

READBACK ERROR DETECTION SOFTWARE

for Rail Radio Safe Working Communications

Venkata Ashita Medam

B.Tech Computer Science, 2025

ashitamv.2003@gmail.com

+91-7874544433

INTRODUCTION

This documentation presents a comprehensive account of my internship tenure at Cyient. It outlines my role, the skills I developed, and the overall contributions that I made to the company during my internship period.

INTERNSHIP ACTIVITIES

ROLE

Junior Data Scientist

- collaborated with senior data scientists and cross-functional teams to understand business objectives and define data analysis requirements.
- conducted exploratory data analysis to identify patterns, trends, and insights that can aid in decision-making of the project.
- supported the development and implementation of machine learning models for predictive and prescriptive analytics.

SKILLS ACQUIRED

Throughout my internship, I acquired and improved upon various skills, including data analysis, project management, and using cloud computing platforms like Amazon Web Services & Microsoft Azure.

COLLABORATION & TEAMWORK

I collaborated with a team comprising of 6 to 8 members.

TECHNOLOGIES USED

SOFTWARE TOOLS USED

Microsoft Azure

- is a public cloud computing platform that can be used for services such as analytics, virtual computing, storage, networking, and much more.

Amazon Web Services

- is a broadly adopted cloud platform that offers several on-demand operations like compute power, database storage, content delivery that helps provide scalable and cost-effective cloud computing solutions.

Google Collaboratory

- is a hosted Jupyter Notebook service that requires no setup to use and provides free access to computing resources, including GPUs and TPUs.

PROGRAMMING LANGUAGES USED

Python

OVERVIEW OF THE PROJECT

DESCRIPTION OF THE USE CASE

In the context of driverless goods trains, a central command centre serves as the hub for overseeing operations. The communication between the command centre and the operator is established through various means, including satellite phones, ensuring seamless connectivity. The command centre relays instructions to the operator regarding the train's projected movement and direction. The operator acknowledges and executes these directives upon confirmation. A critical aspect of this process involves the repetition of instructions for validation; any discrepancies that arise would indicate a potential risk of occurrence of an accident. This scenario illustrates a significant use case where effective communication protocols and confirmation mechanisms are essential to ensure a safe and accurate operation of driverless goods trains.

APPROACH TO RESOLVE THE USE CASE

This three-step process facilitates accurate conversion and evaluation of spoken content into textual data, thereby supporting reliable analysis and decision-making.

- In the initial step, the process begins with taking an audio file and performing noise cancellation techniques to enhance the quality of the audio.
- Subsequently, the next step involves extracting data from an audio file and transforming it into a structured text format, similar to a conversation, using speech-to-text technology.
- Moving on to the final step, the focus is on verifying the comparability of the context between the transcribed dialogues and the original audio content, assessing the level of similarity between them.

IMPLEMENTATION TO RESOLVE THE USE CASE

STEP 1 :

Google Collaboratory Link : [NoiseCancellation_Cyient_1.ipynb](#)

NOISE-CANCELLATION

To perform noise cancellation on the audio file to improve the quality, I used Python libraries called Librosa, Soundfile and Noisereduce.

- Librosa : Python library that provides tools used for audio processing (like analyzing & extracting info)
- Soundfile : Python library used for reading and writing sound files (specially audio data)
- Noisereduce : Python library that uses a noise reducing algorithm called "spectral gating" to remove noise from audio

From my research, I learn that there are 2 types of noise present in an audios -

- Non-Stationary Noise : noise whose statistical properties change over time.
Ex : Environmental Noise (like Traffic, Crowd, Wind etc)
- Stationary Noise : noise whose statistical properties (like mean, variance, and autocorrelation) remain constant over time. Ex : Humming Noise

STEP 2 :

SPEECH-TO-TEXT RECOGNITION

Google Collaboratory Link : [Whisper TTS Comparison Cyient 2.ipynb](#)

The utilization of OpenAI's capabilities helped in providing a transcript from the audio file by speech-to-text recognition. One of OpenAI's standout offerings in this domain is the Whisper AI, an exceptional Speech Recognition & Transcription Model/Software.

Whisper AI is a cutting-edge Speech Recognition & Transcription Solution designed to equip artificial intelligence in the conversion of spoken language to an accurate textual representation.

STEP 2 :

MATCH / MISMATCH OF DIALOGUES

Google Collaboratory Link : [Whisper TTS Comparison Cyient 2.ipynb](#)

In the final step, the comparative evaluation aiming for a minimum degree of similarity, was achieved using a LLM. On using the Transformers python library, data tokenization and embeddings were produced and used to generate similarity using the cosine similarity metric.

APPENDICES OF THE USE CASE

CODE SNIPPET 1 :

NOISE CANCELLATION

Important Python Libraries :

