CSCE 606: Software Engineering

Team Codezilla: Final project report

1. **Two paragraph summary of the project as implemented, including the main customer need and how the application meets it, including who the stakeholders are. This will contrast to what you wrote in Iteration 0.**

   The customer has 20 second video samples for which they need to generate "transitional bilingual observation protocol" reports which includes teaching strategies, curriculum, physical group, activity structure, mode, language content, and language of instruction. These reports are used to analyze the quality of education. Since it takes time to manually make these reports, the customer suggests using machine learning to experiment with making some part of reports generated automatically. We intend to experiment with this idea and do a proof of concept for the same.

   Our first step was to study the feasibility of the project. Out of the many features in the proposal, we have shortlisted two features, namely "**Physical group**" and "**Language of Instruction**" for creating a POC. The meaning of the two terms is explained below:

   Physical Group: System would be using object detection on the pre-recorded videos to distinguish between a small v/s large group of students involved in a certain activity. For separating small groups from large, a predefined limit for the number of students will be defined in the system.

   From the analysis of our object detection model we are able to detect students in the video, but we found this task to be infeasible for the TBOP due to:

   1) Camera is not fixed, significantly improving the complexity in understanding the number of students in a classroom.
   2) Clustering students into groups of 3 or 4 is very challenging. This can't be achieved without a proper truth value in the given dataset which can't be determined from video.

Then most important we can't distinguish between teacher to student without post analysis which cannot be scaled to every video

Language of Instruction: Using voice recognition on the pre-recorded videos, the system will update the language of instruction for the lecture.

Based on the feasibility of implementation under given time constraints, we have decided to work on the “Language of instruction” aspect of the project. The idea of the implementation of the project is as below:
As per the current data, we have two modes of instruction for the classroom: English and Spanish. We can run a text-to-speech recognition model on the videos to obtain the preliminary data. (Link) Post that, we can run another language-classification model to identify the language. (Link)

Some of the APIs are paid and don’t have much free trial available. We have used open source libraries.

Stakeholders of our project:

- Dr. Irby,Beverly
- Dr. Guerrero, Cindy L
- Dr. Robideau, Melanie L
- Dr. Lara-Alecio, Rafael
- Dr. Fuhui Tong
- Dr. Sutton Jones, Kara L
- Dr. Etchells, Matthew
- Dr. Costello, Yvonne M

2. **Description of all user stories (including revised/refactored stories in the case of legacy projects). For each story, explain how many points you gave it, explain the implementation status, including those that did not get implemented. Discuss changes to each story as they went. Show lo-fi UI mockups/storyboards you created and then the corresponding screen shots, as needed to explain to stories.**

   Iteration 0 User Story - Data Analysis - 2 Points
   Feature: Analyze the data provided by the clients. Identify the potentially achievable features using machine learning for one of the above categories.

   - We want to identify one potential feature in the data which can be selected to automatically fill in the TBOP.

Initially we decided to work on two features for this POC - “Language of Instruction” and “Physical Group” but to implement the later feature, object detection was required. But with the camera being fixed distinguishing between teacher and student without post analysis was not possible within the given data and the time limit. So a conscientious decision was made by the team to focus on the “Language of Detection” feature as an initial step in this project.

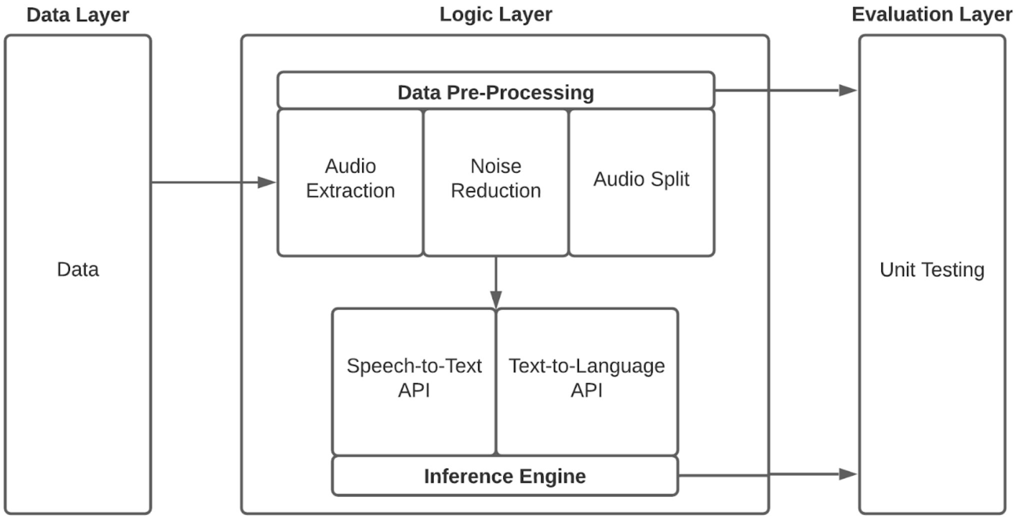
The story was fully implemented in this iteration.

