Quiet Zone

App prototype for noisy environment detection system with Machine Learning

Authors: Constantino-Daniel Boșcu, Mihai Alexandru Bizineche, Andrei-Daniel Boghici, Alexandru Dornea, Dominic Luca Ioan

Category:

COLEGIUL NAȚIONAL
“NICHITA STĂNESCU” Municipiul Ploiești

E-NNOVATE
International Innovation Show

Table of contents

I. Introduction

INTRODUCTION AND PROBLEM STATEMENT

II. Body

1. METHODOLOGY FOR THE SYSTEM AND AI MODEL

2. ANDROID IMPLEMENTATION - QUIET ZONE -

3. FUTURE DEVELOPMENT PLAN

III. Conclusion

OVERALL CONCLUSION OF THE PROJECT

I.Introduction

As a group of students who want to contribute to the well-being of the people, we entered this international competition to share our idea to the world and spread awareness about hearing, with the help of Machine Learning.

Earplugs are required · now
If not available, withdrawal from t...

Problem statement

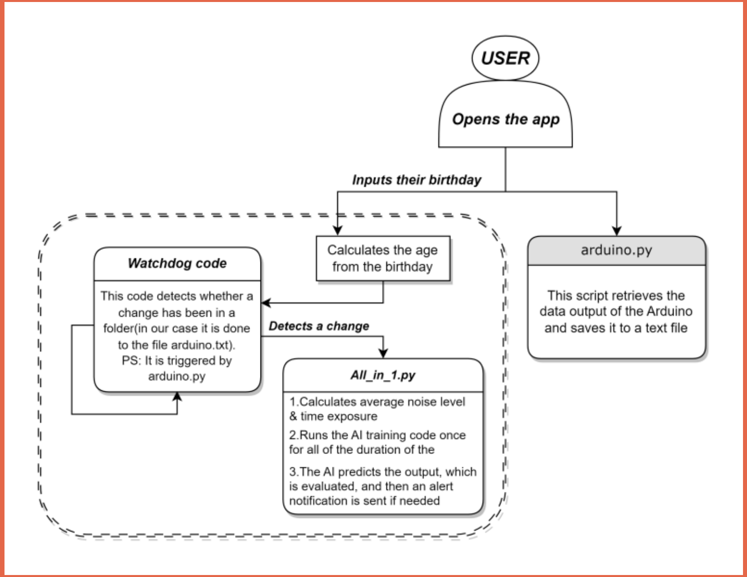
Some of the greatest "silent" threats of the 21st century are those things that make our lives better or more exciting: concerts on huge stadiums, heavy traffic of urban areas, and the roaring of machines in industrial shops or building sites. The result: hearing loss in younger and younger people.

Studies show that over 1 billion people are in danger of losing their hearing abilities. The worst part is that most of them will not do anything to preserve their hearing because they do not realise the decreasing of their auditory perception.

II.The architecture of the system

The program uses multiple scripts to work. Firstly, using a subprocess loop, dB values from the microphone mounted on the Arduino board is received through a txt file. Following, the program will open another subprocess. The user inputs their date of birth, calculating his age, and when the program detects a change in the txt file containing the decibel levels, it will calculate the average value of them.

If it is over 80 decibels, the AI is trained for the whole time this app open in the background and also, a timer will start. If the data processed by the A.I. shows that the environment is dangerous to your auditory system, it will send an alert notification



USER

Opens the app

Inputs their birthday

Calculates the age from the birthday

arduino.py

This script retrieves the data output of the Arduino and saves it to a text file

Watchdog code

This code detects whether a change has been in a folder(in our case it is done to the file arduino.txt). PS: It is triggered by arduino.py

Detects a change

All_in_1.py

1.Calculates average noise level & time exposure

2.Runs the AI training code once for all of the duration of the

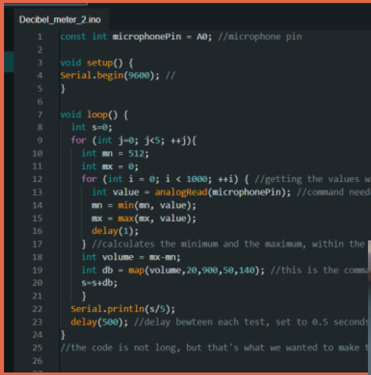
3.The AI predicts the output, which is evaluated, and then an alert notification is sent if needed

The system Arduino implementation

We used Arduino components to make a prototype, with simple hardware and efficient software.

