Speech Recognition Using NLP

#Importing Python Libraries
Line no 1 import speech_recognition as sr
!pip install speech_recognition

Line no 2 import pyttsx3 as pt
!pip install pyttsx3
=> pyttsx3 -> python text to speech

Line no 3 import pywhatkit as pk
!pip install pywhatkit
=> pywhatkit -> python whatsapp kit it uses to automate task.

Line no 4 import subprocess
!pip install subprocess
=> it is installed python library used for command

 # Initialize recognizer and text-to-speech engine
Line no 5 listening = sr.Recognizer()
=> listening -> variable name
=> = -> Assignment Operator
=> sr -> alias name for speech recognition
=> .Recognize() -> it is a object which is used to recognize audio

Line no 6 engine = pt.init()
=> engine -> it is a engine that is responsible to text to speech conversion
=> = -> Assignment Operator
=> pt-> alias name for pyttsX3
=> .init()-> it is a object which is used to initialize text to speech conversion

 # Speak function
Line no 7 def speak(text):
=> def-> it is used to represent user defined function
=> speak() -> it is name of user defined function
=> text-> it is a parameter
=> : -> The block of execution start from here.

Line no 8 engine.say(text)
=> engine-> it is a engine responsible to convert text to speech conversion
=> . -> refer to
=> say ()-> it is a function which is used to take care for text to speech conversion
=> text-> A string which is used to convert text into audio

Line no 9 engine.runAndWait()
=> engine-> it is a engine responsible to convert text to speech conversion
=> . -> refer to
=> runAndWait() -> it is a function which is used to play audio

 # Hear function to capture voice command
Line no 10 def hear():
=> def-> it is used to represent user defined function
=> hear() -> user to define function name
=> : -> The block of execution start from here.

Line no 11 try:
=> try -> it is a keyword used int execution handling mechanism
=> : -> The block of execution for code start from here

Line no 12 with sr.Microphone() as mic:
=> with -> with is the object name which is used to close the file after execution
=> sr -> alias name from speech recognition
=> .Microphone() -> it is a function in the speech recognition module.
=> as -> it used for alias name
=> mic -> it is a alias name for sr.Microphone() function

Line no 13 print('Listening...')
=>print() -> it is a function used to show output on the console.
=> 'Listening...' -> A string that represent ‘text’

Line no 14 voice = listening.listen(mic)
=> voice -> it is a variable name
=> = -> assignment operator
=> listening-> it is a class
=> . -> refer to
=> .listen() -> function used to listen audio from mic
=> mic -> argument

Line no 15 cmd = listening.recognize_google(voice)

=> cmd -> it is a variable name
=> = -> assignment operator
=> listening-> it is a object which is present in recognizable class
=> .recognize_google()-> it is a function from google
=> voice-> it is a argument

Line no 16 cmd = cmd.lower()

=> cmd -> it is a variable name
=> = -> assignment operator
=> cmd.lower() -> it is a function which is used to convert all string to lowercase

Line no 17 if 'pankaj' in cmd:

=> if -> it is a condition
=> ‘pankaj’ -> it is a string
=> in -> ‘in’ is a membership operator (it is checking value exist or not)
=> cmd -> we are calling the variable name cmd
=> : -> The execution for the block starts from here.

Line no 18 cmd = cmd.replace('pankaj', '').strip()

=> cmd -> it is a variable name
=> = -> assignment operator
=> cmd -> we are referring to cmd
=> replace-> it is replacing old value with New value
=> ”pankaj”-> old value
=> “ ” -> new value
=> .strip() -> strip function is used to remove unnecessary space

Line no 19 print(cmd)

=> print() -> print is function used to show output on console

=> cmd -> we are calling the cmd to show output on console

Line no 20 return cmd

=> return -> it terminate the execution of function

=> cmd -> we terminating cmd variable with respect to user defined function

Line no 21 except:

=> execution handling block

=> The execution for a block of code starts from here.

Line no 22 return ""

=> we are terminating the block of code with double we equates

=> empty string

Line no 23 return " "

=> we are terminating the block of code with double quotes

=> empty string

 # Function to open system applications
Line no 24 def open_app(app_name):
=> def ->it is keyword which is used for user-defined function
=> open_app() -> function name which is user defined
=> app_name -> it is a parameter which is used to open apps
=> : -> it is used to execute blocks of code.

Line no 25 if 'notepad' in app_name:
=> if -> it is a condition
=> ‘notepad’ -> it is a string
=> in -> membership operator used to check weather the value is present or not
=> app_name -> it is variable name used to open the file

Line no 26 speak('Opening Notepad')
=> speak() -> it is a function which is used to convert text to speech
=> ‘'Opening Notepad'’ -> a string that consist of text

Line no 27 subprocess.run('notepad')
=> subprocess-> it is a python library used for command pupose
=> .run() -> it is a function used to run the command
=> ‘notepad’ -> it is a string which is in the form of a command.

Line no 28 elif 'calculator' in app_name:
=> elif-> It is a condition Keyword
=> 'calculator' -> it is a string which is in the form of command
=> in -> membership operator
=> app_name -> it is parameter which is used to open file
=> : -> The block of execution code starts from here.

Line no 28   speak('Opening Calculator')
=> speak() -> it is a function which is used to convert text to speech
=> 'Opening Calculator' -> a string that represents text.

Line no 29   subprocess.run('calc')
=> subprocess -> it is a python library for command purpose
=> . -> refer to
=> run() -> it is a function which is used to run the command
=> ‘calc’ -> it is a string in the form of command

Line no 30   elif 'chrome' in app_name:
=> elif-> It is a condition Keyword
=> 'chrome' -> it is a string which is in the form of command
=> in -> membership operator
=> app_name -> it is parameter which is used to open file
=> : -> The block of execution code starts from here.

