**MINI PROJECT**

**MIDTERM REPORT**

**(2019-2020)**

**AI ChatBot for Health Care**

INSTITUTE OF ENGINEERING AND APPLICATIONS



**Submitted To: Submitted By:**

Respected Mr.Amir Khan Nitya Nagpal(171500210)

(Technical Trainer) Kirti Verma(171500162)

AyushiSrivastava(171500080)

## Project Aim:

To make a computer program designed to simulate an intelligent conversation with one or more human users.

## About the Project:

In this project, we will showcase how the user can interact with a chatbot to get a response to their queries. Chatbots are computer programs with the designed purpose to have conversations with human users over the world wide web and internet. Many tech programs use conversational agents in their services to help streamline communication with their target audience. Where human agents would have limits on how many customers they can service, chatbots can handle many more customers at once.

This is a basic project that is developed in Python. In this project, we use some Python Libraries. A person just has to put their query to the chatbot which is used for chatting. The system will give appropriate answers to the user. If the answer is found invalid, then there is a system to declare the answer as invalid.

## Motivation:

We sometimes pass our time by chatting with different chatterboxes available on the internet, so to make one of them was indeed an interesting idea. We also had thought of integrating it with speech-to-text and text-to-speech software.

**What is bot?**

Bots are program that run continuously formulate decisionds, act upon those decisions without human intervention,and are able adapt to the context they operate in.

The most advanced bots are powered by artificial intelligence, hlping it to understand complex requests,personalize responses and improve interactions over time.

**What is chat box?**

Chatbox are a form human computer dialogue system which operates through natural languagevia text or speech.

**Artificial Intelligence HealthCare Chatbot System**

Through chatbots one can communicate with text or voice interface and get reply through artificial intelligence. Typically, a chat bot will communicate with a real person. Chat bots are used in applications such as ecommerce customer service, call centres and Internet gaming. Chatbots are programs built to automatically engage with received messages. Chatbots can be programmed to respond the same way each time, to respond differently to messages containing certain keywords and even to use machine learning to adapt their responses to fit the situation. A developing number of hospitals, nursing homes, and even private centres, presently utilize online Chatbots for human services on their sites. These bots connect with potential patients visiting the site, helping them discover specialists, booking their appointments, and getting them access to the correct treatment. In any case, the utilization of artificial intelligence in an industry where individuals’ lives could be in question, still starts misgivings in individuals. It brings up issues about whether the task mentioned above ought to be assigned to human staff. This healthcare chatbot system will help hospitals to provide healthcare support online 24 x 7, it answers deep as well as

general questions. It also helps to generate leads and automatically delivers the

information of leads to sales. By asking the questions in series it helps patients by guiding what exactly he/she is looking for.

**Types of Chat Bot's**

* Flow oriented chatbox
* Artificially oriented chatbot
* Human supported bots

### Install Packages

Before starting to work on our chatbot we need to download a few python packages. Please note as of writing this these packages will **only work in python 3.6.** Hopefully this will be fixed in the future.

We will simply use pip to install the following:  
-numpy  
-nltk  
-tensorflow  
- tflear

**Pandas**

**matplotlib.pyplot**

Simply go to CMD and type: **pip install "package name"**. Where you will replace "package\_name" with all of the entries listed above

**Numpy**

NumPy is a Python package which stands for ‘Numerical Python’. It is the core library for scientific computing, which contains a powerful n-dimensional array object, provide tools for integrating C, C++ etc. It is also useful in linear algebra, random number capability etc. .

## NLTK module

NLTK is one of the leading platforms for working with human language data and Python, the module NLTK is used for natural language processing. NLTK is literally

an acronym for Natural Language Toolkit.

Install NLTK with Python 3.x using:

sudo pip3 install nltk

# pyplot

# [matplotlib.pyplot](https://matplotlib.org/api/pyplot_api.html#module-matplotlib.pyplot)

Provides a MATLAB-like plotting framework.

pylab combines pyplot with numpy into a single namespace. This is convenient for interactive work, but for programming it is recommended that the namespaces be kept separate, e.g.:

importnumpy**as**np

importmatplotlib.pyplot**as**plt

x=np.arange(0,5,0.1);

y=np.sin(x)

plt.plot(x,y)

matplotlib.pyplot.acorr(x, hold=None, data=None, \*\*kwargs)

parameters:

**x** : sequence of scalar

**hold** :boolean, optional, deprecated, default: True

**detrend** : callable, optional, default: mlab.detrend\_none

x is detrended by the detrend callable. Default is no normalization.