- **Librosa** : Python lib that provides tools used for audio processing (like analyzing & extracting info)
- **Soundfile** : Python lib used for reading and writing sound files (specially audio data)
- **Noisereduce** : Python lib that uses a noise reduce algo called "spectral gating" to remove noise from audio

```
pip install librosa
Requirement already satisfied: librosa in /usr/local/lib/python3.10/dist-packages (0.10.0.post2)
Requirement already satisfied: audioread<=2.1.9 in /usr/local/lib/python3.10/dist-packages (from librosa) (3.0.0)
Requirement already satisfied: numpy<=1.22.0,!=1.22.1,!=1.22.2,>=1.20.3 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.22.4)
Requirement already satisfied: scipy>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.10.1)
Requirement already satisfied: scikit-learn>=0.20.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.2.2)
Requirement already satisfied: joblib>=0.14 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.3.1)
Requirement already satisfied: decorator<=4.3.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (4.4.2)
Requirement already satisfied: numba<=0.51.0,!=0.51.1,>=0.50.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (0.50.4)
Requirement already satisfied: pyparsing<3.1.2,!=3.1.1 in /usr/local/lib/python3.10/dist-packages (from librosa) (3.1.1)
Requirement already satisfied: pioch<1.7,>=1.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.6.0)
Requirement already satisfied: sox<=0.3.2 in /usr/local/lib/python3.10/dist-packages (from librosa) (0.3.5)
Requirement already satisfied: typing-extensions<4.1.1 in /usr/local/lib/python3.10/dist-packages (from librosa) (4.7.1)
Requirement already satisfied: lazy-loader<0.1 in /usr/local/lib/python3.10/dist-packages (from librosa) (0.3)
Requirement already satisfied: msgpack<1.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.0.5)
Requirement already satisfied: llvmlite<0.40,>=0.39.odev0 in /usr/local/lib/python3.10/dist-packages (from numba=>0.51.0->librosa) (0.39.1)
Requirement already satisfied: setuptools<61.0.0,!=60.0.1,>=59.1.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (67.7.2)
Requirement already satisfied: cffi<1.18.0,!=1.17.0,>=1.17.1 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.14.4)
Requirement already satisfied: packaging<20.0 in /usr/local/lib/python3.10/dist-packages (from pioch<1.7,>=1.0->librosa) (23.1)
Requirement already satisfied: requests<=2.19.0 in /usr/local/lib/python3.10/dist-packages (from pioch<1.7,>=1.0->librosa) (2.27.1)
Requirement already satisfied: threadpoolctl<2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn=>0.20.0->librosa) (3.2.0)
Requirement already satisfied: cffi<1.0 in /usr/local/lib/python3.10/dist-packages (from soundfile=>0.12.1->librosa) (1.15.1)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-packages (from cffi=>1.0->soundfile) (2.21)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests=>2.19.0->pioch<1.7,>=1.0->librosa) (1.26.16)
Requirement already satisfied: certifi>2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests=>2.19.0->pioch<1.7,>=1.0->librosa) (2023.7.22)
Requirement already satisfied: charset-normalizer<=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests=>2.19.0->pioch<1.7,>=1.0->librosa) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests=>2.19.0->pioch<1.7,>=1.0->librosa) (3.4)
```

```
pip install soundfile
Requirement already satisfied: soundfile in /usr/local/lib/python3.10/dist-packages (0.12.1)
Requirement already satisfied: cffi<1.0 in /usr/local/lib/python3.10/dist-packages (from soundfile) (1.15.1)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-packages (from cffi=>1.0->soundfile) (2.21)
```

```
pip install noisereduce
Run cell (#/Ctrl+Enter)
cell has not been executed in this session -any.whl (15 KB)
/usr/local/lib/python3.10/dist-packages (from noisereduce) (1.10.1)
o in /usr/local/lib/python3.10/dist-packages (from noisereduce) (3.7.1)
o in /usr/local/lib/python3.10/dist-packages (from noisereduce) (1.11.0)
Requirement already satisfied: numpy<=0.51.0 in /usr/local/lib/python3.10/dist-packages (from librosa->noisereduce) (3.0.0)
Requirement already satisfied: scikit-learn<=0.20.0 in /usr/local/lib/python3.10/dist-packages (from librosa->noisereduce) (1.2.2)
Requirement already satisfied: joblib<=0.14 in /usr/local/lib/python3.10/dist-packages (from librosa->noisereduce) (1.3.1)
Requirement already satisfied: cycler<0.10.0,!=0.10.0,>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->noisereduce) (1.1.2)
Requirement already satisfied: numba<=0.51.0 in /usr/local/lib/python3.10/dist-packages (from librosa->noisereduce) (0.56.4)
Requirement already satisfied: soundfile<=0.12.1 in /usr/local/lib/python3.10/dist-packages (from librosa->noisereduce) (0.12.1)
Requirement already satisfied: pioch<1.7,>=1.0 in /usr/local/lib/python3.10/dist-packages (from librosa->noisereduce) (1.6.0)
Requirement already satisfied: sox<=0.3.2 in /usr/local/lib/python3.10/dist-packages (from librosa->noisereduce) (0.3.5)
Requirement already satisfied: typing-extensions<4.1.1 in /usr/local/lib/python3.10/dist-packages (from librosa->noisereduce) (4.7.1)
Requirement already satisfied: lazy-loader<0.1 in /usr/local/lib/python3.10/dist-packages (from librosa->noisereduce) (0.3)
Requirement already satisfied: msgpack<1.0 in /usr/local/lib/python3.10/dist-packages (from librosa->noisereduce) (1.0.5)
Requirement already satisfied: contourpy<1.0 in /usr/local/lib/python3.10/dist-packages (from matplotliblib->noisereduce) (1.1.0)
Requirement already satisfied: cycler<0.10.0 in /usr/local/lib/python3.10/dist-packages (from matplotliblib->noisereduce) (0.11.0)
Requirement already satisfied: pyparsing<3.1.1,!=3.1.0,>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from matplotliblib->noisereduce) (4.4.1)
Requirement already satisfied: kiwisolver<1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotliblib->noisereduce) (1.0.4)
Requirement already satisfied: packaging<20.0 in /usr/local/lib/python3.10/dist-packages (from matplotliblib->noisereduce) (23.1)
Requirement already satisfied: pillow<6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotliblib->noisereduce) (9.4.0)
Requirement already satisfied: pyparsing<2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotliblib->noisereduce) (3.1.0)
Requirement already satisfied: python-dateutil<2.7.7 in /usr/local/lib/python3.10/dist-packages (from matplotliblib->noisereduce) (2.8.2)
Requirement already satisfied: six<1.16.0,!=1.16.1,>=1.15.2 in /usr/local/lib/python3.10/dist-packages (from matplotliblib->noisereduce) (0.39.1)
Requirement already satisfied: setuptools<61.0.0,!=60.0.1,>=59.1.0 in /usr/local/lib/python3.10/dist-packages (from matplotliblib->noisereduce) (67.7.2)
Requirement already satisfied: appdirs<1.3.0 in /usr/local/lib/python3.10/dist-packages (from pioch<1.7,>=1.0->librosa->noisereduce) (1.4.4)
Requirement already satisfied: requests<=2.19.0 in /usr/local/lib/python3.10/dist-packages (from pioch<1.7,>=1.0->librosa->noisereduce) (2.27.1)
Requirement already satisfied: scikit-learn<=0.22.2 in /usr/local/lib/python3.10/dist-packages (from python-dateutil->noisereduce) (1.16.0)
Requirement already satisfied: cffi<1.0 in /usr/local/lib/python3.10/dist-packages (from requests=>2.19.0->pioch<1.7,>=1.0->librosa->noisereduce) (3.2.0)
Requirement already satisfied: certifi>2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests=>2.19.0->pioch<1.7,>=1.0->librosa->noisereduce) (2023.7.22)
Requirement already satisfied: charset-normalizer<=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests=>2.19.0->pioch<1.7,>=1.0->librosa->noisereduce) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests=>2.19.0->pioch<1.7,>=1.0->librosa->noisereduce) (3.4)
Installing collected packages: noisereduce
Successfully installed noisereduce-2.0.1
```

```

import IPython
from scipy.io import wavfile
import noisereduce as nr
import soundfile as sf
from noisereduce.generate_noise import band_limited_noise
import matplotlib.pyplot as plt
import urllib.request
import numpy as np
import io
import librosa
%matplotlib inline

