SmartHear - Intelligent Hearing for Android

Final Project Report

Submitted on **09th May 2016**

Group 1:

1. **Ragunandan Rao Malangully (13)** 2. **Ravi Kiran Yadavalli (31)**

I. Introduction

This document intends to provide an overall description of the project named "Smart Hear" in detail. The project schedule and the plan of action is also discussed. The proposal document submitted would give an insight about what the project is about. Audio detection and analysis is one of the major fields of research that is yet to reach its fullest potential. A subset of this research is to aid the people challenged in hearing sounds. Hearing dog is one such solution that notifies the user on hearing any sound. With the growth in the audio data size and the need to perform real time analysis on the streaming audio, it is very puzzling as to why it is an unexplored area on the big data platform. These concepts were the motivation behind our goal of aiding the challenged users using an audio detection and analysis application using the big data tools. The focus of the study was to evaluate the various machine learning algorithms performance in terms of audio signal detection and analysis with respect to varying features used for analysis.

II. Project goal and objectives

Overall goal

The overall goal is to provide a hearing aid through the use of the smart phone that is accessible to every person today. The smart phone can be utilized in a way that it can act similar to the hearing dog that is what motivated us to take up this idea and try to implement using the smart phone.

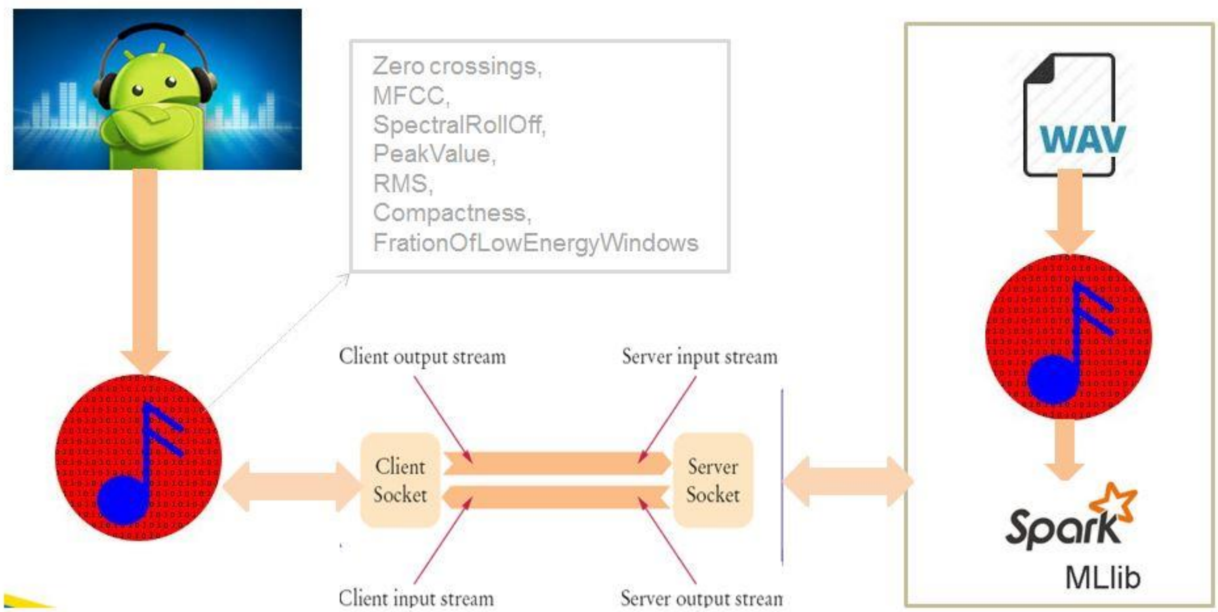
This application can be used to provide benefits to the special abled people who face challenges in listening to sound. All the features come at the cost of installing an application that is freely provided to the user. No costs attached.

Objectives

- To provide a smart hearing aid to the user.
- To implement features to provide the user flexibility in varying certain key features of sound like frequency, tempo etc.
- To provide notifications to the user when there are important events like a door bell ring or an alarm that goes off.
- To make sure that the context is recognized and the user settings are changed accordingly.
- To ensure that the response from the server is at acceptable limits and is accurate.
- To present the ideal set of features for better accuracy of the audio class detection.
- To evaluate the various machine learning algorithms and pick the best one for audio detection and prediction of audio class.
- Provide user customizable settings that the user can adjust to suit his or her needs.
- Ensure that the application can deliver its functionality at minimal cost of operation.

