

PHASE 3: DEVELOPMENT PART 1

Creation of chatbot with python :

To create a chatbot with Python , you need to install some packages. All the packages you need to install to create a chatbot with Machine Learning using the Python programming language are mentioned below:

tensorflow==2.3.1

nltk==3.5

colorama==0.4.3

numpy==1.18.5

scikit_learn==0.23.2

Flask==1.1.2

Defining the Intentions of a Chatbot:

Now we need to define a few simple intents and a group of messages that match those intents and also map some responses based on each intent category. I'll create a JSON file named "intents.json" including this data as follows:

```
{
  "intents": [
    {
      "tag": "greeting",
      "patterns": [
        "Hello!",
        "Hi there! How can I assist you today?",
        "Hey!", "Greetings! What can I help you with?",
        "Good day!"
      ],
      "responses": [
        "Hello! How can I assist you today?",
        "Hi there! I'm here to help. What do you need?",
        "Hey! What can I do for you today?",
        "Greetings! How may I be of assistance?",
        "Good day! How can I assist you?"
      ]
    },
    {
      "tag": "introduction",
```

```

    "patterns": [
        "Tell me about yourself.",
        "Who are you?",
        "Introduce yourself.",
        "What's your name?",
        "Are you a bot or a human?"
    ],
    "responses": [
        "I'm an AI chatbot here to assist you with your programming and learning needs. You can call me LearnProg.",
        "I'm LearnProg, an AI chatbot designed to provide information and recommendations for programming and learning.",
        "I'm LearnProg, a virtual assistant programmed to help you with programming resources and recommendations.",
        "I'm LearnProg, a bot created to assist you in your programming journey.",
        "I'm LearnProg, an AI here to help you with programming and learning resources."
    ]
},
{
    "tag": "thanks",
    "patterns": [
        "Thank you!",
        "Thanks!",
        "Appreciate your help.",
        "Grateful for your assistance.",
        "I'm thankful."
    ],
    "responses": [
        "You're welcome! If you have more questions, feel free to ask.",
        "You're welcome! If you need further assistance, don't hesitate to reach out.",
        "You're welcome! I'm here to help whenever you need.",
        "You're welcome! I'm glad I could assist you.",
        "You're welcome! If you have more inquiries, just let me know."
    ]
},
{
    "tag": "farewell",
    "patterns": [
        "Goodbye!",
        "Farewell!",
        "See you later!",
        "Take care!",
        "Bye for now."
    ],
    "responses": [
        "Goodbye! If you have more questions in the future, don't hesitate to return.",
        "Farewell! Feel free to return if you need assistance later.",
        "See you later! Take care and happy learning!",
        "Take care! If you need help, come back anytime."
    ]
}

```

```

        "Bye for now! If you have more inquiries, just return
to our chat."
    ],
    },
    {
        "tag": "about_chatbot",
        "patterns": [
            "How do you work?",
            "What can you do?",
            "Tell me more about your capabilities.",
            "Explain your functions.",
            "How can you assist me?"
        ],
        "responses": [
            "I work by analyzing your questions and providing
relevant information, recommendations, and assistance related to programming
and learning.",
            "I can assist you with programming language
recommendations, learning resources, and answers to your queries.",
            "I'm here to help you by providing information on
programming languages, learning materials, and more.",
            "My capabilities include offering programming book
recommendations, suggesting online courses, and answering your programming-
related questions.",
            "I'm designed to assist you with programming and
learning-related tasks, from book recommendations to language advice."
        ]
    },
    {
        "tag": "random_conversation",
        "patterns": [
            "How's the weather today?",
            "Tell me a joke.",
            "What's your favorite color?",
            "Do you like movies?",
            "What's your favorite programming language?"
        ],
        "responses": [
            "I'm just a chatbot, so I don't know about the
weather. But I can help with programming questions!",
            "Here's a joke: Why don't programmers like nature?
It has too many bugs!",
            "I don't have a favorite color, but I'm here to
assist you with programming and learning.",
            "I can't watch movies, but I can recommend
programming documentaries!",
            "I don't have preferences, but I can certainly help
you with programming languages."
        ]
    },
    {
        "tag": "chatbot_name",
        "patterns": [
            "What's your name?",
            "Tell me your name.",
            "Who are you, LearnProg?"
        ],
    },