The code sets a pin constant for the microphone, reads data from the sensor every millisecond for a second, calculates minimum and maximum, maps the value from an ARC value to a decibel value, repeats the process five times, and sends the result to a text file every 6 seconds, with little to no error.

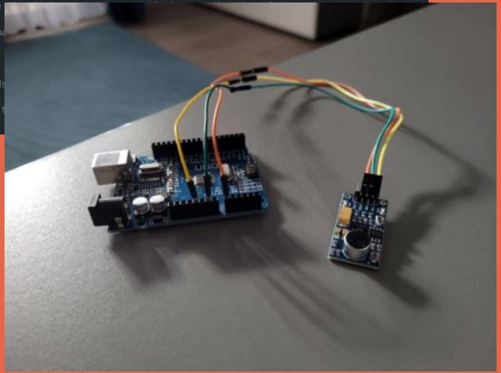
This information is then transmitted to the AI, which provides the diagnostic for the data.



```
const int microphonePin = A0; //microphone pin

void setup() {
Serial.begin(9600); //
}

void loop() {
  int s=0;
  for (int j=0; j<5; ++j){
    int mn = 512;
    int mx = 0;
    for (int i = 0; i < 1000; ++i) { //getting the values will take 1 second so that its as precise as possible
      int value = analogRead(microphonePin); //command needed to read the value from the microphone
      mn = min(mn, value);
      mx = max(mx, value);
      delay(1);
    } //calculates the minimum and the maximum, within the
    int volume = mx-mn;
    int db = map(volume,20,900,50,140); //this is the comma
    s=s+db;
    }
  Serial.println(s/5);
  delay(500); //delay bewteen each test, set to 0.5 seconds
}
//the code is not long, but that's what we wanted to make t
```

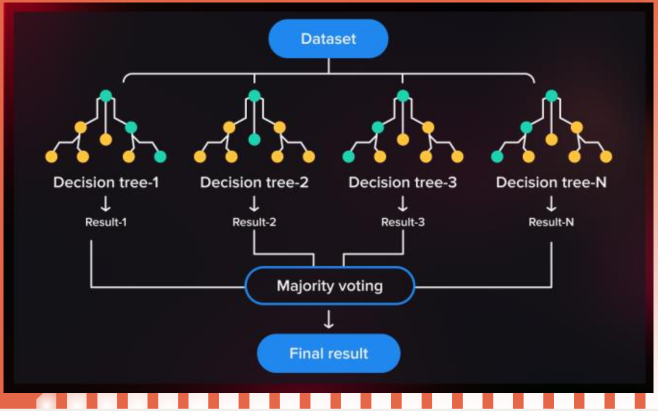


The Machine Learning Model

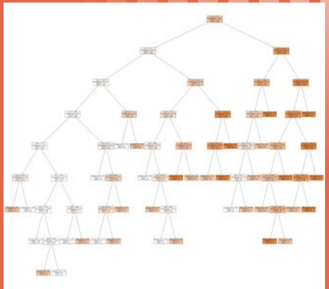
Our Machine Learning algorithm, to ensure precision in its answers, uses the Random Forest Classifier framework for its training. This technology is combination of Decision Trees, Ensembled Learning, and Bootstrapping.

The data set we used for our algorithm has 4 parameters, each consisting of 398 rows of data we created with the help of NIOSH and OSHA. These 4 parameters are: age, noise level, time exposure, and, of course, the labels. The 3 possible labels are: "Negative", "Recommended", "Required". With this training, we reached the outstanding accuracy of 94%.

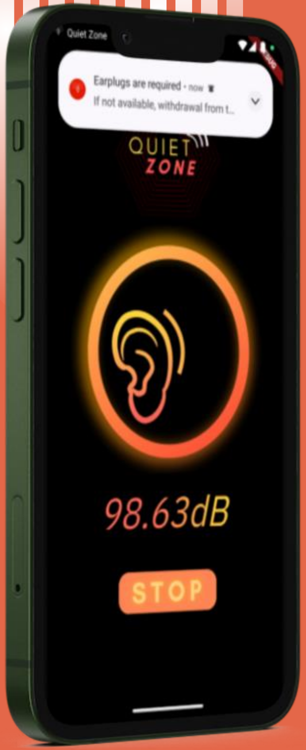
The age is input by the user, the noise level is extracted from the Arduino's txt file, and the time exposure is based on the on amount of dB values in the file.