Iteration 1 User Story - Model Development, Initial phase - 3 Points

Feature: Based on the feature selected, process the data which included cleaning of the data and converting the raw video file into the correct format for the ML model.

- Independent models were created for extracting the audio, splitting into a required number of audio files and decision model which makes API calls to Audio to Text google API and Text to Language Python API.

This story has been implemented for this iteration. Individual pieces of code have been developed and Unit testing has been done for them



Iteration 2 User Story - Model Development, integration and testing - 2 Points

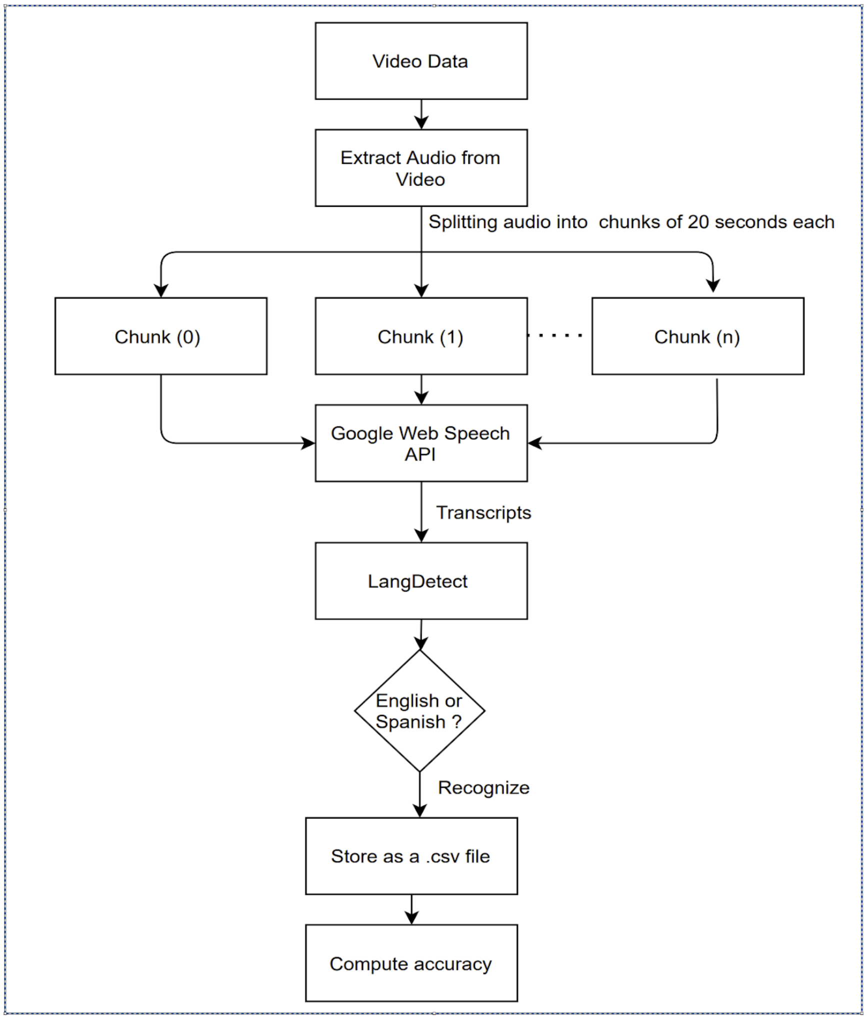
Feature: Apply the models involved in audio/video processing to classify language/physical groups.

- Integration of the separate modules were done to form a pipeline which takes in the input video file and outputs the encoded video name, timestamp and the detected language which then gets saved in the csv output file.