```

```

    "responses": [
        "You got it right! I'm LearnProg, your programming and
learning assistant.",
        "My name is LearnProg, and I'm here to help you with
programming and learning resources.",
        "Yes, I'm LearnProg. How can I assist you today?"
    ]
},
{
    "tag": "out_of_context",
    "patterns": [
        "What's the weather like today?",
        "Tell me a joke.",
        "Do you like pizza?"
    ],
    "responses": [
        "I apologize, but that's a bit out of my area of
expertise. If you have any programming or learning-related questions, feel
free to ask, and I'll be happy to help!",
        "It's great to chat, but my expertise lies in
programming and learning. If you have questions in those areas, please ask
away."
    ]
},
{
    "tag": "out_of_context",
    "patterns": [
        "Order a pizza.",
        "What's the time?",
        "Sing me a song."
    ],
    "responses": [
        "I appreciate your request, but I'm here to assist you
with programming and learning. If you have any questions related to those
topics, please feel free to ask, and I'll be glad to help."
    ]
},
{
    "tag": "python_learning",
    "patterns": [
        "I want to learn Python.",
        "I'm completely new to programming."
    ],
    "responses": [
        "Great choice! Python is a versatile language. Are you
new to programming, or do you have experience in other languages?",
        "Perfect! I can recommend some beginner-friendly Python
resources. Do you prefer books or online tutorials?"
    ]
},
{
    "tag": "python_book_recommendation",
    "patterns": [
        "I'd like a book recommendation, please."
    ],
    "responses": [
        "Sure, 'Automate the Boring Stuff with Python' by Al
Sweigart is an excellent choice for beginners. It's available online for free

```

```

at [Automate the Boring Stuff with
Python] (https://automatetheboringstuff.com/)."
```

],

```

    {
      "tag": "web_development_resources",
      "patterns": [
        "Can you suggest resources for learning web
development?"
      ],
      "responses": [
        "Certainly! Web development is a popular choice.
There are many online courses and tutorials available. What specific aspect
of web development interests you?"
      ]
    },
    {
      "tag": "front_end_development",
      "patterns": [
        "I'm interested in front-end development."
      ],
      "responses": [
        "That's a good starting point! You can explore HTML,
CSS, and JavaScript. I recommend freeCodeCamp's front-end development
curriculum at [freeCodeCamp] (https://www.freecodecamp.org/)."
```

],

```

    },
    {
      "tag": "javascript_resources",
      "patterns": [
        "What are the best resources for mastering
JavaScript?"
      ],
      "responses": [
        "JavaScript is a versatile language. You can check
out 'Eloquent JavaScript' by Marijn Haverbeke for an in-depth understanding
of JavaScript at [Eloquent JavaScript] (https://eloquentjavascript.net/)."
```

],

```

    },
    {
      "tag": "tech_blogs_recommendation",
      "patterns": [
        "Can you recommend any blogs for staying updated
in the tech industry?"
      ],
      "responses": [
        "Certainly! 'TechCrunch,' 'The Verge,' and
'Wired' are popular tech news sources. Is there a specific tech niche you're
interested in?"
      ]
    },
    {
      "tag": "ai_ml_resources",
      "patterns": [
        "I'm into artificial intelligence and machine
learning."
      ],
    },
  ],
}

```

```

    "responses": [
        "Great choice! 'Towards Data Science' on Medium and
'AI Trends' are excellent sources for AI and machine learning updates at
[Towards Data Science] (https://towardsdatascience.com/) and [AI
Trends] (https://www.aitrends.com/)."
```

],

```

    {
        "tag": "programming_motivation",
        "patterns": [
            "How can I stay motivated while learning
programming?"
        ],
        "responses": [
            "Staying motivated is important. Setting goals,
tracking progress, and joining coding communities can help. Would you like
more tips?"
        ],
        {
            "tag": "programming_motivation_tips",
            "patterns": [
                "Yes, please!"
            ],
            "responses": [
                "Consider participating in coding challenges and
projects. Also, break your learning into small, manageable tasks."
            ],
            {
                "tag": "data_science_books",
                "patterns": [
                    "I'm interested in data science. Any book
recommendations?",
                    "Can you suggest data science books?"
                ],
                "responses": [
                    "Absolutely! Consider reading 'Python for Data
Analysis' by Wes McKinney. You can find it
[here] (https://www.oreilly.com/library/view/python-for-data/9781491957653/).
It's a great resource for aspiring data scientists."
                ],
                {
                    "tag": "coding_skills_improvement",
                    "patterns": [
                        "How can I improve my coding skills?",
                        "Any tips for becoming a better coder?"
                    ],
                    "responses": [
                        "Improving coding skills requires practice. Try sites
like LeetCode, HackerRank, and CodeSignal for coding challenges and
competitions. They're excellent for skill enhancement."
                    ],
                },
            },
        },
    ],