**normed** :boolean, optional, default: True

if True, input vectors are normalised to unit length.

**usevlines** :boolean, optional, default: True

if True, Axes.vlines is used to plot the vertical lines from the origin to the acorr. Otherwise, Axes.plot is used.

**maxlags** : integer, optional, default: 10

number of lags to show. If None, will return all 2 \* len(x) - 1 lags.

Returns:

**(lags, c, line, b)** : where:

* lags are a length 2`maxlags+1 lag vector.
* c is the 2`maxlags+1 auto correlation vectorI
* line is a [Line2D](https://matplotlib.org/api/lines_api.html#matplotlib.lines.Line2D) instance returned by [plot](https://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot).
* b is the x-axis.

Other parameters:

**linestyle** : [Line2D](https://matplotlib.org/api/lines_api.html#matplotlib.lines.Line2D) prop, optional, default: None

Only used if usevlines is False.

**marker** : string, optional, default: ‘o’

**Import libraries**

* **label Encoder**

[**LabelEncoder**](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html#sklearn.preprocessing.LabelEncoder) is a utility class to help normalize labels such that they contain only values between 0 and n\_classes-1. This is sometimes useful for writing efficient Cython routines

## train\_test\_split

train\_test\_split is a function in **Sklearn model selection** for splitting data arrays into **two subsets**: for training data and for testing data. With this function, you don't need to divide the dataset manually.

By default, Sklearn **train\_test\_split** will make random partitions for the two subsets. However, you can also specify a random state for the operation.

“Splitting the dataset into training set and test set”

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.25, random\_state = 0)

* **Decision Tree Classifier**

Decision Trees can be used as classifier or regression models.

A tree structure is constructed that breaks the dataset down into smaller subsets eventually resulting in a prediction.

**Decision Tree Algorithm**

A decision tree is a flowchart-like tree structure where an internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in recursively manner call recursive partitioning. This flowchart-like structure helps

you in decision making. It's visualization like a flowchart diagram which easily

mimics the human level thinking. That is why decision trees are easy to understand and interpret.



Decision Tree is a white box type of ML algorithm. It shares internal decision-making logic, which is not available in the black box type of algorithms such as Neural Network. Its training time is faster compared to the neural network algorithm. The time complexity of decision trees is a function of the number of records and number of attributes in the given data. The decision tree is a distribution-free or non-parametric method, which does not depend upon probability distribution assumptions. Decision trees can handle high dimensional data with good accuracy.

* **Tkinter**

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

**read\_csv() function:**

Now to import the dataset, we will use read\_csv() function of pandas library, which is used to read a csv file and performs various operations on it. Using this function, we can read a csv file locally as well as through an URL.

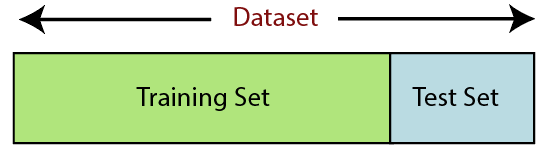
To handle missing values, we will use **Scikit-learn** library in our code, which contains various libraries for building machine learning models

we will convert the country variables into categorical data by using **LabelEncoder()** class from **preprocessing** library.

In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model.

Suppose, if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models.

If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:



**Training Set:** A subset of dataset to train the machine learning model, and we already know the output.

**Test set:** A subset of dataset to test the machine learning model, and by using the test set, model predicts the output

## Requirements:

**Hardware Requirement:**

* i3 Processor Based Computer or higher
* Memory: 1 GB
* Hard Drive: 50 GB
* Monitor
* Internet Connection

**Software Requirement:**

* Windows 7 or higher
* Google Chrome Browser
* **Advantages**
* Save time and money
* Generate new leads
* Guide users
* It provides support 24 x 7
* **Limitation**
* It requires active internet connection.
* Not all business can use chatbot.
* **Application**
* This system can be used by the multiple peoples to get the counselling sessions online.

**Bibliography**

**Books**

Learn PYTHON the HARD WAY(Third Edition)

Introduction to Machine Learning

Machine Learning with Python Cookbook

**Website**

www.we3schools.com

www.stackoverflow.com

www.it-ebooks.com