III. Architecture & Application

The architecture of the application is pretty simple. We have an android client that records the audio sounds and sends the audio to JAudio library for feature extraction. The JAudio library extracts the seven features mentioned in the architecture and also adds the contextual information. This information is then sent to the spark server engine through a socket connection. The spark server is trained with .wav audio files and real time data from the device. The .wav files are sent to the JAudio library for feature extraction. These features are then used to train the model at the server end. Trained models are then saved to files. Each context has a different model that is trained and saved for that context. Based on the context information a model for prediction is selected. Then the features sent from the client are used to predict the audio class. This information then is sent back to the android client through a socket connection. Based on the information sent from the server the android client creates a notification for the user to alert him regarding the audio event.



Machine Learning (Algorithms)

- Real time dynamic model training by collecting features using Spark MLlib from client side feature set.
- Multiple models have been built based on several parameters like efficient set of feature set, machine learning model, number of classes.
- Server client connection established through socket connection.
- The serval experiments conducted among Naive Bayes, Random Forest and Decision Tree algorithms with varying number of features.

IV. Data

In the training phase of the application twenty audio classes were used in the model. Each model had fifteen audio files for training and validation purpose. To improve the accuracy and to add the real time nature to the application some real time data collected from the device was also provided to the model as training data. Context aware systems are a component of ubiquitous computing or pervasive computing environment. In our current work, we have devised four important contexts based on the geographical prevalence.

- Home Context
- Classroom Context
- Outdoor Context
- Office Context

Design of context aware model captures important aspects in audio analysis such as a noise level and mining and training of sounds in location specific model. Our experiments have proved that a context aware model has 2.19 % more accuracy than an aggregation of all the contexts as a single general context.

