**ANNA UNIVERSITY REGIONAL CAMPUS, COIMBATORE - 641 041**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING ACADEMIC YEAR 2023-2024 (SEMESTER 5)**

**IBM NAAN MUDHALVAN ARTIFICIAL INTELLIGENCE PROJECT 3: CREATE A CHATBOT IN PYTHON**

**PHASE 3: DEVELOPMENT PART 1**

**PRESENTED BY**

SHRIMAN K

710021106301

# ABSTRACT:

The purpose of this project is to **design** and **implement** a **chatbot using Python** for which utilize **machine learning algorithms** to **analyze medical data** and **predict** the likelihood of an **individual developing diabetes.**

The **chatbot aims** and the **main goal** is to **provide/assist early risk assessment** and **personalized preventive measures, enabling individuals** to take **proactive actions** to manage **their health effectively**. By using **Artificial Intelligence (AI)** and **Natural Language Processing (NLP).** This project will enhance the **overall health service**.

# The chatbot will be developed using Python programming language, Natural Language Processing (NLP) libraries , and possibly TensorFlow for advanced chat bot and also integrated with existing customer service platforms to ensure seamless customer interactions.

**INTRODUCTION:**

We can provide you with a basic code structure for building a **chatbot for diabetes patients** using **Python** in **Visual Studio**. However, note that we are creating a complete chatbot system with all the necessary functionalities is beyond the scope of a single code snippet.

We might need to integrate various **APIs, databases,** and **natural language processing libraries** to enhance its **capabilities.** Here's we simplified example to get started.

**CHATBOT OBJECTIVES:**

Using natural language processing, It can provide and understand your queries and respond accordingly. Whether you need information, help with a task, or just want to have a conversation like I'm here to assist you.

It can provide general knowledge, answer factual questions, offer recommendations, and even engage in casual discussions. Please keep in mind that while It strive to provide accurate and helpful responses, It might occasionally make errors or lack the most up-to-date information.

If in doubt, it's always a good idea to verify any critical or time-sensitive information from reliable sources. It allows businesses to connect with customers in a personal way without the expense of human representatives.

**CHATBOT IN MEDICAL FIELD:**

Chatbots for healthcare allow patients to communicate with specialists using traditional methods, including phone calls, video calls, messages, and emails. By doing this, engagement is increased, and medical personnel have more time and opportunity to concentrate on patients who need it more.

Healthcare chatbots can use information about the patient's condition, allergies, and insurance information to schedule appointments faster and better. This includes: Finding a slot at a specialized health facility or lab test center.

Chatbots — computer programs that simulate conversations with humans — are being employed to monitor the health of pregnant women as they approach delivery dates and of orthopedic surgery patients after discharge, and to answer messages that come in through online patient portals about everything from appointments too.

# PROCEDURE TO RUN THE CODE IN Google colab:

**Step 1:**

Login with google account for google colab

# Step 2:

# Login and open in google colab in the PC

# Step 3:

Datasheet downloaded in the Kaggle application and get the dialog.txt file.

# Step 4:

Datasheet uploaded the google drive and read the file in the colab.

# Step 5:

# Save the file and execute the code in the google colab.

# PROGRAM:

# from google.colab import drive

drive.mount('/content/drive')

import tensorflow

import pandas as pd

import sklearn

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.svm import SVC

data = pd.read\_csv('/content/drive/MyDrive/colabdata/dialogs.txt',sep='\t')

data.head()

corpus=data['Question']

vectorizer = CountVectorizer()

svmsvc = SVC(kernel='linear')

X = vectorizer.fit\_transform(corpus)

vectorizer = CountVectorizer(binary=True)

train\_annot = vectorizer.fit\_transform(data['Question'])

#print(vectorizer.get\_feature\_names\_out())

#print(vectors.toarray())

from sklearn import svm

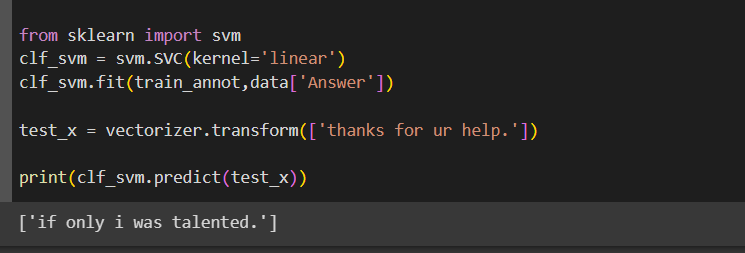
clf\_svm = svm.SVC(kernel='linear')

clf\_svm.fit(train\_annot,data['Answer'])

test\_x = vectorizer.transform(['how are you.'])

**print(clf\_svm.predict(test\_x)**

**OUTPUT IN COLAB:**

****