Feature: Perform post-processing analytics to the outcome of the ML model to provide relevant results as feedback

- Post-processing involved outputting metrics to analyse the working of the pipeline.The metrics include the overall accuracy of the detected languages, the accuracy of predicting english and the percentage of english in the video.

Flow of code:



The story was fully implemented for this iteration.

Note regarding lo-fi mockups/storyboards/screenshots:
The project would be based solely on a Machine Learning model to observe and analyze the data provided by the customer and aim to improve the performance measurement **without the need** or requirement of any **User Interface.** This has been mentioned in the custom grading approach in all our preceding reports. As such, we do not have any lo-fi mockups or user-interfaces built for this project.
We did not change anything for the user stories and each story has been completely implemented.

3. **For legacy projects, include a discussion of the process for understanding the existing code, and what refactoring/modification was performed on the code, in addition to the user stories listed above.**

   Not applicable. Not a legacy project.

4. **List who held each team role, e.g. Scrum Master, Product Owner. Describe any changes in roles during the project.**

   Scrum Master & Developer: Rohan Mahesh Jagiasi
   Product Owner & Developer: Aishwarya Teegulla
   Developer: Sunil Krishna Kumar Komadam
   Developer: Sanjiti Bhargava
   Developer: Spoorthy Kanduri
   Developer: Zaid Pervaiz Bhat
   Developer: Samiksha Marne

   There were no changes in the roles of the team throughout the iterations.

5. **For each scrum iteration, summarize what was accomplished and points completed.**

   Iteration 0:(Completed 2/2 points) For this iteration, we studied the feasibility of the project. Out of the many features in the proposal, we shortlisted two features, first "Physical group" and the second "Language of Instruction" for creating a POC. As the data wasn't available for this iteration, a decision was made to select one of the features for initial experimentation based on available data in the next iteration.

   Iteration 1: (Completed 2/2 points) Based on the data provided, all the team members decided to choose the "language of instruction" feature for the POC and worked together to perform the following tasks:

   1. Extract Audio from Video
      It takes in a video file as input and extracts the audio stream from it and stores in the '.wav' format. We can optionally choose to specify the format of the desired audio file. The code was tested on the sample video files
   2. Split into 20 seconds
      It divides the extracted audio file in the '.wav' format into small chunks of 20 seconds each.The code was tried and tested on various audio files to determine its reliability

3. Remove Noise
   There are two ways to remove noise. One is by using the integrated features of Google API and mentioning an extra parameter in the next step.
4. The other is a command utility called SoX (Sound Exchange). In SoX, we can apply a low pass filter to reduce all the background noise below a certain threshold. Silent parts of the audio can also be clipped by specifying a certain energy level below which the audio will be removed. You can visually see the difference by plotting a spectrogram.
5. Use Audio to text API
   We used a Google Web Speech API for speech-to-text conversion. Each chunk of the audio file is provided to the API in the wav format and the output is obtained in the form of text transcript.
   The audio to text feature is implemented using the 'Speech Recognition' library. It provides us with a Recognizer class and each instance of this class offers us with seven methods for recognizing speech from an audio source (which is previously extracted from the classroom video recordings in our project). We are using Google Web Speech API for speech-to-text. The audio file is provided to the API in the wav format and the output is obtained in the form of text.
6. Use Text to Language API
   Once the transcript is generated in the above step, it is fed to a language detector library, "langdetect", to recognize English / Spanish.

Iteration 2:(Completed 3/3 Points) In this iteration, all the independent working modules in iteration 1 were integrated together which basically took the video file as the input and returned the encoded timestamp and the language which has been detected as output. Both the outputs were stored in a csv file along with the video filename.We then tested the end-to-end pipeline for all the sample videos provided.

After storing the above output in a csv file, we designed a testing framework that reads the above csv file stored and compares it with the ground truth encoded files provided by the client. We provide an output analysis illustrating the accuracy with which we are able to predict the languages correctly. We additionally calculated the percentage of English in video to satisfy the requirements of the TBOP.

6. List of customer meeting dates, and description of what happened at the meetings, e.g. what software/stories did you demo.

Iteration 0 :- In this iteration the client gave a basic overview of the project including the dataset and described the final objective of the project.

Iteration 1:- Potential achievable feature - “Language of Instruction ” and how we can possibly achieve the desired result for the POC was discussed.