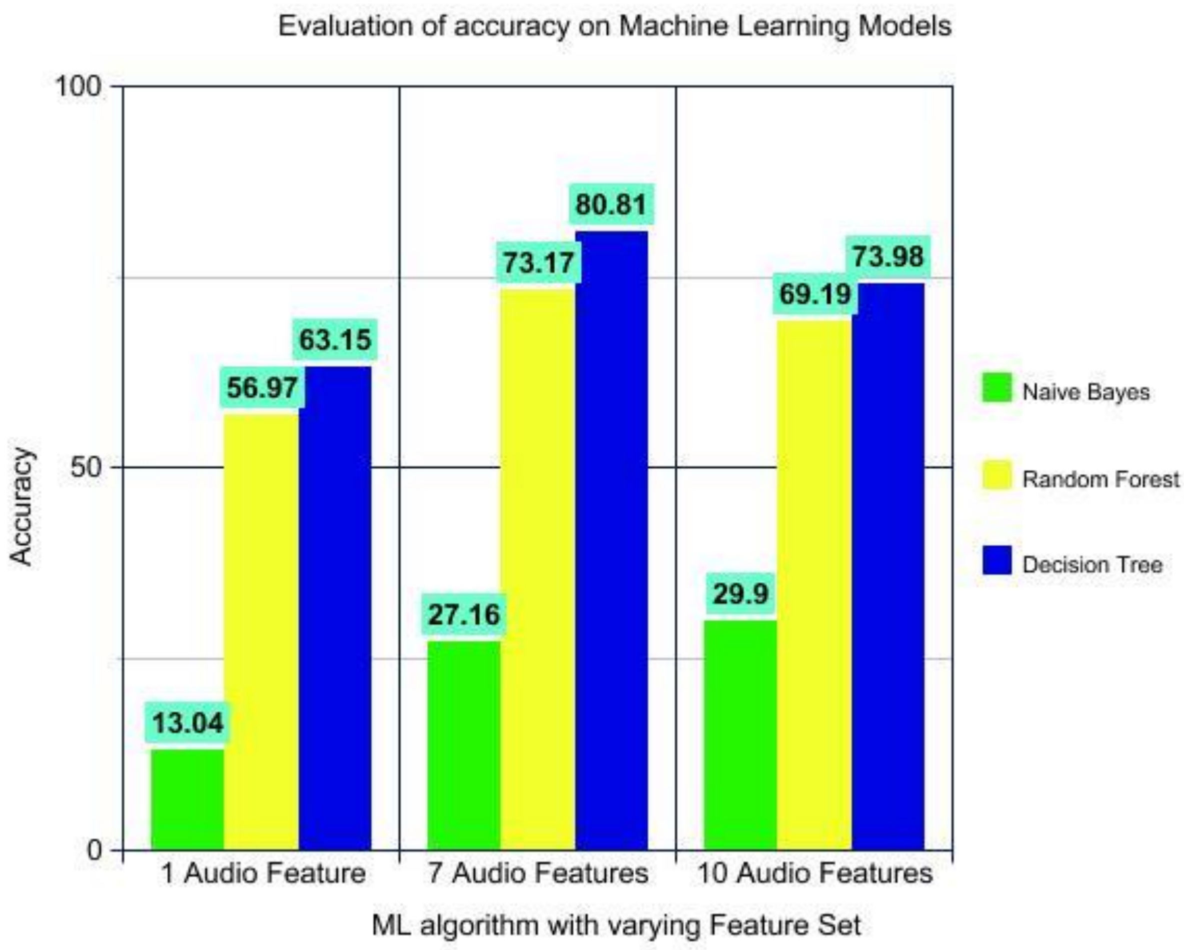
Feature Extraction

- Audio feature extraction is extracting properties, such as beat points, statistical summaries, along with many other less obviously useful properties.
- These properties can then be fed to machine learning toolkits (Here it is Apache Spark) to automatically extract properties (such as artist or genre) from unknown music.
- More interesting applications include prediction of sounds based on the location and redundant time-based daily activities or events.
- Based on the AudioRecorder and MediaRecorder API's provided by Android the audio byte data is captured and analyzed to extract feature set using JAudio library and send to spark for machine learning.
- The features used in the application are Zero crossings, MFCC, SpectralRollOff, PeakValue, RMS, Compactness, and FrationOfLowEnergyWindows as they resulted in the most accuracy for the analysis.

