Create a Chat bot in python

Date	1 November 2023
Team ID	329
Project Name	Create a Chat bot in python
Name	Dilip

PROBLEM DEFINITION

Defining Problem Statement and prioritizing ideas based on Project.
It is a project that involves a computer program that can interact with user. With basic conversation.

“This Project is a software application designed to simulate human conversation and They can use our chatbot at any time”.

You May ask why this project is used for, we have given some list of ideas and thoughts that those are the reasons for creating this project.

Those reasons are given below,

PROBLEMS

Problems

Why we should use Chat Bot
&
Approaches used for Chat Bot

Ideas and Approaches

Dhanasekaren

- 24/7 Accessibility
- NLP(Natural Language Processing)
- Innovation and Modernization

Hemanth

- Efficient Information Retrieval
- Generative Models
- Assistance for International Students

Gunasekar

- Personalized Assistance
- Machine Learning
- Streamlined Admissions Process

Dilip

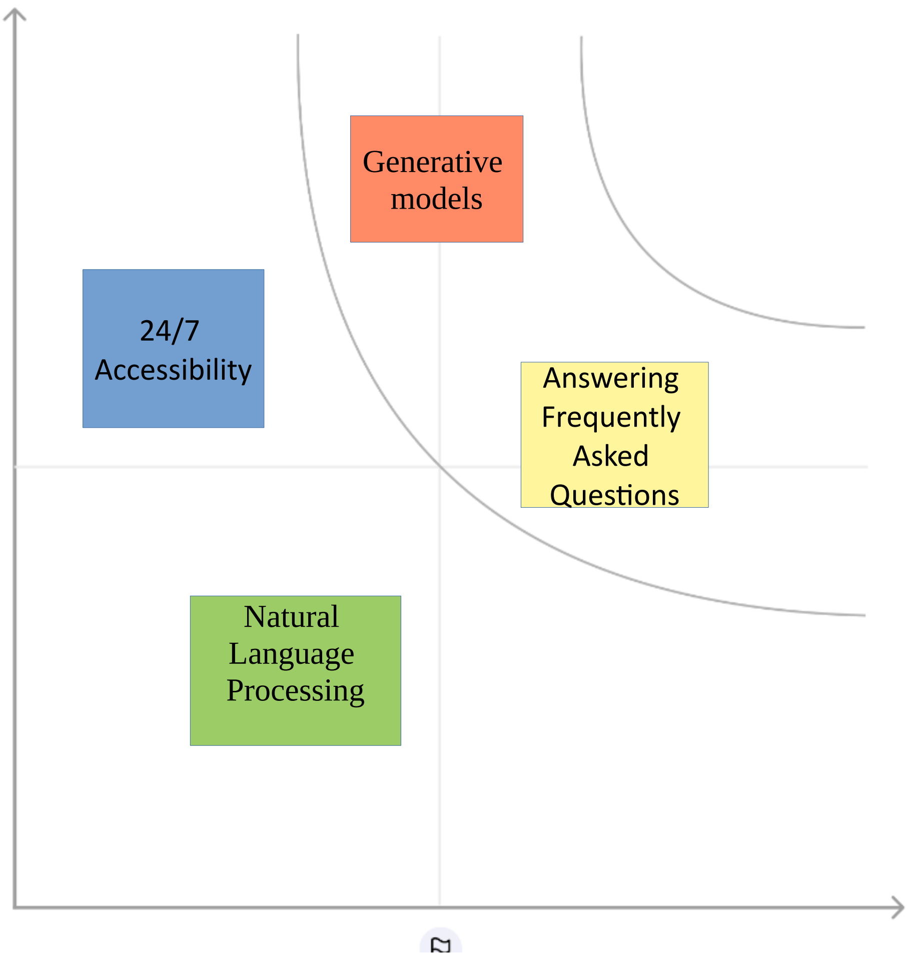
- Campus Navigation
- Voice-Based

Harish MD

- Event Reminders & Updates
- Rule-Based

Prioritizing Ideas

Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact


Generative models
24/7 Accessibility
Answering Frequently Asked Questions
Natural Language Processing

Feasibility
Regardless of their importance, which tasks are more feasible than others?
(Cost, time, effort, complexity, etc)

phases of development

phase 1: Problem statement, Design thinking process.

phase 2: Use a model, here i used Decision Tree.

Phase 3: Then use flask to create a Webpage

phase 4: Then use HTML and CSS to enhance the page

Libraries used

1.Flask

Flask is used for developing web applications using python

2.string

Strings can be used to handle textual data in Python

3.re

Regular Expression Syntax. A regular expression (or RE) specifies a set of strings that matches it; the functions in this module let you check if a particular string matches a given regular expression