```

Loading Input Audio File

```

[ ] url_1 = "/content/conversation-match-noise.wav"
url_2 = "/content/conversation-mismatch-noise.wav"

[ ] data_1, rate_1 = librosa.load(url_1)
data_2, rate_2 = librosa.load(url_2)

[ ] IPython.display.Audio(data=data_1, rate=rate_1)

▶ 0:00 / 0:10 ━━━━ ⏪ ⏴

[ ] IPython.display.Audio(data=data_2, rate=rate_2)

▶ 0:00 / 0:13 ━━━━ ⏪ ⏴

[ ] noise_len1 = 10# seconds
noise1 = band_limited_noise(min_freq=2000, max_freq = 12000, samples=len(data_1), samplerate=rate_1)*10
noise_clip1 = noise1[:rate_1*noise_len1]
audio_clip_band_limited1 = data_1+noise1

[ ] noise_len2 = 13# seconds
noise2 = band_limited_noise(min_freq=2000, max_freq = 12000, samples=len(data_2), samplerate=rate_2)*10
noise_clip2 = noise2[:rate_2*noise_len2]
audio_clip_band_limited2 = data_2+noise2

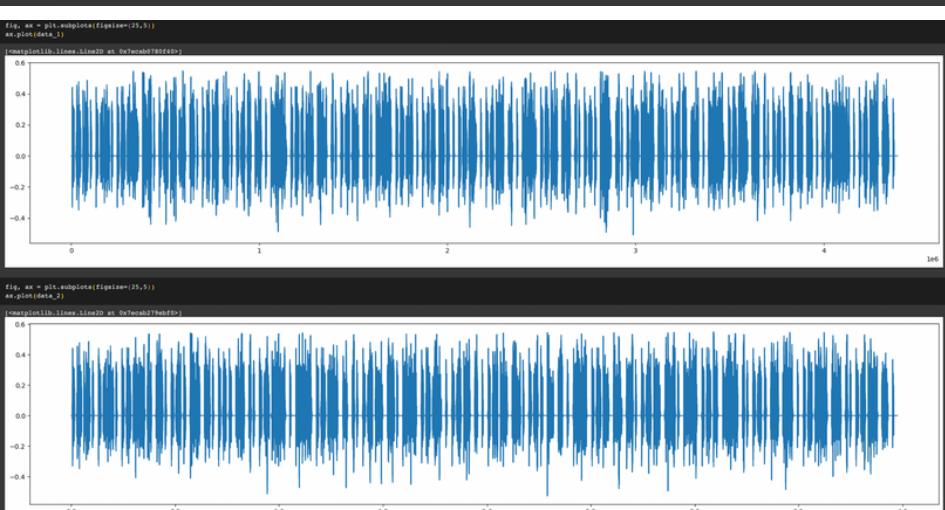
[ ] IPython.display.Audio(data=audio_clip_band_limited1, rate=rate_1)

▶ 0:00 / 0:10 ━━━━ ⏪ ⏴

[ ] IPython.display.Audio(data=audio_clip_band_limited2, rate=rate_2)

▶ 0:00 / 0:13 ━━━━ ⏪ ⏴

```



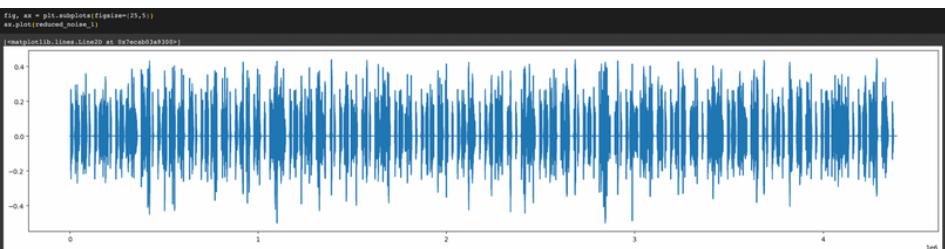
Removing Stationary Noise

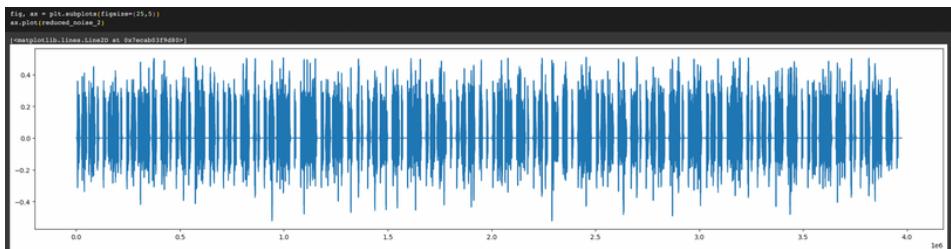
- only uses the initially estimated noise threshold
 - Stationary Noise** - noise whose statistical properties (like mean, variance, and autocorrelation) remain constant over time.
- Ex : Humming Noise

```

reduced_noise_1 = nr.reduce_noise(y = data_1, sr=rate_1, n_std_thresh_stationary=1,stationary=True)
reduced_noise_2 = nr.reduce_noise(y = data_2, sr=rate_2, n_std_thresh_stationary=0.5,stationary=True)

```





```
IPython.display.Audio(data=reduced_noise_1, rate=rate_1)
```

▶ 0:00 / 3:19 ━━━━ 🔍 ⏮

```
IPython.display.Audio(data=reduced_noise_2, rate=rate_2)
```