Iteration 2:- We presented the complete integrated model, testing and the results of the whole pipeline and showed how certain accuracy is achieved for different videos when given as the input. The POC was very well received and appreciated by the client.

Customer interactions across all iterations:
Meeting 1: April 7th, 2021
Time: 4:15-5:00pm
Zoom Video Recording:
https://drive.google.com/file/d/1i4PA_DTs95A2XlI9OgbvI2tnWHx2Y7dt/view?usp=sharing

Meeting 2: April 13th, 2021
Time: 4:15-5:00pm
Zoom Video Recording:
https://drive.google.com/file/d/1wvgtwXFUtHjIVjcIESBdSj_Ctp53U5VG/view?usp=sharing

Meeting 3: April 20th, 2021
Time: 4:15-5:00pm
Zoom Video Recording:
https://drive.google.com/file/d/15LYUEf1I2PUWoQhOpw0pjrA5XOPjIS69/view?usp=sharing

Meeting 4: April 29th, 2021
Time: 4:30-5:15pm
Zoom Video Recording:
https://drive.google.com/file/d/15LYUEf1I2PUWoQhOpw0pjrA5XOPjIS69/view?usp=sharing

7. Explain your BDD/TDD process, and any benefits/problems from it.

The tech stack that our project uses is Python. Being a machine learning POC, our project’s tests have more to do with accuracy as compared to conventional BDD/TDD tests. We have tested the performance of our software against the data that the client has provided and have measured accuracy.

8. Discuss your configuration management approach. Did you need to do any spikes? How many branches and releases did you have?

Our project is a new POC. We do not have any legacy code/repo. As such, our software has a single version 1.0 at the moment. We have one master branch in the git repository where we are storing our code. Our project has fairly straightforward installation steps, most packages are easily installable using pip commands through the cmd, as such we did not require any exclusive config file.

9. Discuss any issues you had in the production release process to Heroku.

Our project is a new POC. The python program has to be run by using commands in the cmd and it has no user interface. Our deliverables towards this POC did not include deployment to Heroku. This was also mentioned in the custom grading approach for this project.

10. Describe any issues you had using AWS Cloud9 and GitHub and other tools.

We did not face any issues with respect to version control using git. We have created a repository to store our project code and documentation. Our code also includes an exhaustive readme.md file which lists the installation instructions, dependencies and tree structure of the project which makes it easy for anyone to clone and run the project at ease. Our project does not have cloud-based deployments, and the developers of our team have used IDE's like Pycharm/Spyder/VS code for development purposes instead of using an cloud IDE like AWS Cloud 9.

11. Describe the other tools/GEMs you used, such as CodeClimate, or SimpleCov, and their benefits.

MoviePy library It was used to extract the Audio from the Video - It takes in a video file as input and extracts the audio stream from it and stores in the '.wav' format. With this Library we can optionally choose to specify the format of the desired audio file.

Pydub library It divides the audio file into chunks of specified length. So the audio extracted in the above step was split into chunks of 20 seconds each.

SoX (Sound Exchange) library It applies a low pass filter to reduce all the background noise below a certain threshold. Silent parts of the audio can also be clipped by specifying a certain energy level below which the audio will be removed.

Speech Recognition Library It provides us with a Recognizer class and each instance of this class offers us with seven methods for recognizing speech from an audio source.

Langdetect Library The text to language feature was implemented using this library. It has better accuracy and also the inference time is very fast. This library returns the detected language of longer sentences.

12. **Make sure all code (including Cucumber and RSpec!) is pushed to your public GitHub repo.**

The working code has been pushed to the GitHub repo along with the instructions to install and run the project.

13. **Links to your Pivotal Tracker, public GitHub repo, and Heroku deployment, as appropriate. Make sure these are up-to-date.**

Github Repository: https://github.com/rjagiasi/TBOP
Pivotal tracker account: https://www.pivotaltracker.com/n/projects/2495399
As mentioned on the custom grading approach in each previous iteration, our deliverables towards this POC did not include deployment to Heroku

14. **Links to your poster video and demo video.**

Link to the poster :
https://drive.google.com/file/d/13X2TsCD8uS1NyCoNTa1Bcybny4dV_3fO/view?usp=sharing
Link to the demo/poster video:
Youtube video: https://youtu.be/yQdIHDuXR5I

15. **Integration & Testing:**

As shown above, individual modules were built and integrated as per the given flow using python.