4.unicodedata

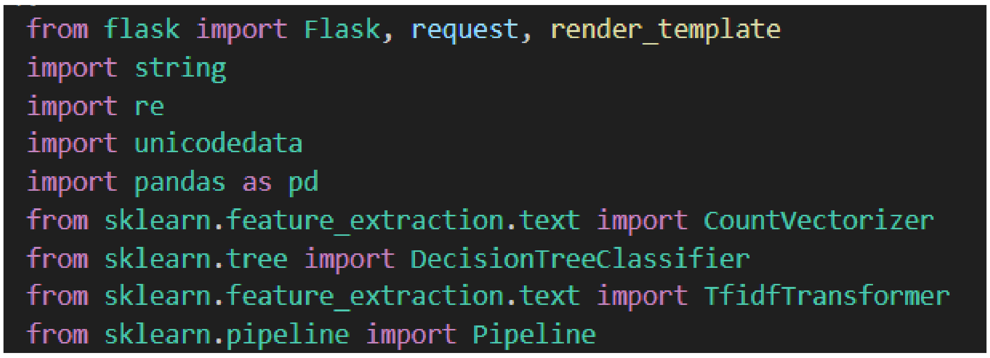
This module provides access to UCD and uses the same symbols and names as defined by the Unicode Character Database

5.pandas

It provides many functions and methods to expedite the data analysis process

6.sklearn

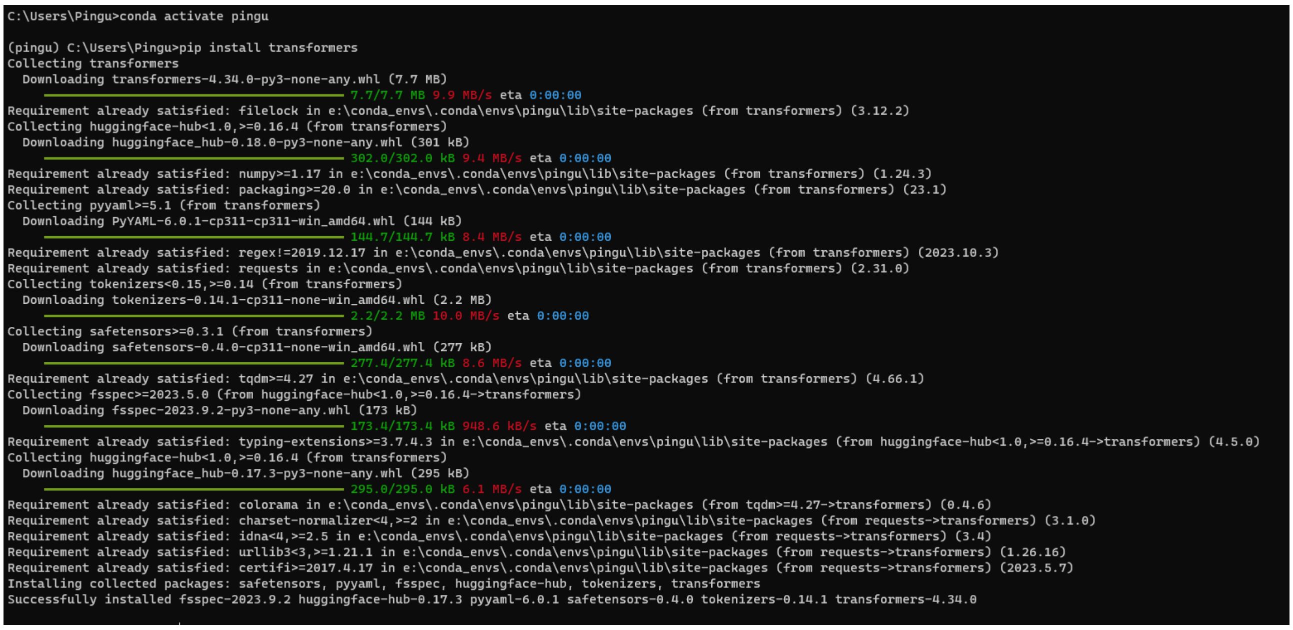
Scikit-Learn, also known as sklearn is a python library to implement machine learning models and statistical modelling



```python
from flask import Flask, request, render_template
import string
import re
import unicodedata
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.tree import DecisionTreeClassifier
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.pipeline import Pipeline
```

Integration of NLP techniques

Package name: transformers
Use: For GPT-3 integration
Command to install: pip install transformers