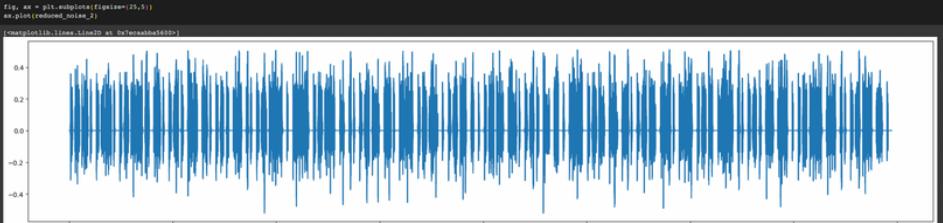
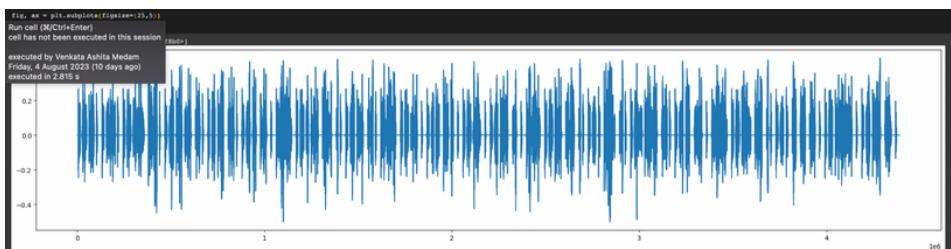
▶ 0:00 / 3:00 ━━━━ 🔍 ⏮

Removing Non-Stationary Noise

- keeps updating the initially estimated noise threshold during the process of noise reduction.
- **Non-Stationary Noise** - noise whose statistical properties change over time.

EX: Environmental Noise (like Traffic, Crowd, Wind etc)

```
[1] reduced_noise_1 = nr.reduce_noise(y = data_1, sr=rate_1, n_std_thresh_stationary=1,stationary=True)
reduced_noise_2 = nr.reduce_noise(y = data_2, sr=rate_2, n_std_thresh_stationary=0.5,stationary=False)
```



```
IPython.display.Audio(data=reduced_noise_1, rate=rate_1)
```

▶ 0:00 / 3:19 ━━━━ 🔍 ⏮

```
IPython.display.Audio(data=reduced_noise_2, rate=rate_2)
```

▶ 0:00 / 3:00 ━━━━ 🔍 ⏮

```
folder_prefix2 = 'noise-cancelled-audios/'
```

```
try:
    # Upload each output file to the 'noise-cancelled-audios' folder in S3
    for file_path in output_file_paths:
        # Use the basename of the file as the S3 object key
        object_key = folder_prefix2 + file_path.split('/')[-1]

        # Upload the file to S3
        with open(file_path, 'rb') as file:
            s3_client.upload_fileobj(file, bucket_name, object_key)

        print("Upload complete:", object_key)

except Exception as e:
    print("An error occurred:", e)
```

```
Upload complete: noise-cancelled-audios/conversation_1.wav
Upload complete: noise-cancelled-audios/conversation_2.wav
```

CODE SNIPPET 2 :

SPEECH-TO-TEXT RECOGNITION

&

MATCH / MISMATCH OF DIALOGUES

Install & Update of OpenAI's Whisper AI

Whisper AI: It is a Speech Recognition & Transcription Model/Software that uses AI to convert speech to text.

Running Whisper AI on the Audio File

[1] I whisper /content/conversation 1-way --medium --language English --translatetext
[0]:00.000... [01:00.220] Person A, sender Person B, this is Person A come in, over.
[0]:00.000... [01:00.220] Person B, receive this is Person B loud and clear, over.
[0]:11.320... [01:17.220] Person A, sender Person B, we need to proceed to the extraction point immediately.
[0]:11.320... [01:17.220] Person B, receive understood, I am proceeding to extraction point, over.
[0]:21.440... [01:28.440] Person A, sender Person B, confirming the location, over.
[0]:28.640... [01:31.840] Person B, receiver copy, that Person A Grid Alpha 5, proceeding to extraction point.
[0]:35.860... [01:38.860] Person A, sender Person B, Grid Alpha 5, proceeding to extraction point, over.
[0]:36.860... [01:40.860] Person B, receiver confirmed, Grid Alpha 5, proceeding to extraction point, over.
[0]:44.440... [01:49.440] Person A, sender Person B, confirming extraction point, over.
[0]:48.420... [01:53.420] Person B reaches Grid Alpha 5 and waits for further instructions.
[0]:53.220... [01:58.220] Person A, sender Person B, be advised, there is a potential threat in the area.
[0]:53.220... [01:58.220] Person B, receiver caution, potential threat.
[0]:10.542... [01:05.240] Person A, sender Person B, confirming extraction point, over.
[0]:10.542... [01:05.240] Person B, receiver copy, that Person A proceeding with caution.
[0]:13.740... [01:18.740] Person A, sender Person B, confirming potential threat, over.
[0]:13.740... [01:18.740] Person B, receiver copy, that Person A proceeding with caution.
[0]:13.760... [01:18.760] Person A, sender Person B, encounters the potential threat but fails to confirm properly.
[0]:13.760... [01:18.760] Person B, receiver copy, that Person A encounters the potential threat, over.
[0]:12.900... [01:13.900] Person A, sender Person B, requesting extraction point, over.
[0]:12.900... [01:13.900] Person B, receiver copy, this is Person B, I think I have extracted but not sure, over.
[0]:13.660... [01:14.660] Person A, sender Person B, acknowledged and reconfirm immediately, over.
[0]:13.660... [01:14.660] Person B, receiver copy, that Person A confirming potential threat.
[0]:14.570... [01:18.490] Person A, sender Person B, proceeding with caution, over.
[0]:14.570... [01:18.490] Person B, receiver copy, that Person A proceeding with caution, over.
[0]:14.980... [01:15.490] Person A, sender Person B, Person B verifies the potential threat and proceeds cautiously.
[0]:14.980... [01:15.490] Person B, receiver copy, that Person A proceeding with caution, update on the extraction point.
[0]:15.030... [01:20.030] Person A, sender Person B, it has been changed to Grid Bravo 7.
[0]:15.030... [01:20.030] Person B, receiver copy, that Person A proceeding with caution, update on the extraction point.
[0]:01.220... [01:03.940] Person A, sender Person B, confirming extraction point, over.
[0]:01.220... [01:03.940] Person B, receiver copy, that Person A confirming extraction point, over.
[0]:11.220... [01:22.020] Person A, sender Person B, Please reconfirm, over.
[0]:11.220... [01:22.020] Person B, receiver copy, that Person A, I still have Grid Alpha 5 in my coordinates.
[0]:21.960... [01:24.980] Person A, sender Person B, confirming the extraction point, over.
[0]:21.960... [01:24.980] Person B, receiver copy, that Person A confirming the extraction point, over.
[0]:24.980... [01:29.780] Person A, sender Person B, confirming the new location, over.
[0]:24.980... [01:29.780] Person B, receiver copy, that Person A confirming the new location, over.
[0]:33.860... [01:38.700] Person A, sender Person B, reaches Grid Bravo 7 and awaits further instructions.
[0]:33.860... [01:38.700] Person B, receiver copy, that Person A reaches Grid Bravo 7 and awaits further instructions.
[0]:38.700... [01:43.980] Person A, sender Person B, we have encountered unexpected resistance.
[0]:38.700... [01:43.980] Person B, receiver copy, that Person A unexpected resistance and fall back, over.
[0]:47.340... [01:49.240] Person A, sender Person B, confirming extraction point, over.
[0]:47.340... [01:49.240] Person B, receiver copy, that Person A cannot shoot short, over.
[0]:49.440... [01:55.140] Person A, sender Person B, confirming medical support, over.
[0]:49.440... [01:55.140] Person B, receiver copy, that Person A confirming medical support, over.
[0]:01.260... [01:06.260] Person A, sender understood, Person B confirming injured Person L and requesting immediate medical support, over.
[0]:06.740... [01:10.220] Person A, sender Person B, confirming affirmative, Person A injured Person L confirmed.
[0]:06.740... [01:10.220] Person B, receiver copy, that Person A confirming affirmative, Person A injured Person L confirmed.
[0]:15.750... [01:18.420] Person A, sender Person B, Need medical support urgently, over.

```
!whisper /content/conversation_2.wav --model medium --language English --task transcribe
```