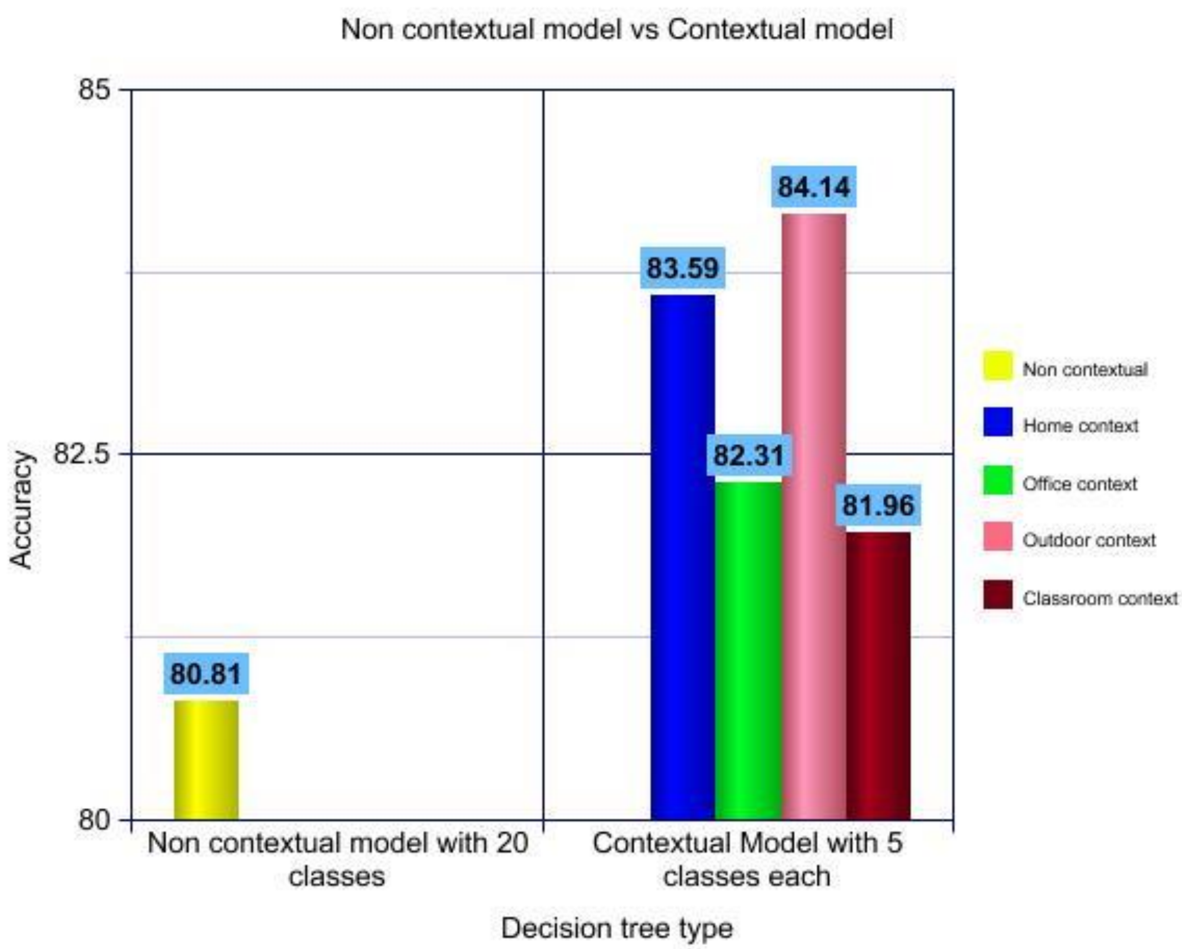
V. Evaluation and Accuracy

There were a series of experiments performed in the study of the application improvements. The following are the evaluation results and analysis for the experiments.

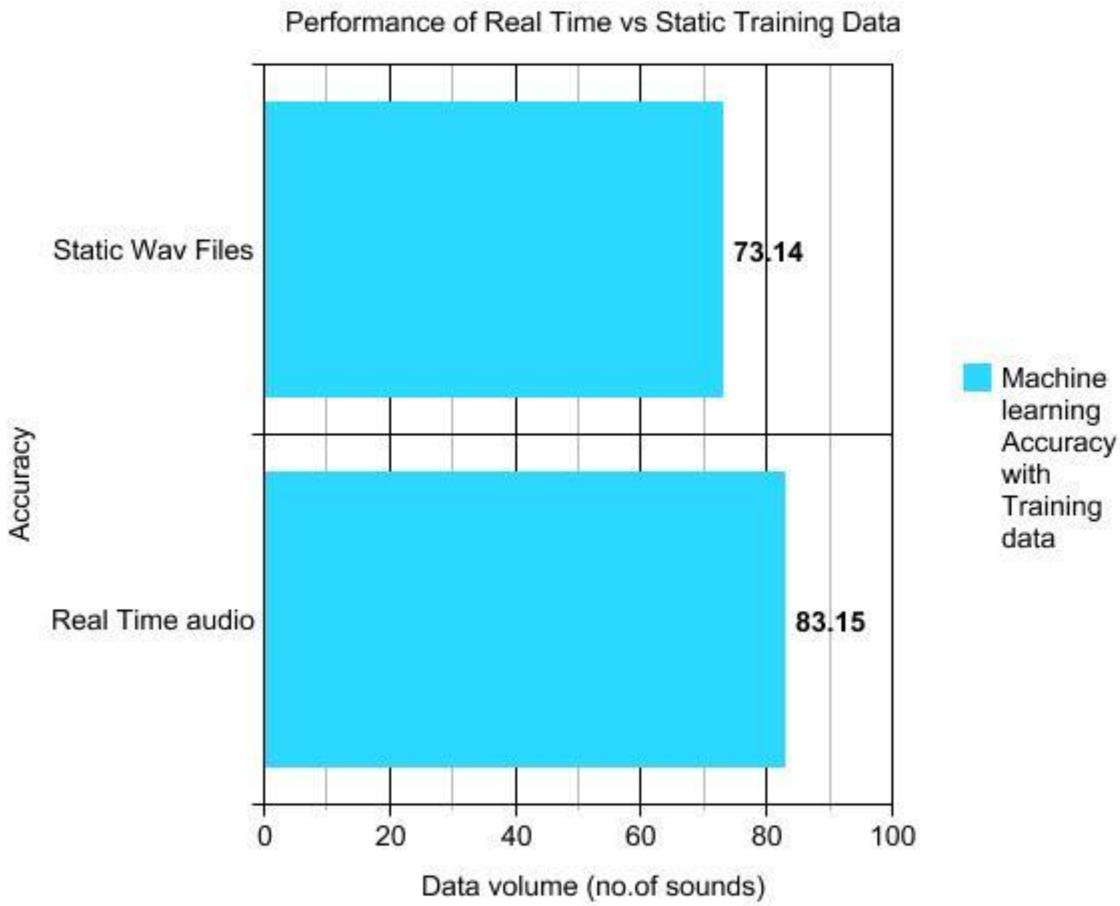
1. Finding the ideal features set for improving the accuracy of the prediction model. In this evaluation the focus was to find a match for the best model in terms of highest accuracy. At the same time varying the features set size to minimize the effort on the model while trying to increase the accuracy. The figure below presents the case that based on the comparison for accuracy for Naïve Bayes, Random Forest and Decision Tree algorithms with 1, 7 and 10 features respectively. From the figure it can be deduced that Decision Tree with 7 features (Zero crossings, MFCC, SpectralRollOff, PeakValue, RMS, Compactness, and FrationOfLowEnergyWindows) resulted in the best accuracy. So it was used in the application.