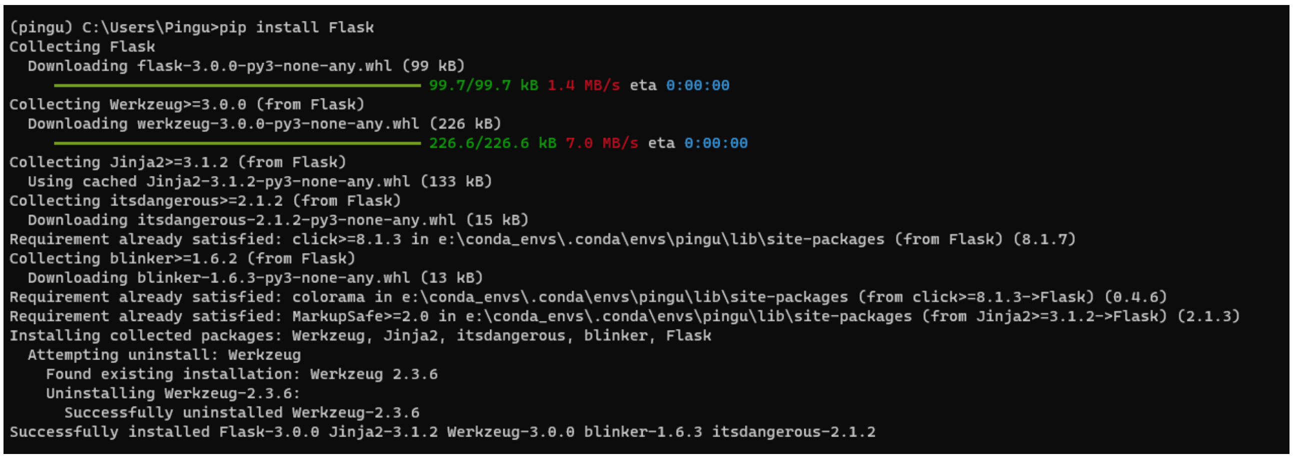
```
C:\Users\Pingu>conda activate pingu

(pingu) C:\Users\Pingu>pip install transformers
Collecting transformers
  Downloading transformers-4.34.0-py3-none-any.whl (7.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.7/7.7 MB 9.9 MB/s eta 0:00:00
Requirement already satisfied: filelock in e:\conda_envs\.conda\envs\pingu\lib\site-packages (from transformers) (3.12.2)
Collecting huggingface-hub<1.0,>=0.16.4 (from transformers)
  Downloading huggingface_hub-0.18.0-py3-none-any.whl (301 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 302.0/302.0 kB 9.4 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.17 in e:\conda_envs\.conda\envs\pingu\lib\site-packages (from transformers) (1.24.3)
Requirement already satisfied: packaging>=20.0 in e:\conda_envs\.conda\envs\pingu\lib\site-packages (from transformers) (23.1)
Collecting pyyaml>=5.1 (from transformers)
  Downloading PyYAML-6.0.1-cp311-cp311-win_amd64.whl (144 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 144.7/144.7 kB 8.4 MB/s eta 0:00:00
Requirement already satisfied: regex!=2019.12.17 in e:\conda_envs\.conda\envs\pingu\lib\site-packages (from transformers) (2023.10.3)
Requirement already satisfied: requests in e:\conda_envs\.conda\envs\pingu\lib\site-packages (from transformers) (2.31.0)
Collecting tokenizers<0.15,>=0.14 (from transformers)
  Downloading tokenizers-0.14.1-cp311-none-win_amd64.whl (2.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.2/2.2 MB 10.0 MB/s eta 0:00:00
Collecting safetensors>=0.3.1 (from transformers)
  Downloading safetensors-0.4.0-cp311-none-win_amd64.whl (277 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 277.4/277.4 kB 8.6 MB/s eta 0:00:00
Requirement already satisfied: tqdm>=4.27 in e:\conda_envs\.conda\envs\pingu\lib\site-packages (from transformers) (4.66.1)
Collecting fsspec>=2023.5.0 (from huggingface-hub<1.0,>=0.16.4->transformers)
  Downloading fsspec-2023.9.2-py3-none-any.whl (173 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 173.4/173.4 kB 948.6 kB/s eta 0:00:00
Requirement already satisfied: typing-extensions>=3.7.4.3 in e:\conda_envs\.conda\envs\pingu\lib\site-packages (from huggingface-hub<1.0,>=0.16.4->transformers) (4.5.0)
Collecting huggingface-hub<1.0,>=0.16.4 (from transformers)
  Downloading huggingface_hub-0.17.3-py3-none-any.whl (295 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 295.0/295.0 kB 6.1 MB/s eta 0:00:00
Requirement already satisfied: colorama in e:\conda_envs\.conda\envs\pingu\lib\site-packages (from tqdm>=4.27->transformers) (0.4.6)
Requirement already satisfied: charset-normalizer<4,>=2 in e:\conda_envs\.conda\envs\pingu\lib\site-packages (from requests->transformers) (3.1.0)
Requirement already satisfied: idna<4,>=2.5 in e:\conda_envs\.conda\envs\pingu\lib\site-packages (from requests->transformers) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in e:\conda_envs\.conda\envs\pingu\lib\site-packages (from requests->transformers) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in e:\conda_envs\.conda\envs\pingu\lib\site-packages (from requests->transformers) (2023.5.7)
Installing collected packages: safetensors, pyyaml, fsspec, huggingface-hub, tokenizers, transformers
Successfully installed fsspec-2023.9.2 huggingface-hub-0.17.3 pyyaml-6.0.1 safetensors-0.4.0 tokenizers-0.14.1 transformers-4.34.0
```

As per the phase 3 submission they want me to install transformers and flask and i installed.
In phase 3 they provided like integrate with GPT – 3 Using transformers but there is no GPT-3 in transformers only GPT – 2 function is present. So, I did not use the transformers libraries instead i used the Decision Tree in Sklearn library and made a pipe.

web application

Flask library is used to integrate with the web

Package name: Flask
Use: For web app development
Command to install: pip install Flask