00:06.200 → 00:06.300 Person A, render Person B, this is Person A come in, over.
00:06.200 → 00:07.1200 Person B, receive this is Person B loud and clear, over.
00:11.320 → 00:11.330 Person A, render Person B, we need to rendezvous at the designated meeting point, Grid Charlie 3.
00:18.840 → 00:19.840 Confirm your current location, over.
00:22.080 → 00:22.940 Person B, receive Roger that, Person A. My current location is Grid Delta 2.
00:23.000 → 00:23.100 Person A, render Person B, we need to rendezvous at the designated meeting point, Grid Charlie 3.
00:32.400 → 00:33.400 Person A, render Negative, Person B. The meeting point is Grid Charlie 3.
00:34.900 → 00:35.100 Confirm, over.
00:36.000 → 00:36.100 Person A, render Apologies, Person A confirming Grid Charlie 3 as the meeting point, over.
00:49.280 → 00:51.760 Person B acknowledges the correct meeting point and adjusts their course.
00:51.800 → 00:52.100 Person A, render Person B, be aware of potential hostiles in the area.
00:59.250 → 00:59.550 Person A, render Person B, be aware of potential hostiles in the area.
01:00.120 → 01:00.940 Person A, render Person B, be aware of potential hostiles, over.
01:00.940 → 01:01.940 Person A, render Person B, be aware of potential hostiles, over.
01:12.600 → 01:14.640 Person B encounters potential hostiles and takes cover.
01:14.640 → 01:15.200 Person A, render Person B, do you confirm the presence of hostiles?
01:23.800 → 01:24.200 Over.
01:23.300 → 01:28.200 Person B, receive Affirmative, Person A hostiles spotted.
01:28.200 → 01:29.200 Person A, render Stay in cover, Person B backup is en route.
01:33.240 → 01:34.640 Person A, render Stay in cover, Person B backup is en route.
01:34.560 → 01:46.140 Confirm you are well hidden and secure, over.
01:46.140 → 01:47.140 Person A, render Stay in cover, Person A : I am well hidden and secure.
01:46.520 → 01:49.520 Waiting for backup, over.
01:49.840 → 01:50.940 Person B confide their safety while waiting for backup.
01:50.940 → 01:51.940 Person A, render Person B, update on the situation.
01:58.640 → 02:01.320 Backup ETA is 5 minutes.
02:01.340 → 02:04.480 Hold position and confirm, over.
02:04.480 → 02:05.480 Person A, render Person B a holding position.
02:05.480 → 02:07.480 Confirming 5 minutes ETA for backup, over.
02:09.980 → 02:11.980 Person B receives the backup and confirms their arrival.
02:11.980 → 02:12.980 Person A, render Person B a holding position.
02:12.980 → 02:24.980 Lead with the original plan to rendezvous at grid Charlie 3.
02:24.500 → 02:27.000 Confirm, over.
02:27.000 → 02:28.000 Person A, render Negative, Person A. The hostiles have increased in number.
02:35.300 → 02:46.300 Requesting alternative extraction point, over.
02:46.300 → 02:46.900 Person A, render Understood, Person B confirming the change in extraction point.
02:46.900 → 02:47.900 New extraction point is grid Echo 4.
02:48.940 → 02:51.940 Confirm, over.
02:51.500 → 02:54.600 Person B, receiver Confirmed, Person A rendezvous at grid Echo 4.

```
transcript_1 = "Person A, sender Person B, this is Person A come in, over.\n" \
    "Person B, receiver this is Person B and clear, over.\n" \
    "Person A, we have encountered unexpected resistance, over.\n" \
    "Our current location is Grid Alpha 5, over.\n" \
    "Person B, receiver copy that, Person A Grid Alpha 5, proceeding to extraction point.\n" \
    "Confirming the location, over.\n" \
    "Person A, receiver confirmation, over.\n" \
    "Person B, receiver Grid Alpha 5, over.\n" \
    "Confirming extraction point, over.\n" \
    "Person B, receiver confirmed, over.\n" \
    "Person A, sender Person B, this is Person A come in, over.\n" \
    "Person B, receiver Grid Alpha 3 and wait for further instructions, over.\n" \
    "Person A, sender Person B, be advised, there is a potential threat in the area.,\n" \
    "Exercise caution and proceed with stealth.,\n" \
    "Confirm, over.\n" \
    "Person B, receiver that, Person A proceeding with caution.,\n" \
    "Confirming potential threat, over.\n" \
    "Person B encounters the potential threat but fails to confront properly.,\n" \
    "Person A, receiver Person B, update on the extraction point.,\n" \
    "Person B, receiver alpha, this is Person B. I think I see something, but not sure, over.\n" \
    "Person A, sender Person B, acknowledges and reconfirms immediately, over.\n" \
    "Person B, receiver negative, Person A, proceed to Grid Alpha 5, over.\n" \
    "Person A, receiver caution, Person B confirming potential threat.,\n" \
    "Person B verifies the potential threat and proceeds cautiously, over.\n" \
    "Person A, sender Person B, update on the extraction point.,\n" \
    "Person B, receiver Grid Bravo 7 and wait for further instructions.,\n" \
    "Person A, sender Person B, we have encountered unexpected resistance.,\n" \
    "Abort the extraction and fall back.,\n" \
    "Confirm, over.\n" \
    "Person B, receiver negative, Person A. We cannot abort yet.,\n" \
    "We have injured Person L. Requesting medical support, over.\n" \
    "Person A, receiver Person B confirming injured person L and requesting immediate\n    medical report, over.\n" \
    "Person B, receiver affirmative, Person A injured Person L confirmed.,
```

