PYTHON PROJECT

TOPIC :
SPEECH RECOGNITION

Bhawna
2210990223
G(6)

Index

Objective

- **Developing a Speech-to-Text System:** The primary objective of many speech recognition projects is to build a system that can accurately convert spoken language into written text. This involves processing audio input and generating corresponding textual output.
- **Improving Accuracy:** Another objective might be to enhance the accuracy of speech recognition models. This can involve experimenting with different algorithms, architectures, training techniques, and datasets to achieve higher recognition rates.
- **Real-Time Recognition:** Some projects aim to develop real-time speech recognition systems capable of processing audio input and providing text output instantaneously. This objective often involves optimizing algorithms for speed and efficiency.
- **Building Applications:** Many speech recognition projects have the objective of creating practical applications that leverage speech recognition technology. This could include voice-controlled assistants, transcription tools, dictation software, or voice-activated commands for devices and applications.

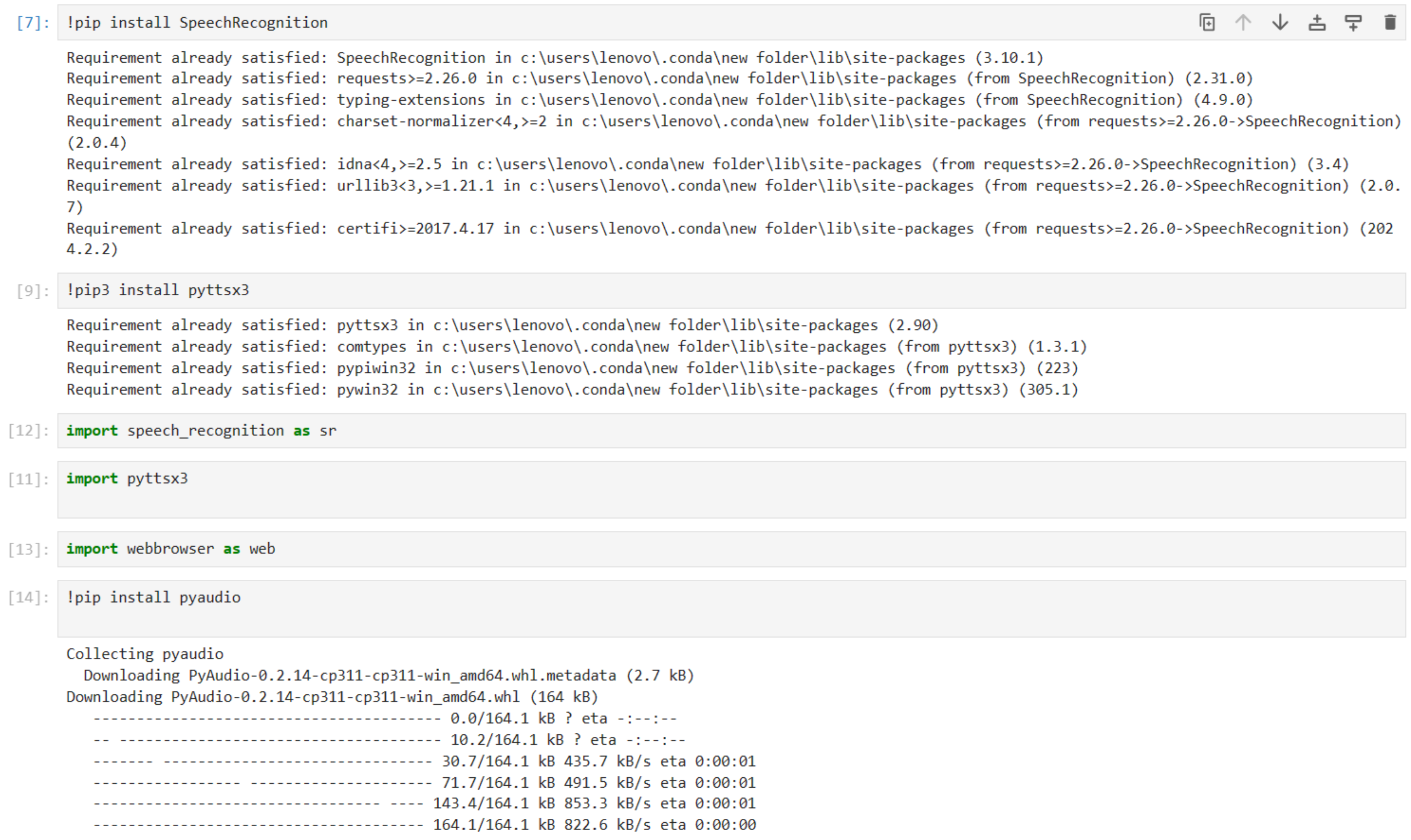
Introduction

Speech recognition technology has rapidly evolved in recent years, enabling machines to understand and interpret human speech with increasing accuracy. This advancement has opened up numerous possibilities in various fields such as virtual assistants, automation, accessibility tools, and more. Speech recognition technology holds immense potential in revolutionizing human-computer interaction and improving accessibility for individuals with disabilities. The primary objective of this project is to develop a speech recognition system capable of accurately transcribing spoken words into text. Through this project, we aim to explore the capabilities of speech recognition using Python and empower developers to integrate this technology into their applications. By understanding the fundamentals of speech recognition and leveraging Python's libraries and tools, we can create innovative solutions that enhance productivity, convenience, and accessibility for users worldwide.