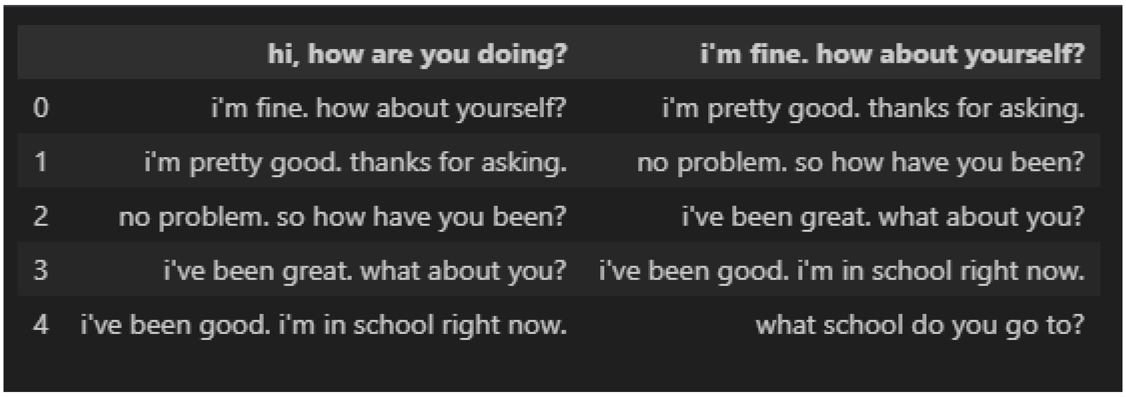
```
(pingu) C:\Users\Pingu>pip install Flask
Collecting Flask
  Downloading flask-3.0.0-py3-none-any.whl (99 kB)
     ---------------------------------------- 99.7/99.7 kB 1.4 MB/s eta 0:00:00
Collecting Werkzeug>=3.0.0 (from Flask)
  Downloading werkzeug-3.0.0-py3-none-any.whl (226 kB)
     ---------------------------------------- 226.6/226.6 kB 7.0 MB/s eta 0:00:00
Collecting Jinja2>=3.1.2 (from Flask)
  Using cached Jinja2-3.1.2-py3-none-any.whl (133 kB)
Collecting itsdangerous>=2.1.2 (from Flask)
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Requirement already satisfied: click>=8.1.3 in e:\conda_envs\.conda\envs\pingu\lib\site-packages (from Flask) (8.1.7)
Collecting blinker>=1.6.2 (from Flask)
  Downloading blinker-1.6.3-py3-none-any.whl (13 kB)
Requirement already satisfied: colorama in e:\conda_envs\.conda\envs\pingu\lib\site-packages (from click>=8.1.3->Flask) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in e:\conda_envs\.conda\envs\pingu\lib\site-packages (from Jinja2>=3.1.2->Flask) (2.1.3)
Installing collected packages: Werkzeug, Jinja2, itsdangerous, blinker, Flask
  Attempting uninstall: Werkzeug
    Found existing installation: Werkzeug 2.3.6
    Uninstalling Werkzeug-2.3.6:
      Successfully uninstalled Werkzeug-2.3.6
Successfully installed Flask-3.0.0 Jinja2-3.1.2 Werkzeug-3.0.0 blinker-1.6.3 itsdangerous-2.1.2
```

Preprocessing the dataset

I provided source code file called “AI_Phase3_source_code.ipynb” in my git hub repository

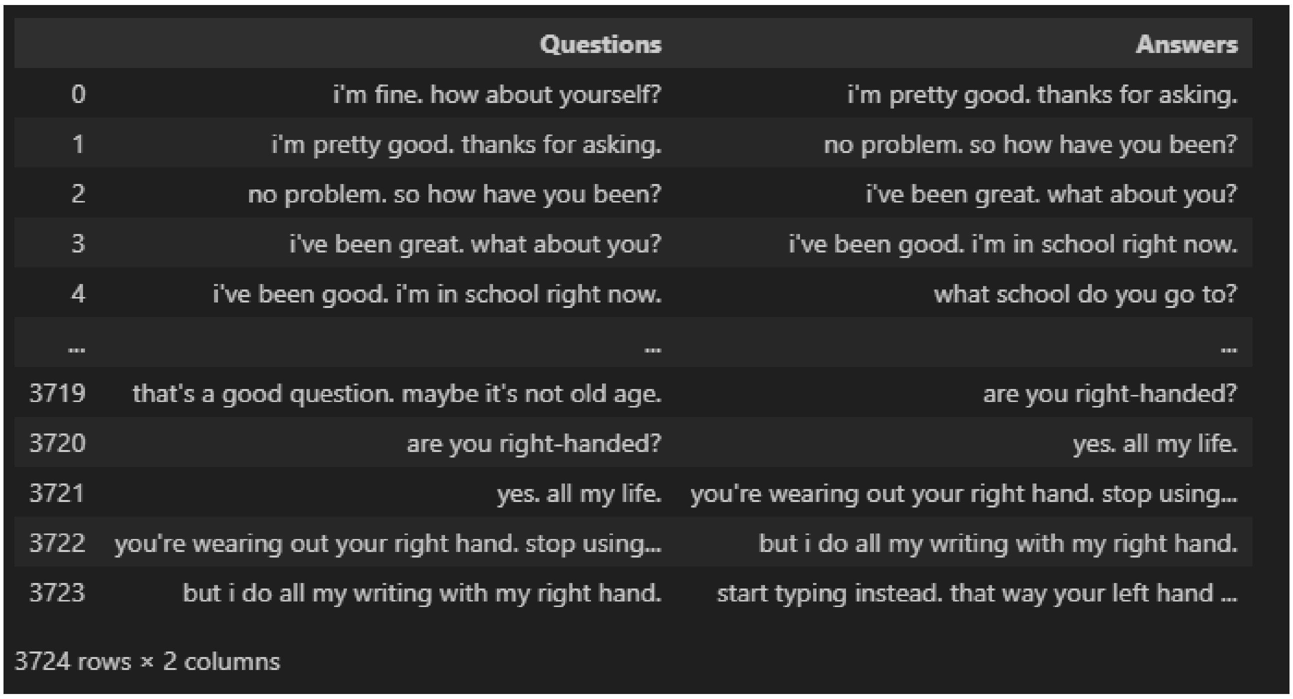
```
# import all required libraries

import string
from nltk.corpus import stopwords
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.tree import DecisionTreeClassifier
from sklearn.feature_extraction.text import TfidfTransformer,TfidfVectorizer
from sklearn.pipeline import Pipeline
```

```
# importing the dataset

df = pd.read_csv(r"C:\Users\Pingu\Desktop\IBM\AI_Phase3\dialogs.txt", sep='\t')


df.head()
```



