PROJECT

SPEECH TO TEXT CONVERTER

Why Python?

Python has -

- clean object-oriented design
- provides enhanced process control capabilities
- possesses strong integration
- text processing capabilities
- its own unit testing framework

All of which contribute to the increase in its speed and productivity.

Why do we need this type of system?

OBJECTIVES:

- HOW IT WORKS
- WORKING WITH MICROPHONE
- USE CASE

How It Works?

How It Works?

CHALLENGES FACED BY SPEECH RECOGNITION

- STYLE OF SPEAKING
- ENVIRONMENT

How It Works?

Following are the packages which can be used for speech recognition in python.

- Apiai
- Assemblyai
- Google-cloud-speech
- SpeechRecognition
- Pocketsphinx
- Watson-developer-cloud

WORKING WITH MICROPHONE

To work with microphone we will have to install PyAudio package in python

Microphone class

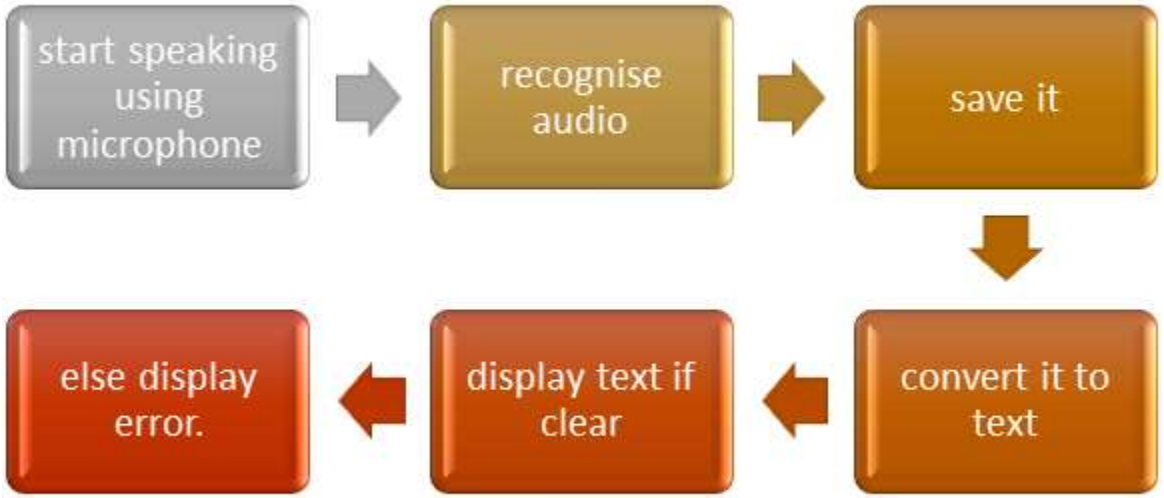
To record audio using the microphone we will have a microphone class.

Use Case

We will make a program using the speech recognition python to execute the following:

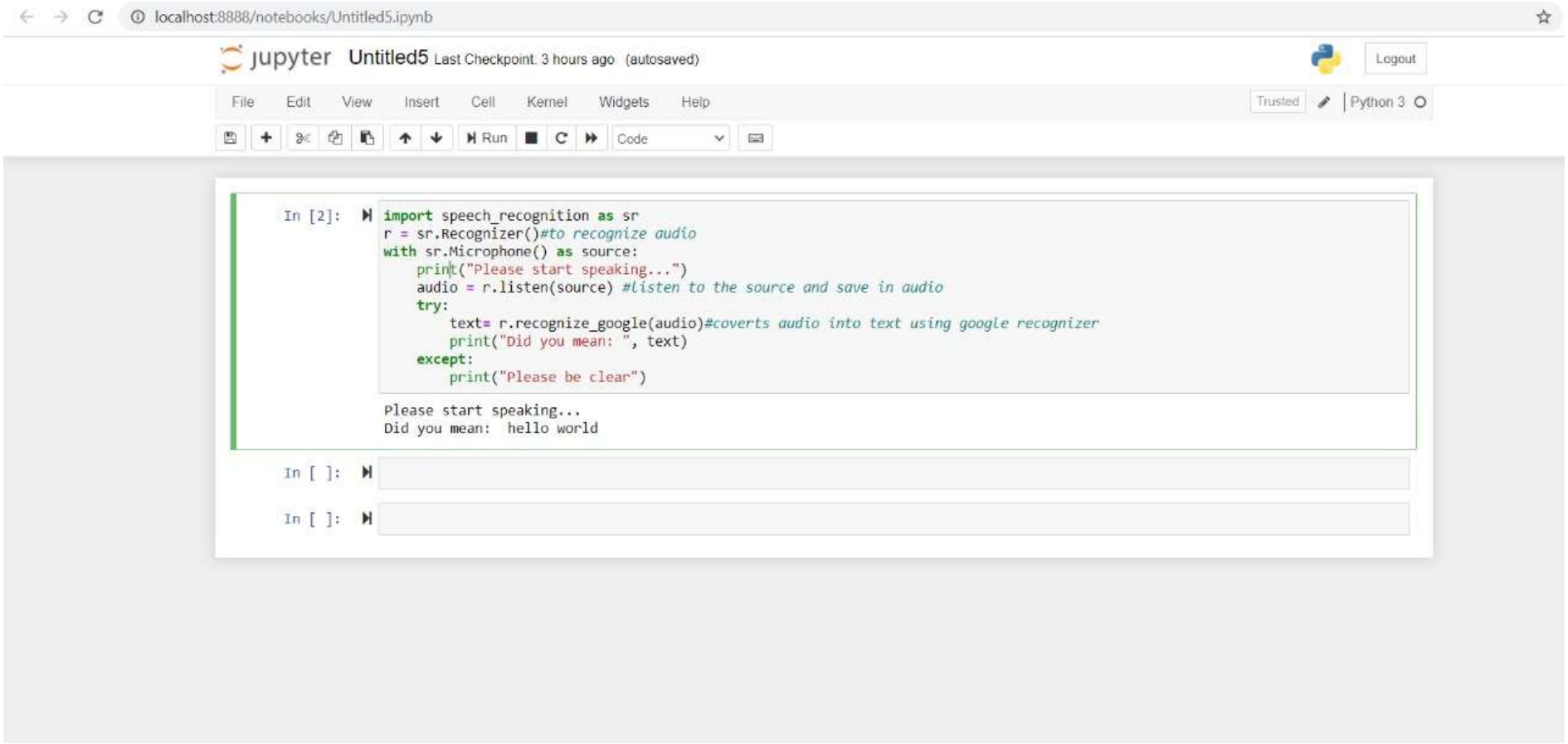
- Converting speech to text using speech recognition

FLOW CHART


start speaking using microphone
recognise audio
save it
convert it to text
display text if clear
else display error.

CODE-

```
import speech_recognition as sr
r = sr.Recognizer() #to recognize audio
with sr.Microphone() as source:
    print("Please start speaking...")
    audio = r.listen(source)
    #listen to the source and save in audio
    try:
        text= r.recognize_google(audio) #coverts audio into text using google recognizer
        print("Did you mean: ", text)
    except:
        print("Please be clear")
```


localhost:8888/notebooks/Untitled5.ipynb
jupyter Untitled5 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code
In [2]:
import speech_recognition as sr
r = sr.Recognizer()#to recognize audio
with sr.Microphone() as source:
print("Please start speaking...")
audio = r.listen(source) #listen to the source and save in audio
try:
text= r.recognize_google(audio)#coverts audio into text using google recognizer
print("Did you mean: ", text)
except:
print("Please be clear")
Please start speaking...
Did you mean: hello world
In [ ]:
In [ ]:

localhost:8888/notebooks/Untitled5.ipynb

jupyter Untitled5 Last Checkpoint: 3 hours ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Run Code

In [3]:

```
import speech_recognition as sr
r = sr.Recognizer()#to recognize audio
with sr.Microphone() as source:
    print("Please start speaking...")
    audio = r.listen(source) #listen to the source and save in audio
    try:
        text= r.recognize_google(audio)#coverts audio into text using google recognizer
        print("Did you mean: ", text)
    except:
        print("Please be clear")
```

```
Please start speaking...
Please be clear
```

In [ ]:

ADVANTAGES:

-> Able to write text through both keyboard and voice input.

->lower operational costs

->provide significant help for the people with disabilities.

->requires less consumption of time in writing text.

APPLICATIONS

In-car systems.

Usage in Education and Daily life

People with disabilities.

illiterate

CONCLUSION

Speech to text is a powerful technology that will soon be ubiquitous. Its reasonably straightforward usability in conjunction with Python (one of the most popular programming languages in the world) makes creating its applications easier. As we make strides in this field, we are paving the path to a world where access to the digital world is not just finger tipped away but also a spoken word.