Code description

Python Files present:

1. Record audio.py
2. Order.py

Record_audio.py:

1. **get_features(): (Code is fully implemented/given)**
   - **– Parameters:**
     - filepath: path of the audio file
     - sr(samplingrate = 8000) : for all the recordings given and newly recorded audio files
     - using the program are recorded with a sampling rate of 8000
     - n mfcc: 30
     - n mels: 128
     - frames: 1
   - **– Details:**
     - returns deep learned features, given the file path of the audio
2. **get_network(): (Code is fully implemented/given)**
   - **– Parameters :** NA
   - **– Details:**
     - This function returns a deep neural network for deep features used in the get features() function
3. **record voice() : (Code is fully implemented/given)**
   - **– Parameters:**
     - Username: an id given to a person
     - j: label of the sample
     - v: number of the sample recorded by a person with a particular ID
   - **– Details:**
     - This function when invoked records the audio from the system's default microphone for a second

4. play audio():(Code is fully implemented/given)

– Parameters:

- Path: path of the audio file to be played

– Details:

- This function plays an audio (.wav) file given it path

5. Plotchart():(Code is fully implemented/given)

– Parameters:

- Objects: list of elements on X-axis
- Confidence: list of confidence values respectively for corresponding objects

– Details:

- Plots bar chart for each and every sample vs its latest confidence metric

Order.py: (Code partially implemented/ training classifier has to be coded by you)

Class order ():

Methods:

1. **classify input(): (You have to Complete this function so that it returns predicted label and confidence measure in the same order)**

   Parameters:

   - Define your own parameters, as per your classifier model

   Details:

   - While ordering the food, this method has to classify the input voice sample

2. **confirm input(): (Code is fully implemented/given)**

   Parameters:

   - digit: predicted label for the given voice sample

- confidence: confidence measure obtained from the classify input() function's output
- flag : to track the menu and quantity

3. take user input(): (Code is fully implemented/given)

Parameters:

- list task
- flag

Details:

- It prompts for the menu/quantity, records the value using record voice function in Record audio.py files and extracts features from getfeatures() function in Record audio.py files
- By passing these extracted features to classify input() method, you have to predict the label and obtain a confidence measure
- Then the confidence measure and predicted digit are passed to confirm input() function to check whether the confidence measure is high enough. If the confidence is high enough and predicted label is valid then this function returns the menu choice and label, otherwise calls the take user input() again to start the whole process once more