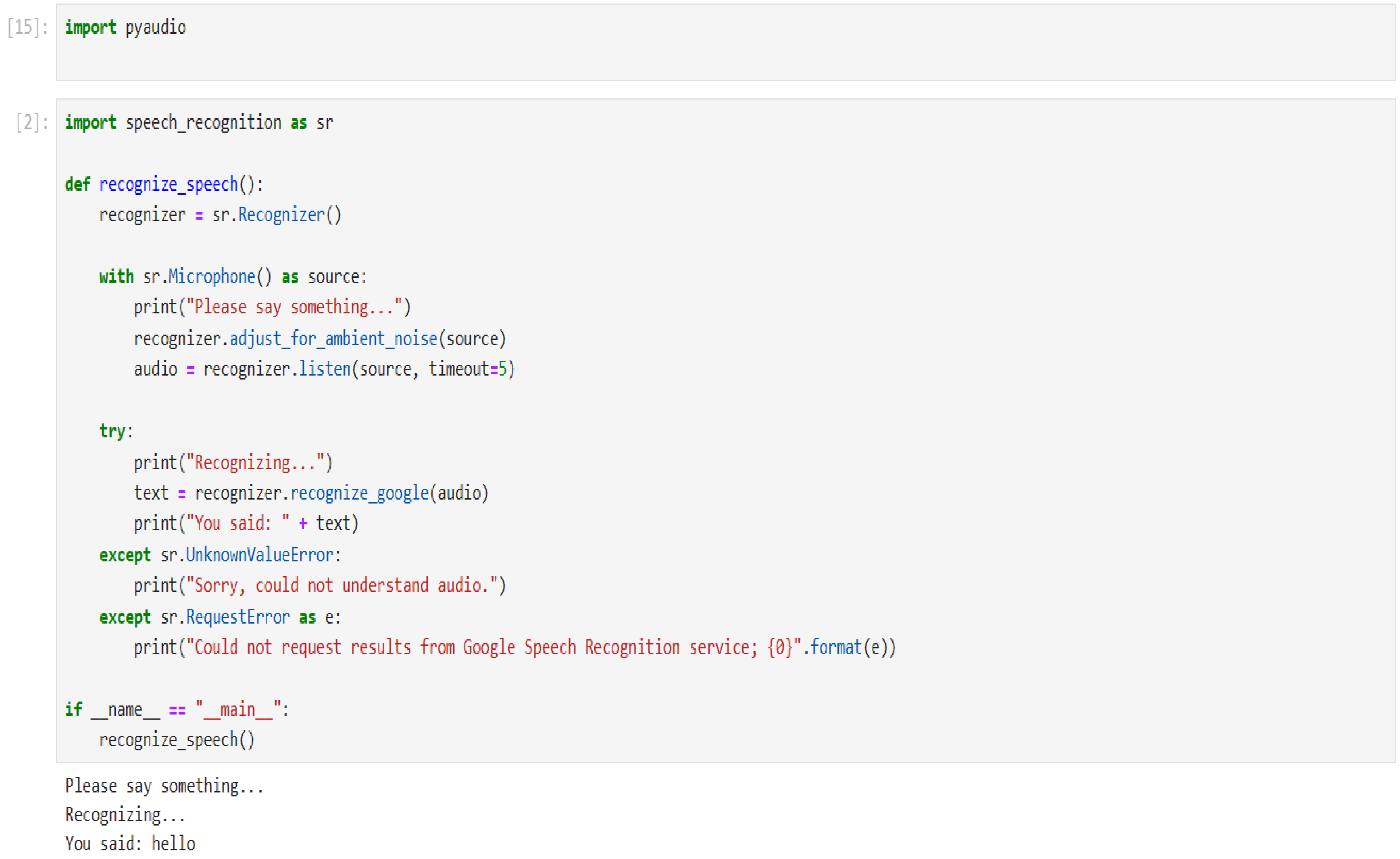
Methodology , Approach & Techniques

- **Data Collection and Preprocessing**:
  - Gather audio data for training (if applicable) and testing your speech recognition model. This may involve recording your own voice or using existing datasets.
  - Preprocess the audio data as needed (e.g., noise reduction, normalization) to improve the accuracy of the speech recognition system.
- **Research and Select Tools**:
  - **SpeechRecognition** : A Python library for performing speech recognition, which supports several speech recognition engines and APIs.
  - **PyAudio** : PyAudio is a Python library that provides bindings for PortAudio, a cross-platform audio I/O library. We will use PyAudio to capture audio input from the microphone and process it for speech recognition.
- **Testing and Evaluation**:
  - Test your speech recognition system thoroughly to ensure its accuracy and robustness.
  - Evaluate the performance of the system based on predefined metrics (e.g., accuracy, response time) and make necessary adjustments.

Source Code



```
[7]: !pip install SpeechRecognition

     Requirement already satisfied: SpeechRecognition in c:\users\lenovo\.conda\new folder\lib\site-packages (3.10.1)
     Requirement already satisfied: requests>=2.26.0 in c:\users\lenovo\.conda\new folder\lib\site-packages (from SpeechRecognition) (2.31.0)
     Requirement already satisfied: typing-extensions in c:\users\lenovo\.conda\new folder\lib\site-packages (from SpeechRecognition) (4.9.0)
     Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\lenovo\.conda\new folder\lib\site-packages (from requests>=2.26.0->SpeechRecognition)
     (2.0.4)
     Requirement already satisfied: idna<4,>=2.5 in c:\users\lenovo\.conda\new folder\lib\site-packages (from requests>=2.26.0->SpeechRecognition) (3.4)
     Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\lenovo\.conda\new folder\lib\site-packages (from requests>=2.26.0->SpeechRecognition) (2.0.
     7)
     Requirement already satisfied: certifi>=2017.4.17 in c:\users\lenovo\.conda\new folder\lib\site-packages (from requests>=2.26.0->SpeechRecognition) (202
     4.2.2)

[9]: !pip3 install pyttsx3

     Requirement already satisfied: pyttsx3 in c:\users\lenovo\.conda\new folder\lib\site-packages (2.90)
     Requirement already satisfied: comtypes in c:\users\lenovo\.conda\new folder\lib\site-packages (from pyttsx3) (1.3.1)
