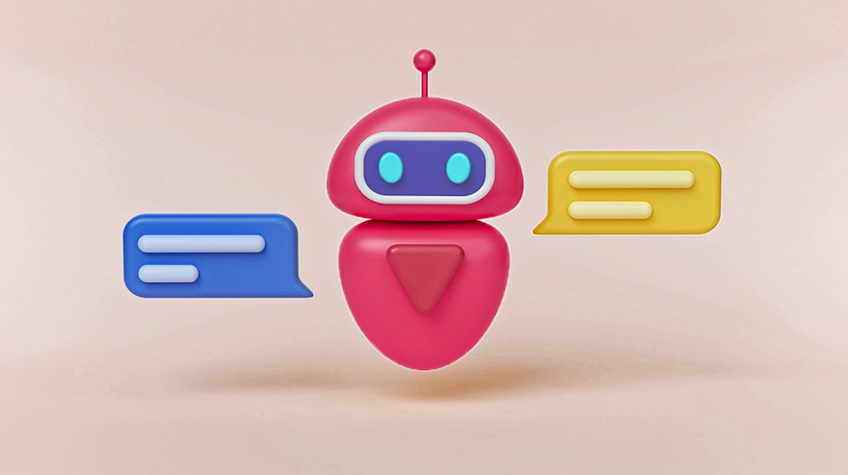
**Develop Part- 2**

|  |  |
| --- | --- |
| Domain AI | **PYTHON - CHATBOT** |

|  |
| --- |
|  |



**Group Members:**

Naveen kumar.T,

Kavitha.R,

Balaji.R,

Kaviyarasu.M.

**Faculty Mentor Mail ID:** [**ezhilmathia@jkkn.ac.in**](mailto:ezhilmathia@jkkn.ac.in)

**Industry Evaluator:**

[**IndustryEvaluator@skillup.online**](mailto:IndustryEvaluator@skillup.online)

**AI-CHATBOT**

import random

import json

import pickle

import numpy as np

import tensorflow as tf

import nltk

from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

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

words = []

classes = []

documents = []

ignoreLetters = ['?', '!', '.', ',']

for intent in intents['intents']:

    for pattern in intent['patterns']:

        wordList = nltk.word\_tokenize(pattern)

        words.extend(wordList)

        documents.append((wordList, intent['tag']))

        if intent['tag'] not in classes:

            classes.append(intent['tag'])

words = [lemmatizer.lemmatize(word) for word in words if word not in ignoreLetters]

words = sorted(set(words))

classes = sorted(set(classes))

pickle.dump(words, open('words.pkl', 'wb'))

pickle.dump(classes, open('classes.pkl', 'wb'))

training = []

outputEmpty = [0] \* len(classes)

for document in documents:

    bag = []

    wordPatterns = document[0]

    wordPatterns = [lemmatizer.lemmatize(word.lower()) for word in wordPatterns]

    for word in words:

        bag.append(1) if word in wordPatterns else bag.append(0)

    outputRow = list(outputEmpty)

    outputRow[classes.index(document[1])] = 1

    training.append(bag + outputRow)

random.shuffle(training)

training = np.array(training)

trainX = training[:, :len(words)]

trainY = training[:, len(words):]

model = tf.keras.Sequential()

model.add(tf.keras.layers.Dense(128, input\_shape=(len(trainX[0]),), activation = 'relu'))

model.add(tf.keras.layers.Dropout(0.5))

model.add(tf.keras.layers.Dense(64, activation = 'relu'))

model.add(tf.keras.layers.Dropout(0.5))

model.add(tf.keras.layers.Dense(len(trainY[0]), activation='softmax'))

sgd = tf.keras.optimizers.SGD(learning\_rate=0.01, momentum=0.9, nesterov=True)

model.compile(loss='categorical\_crossentropy', optimizer=sgd, metrics=['accuracy'])

hist = model.fit(np.array(trainX), np.array(trainY), epochs=200, batch\_size=5, verbose=1)

model.save('chatbot\_model.h5', hist)

print('Done')