Run cell (⌘/Ctrl+Enter)

```
transcript_3 = "Person A, sender Person B, this is Person A come in, over.\n" \
    "Person B, receiver this is Person B loud and clear, over.\n" \
    "Person A, sender Person B, we need to rendezvous at the designated meeting point, Grid Charlie" \
    ".\n" \
    "Confirm your current location, over.\n" \
    "Person B, receiver Roger that, Person A. My current location is Grid Delta 2.\n" \
    "Confirming the meeting point, over.\n" \
    "Person A, sender Negative, Person B. The meeting point is Grid Charlie 3.\n" \
    "Lead the extraction point, over.\n" \
    "Person B, receiver Apologies, Person A confirming Grid Charlie 3 as the meeting point, over.\n" \
    "Person B acknowledges the correct meeting point and adjusts their course.\n" \
    "Person A, sender Person B, be aware of potential hostiles in the area.\n" \
    "Proceed with caution.\n" \
    "Contact Roger, over.\n" \
    "Person B, receiver Copy that, Person A proceeding with caution.\n" \
    "Confirming potential hostiles, over.\n" \
    "Person B encounters potential hostiles and takes cover.\n" \
    "Person A, sender Person B, do you confirm the presence of hostiles?\n" \
    "Person B, over.\n" \
    "Person B, receiver Affirmative, Person A hostiles spotted.\n" \
    "Confirming cover, over.\n" \
    "Person A, sender Stay in cover, Person B backup is en route.\n" \
    "Contact Roger, he will hide and secure, over.\n" \
    "Person B, receiver Confirm, Person A, I am well hidden and secure.\n" \
    "Waiting for backup, over.\n" \
    "Person A confirms their safety while waiting for backup.\n" \
    "Person A, sender Roger, update on the situation.\n" \
    "Backup ETA is 5 minutes.\n" \
    "Hold position and confirm, over.\n" \
    "Person B, receiver Roger that, Person A holding position.\n" \
    "Confirming 5 minutes ETA for backup, over.\n" \
    "Person A, receiver Roger that, Person A awaiting arrival.\n" \
    "Person A, sender Person B, backup has arrived.\n" \
    "Lead with the original plan to rendezvous at grid Charlie 3.\n" \
    "Confirm, over.\n" \
    "Person A, receiver Negative, Person A. The hostiles have increased in number.\n" \
    "Person A, sender Understood, Person B confirming the change in extraction point.\n" \
    "New rendezvous point is grid Echo 4.\n" \
    "Confirm, over.\n" \
    "Person B, receiver confirmed, Person A rendezvous at grid Echo 4.\n" \
    "Rendezvous point is grid Echo 4, over.\n"
```

```

def new_transcript(transcript):
    new_transcript = []

    lines = transcript.strip().split("\n")

    for line in lines:
        speaker_start_idx = line.find("Person ")
        if speaker_start_idx != -1:
            speaker_end_idx = line[speaker_start_idx:].find("'", speaker_start_idx)
            if speaker_end_idx != -1:
                speaker_end_idx += 1
                modified_line = line[:speaker_end_idx] + ":" + line[speaker_end_idx + 1:]
                new_transcript.append(modified_line)
            else:
                new_transcript.append(line)
        else:
            new_transcript.append(line)

    return new_transcript