     Requirement already satisfied: pypiwin32 in c:\users\lenovo\.conda\new folder\lib\site-packages (from pyttsx3) (223)
     Requirement already satisfied: pywin32 in c:\users\lenovo\.conda\new folder\lib\site-packages (from pyttsx3) (305.1)

[12]: import speech_recognition as sr

[11]: import pyttsx3

[13]: import webbrowser as web

[14]: !pip install pyaudio

     Collecting pyaudio
       Downloading PyAudio-0.2.14-cp311-cp311-win_amd64.whl.metadata (2.7 kB)
     Downloading PyAudio-0.2.14-cp311-cp311-win_amd64.whl (164 kB)
        ---------------------------------------- 0.0/164.1 kB ? eta -:--:--
        -- ------------------------------------- 10.2/164.1 kB ? eta -:--:--
        ------- -------------------------------- 30.7/164.1 kB 435.7 kB/s eta 0:00:01
        ----------------- ---------------------- 71.7/164.1 kB 491.5 kB/s eta 0:00:01
        ---------------------------------- ---- 143.4/164.1 kB 853.3 kB/s eta 0:00:01
        ---------------------------------------- 164.1/164.1 kB 822.6 kB/s eta 0:00:00
```



```
[15]: import pyaudio
```

```
[2]: import speech_recognition as sr

     def recognize_speech():
         recognizer = sr.Recognizer()

         with sr.Microphone() as source:
             print("Please say something...")
             recognizer.adjust_for_ambient_noise(source)
             audio = recognizer.listen(source, timeout=5)

         try:
             print("Recognizing...")
             text = recognizer.recognize_google(audio)
             print("You said: " + text)
         except sr.UnknownValueError:
             print("Sorry, could not understand audio.")
         except sr.RequestError as e:
             print("Could not request results from Google Speech Recognition service; {0}".format(e))

     if __name__ == "__main__":
         recognize_speech()
```

```
Please say something...
Recognizing...
You said: hello
```

Conclusion

In conclusion, developing a speech recognition project in Python involves a multifaceted approach encompassing data collection, preprocessing, feature extraction, model selection, training, evaluation, integration, and continuous improvement. By following these methodologies and techniques, one can create a robust and accurate speech recognition system capable of transcribing spoken language into text with high fidelity. Overall, a well-executed speech recognition project in Python can significantly enhance communication and productivity in various domains.

The End