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

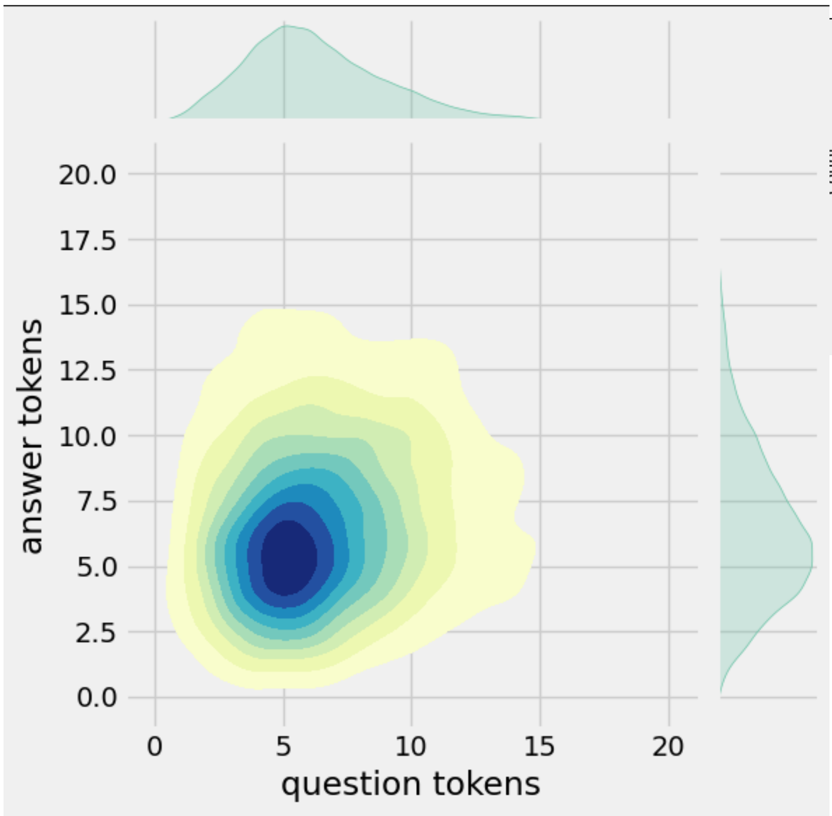
```
# add column names

df.columns=['Questions','Answers']
df
```

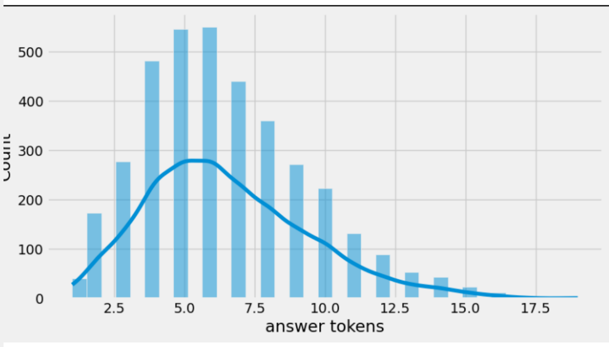


	Questions	Answers
0	i'm fine. how about yourself?	i'm pretty good. thanks for asking.
1	i'm pretty good. thanks for asking.	no problem. so how have you been?
2	no problem. so how have you been?	i've been great. what about you?
3	i've been great. what about you?	i've been good. i'm in school right now.
4	i've been good. i'm in school right now.	what school do you go to?
...	...	...
3719	that's a good question. maybe it's not old age.	are you right-handed?
3720	are you right-handed?	yes. all my life.
3721	yes. all my life.	you're wearing out your right hand. stop using...
3722	you're wearing out your right hand. stop using...	but i do all my writing with my right hand.
3723	but i do all my writing with my right hand.	start typing instead. that way your left hand ...