```

```

{
  "tag": "coding_competitions",
  "patterns": [
    "Can you suggest online platforms for coding competitions?",
    "Where can I participate in coding challenges?"
  ],
  "responses": [
    "Certainly! You can explore Codeforces, AtCoder, and TopCoder. They host a variety of coding competitions and challenges to hone your skills."
  ]
},

{
  "tag": "data_science_resources",
  "patterns": [
    "Do you have any recommendations for learning data science?"
  ],
  "responses": [
    "Data science is a rewarding field. I recommend 'Python for Data Analysis' by Wes McKinney for a strong foundation in Python and data analysis at [Python for Data Analysis] (https://www.oreilly.com/library/view/python-for-data/9781491957653/)."

```

```

        "Certainly! 'Scratch' is a great platform for kids to
start learning programming in a fun and visual way at
[Scratch] (https://scratch.mit.edu/)."
```

],

```

    {
        "tag": "coding_interview_prep",
        "patterns": [
            "Are there any resources to prepare for coding
interviews?"
        ],
        "responses": [
            "Preparing for coding interviews is crucial.
'Cracking the Coding Interview' by Gayle Laakmann McDowell is a popular
choice among candidates."
        ]
    },
    {
        "tag": "learn_devops",
        "patterns": [
            "I want to learn more about DevOps. What resources
should I explore?"
        ],
        "responses": [
            "DevOps is an important field. 'The DevOps Handbook' by
Gene Kim, Jez Humble, and Patrick Debois is an excellent resource to start
with."
        ]
    },
    {
        "tag": "coding_competitions",
        "patterns": [
            "Can you suggest online platforms for coding
competitions?"
        ],
        "responses": [
            "Certainly! Platforms like Codeforces, LeetCode, and
HackerRank host coding competitions and challenges. They're great for honing
your coding skills."
        ]
    },
    {
        "tag": "game_development_resources",
        "patterns": [
            "I'm interested in game development. Any
recommendations for game development books?"
        ],
        "responses": [
            "Game development is exciting! 'Unity in Action' by Joe
Hocking is a great book for learning game development using the Unity
engine."
        ]
    },
    {
        "tag": "learn_sql",
        "patterns": [

```



```

        "I need to learn SQL for my job. Any good resources for
SQL beginners?"
    ],
    "responses": [
        "Learning SQL is essential for database management.
'Learn SQL in One Day and Learn It Well' by Jamie Chan is an excellent choice
for beginners at [Learn SQL in One Day and Learn It
Well] (https://www.goodreads.com/book/show/31947289-learn-sql-in-one-day-and-learn-it-well)."https://www.goodreads.com/book/show/18197267-don-t-make-me-think-revisited)."

```

Data preparation:

The second step of this task to create a chatbot with Python and Machine Learning is to prepare the data to train our chatbot. I'll start this step by importing the necessary libraries and packages:

```
import json
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Embedding, GlobalAveragePooling1D
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from sklearn.preprocessing import LabelEncoder
```

Now I will read the JSON file and process the required files:

```
with open('intents.json') as file:
    data = json.load(file)
```

```
training_sentences = []
training_labels = []
labels = []
responses = []
```

```
for intent in data['intents']:
    for pattern in intent['patterns']:
        training_sentences.append(pattern)
        training_labels.append(intent['tag'])
    responses.append(intent['responses'])
```

```
if intent['tag'] not in labels:
    labels.append(intent['tag'])
```

```
num_classes = len(labels)
```

Now we need to use the label encoder method provided by the Scikit-Learn library in Python:

```
lbl_encoder = LabelEncoder()
lbl_encoder.fit(training_labels)
training_labels = lbl_encoder.transform(training_labels)
```

Tokenization:

Now we need to vectorize the data using the Tokenization method to create a chatbot with Python

```
vocab_size = 1000
embedding_dim = 16
max_len = 20
oov_token = "<OOV>"

tokenizer = Tokenizer(num_words=vocab_size, oov_token=oov_token)
tokenizer.fit_on_texts(training_sentences)
word_index = tokenizer.word_index
sequences = tokenizer.texts_to_sequences(training_sentences)
padded_sequences = pad_sequences(sequences, truncating='post', maxlen=max_len)
```