2. Comparison of the accuracy for Non contextual model versus the contextual model. In this exercise we had two models, one where there was no context information for the model. There was only a single model with 20 classes and training data for all the 20 classes. On the other hand we had 4 different models each for one of four contexts. Each contextual model had 5 classes each and training data was limited only to these 5 classes for each model. Then we compared the accuracy of non-contextual model vs each of the contextual model. As the following figure depicts the contextual models outperform the non-contextual model as the contextual models each have limited classes to predict which in turn improves accuracy of the model.



3. In this experiment the focus was on the study of the accuracy of the model prediction with only static data in training versus model prediction of model with real time data included in training. It was clearly evident that with static data in training the accuracy of the model was only 73 percent as audio data varies with a lot of factors such as noise, context etc. When real time data was also added for training of the model there was an increase in the accuracy to 83 percent as most of the real time data captured the variance of the audio data in terms of noise and other factors. This is depicted in the figure below.



VI. Runtime Performance

The application had acceptable performance in terms of the run time required for the entire work flow. In terms of the model training and model saving the run time required was 20 minutes and 34 seconds for the non-contextual model. While it around 6 minutes for each of the four contextual models. The runtime for one cycle of audio recording, feature extraction, prediction of audio class and notification to the user was as following.

- Audio recording – 4 seconds
- Feature extraction and audio class detection at server – 3.42 seconds.
- Response from server to client and notification to the user – 1.89 seconds.

VII. Future Work

The current model predicts the sound class in a particular context from a set of four contexts and 5 classes each with 83.00 % accuracy on average from several experiments conducted from thousands of .wav sound files and over 250 hours of real time audio data.

- Based on the current analysis, the audio features apparently vary from device to device and the noise levels. Arriving at a common model for prediction for all the devices.
- Improving the accuracy of the model with more classes and dynamical addition of data for training of the model.
- The challenge ahead of us is to devise a model to handle device configuration dynamically and adjust the noise levels accordingly.
- Collaborative learning from multiple devices to improve the prediction of model.
- Automatically detecting the user context and efficient usage of the battery for optimal performance while saving the energy consumption.