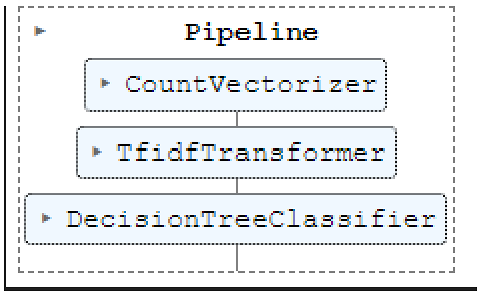
3724 rows × 2 columns

```
# Data Preprocessing

df['question tokens']=df['Questions'].apply(lambda x:len(x.split()))
df['answer tokens']=df['Answers'].apply(lambda x:len(x.split()))
plt.style.use('fivethirtyeight')
fig,ax=plt.subplots(nrows=1,ncols=2,figsize=(20,5))
sns.set_palette('Set2')
sns.histplot(x=df['question tokens'],data=df,kde=True,ax=ax[0])
sns.histplot(x=df['answer tokens'],data=df,kde=True,ax=ax[1])
sns.jointplot(x='question tokens',y='answer tokens',data=df,kind='kde',fill=True,cmap='YlGnBu')
plt.show()
```

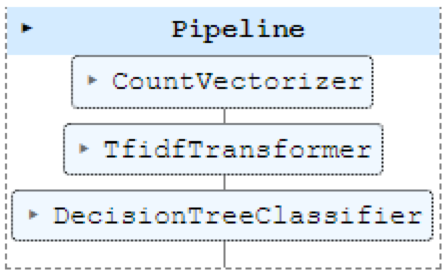




```
# Function for converting upper to lower case

def cleaner(x):
    return [a for a in (''.join([a for a in x if a not in string.punctuation])).lower().split()]


# Model

Pipe = Pipeline([
            ('bow',CountVectorizer(analyzer=cleaner)),
            ('tfidf',TfidfTransformer()),
            ('classifier',DecisionTreeClassifier())
])

Pipe.fit(df['Questions'],df['Answers'])
```

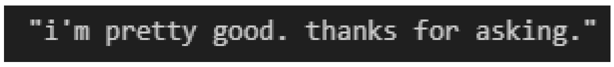


```
Pipe.fit(df['Questions'],df['Answers'])
```



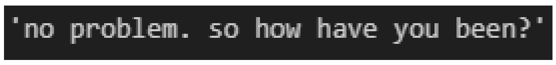
Testing

Pipe.predict(['im fine. how about yourself'])[0]



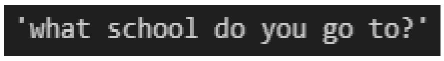
```
"i'm pretty good. thanks for asking."
```

Pipe.predict(['im pretty good. thanks for asking.'])[0]



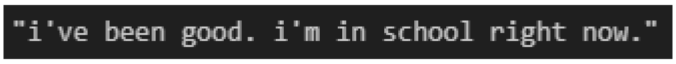
```
'no problem. so how have you been?'
```

Pipe.predict(['ive been good. im in school right now.'])[0]



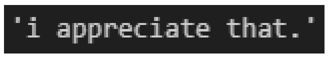
```
'what school do you go to?'
```

Pipe.predict(['ive been great. what about you?'])[0]



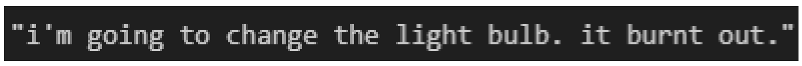
```
"i've been good. i'm in school right now."
```

Pipe.predict(['great'])[0]



```
'i appreciate that.'
```

Pipe.predict(['What are you doing'])[0]



```
"i'm going to change the light bulb. it burnt out."
```

Adding new data to the existing dataset

I have attached the dataset in my GitHub
GitHub link: https://github.com/dhanasekarenb/IBM_AI_Chatbot.git

Python Code:

```
from flask import Flask, request, render_template
import numpy as np
import string
from nltk.corpus import stopwords
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.tree import DecisionTreeClassifier
from sklearn.feature_extraction.text import TfidfTransformer,TfidfVectorizer
from sklearn.pipeline import Pipeline

app = Flask(__name__)

# importing the dataset
df = pd.read_csv(r"C:\Users\zerobroz\Desktop\AI_Phase4\data_set\dialogs.txt", sep='\t')

#add column names
df.columns=['Questions','Answers']

# Function for converting upper to lower case
def cleaner(x):
    return [''.join([a for a in x if a not in string.punctuation]).lower()]

# Load your pre-trained pipeline model
Pipe = Pipeline([
    ('bow', CountVectorizer(analyzer=cleaner)),
    ('tfidf', TfidfTransformer()),
    ('classifier', DecisionTreeClassifier())
])

# Load your pre-trained model weights
Pipe.fit(df['Questions'], df['Answers'])

# Define route for the home page
@app.route('/')
def index():
    return render_template('index.html')

# Define route for processing user input and generating chatbot response
@app.route('/get_response', methods=['POST'])
def get_response():
    user_input = request.form['user_input']
    response = Pipe.predict([user_input])[0]
    return render_template('index.html', user_input=user_input, response=response)

if __name__ == '__main__':
    app.run(debug=True)
```

HTML file:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Chatbot</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
    <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Play&display=swap" rel="stylesheet">
</head>

