

# Hackathon – II

## Voice Commands Based Ordering System

This hackathon has been designed to help you practice, reinforce and apply various concepts learned in Module - 1,2.

<b>Objective:</b>	<b>1</b>
<b>Details</b>	<b>1</b>
Understanding the data:	1
Setting up:	3
Understanding the Application:	3
How and where to Train:	4
Deployment of your model on the application:	4
<b>Problem statement:</b>	<b>4</b>
Stage 1: Loading data (On colab)	4
Stage 2: Training classifier on the studio_data (On colab)	5
Stage 3: Deploy your classifier (on server)	5
Stage 4: Train your model on Noisy_data(on colab) and deploy your classifier (on server)	5
Stage 5: Collect the voice samples and refine the classifier trained on noisy_data, by now using your team's data	6

## 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 of class is denoted by a numerical label 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 recorded in the studio are saved with the following naming convention:

- Class Representation + user\_id + 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

The audio files recorded in a noisy environment with built-in noise cancellation microphone are saved with following the naming convention:

- Class Representation + “n” + user\_id + sample\_ID
- For example: The voice sample by the group g3, which is “no”, is saved as 11\_g3\_38.wav. Here 38 is sample\_ID
- The ‘11’ that you see above is the label of that sample

## Setting up:

In the current hackathon on 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. 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 filesystem use the Document "[Server\\_Access\\_and\\_File\\_transfer.pdf](#)"

## Understanding the Application:

The Web application for the Voice based food ordering System can be accessed with this [link](#). The entry screen asks you to provide the Group Id. The Group Id that you will enter here will be 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 deploy in the server. All the training will happen in the 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”**

- You need to change only the methods take\_user\_input and classify\_input
- Upload the model into the server using FTP to **“./Hackathon-setup”**
- 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:

### Tasks:

### Stage 1: Loading data (On colab)

- The larger goal is to train your model on different types of voice data (such as clean studio data, noisy data and finally your own data)
- In order to do that, you should extract features from the data
- Evaluation Criteria (**Max. Marks: 5**)
  - Load the data and get features (refer colab notebook)

## Stage 2: Training classifier on the studio\_data (On colab)

- The goal here is to train your model on voice samples collected in a noiseless studio setup
- Evaluation Criteria (**Max. Marks: 15**)
  - Train the classifier, save the model
  - The score you get: Validation accuracy percentage of 15 (Validation data should be at least 20% of the total data)
  - Example: If a team gets, 80 % accuracy on the validation set, then the marks will be 80% of 15 marks i.e. 12 marks (will round of the score, in case of non - integer scores).
  - and deploy the (refer colab notebook)

## Stage 3: Deploy your classifier (on the server)

- Deploy your model on the server, check deployment section in the same document for details
- Evaluation Criteria (**Max. Marks: 5**)
  - There are two stages in the food ordering application
    - Ordering Item
    - Providing the number of servings
  - If both the stages are cleared with correct predictions you will get complete marks
  - Otherwise, no marks will be awarded

## Stage 4: Train your model on Noisy\_data(on colab) and deploy your classifier (on the server)

- The goal here is to train your model on voice samples collected in a noisy environment and save the model
- Evaluation Criteria (**Max. Marks: 5**)

## Stage 5: Collect the voice samples and refine the classifier trained on noisy\_data, by now using your team's data

- The goal here is to refine the model that you trained on voice samples collected in a noisy environment
- You will refine your model trained on noisy\_data, save and download it.
- Deploy your model on the server, check deployment section in the same document for details
- Evaluation Criteria (**Max. Marks: 10**)
  - There are two stages in the food ordering application
    - Ordering Item
    - Providing the number of servings
  - If both the stages are cleared with correct predictions you will get complete marks
  - Otherwise, no marks will be awarded