Training a Neural Network

Now the next and most important step in the process of building a chatbot with Python and Machine Learning is to train a neural network. Now, I will train and create a neural network to train our chatbot:

```
model = Sequential()
model.add(Embedding(vocab_size, embedding_dim, input_length=max_len))
model.add(GlobalAveragePooling1D())
model.add(Dense(16, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

model.summary()
epochs = 500
history = model.fit(padded_sequences, np.array(training_labels), epochs=epochs)
```

Saving The Neural Network:

We've trained the model, but before we go any further in the process of building a chatbot with Python and Machine Learning, let's save the model so that we can use this neural network in the future as well:

```
# to save the trained model
model.save("chat_model")

import pickle
```

```
# to save the fitted tokenizer
with open('tokenizer.pickle', 'wb') as handle:
    pickle.dump(tokenizer, handle, protocol=pickle.HIGHEST_PROTOCOL)

# to save the fitted label encoder
with open('label_encoder.pickle', 'wb') as ecn_file:
    pickle.dump(lbl_encoder, ecn_file, protocol=pickle.HIGHEST_PROTOCOL)
```

Now let's Build a Chatbot with Python and our Trained Model:

Now I am going to implement a chat function to interact with a real user. When the message from the user will be received, the chatbot will compute the similarity between the sequence of the new text and the training data.

Taking into account the trust scores obtained for each category, it categorizes the user's message according to an intention with the highest trust score:

```
import json
import numpy as np
from tensorflow import keras
from sklearn.preprocessing import LabelEncoder
```

```
import colorama
colorama.init()
from colorama import Fore, Style, Back
```

```
import random
import pickle
```

```
with open("intents.json") as file:
    data = json.load(file)
```

```
def chat():
    # load trained model
    model = keras.models.load_model('chat_model')

    # load tokenizer object
    with open('tokenizer.pickle', 'rb') as handle:
        tokenizer = pickle.load(handle)

    # load label encoder object
    with open('label_encoder.pickle', 'rb') as enc:
        lbl_encoder = pickle.load(enc)

    # parameters
```

```

max_len = 20

while True:
    print(Fore.LIGHTBLUE_EX + "User: " + Style.RESET_ALL, end="")
    inp = input()
    if inp.lower() == "quit":
        break

    result =
model.predict(keras.preprocessing.sequence.pad_sequences(tokenizer.texts_to_sequences([inp]),
                                                         truncating='post', maxlen=max_len))
    tag = lbl_encoder.inverse_transform([np.argmax(result)])

    for i in data['intents']:
        if i['tag'] == tag:
            print(Fore.GREEN + "ChatBot:" + Style.RESET_ALL ,
np.random.choice(i['responses']))

    # print(Fore.GREEN + "ChatBot:" + Style.RESET_ALL,random.choice(responses))

print(Fore.YELLOW + "Start messaging with the bot (type quit to stop)!" + Style.RESET_ALL)
chat()

```

OUTPUT :

```

(base) C:\Users\DHARSHINI S>pip install flask
Requirement already satisfied: flask in c:\users\धारशनी s\anaconda3\lib\site-packages (2.2.2)
Requirement already satisfied: Werkzeug>=2.2.2 in c:\users\धारशनी s\anaconda3\lib\site-packages (from flask) (2.2.3)
Requirement already satisfied: Jinja2>=3.0 in c:\users\धारशनी s\anaconda3\lib\site-packages (from flask) (3.1.2)
Requirement already satisfied: itsdangerous>=2.0 in c:\users\धारशनी s\anaconda3\lib\site-packages (from flask) (2.0.1)
Requirement already satisfied: click>=8.0 in c:\users\धारशनी s\anaconda3\lib\site-packages (from flask) (8.0.4)
Requirement already satisfied: colorama in c:\users\धारशनी s\anaconda3\lib\site-packages (from click>=8.0->flask) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\धारशनी s\anaconda3\lib\site-packages (from Jinja2>=3.0->flask) (2.1.1)

(base) C:\Users\DHARSHINI S>pip install tensorflow
Collecting tensorflow
  Obtaining dependency information for tensorflow from https://files.pythonhosted.org/packages/80/6f/57d36f6507e432d7fc1956b2e9e8530c5c2d2bfcdb8821bcbfae271cd6688/tensorflow-2.14.0-cp311-cp311-win_amd64.whl.metadata
  Downloading tensorflow-2.14.0-cp311-cp311-win_amd64.whl.metadata (3.3 kB)
Collecting tensorflow-intel==2.14.0 (from tensorflow)
  Obtaining dependency information for tensorflow-intel==2.14.0 from https://files.pythonhosted.org/packages/ad/6e/1bfe367855dd87467564f7bf9fa14f3b17889988e79598bc37bf18f5ffb6/tensorflow_intel-2.14.0-cp311-cp311-win_amd64.whl.metadata
  Downloading tensorflow_intel-2.14.0-cp311-cp311-win_amd64.whl.metadata (4.8 kB)
Collecting absl-py==1.0.0 (from tensorflow-intel==2.14.0->tensorflow)
  Obtaining dependency information for absl-py==1.0.0 from https://files.pythonhosted.org/packages/01/e4/dc0a1dcc4e74e08d7abedab278c795eef54a224363bb18f5692f416d834f/absl_py-2.0.0-py3-none-any.whl.metadata
  Downloading absl_py-2.0.0-py3-none-any.whl.metadata (2.3 kB)
Collecting astunparse==1.6.0 (from tensorflow-intel==2.14.0->tensorflow)
  Downloading astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Collecting flatbuffers>=23.5.26 (from tensorflow-intel==2.14.0->tensorflow)
  Obtaining dependency information for flatbuffers>=23.5.26 from https://files.pythonhosted.org/packages/6f/12/d5c79ee25

