**AI-CHATBOX PSEUDO CODE**

import random

import json

import pickle

import numpy as np

import nltk

from nltk.stem import WordNetLemmatizer

from keras.models import load\_model

# Import necessary libraries and modules

lemmatizer = WordNetLemmatizer()

intents = json.loads(open('C:\Simplilearn\Python\Python projects\chatbot using python\chatbot\intents.json').read())

words = pickle.load(open('words.pkl', 'rb'))

classes = pickle.load(open('classes.pkl', 'rb'))

model = load\_model('chatbot\_model.h5')

# Load necessary data and model

def clean\_up\_sentence(sentence):

sentence\_words = nltk.word\_tokenize(sentence)

sentence\_words = [lemmatizer.lemmatize(word) for word in sentence\_words]

return sentence\_words

# Function to clean up the sentence by tokenizing and lemmatizing the words

def bag\_of\_words(sentence):

sentence\_words = clean\_up\_sentence(sentence)

bag = [0] \* len(words)

for w in sentence\_words:

for i, word in enumerate(words):

if word == w:

bag[i] = 1

return np.array(bag)

# Function to create a bag of words representation of the sentence

def predict\_class(sentence):

bow = bag\_of\_words(sentence)

res = model.predict(np.array([bow]))[0]

ERROR\_THRESHOLD = 0.25

results = [[i, r] for i, r in enumerate(res) if r > ERROR\_THRESHOLD]

results.sort(key=lambda x: x[1], reverse=True)

return\_list = []

for r in results:

return\_list.append({'intent': classes[r[0]], 'probability': str(r[1])})

return return\_list

# Function to predict the class of the sentence based on the trained model

def get\_response(intents\_list, intents\_json):

tag = intents\_list[0]['intent']

list\_of\_intents = intents\_json['intents']

for i in list\_of\_intents:

if i['tag'] == tag:

result = random.choice(i['responses'])

break

return result

# Function to get a random response based on the predicted intent

print("GO! Bot is running!")

while True:

message = input("")

ints = predict\_class(message)

res = get\_response(ints, intents)

print(res)

# Main loop to continuously take user input, predict the intent, and provide a response

**DESCRIPTION:**

The Objective of the above pseudo code is to take user input, predict the intend, and provide a response.