Hackathon
Voice Commands Based Ordering System

This hackathon has been designed to help you practice, reinforce, and apply various concepts learned so far.

Objective:

Upon successful completion of this Hackathon, you will integrate a classifier, which can classify the voice commands, into a food ordering system.

Details

Understanding the data:

The data contains voice samples of (classes) ‘Zero’, ‘One’, ‘Two’, ‘Three’, ‘Four’, ‘Five’, ‘Six’, ‘Seven’, ‘Eight’, ‘Nine’, ‘Yes’ and ‘No’. Each class is denoted by a numerical label from 0 to 11 in the following way.

Class of samples	Labels
Zero	0
One	1
Two	2
Three	3
Four	4
Five	5
Six	6
Seven	7
Eight	8
Nine	9
Yes	10
No	11

The audio files collected in a Studio dataset consist of very few noise samples whereas the audio files collected in a Noisy dataset contain more noise samples. In both datasets, noise and speech are mixed and are in .wav format.

The audio files collected in the studio and noisy data are saved with the following naming convention:

- Class Representation + user_id + sample_ID (or n+sample_ID)
- For example, The voice sample by the user b2, which is “Yes”, is saved as 10_b2_35.wav. Here 35 is sample ID
- The ‘10’ that you see above is the label of that sample

Setting up:

In the current hackathon on the Voice-based Food ordering system, we will deploy the code on the server by placing the trained model from Google's Colab. The user id and password for the server login will be provided for each team by the mentors in the lab, for connecting to the server. Please find the steps below to connect to the server. Once you have connected to the server, you will find the files related to the hackathon. If you want to make any changes to the existing files or to add new data you need to use FileZilla and upload them in the corresponding folders.

Caution:

- You are allowed only to operate on the folder "Hackathon-setup". It is neither recommended nor needed to make multiple copies of any of the files or folders.
- Training should only happen in google's colab, Server is strictly meant only for application deployment and team's data collection only

For detailed instructions to set up the server and to access the file system use the document **"2-Server Access and File Transfer for Voice-based food ordering system".**

Understanding the Application:

The Web application for the Voice based food ordering system can be accessed with **"3-Hackathon 1 Application Interface Documentation"** shared in your drive.

The entry screen asks you to provide the Group Id. The Group Id that you will enter here will be the same as the one you used to login to the application. After that you will find two options:

1. **Record Sample data:** This leads you to a screen to help you to collect the data. The data recorded here will be saved in the **"./Hackathon-setup/team_data"** with the same naming convention provided in the data section. This data will be useful to you to fine-tune the classifier
2. **Order Food:** This application takes user response, when uploaded it will call the classifier of that group deployed in the server and returns a response

For more details: Kindly check the Document

How and where to Train?

You will be provided with Studio_data and Noisy_data (voice samples). You need to train your model on this and further you need to enhance them using your own team's data to be collected from the application. These trained models should be saved and deployed on the server. All the training will happen in Google Colab. Starter code for this purpose along with the instructions is provided to you in the shared google drive folder for experiments.

Deployment of your model on the application:

The files that you need to change to deploy your model are present in the **"./Hackathon-setup"**

- Upload the model into the server using FTP to **"./Hackathon-setup"**
- You need to change only the methods take_user_input and classify_input in Order.py
- take_user_input: load the trained model in this method and it will be passed to the classify_input method
- classify_input: Using the model passed from the take_user_input method predict the label for the features of the audio data. This should return predict label and confidence measure (in the range of 0-1) for that prediction

Problem statement:

- Please find the problem statement in "Hackathon1 - Voice Food Ordering System.ipynb" shared in your drive