<body>
    <div class="container">
        <div class="word-cloud">
            <h2>Popular keywords</h2>
            <!-- Add your most used words content here -->
            <ul>
                <li>hi</li>
                <li>hey</li>
                <li>hello</li>
                <li>Tell me a joke</li>
                <li>who created you?</li>
                <li>what is the meaning of life?</li>
                <li>tell me a fun fact.</li>
                <li>another fun fact.</li>
                <li>what do you like to do in your free time?</li>
                <li>what languages do you speak?</li>
                <li>tell me a riddle.</li>
                <li>another riddle.</li>
            </ul>
        </div>
        <div class="chat-box">
            <h1>Chatbot</h1>
            <div class="chat-container">
                {% if user_input %}
                <div class="message user-message">
                    You: {{ user_input }}
                </div>
                {% endif %}
                {% if response %}
                <div class="message bot-message">
                    Bot: {{ response }}
                </div>
                {% endif %}
            </div>
            <form id="user-form" method="POST" action="/get_response">
                <input type="text" id="user_input" name="user_input" placeholder="Type your
message...">
                <button type="submit">Send</button>
            </form>
        </div>
    </div>
</body>

</html>
```

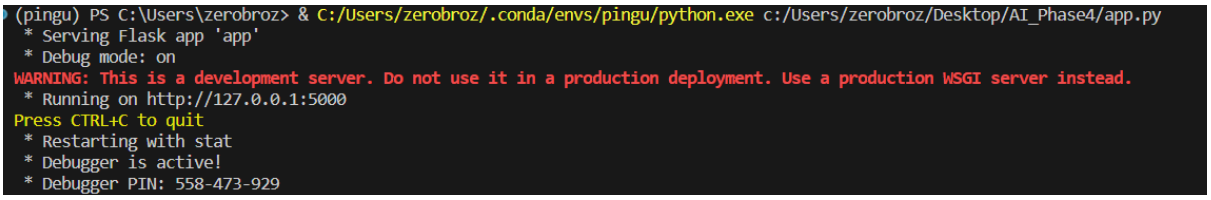
```
}
```

CSS File:

```
body {
   /* font-family: 'Times New Roman', Times, serif; */
   font-family: 'Play', sans-serif;
   margin: 0;
   padding: 0;
   background-color: #f2f2f2;
}

.container {
   display: grid;
   grid-template-columns: 1fr 2fr;
   grid-gap: 20px;
   max-width: 1200px;
   height: 700px;
   margin: 0 auto;
   padding: 20px;
}

.word-cloud {
   background-color: #fff;
   padding: 50px;
   box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

.chat-box {
   background-color: #fff;
   padding: 20px;
   box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

h1 {
   font-size: 24px;
   margin-bottom: 20px;
}

.chat-container {
   height: 500px;
   overflow-y: auto;
   padding: 10px;
   border: 1px solid #ccc;
   margin-bottom: 20px;
}

.message {
   margin-bottom: 10px;
   padding: 10px;
```

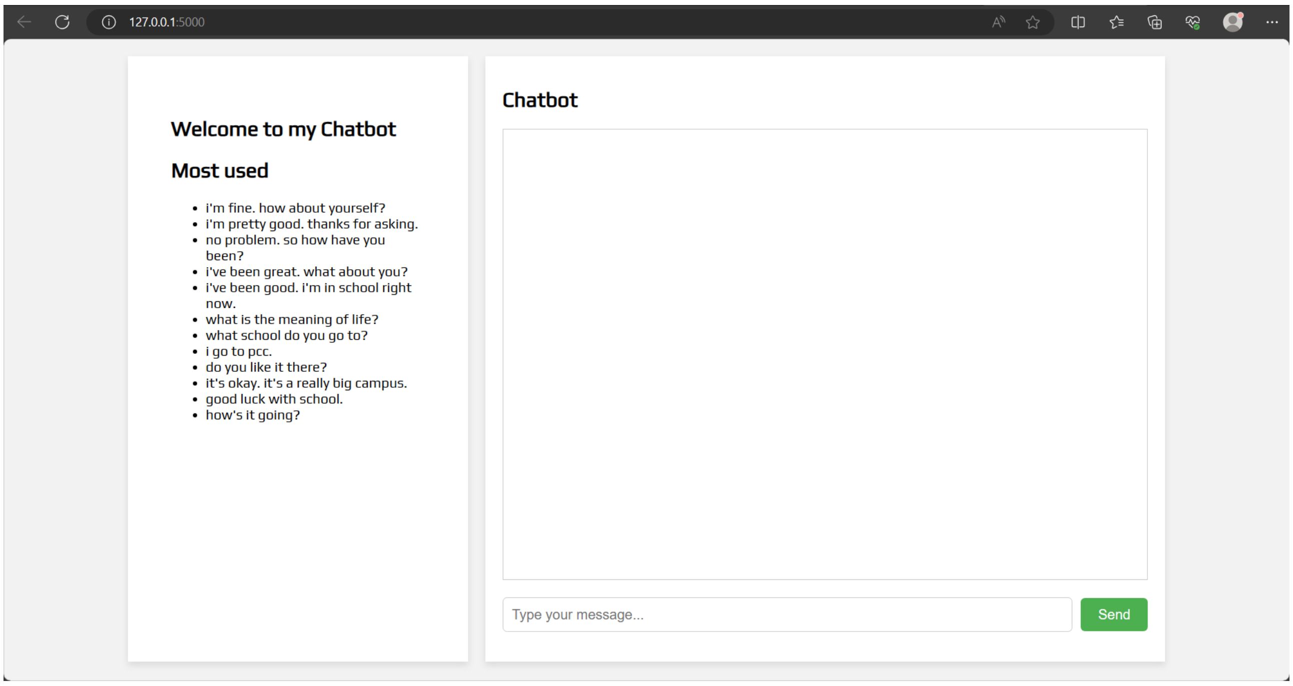
```
    border-radius: 5px;
}