```



```

# Import transformers library, AutoTokenizer, and Model
import torch
import torch.nn.functional as F
from sentence_transformers import SentenceTransformer
from sentence_transformers import CosineSimilarity

# Compute mean - Take attention mask into account for current averaging
def mean_pooling(model_output, attention_mask):
    token_embeddings = model_output[0] #First element of model_output contains all token embeddings
    input_mask_expanded = attention_mask.unsqueeze(-1).expand(token_embeddings.size(0), -1, -1)
    return torch.sum(token_embeddings * input_mask_expanded, 1) / torch.clamp(input_mask.sum(1), min=1e-9)

# Load model from Huggingface hub
tokenizer = AutoTokenizer.from_pretrained('sentence-transformers/all-mpnet-base-v2')
model = AutoModel.from_pretrained('sentence-transformers/all-mpnet-base-v2')

for i in range(len(dialogues_1)):
    sentences = dialogues_1[i], dialogues_2[i]
    print(sentences)

    # Tokenize sentences
    encoded_input = tokenizer(sentences, padding=True, truncation=True, return_tensors='pt')

    # Compute token embeddings
    with torch.no_grad():
        model_output = model(**encoded_input)

    # Perform pooling
    sentence_embeddings = mean_pooling(model_output, encoded_input['attention_mask'])

    # Normalize embeddings
    sentence_embeddings = F.normalize(sentence_embeddings, p=2, dim=1)

    # Print sentence embeddings
    similarity = cosine_similarity(sentence_embeddings[0].reshape(1,-1), sentence_embeddings[1].reshape(1,-1))
    print(f'({i}) : {similarity}')

    #<sender Person B, this is Person A come in, over., 'receiver this is Person B loud and clear, over.'>
    #<sender Person B, we need to proceed to the extraction point immediately,our current location is Grid Alpha 5, over., 'receiver copy that, Person A Grid Alpha 5, proceeding to extraction point,confirming the location, over.'>
    #<receiver Person B, I am affirmative, Person B Grid Alpha 5.Confirming extraction point, over., 'receiver confirmed, Grid Alpha 5.on route to extraction point, over.'>
    #<sender Person A, I am advised, there is a potential threat in the area.Exercise caution and proceed with stealth,Confirm, over., 'receiver Roger that, Person A proceeding with caution.Confirming potential threat, over.'>
    #<receiver Person B, do you confirm the potential threat, over., 'receiver a, this is Person B. I think I see something, but not sure, over.'>
    #<sender Person B, I acknowledge and reconfirm immediately, over., 'receiver Roger, Person A confirming potential threat,Proceeding with caution, over.'>
    #<receiver Person B, update on the extraction point,it has been changed to Grid Bravo 7,Confirm, over., 'receiver negative, Person A. I still have Grid Alpha 5 in my coordinates.Please reconfirm, over.'>
    #<sender Person B, I am sorry for the confusion, Person B extraction point is indeed Grid Bravo 7,Confirming the new location, over., 'receiver confirmed, Grid Bravo 7.Proceeding to the new extraction point, over.'>
    #<receiver Person B, we have encountered unexpected resistance,Abort the extraction and fall back,Confirm, over., 'receiver negative, Person A. We cannot abort yet,We have injured Person L. Requesting medical support, over.'>
    #<sender Person B, Person B confirming injured Person L and requesting immediate medical support, over., 'receiver affirmative, Person A injured Person L confirmed,Need medical support urgently, over.'>

for i in range(len(dialogues_2)):
    sentences = dialogues_1[i], dialogues_2[i]
    print(sentences)

    # Tokenize sentences
    encoded_input = tokenizer(sentences, padding=True, truncation=True, return_tensors='pt')

    # Compute token embeddings
    with torch.no_grad():
        model_output = model(**encoded_input)

    # Perform pooling
    sentence_embeddings = mean_pooling(model_output, encoded_input['attention_mask'])

    # Normalize embeddings
    sentence_embeddings = F.normalize(sentence_embeddings, p=2, dim=1)

    # Print sentence embeddings
    similarity = cosine_similarity(sentence_embeddings[0].reshape(1,-1), sentence_embeddings[1].reshape(1,-1))
    print(f'({i}) : {similarity}')

    #<sender Person B, this is Person A come in, over., 'receiver this is Person B loud and clear, over.'>
    #<sender Person B, we need to proceed to the extraction point immediately,our current location is Grid Alpha 5, over., 'receiver copy that, Person A Grid Alpha 5, proceeding to extraction point,confirming the location, over.'>
    #<receiver Person B, I am affirmative, Person B Grid Alpha 5.Confirming extraction point, over., 'receiver confirmed, Grid Alpha 5.on route to extraction point, over.'>
    #<sender Person A, I am advised, there is a potential threat in the area.Exercise caution and proceed with stealth,Confirm, over., 'receiver Roger that, Person A proceeding with caution.Confirming potential threat, over.'>
    #<receiver Person B, do you confirm the potential threat, over., 'receiver a, this is Person B. I think I see something, but not sure, over.'>
    #<sender Person B, I acknowledge and reconfirm immediately, over., 'receiver Roger, Person A confirming potential threat,Proceeding with caution, over.'>
    #<receiver Person B, update on the extraction point,it has been changed to Grid Bravo 7,Confirm, over., 'receiver negative, Person A. I still have Grid Alpha 5 in my coordinates.Please reconfirm, over.'>
    #<sender Person B, I am sorry for the confusion, Person B extraction point is indeed Grid Bravo 7,Confirming the new location, over., 'receiver confirmed, Grid Bravo 7.Proceeding to the new extraction point, over.'>
    #<receiver Person B, we have encountered unexpected resistance,Abort the extraction and fall back,Confirm, over., 'receiver negative, Person A. We cannot abort yet,We have injured Person L. Requesting medical support, over.'>
    #<sender Person B, Person B confirming injured Person L and requesting immediate medical support, over., 'receiver affirmative, Person A injured Person L confirmed,Need medical support urgently, over.'>

# Output transcript file paths = ['content/conversation_1.txt', 'content/conversation_2.txt']

folder_prefix1 = 'transcript/'

for file_path in output_transcript_file_paths:
    # Use the basename of the file as the S3 object key
    object_key = folder_prefix1 + file_path.split('/')[-1]

    # Upload the file to S3
    with open(file_path, 'rb') as file:
        s3_client.upload_fileobj(file, bucket_name, object_key)

    print("Upload complete!", object_key)

except Exception as e:
    print("An error occurred:", e)

Upload complete: transcript/conversation_1.txt
Upload complete: transcript/conversation_2.txt

```

UTILIZATION OF CLOUD SERVICES

AMAZON WEB SERVICES

Amazon Web Services (AWS) played a pivotal role in seamlessly integrating the project's backend with its frontend. Specifically, AWS's S3 bucket service proved instrumental in fulfilling the project's requirements.

The process commenced with the installation and utilization of the Boto3 library, which facilitates interaction with S3 buckets. Boto3, an AWS SDK for Python, offers a comprehensive Python API for various AWS infrastructure services, including Amazon S3, Amazon EC2, and Amazon DynamoDB, among others.

Within this project, the Boto3 library was leveraged for the purpose of uploading and retrieving files intended for display on the frontend. To initiate this process, the user's AWS account details, including the AWS secret access key, AWS access key ID, and the region name, needed to be provided. Subsequently, Boto3 was utilized to upload the requisite files to the S3 bucket, enabling public access to these files. This approach eliminates the restriction of file access solely to a user's local desktop, thereby defining the end-to-end flow between the project's backend and frontend.

Uploading the NOISE-CANCELLED AUDIOS into the S3 BUCKET

```

[ ] output_file_paths = [
    "/content/conversation_1.wav",
    "/content/conversation_2.wav"
]

[ ] folder_prefix1 = 'noise-cancelled-audios/'

try:
    # Upload each output file to the 'noise-cancelled-audios' folder in S3
    for file_path in output_file_paths:
        # Use the basename of the file as the S3 object key
        object_key = folder_prefix1 + file_path.split('/')[-1]

        # Upload the file to S3
        with open(file_path, 'rb') as file:
            s3_client.upload_fileobj(file, bucket_name, object_key)

        print("Upload complete!", object_key)

except Exception as e:
    print("An error occurred:", e)

Upload complete: noise-cancelled-audios/conversation_1.wav
Upload complete: noise-cancelled-audios/conversation_2.wav