Line no 31   speak('Opening Google Chrome')
=> speak() -> it is a function which is used to convert text to speech
=> ‘Opening Google Chrome’ -> a string that represents text.

Line no 32   subprocess.run('start chrome', shell=True)
=> subprocess -> it is a python library for command purpose
=> . -> refer to
=> run() -> it is a function which is used to run the command
=> ‘start chrome’ -> it is a string in the form of command
=> shell=True -> we are using shell functionality of window that's why we used this

Line no 33 elif 'command prompt' in app_name or 'cmd' in app_name:

=> elif-> It is a condition Keyword

=> 'command prompt' -> it is a string which is in the form of command

=> in -> membership operator

=> app_name -> it is parameter which is used to open file

=> : -> The block of execution code starts from here.

=> or ‘cmd’ -> it is a string which is in the form of command

=>’in app_name’-> it is a parameter which is used to open the file.

=> : -> The block of execution starts from here.

Line no 34 speak('Opening Command Prompt')

=> speak() -> it is a function which is used to convert text to speech

=> ‘Opening Command Prompt’ -> a string that represents text.

Line no 35 subprocess.run('cmd', shell=True)

=> subprocess -> it is a python library for command purpose

=> . -> refer to

=> run() -> it is a function which is used to run the command

=> ‘cmd’ -> it is a string in the form of command

=> shell=True -> we are using shell functionality of window that's why we used this

Line no 36 elif 'explorer' in app_name:

=> elif-> It is a condition Keyword

=> 'explorer' -> it is a string which is in the form of command
=> in -> membership operator
=> app_name -> it is parameter which is used to open file
=> : -> The block of execution code starts from here.

Line 37 speak('Opening File Explorer')
=> speak() -> it is a function which is used to convert text to speech
=> ‘Opening File Explorer’ -> a string that represents text.

Line no 38 subprocess.run('explorer')
=> subprocess -> it is a python library for command purpose
=> . -> refer to
=> run() -> it is a function which is used to run the command
=> 'explorer' -> it is a string in the form of command

Line no 39 elif 'task manager' in app_name:
=> elif-> It is a condition Keyword
=> 'task manager' -> it is a string which is in the form of command
=> in -> membership operator
=> app_name -> it is parameter which is used to open file
=> : -> The block of execution code starts from here.

Line no 40 speak('Opening Task Manager')
=> speak() -> it is a function which is used to convert text to speech
=> ‘Opening Task Manager’ -> a string that represents text.

Line no 41 subprocess.run('taskmgr')

=> subprocess -> it is a python library for command purpose

=> . -> refer to

=> run() -> it is a function which is used to run the command

=> ‘taskmgr’ -> it is a string in the form of command

Line no 42 else:

=> else -> it is a condition keyword

=> : -> The block of execution code starts from here.

Line no 43 speak(f"Sorry, I can't open {app_name} at the moment")

=> speak() -> it is a function which is used to convert text to speech

=> f ->Format string

=> ‘Sorry, I can't open’ -> A string that represent text

=> {app_name} -> parameter which is shows value for parameter

=> ‘at the moment’ -> A string that represent tex

 # Run function to execute commands
Line no 44 def run():
=> def-> it is used to represent user defined function
=> run() -> user to define function name
=> : -> The block of execution start from here.

Line no 45 cmd = hear()
=> cmd -> variable name
=> = -> Assignment Operator
=> hear() -> we are assigning Hear)() Function to cmd

Line no 46 print(cmd)
=> print() -> it is a function used to show output on console.
=> cmd -> we are calling the ‘cmd’ variable to display output on the console.

 # Play YouTube video
Line no 47 if 'play' in cmd:
=> if -> if is a condition keyword
=> ‘play’ -> it is a substring in cmd if present then execute else not
=> in -> membership operator

Line no 48 song = cmd.replace('play', '').strip()
=> song-> it is a variable name
=> = -> assignment operator
=> cmd -> we are referring to cmd
=> replace-> it is replacing old value with New value
=> ”play”-> old value
=> “ ” -> new value
=> .strip() -> strip function is used to remove unnecessary space

Line no 49 speak(f ‘Playing {song} on YouTube')
=> speak() -> it is a function which is used to convert text to speech
=> f ->Format string
=> ‘Playing ’ -> A string that represent text
=> {song} -> parameter which is shows value for parameter
=> ‘on YouTube’ -> A string that represent text

Line no 50 pk.playonyt(song)
=> pk -> alias name for pywhatkit
=> .playout -> Function which is used to play videos on youtube.
=> song -> it is a parameter value for ‘cmd’ we are calling song variable name.

 # Open system applications
Line no 51 elif 'open' in cmd:
=> elif-> It is a condition Keyword
=> 'open' -> it is a string which is in the form of command
=> in -> membership operator
=> cmd-> it is parameter which is used to open file
=> : -> The block of execution code starts from here.

Line no 52 app_name = cmd.replace('open', '').strip()
=> app_name-> it is a variable name
=> = -> assignment operator
=> cmd -> we are referring to cmd
=> replace-> it is replacing old value with New value
=> ”open'”-> old value
=> “ ” -> new value
=> .strip() -> strip function is used to remove unnecessary space

Line no 53 open_app(app_name)
=> open_app() -> user defined function name
=> app_name -> it is a parameter used to open apps

Line no 54 else:
=> else -> it is a condition keyword
=> : -> The block of execution code starts from here.

Line no 55 speak("I didn't understand that. Please try again.")

=> speak() -> It is a function used to convert text to speech

=> "I didn't understand that. Please try again." -> A string that represent text

 # Main function call