.user-message {
    background-color: #4caf50;
    color: #fff;
    text-align: right;
}

.bot-message {
    background-color: #008CBA;
    color: #fff;
}

form {
    display: flex;
    align-items: center;

}

input[type="text"] {
    flex: 1;
    padding: 10px;
    font-size: 16px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

button {
    padding: 10px 20px;
    font-size: 16px;
    background-color: #4caf50;
    color: #fff;
    border: none;
    border-radius: 5px;
    margin-left: 10px;
    cursor: pointer;
}

button:hover {
    background-color: #45a049;
}
.user-message {
    color: solid black;
}

.bot-message {
    color: solid black;
}
```

OUTPUT:



```
(pingu) PS C:\Users\zerobroz> & C:/Users/zerobroz/.conda/envs/pingu/python.exe c:/Users/zerobroz/Desktop/AI_Phase4/app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 558-473-929
```

After pasting the http://127.0.0.1:5000 in browser.



127.0.0.1:5000

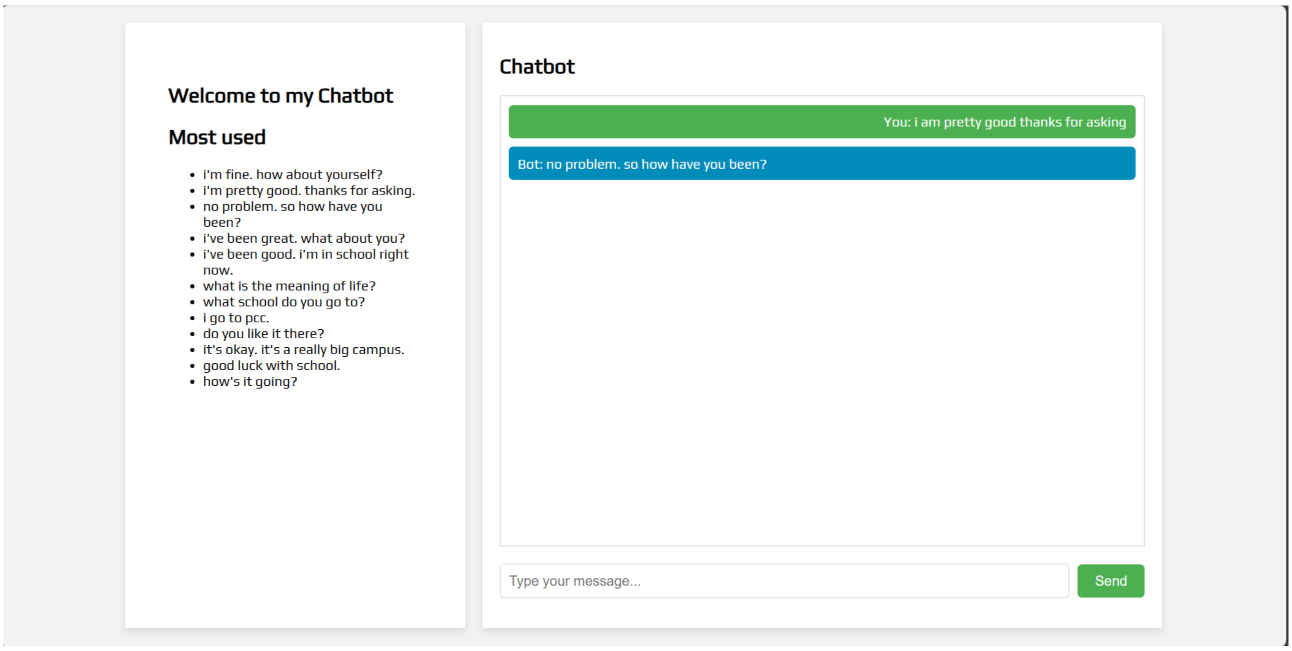
Welcome to my Chatbot

Most used

- i'm fine. how about yourself?
- i'm pretty good. thanks for asking.
- no problem. so how have you been?
- i've been great. what about you?
- i've been good. i'm in school right now.
- what is the meaning of life?
- what school do you go to?
- i go to pcc.
- do you like it there?
- it's okay. it's a really big campus.
- good luck with school.
- how's it going?

Chatbot

After give in the input.



Welcome to my Chatbot

Most used

- i'm fine. how about yourself?
- i'm pretty good. thanks for asking.
- no problem. so how have you been?
- i've been great. what about you?
- i've been good. i'm in school right now.
- what is the meaning of life?
- what school do you go to?
- i go to pcc.
- do you like it there?
- it's okay. it's a really big campus.
- good luck with school.
- how's it going?

Chatbot

You: i am pretty good thanks for asking

Bot: no problem. so how have you been?