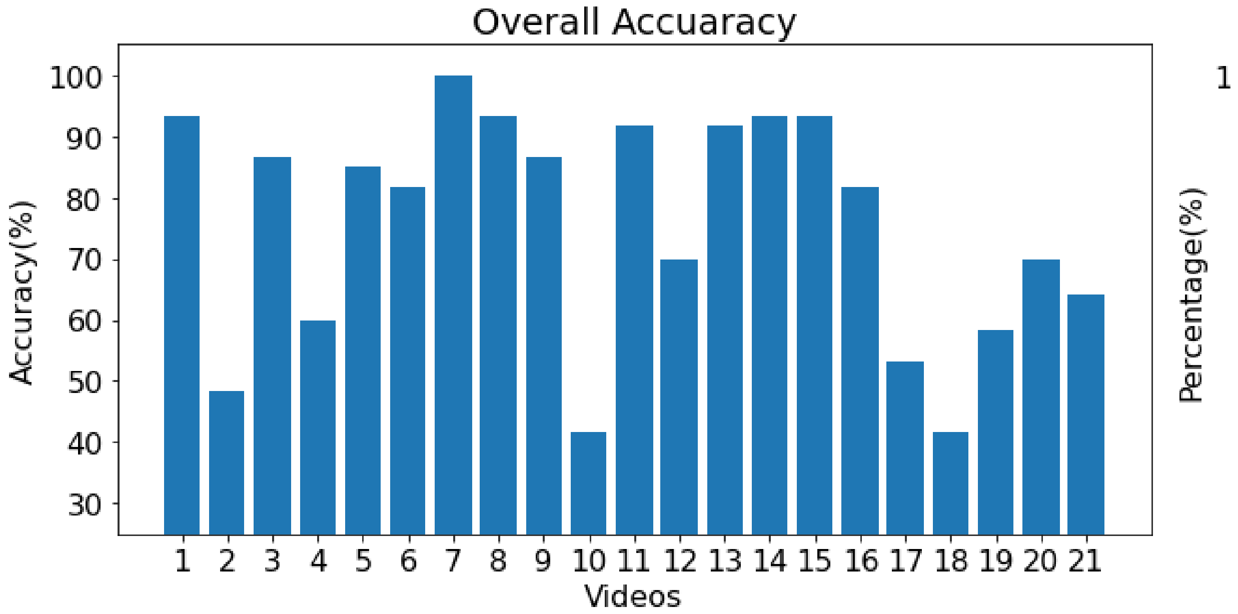
Unit Testing - Within each module, Independent functions have been written for unit testing.
Integration testing - Integration testing is done by executing the pipeline for the videos provided by the client.
Model Validation - Based on the output obtained by executing the software for each video, a CSV file with predictions is generated. This file is then compared with the file given by the client in order to calculate accuracy.