VIII. Related work(References)

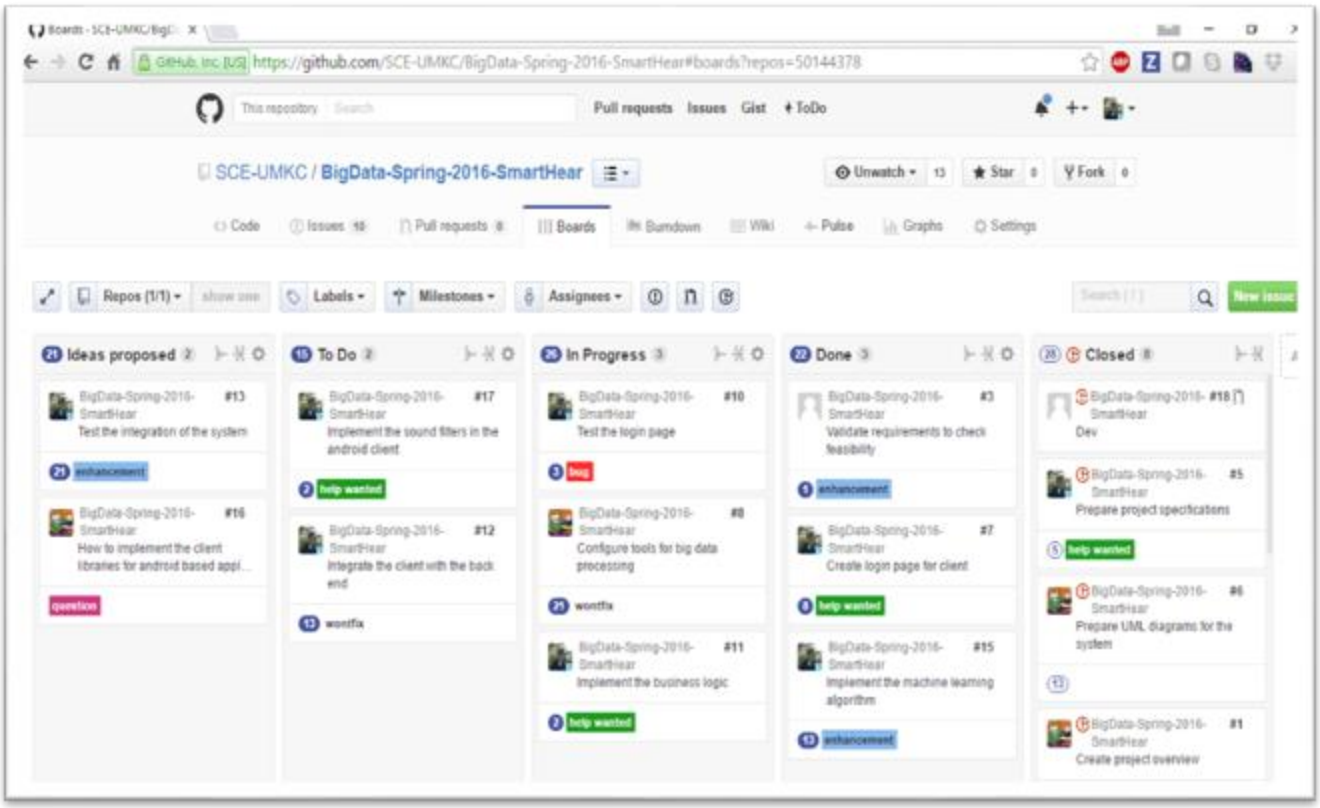
- https://play.google.com/store/apps/details?id=mg.locations.track5&hl=en
- https://play.google.com/store/apps/details?id=com.fsp.android.friendlocator&hl=en
- http://www.raywenderlich.com/120177/beginning-android-development-tutorial-installing-android-studio
- http://developer.android.com/tools/building/building-studio.html
- https://usa.bestsoundtechnology.com/ces/#
- http://www.hearingreview.com/products/new-product-technology/
- http://www.resound.com/en-US/hearing-aids/resound-linx2#.Vse5ZPkrLcs
- http://jmir.sourceforge.net/publications/ISMIR_2005_jAudio.pdf

IX. Project Management

The project was managed mainly as two parts. The client side part and the server side part. As there were two members in the team the work was divided equally for both the members. Ragunandan Rao was responsible for development and testing of the client side module. While Ravi Kiran was responsible for the development and testing of the server side module. The ZenHub was used for creating milestones for each increment and the issues were assigned to each team member. This was used for tracking the contribution. Even the documentation was divided into parts for contribution. PFB the screenshot of ZenHub for the management throughout the project.

Ragunandan Rao Malangully – 50 percent

Ravi kiran Yadavalli – 50 percent.



X. Final project evaluation

The project completed has been a very good implementation of the inception concept. Most of the requirements that were drafted were implemented. The design of the project was not straight forward. The use of agile process really helped the cause as we had to change some of the design policies intermediately. This would not have been possible if agile process was not in place. Agile process is definitely something that we would practice for every project that we would take up. The schedule of the project was not fully practiced as there were some points in the timeline where the required amount of work or goal was not achieved. This caused some alterations in the work schedule but in the end things fell into place. Mostly team meetings weekly was the main source of the project progress and management discussion. As a team we would be updating each other weekly regarding the previous week work completion as well as goals for next week. Also we had the ZenHub for our reference. We did have conflicts on how we can divide the work for each team member but once we had a discussion within the team the problem was solved. For the project the most time consuming part was to figure out to use the best model and to extract features. We believe a little more study on the topi c of the project before taking up the task would really help the cause. We believe if this was a real world project then there would be more refinement in terms of the flow of the project like preprocessing of the audio signal before feature extraction. Improvement of the UI. More customization options. Also we could have seen a mixture of models for prediction of audio class. The recommendation for next year would be that if the topics of interest were provided to the students then it would be better. As w e are of the view that students would be having a limited scope in choosing the topic for their project. This would limit the functionality of the project and also the application would not give an impression of solving real time challenges.