```

```
[ ] output_file_paths = [
    "/content/conversation_1.wav",
    "/content/conversation_2.wav"
]

[ ] folder_prefix2 = 'noise-cancelled-audios/'

try:
    # Upload each output file to the 'noise-cancelled-audios' folder in S3
    for file_path in output_file_paths:
        # Use the basename of the file as the S3 object key
        object_key = folder_prefix2 + file_path.split('/')[-1]

        # Upload the file to S3
        with open(file_path, 'rb') as file:
            s3_client.upload_fileobj(file, bucket_name, object_key)

        print("Upload complete:", object_key)

except Exception as e:
    print(f"An error occurred:", e)

Upload complete: noise-cancelled-audios/conversation_1.wav
Upload complete: noise-cancelled-audios/conversation_2.wav
```

Accessing the NOISE-CANCELLED AUDIOS from the S3 BUCKET to perform SPEECH-TO-TEXT

```
a3_client = boto3.client('s3', aws_access_key_id=aws_access_key_id, aws_secret_access_key=aws_secret_access_key, region_name=region_name)

bucket_name = 'richtext-readback-test'
folder_prefix = 'noise-cancelled-audio'

try:
    # Use the S3 client to list objects inside the folder
    response = a3_client.list_objects_v2(Bucket=bucket_name, Prefix=folder_prefix)

    # Check if the 'Contents' key exists in the response
    if 'Contents' in response:
        objects = response['Contents']
        for obj in objects:
            print(obj)

    else:
        print("No objects found in the folder.")

except Exception as e:
    print(f"An error occurred:", e)

('Key': 'noise-cancelled-audio', 'LastModified': datetime.datetime(2023, 8, 3, 10, 56, 27, tzinfo=tzlocal()), 'ETag': '"$Id$09ff02021a980599ae78127e"', 'Size': 0, 'StorageClass': 'STANDARD'}
('Key': 'noise-cancelled-audio/conversation_1.wav', 'LastModified': datetime.datetime(2023, 8, 4, 7, 5, 13, tzinfo=tzlocal()), 'ETag': '"$Id$2c7b6d3f787f8fe93440d81-2"', 'Size': 873032, 'StorageClass': 'STANDARD'}
('Key': 'noise-cancelled-audio/conversation_2.wav', 'LastModified': datetime.datetime(2023, 8, 4, 7, 5, 14, tzinfo=tzlocal()), 'ETag': '"$Id$ff934a3ea1bb57bd1ed73dd"', 'Size': 7950030, 'StorageClass': 'STANDARD')
```

Uploading the SPEECH-TO-TEXT TRANSCRIPT into the S3 BUCKET

```
if 'Contents' in response:
    objects = response['Contents']
    for obj in objects:
        object_key = obj['Key']

        # Skip the folder entry
        if object_key == folder_prefix:
            continue

        print("Downloading object", object_key)

        # Use the S3 client to get the object data
        object_response = a3_client.get_object(Bucket=bucket_name, Key=object_key)

        # Serialize the data from the response
        object_data = object_response['Body'].read()

        # Process the object data as needed
        # For example, save it to a file
        with open(f'transcript/{object_key}', 'wb') as file:
            file.write(object_data)

    print("Download complete", object_key)

else:
    print("No objects found in the folder.")

Downloading object noise-cancelled-audio/conversation_1.wav
Downloading object noise-cancelled-audio/conversation_1.wav
Downloading object noise-cancelled-audio/conversation_2.wav
Downloading object noise-cancelled-audio/conversation_2.wav
```

```
output_transcript_file_paths = ['content/conversation_1.txt', 'content/conversation_2.txt']

folder_prefix2 = 'transcript/'

try:
    # Upload each output file to the 'noise-cancelled-audios' folder in S3
    for file_path in output_transcript_file_paths:
        # Use the basename of the file as the S3 object key
        object_key = folder_prefix2 + file_path.split('/')[-1]

        # Upload the file to S3
        with open(file_path, 'rb') as file:
            s3_client.upload_fileobj(file, bucket_name, object_key)

        print("Upload complete:", object_key)

except Exception as e:
    print(f"An error occurred:", e)

Upload complete: transcript/conversation_1.txt
Upload complete: transcript/conversation_2.txt
```

MICROSOFT AZURE

Microsoft Azure is a cloud computing platform offered by Microsoft, encompassing a multitude of cloud services. For this project, we tried utilising the azure's speech services to achieve noise cancellation and speech to text recognition for our conversational audios.

However, we could not yield the intended results as we encountered challenges. Specifically, we faced difficulties in generating viable output audio that effectively eliminated background noise. Additionally, the speech-to-text recognition capabilities did not align with our project's requirements, wherein we wanted to achieve a conversational dialogue format.

Given these limitations, our course of action shifted towards an alternative approach. We opted to harness the capabilities of Python libraries, enabling us to address the same objectives with favourable outcomes. Thus, our efforts were successful in accomplishing this goal.

Developing APIs using Python is a common practice, and one efficient web framework for this purpose is Flask. Flask is a robust Python web framework specifically designed for creating APIs. It provides a streamlined way to design and implement APIs, making it a popular choice among developers.

Once you've built your APIs using Flask, testing them is crucial to ensure they work as intended. A powerful tool for API testing is Postman. With Postman, you can interact with your APIs by providing inputs and checking the outputs directly from the backend. This allows you to validate the functionality and correctness of your APIs before deploying them for actual use. Postman simplifies the testing process, making it an essential tool in the API development workflow.

CONCLUSION

In conclusion, my technical internship experience has been a transformative journey that has significantly enriched my skills, knowledge, and perspective in the field of Data Science. This project allowed me to apply theoretical concepts to real-world challenges. Via this involvement, I have not only deepened my understanding of complex technical concepts but also improved my collaboration skills.

Throughout the internship, I encountered several challenges, such as getting acquainted to the use of cloud computing platforms, which initially seemed daunting. However, with the guidance of my mentors and the support of my colleagues, I was able to navigate these challenges and turn them into opportunities for growth.

Moreover, collaborating with my team members provided insights into effective, communication, teamwork and project management practices, which I consider invaluable assets for my future endeavours.

Looking ahead, I am excited to leverage the skills and knowledge gained during this internship as I embark on the next phase of my academic and professional journey. I am confident that the lessons learned and the accomplishments achieved during this internship will propel me forward in achieving my goals.

I am deeply grateful for the guidance and support of my mentors, colleagues, and the entire CYIENT team. This experience would not have been as enriching without their continuous mentorship.