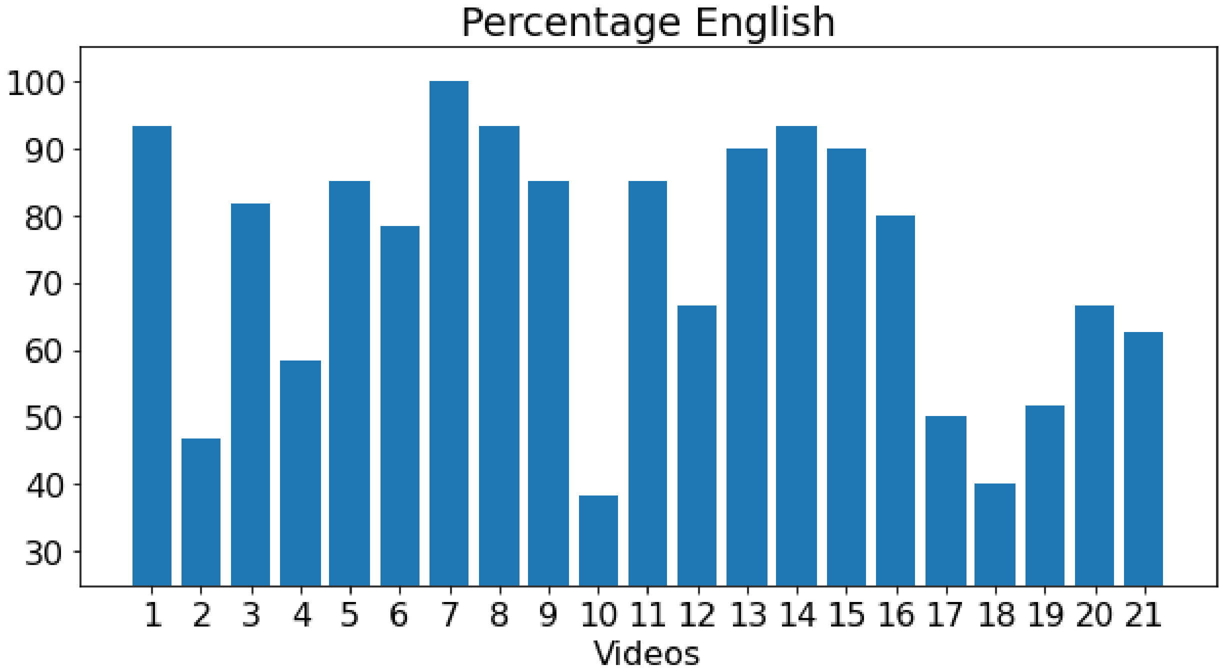
16. **Results** -

The model has been tested on label files provided by the client. The processing times vary anywhere from 1 to 2 minutes. We have compared the output of the code against the language of instruction data and have observed the following accuracy percentages:

Filename	English Accuracy	Percentage English Predicted	Actual English	Accuracy of the Code
10112_R2.asf	93.33	93.33	100.0	93.33
10812_R3.asf	50.0	46.66	93.33	48.33
11121_R2.asf	100.0	81.66	81.66	86.66
11711_R3.asf	66.03	58.33	88.33	60.0
20532_R1.asf	94.44	85.0	90.0	85.0
20532_R3.asf	88.67	78.33	88.33	81.66
30131_R1.asf	100.0	100.0	100.0	100.0
30421_R2.asf	100.0	93.33	93.33	93.33
40531_R1.asf	94.44	85.0	90.0	86.66
40631_R1.asf	51.11	38.33	75.0	41.66
41232_R2.asf	96.22	85.0	88.33	91.66
60331_R2.asf	80.0	66.66	83.33	70.0
60512_R2.asf	96.42	90.0	93.33	91.66
70232_R2.asf	96.55	93.33	96.66	93.33
70232_R3.asf	96.42	90.0	93.33	93.33
80222_R2.asf	85.71	80.0	93.33	81.66
90132_R1.asf	65.21	50.0	76.66	53.33
90312_R1.asf	47.05	40.0	85.0	41.66

91132_R1.asf	79.48	51.66	65.0	58.33
91132_R2.asf	93.02	66.66	71.66	70.0
91132_R3.asf	85.45	62.66	73.33	64.0





As we see, the accuracy of the language prediction is decent enough for most of the videos. An interesting observation is that the accuracy closely follows with the percentage of english in the video. It recognizes English really well but struggles in identifying Spanish, though it does recognize that Spanish parts are non English

For further analysis: We took a sample video completely in Spanish from Youtube to analyze the performance of the model on a Spanish video and we found it to be 78% accurate. Since it works decent on the Youtube video, we assume the quality of recorded audio is an issue here.

17. **Future works** -

Special audio boosting and noise removal mechanisms can be used in the pre processing phase which are generally used in signal analysis. In order to improve the model, a model designed with a smaller subset of languages can be chosen so as to better identify the audio.

Specific Natural Language Processing models can also be used extensively in order to improve the model as well as for the different other categories in TBOP sheet

18. Conclusion

The client expected a proof of concept application to evaluate languages in lecture videos and the same has been successfully implemented. We learnt that the English language works well with the existing model but the Spanish language needs some future work in the preprocessing phase. We learnt about the various libraries and frameworks used to process videos and audios. We also learnt about the various phases in Software Engineering and the importance of them in order to create a successful project. This project can be a good starting point for the eventual aim of education department to automate evaluation of recorded lectures.

19. **Acknowledgements** -

We would like to thank Dr Walker and Dr Beverly to give an opportunity to work on such an exciting project. The vision of the education department is to give realtime feedback to the professor teaching in the class, which would greatly improve the quality of students. This also gave us an insight about how the education team is striving to benefit and improve the life of students and teachers. This is a small contribution in the journey towards achieving it.