_sample
s, len(lines) - 1)]:

  input, target = line.split('\t')
  inputs.append(input)
  targets.append(target)
```

```
lines = pd.DataFrame({'input':inpu
ts, 'target':targets})
```

```
lines.shape
```

```
lines.head()
```

Challenge involved in loading preprocessing a create a Chatbot in python

Data set:

Data Collection: Gathering and curating a diverse and comprehensive dataset for training the chatbot is crucial. It may involve web scraping, using existing chat logs, or generating synthetic data.
Data Cleaning: The collected data often requires thorough cleaning to remove noise, irrelevant information, and potentially offensive content. This can be a time-consuming process.
Data Preprocessing: Tokenization: Breaking down text into tokens (words, phrases, or sentences).Stopword Removal: Eliminating common words that don't carry much meaning.Lemmatization or Stemming: Reducing words to their base or roo.

How to overcome the challenge involved in loading and preprocessing a create a Chatbot in python:

- Tokenization: Breaking down text into tokens (words, phrases, or sentences).
- Stopword Removal: Eliminating common words that don't carry much meaning.
- Lemmatization or Stemming: Reducing words to their base or root forms.
- Handling Spelling Errors: Dealing with misspelled words.
- Sentence Segmentation: Dividing text into sentences.

Loading the dataset

➢ Choose a Dataset:

Organize your dataset into a format suitable for training. Common formats are JSON, CSV, or plain text.

Typically, you'll need pairs of input (user messages) and output (bot responses) for training.

➢ Load the Dataset:

You can use Python's standard file I/O operations or a library like pandas (for CSV) or json (for JSON) to load your dataset into your code.

Program:

```
def cleanup(lines):
  # Since we work on word level, if we normalize the text to lower case, this will re
e the vocabulary. It's easy to recover the case later.
  lines.input=lines.input.apply(lambda x: x.lower())
  lines.target=lines.target.apply(lambda x: x.lower())

  # To help the model capture the word separations, mark the comma with special token
  lines.input=lines.input.apply(lambda x: re.sub("'", '', x)).apply(lambda x: re.su
(",", ' COMMA', x))
  lines.target=lines.target.apply(lambda x: re.sub("'", '', x)).apply(lambda x: re.
(",", ' COMMA', x))

  # Clean up punctuations and digits. Such special chars are common to both domains,
can just be copied with no error.
  exclude = set(string.punctuation)
  lines.input=lines.input.apply(lambda x: ''.join(ch for ch in x if ch not in exclu
e))
  lines.target=lines.target.apply(lambda x: ''.join(ch for ch in x if ch not in exc
e))

  remove_digits = str.maketrans('', '', digits)
  lines.input=lines.input.apply(lambda x: x.translate(remove_digits))
  lines.target=lines.target.apply(lambda x: x.translate(remove_digits))
```

```
st_tok = 'START_'
end_tok = '_END'
def data_prep(lines):
  cleanup(lines)
  lines.target = lines.target.apply(lambda x : st_tok + ' ' + x + ' ' + end_tok)
```

```
data_prep(lines)
```

```
lines.head()
```

Loading the dataset:

Output:

	input	target
0	hi COMMA how are you doing	START_ im fine how about yourself _END
1	im fine how about yourself	START_ im pretty good thanks for asking _END
2	im pretty good thanks for asking	START_ no problem so how have you been _END
3	no problem so how have you been	START_ ive been great what about you _END
4	ive been great what about you	START_ ive been good im in school right now _END