```

```

Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in c:\users\dhharshini s\anaconda3\lib\site-packages ^
(from tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (0.7.1)
Requirement already satisfied: werkzeug>=1.0.1 in c:\users\dhharshini s\anaconda3\lib\site-packages (from tensorboard<2.1
5,>=2.14->tensorflow-intel==2.14.0->tensorflow) (2.2.3)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in c:\users\dhharshini s\anaconda3\lib\site-packages (from google-a
uth<3,>=1.6.3->tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (5.3.1)
Requirement already satisfied: pyasn1-modules>=0.2.1 in c:\users\dhharshini s\anaconda3\lib\site-packages (from google-au
th<3,>=1.6.3->tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (0.2.8)
Requirement already satisfied: rsa<5,>=3.1.4 in c:\users\dhharshini s\anaconda3\lib\site-packages (from google-auth<3,>=1
.6.3->tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in c:\users\dhharshini s\anaconda3\lib\site-packages (from google
-auth-oauthlib<1.1,>=0.5->tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (1.3.1)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\dhharshini s\anaconda3\lib\site-packages (from reques
ts<3,>=2.21.0->tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dhharshini s\anaconda3\lib\site-packages (from requests<3,>=2.21.
0->tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\dhharshini s\anaconda3\lib\site-packages (from requests<3,>
=2.21.0->tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dhharshini s\anaconda3\lib\site-packages (from requests<3,>
=2.21.0->tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (2023.7.22)
Requirement already satisfied: MarkupSafe>=2.1.1 in c:\users\dhharshini s\anaconda3\lib\site-packages (from werkzeug>=1.0
.1->tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (2.1.1)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in c:\users\dhharshini s\anaconda3\lib\site-packages (from pyasn1-mod
ules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in c:\users\dhharshini s\anaconda3\lib\site-packages (from requests-oauthl
ib>=0.7.0->google-auth-oauthlib<1.1,>=0.5->tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (3.2.2)

(base) C:\Users\DHARSHINI S>pip install keras
Requirement already satisfied: keras in c:\users\dhharshini s\anaconda3\lib\site-packages (2.14.0)

```

```

(base) C:\Users\DHARSHINI S>pip install nltk
Requirement already satisfied: nltk in c:\users\dhharshini s\anaconda3\lib\site-packages (3.8.1)
Requirement already satisfied: click in c:\users\dhharshini s\anaconda3\lib\site-packages (from nltk) (8.0.4)
Requirement already satisfied: joblib in c:\users\dhharshini s\anaconda3\lib\site-packages (from nltk) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\dhharshini s\anaconda3\lib\site-packages (from nltk) (2022.7.9
)
Requirement already satisfied: tqdm in c:\users\dhharshini s\anaconda3\lib\site-packages (from nltk) (4.65.0)
Requirement already satisfied: colorama in c:\users\dhharshini s\anaconda3\lib\site-packages (from click->nltk) (0.4.6)

```

Conclusion

The development phase was the heartbeat of the project, where concepts and designs were brought to life through code. We successfully set up the development environment, organized the codebase, and implemented crucial features such as user interaction, natural language processing, response generation, and database management. Our codebase is now a functional representation of the envisioned chatbot, ready for testing and refinement.