	Precision	Recall	F1-score	Support
Negative	0.98	1.00	0.99	44
Recommended	0.87	0.87	0.87	23
Required	0.94	0.91	0.92	33



The Front-End of "Quiet Zone"

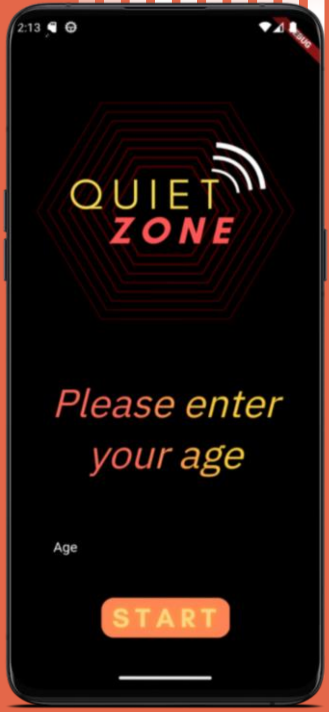


Quiet Zone

Earplugs are required • now

If not available, withdrawal from t...

In the front end, the code will run in this exact order. When the app is opened, it asks for permission to use the microphone and to have access to sending notifications. Then, the app will require from the user their age, which will be sent to the AI. Quickly after, the app goes to its "home page", where a button is located at the center of it. When the button is pressed, the app starts listening and gives every few seconds a decibel reading that is shown on screen. Besides being shown, the reading is also given to the AI so it can make its predictions, and based on the user's age and the decibel reading, it will notify the user if its recommended or necessary to use ear plugs or leave their environment



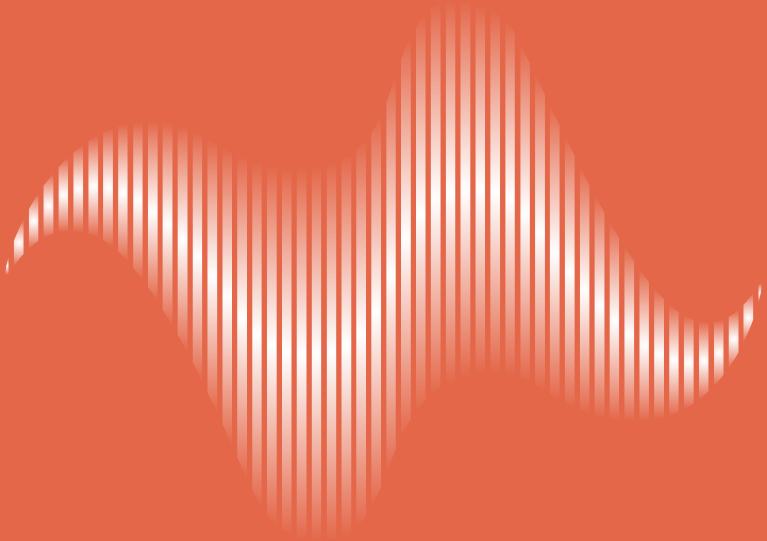
Back-end, APIs, and the 24/7 server

Back-end

Here's the only the modification done to the system is actually the removal of Model_starter.py, the one that checks for changes, and the system's paths

APIs and the 24/7 server

To store and run the back-end we used the pythonanywhere.com site that runs 24/7 server. To connect the scripts and files of the back-end with the front-end we got to use the REST APIs technology.



Future development plan of the project

- Optimize the battery usage and the overall efficiency of the program
- Build an implementation of the app on a smart watch
- Create a database for the user's info for further utilizations
- Integrate with other health and wellness apps
- Allow users to track historical data on sound levels in their environment, helping them to identify patterns and trends
- Partner with hearing health professionals

III. Conclusion

To conclude, our product represents a safe and natural way in which the users can protect their hearing without resorting to hearing aids that would cost them a fortune. In addition, this would also raise awarness about the problem, which can positively improve their overall life because many people do not know how grave and frequent this issue is around the globe.

Aside from the reduced cost compared to hearing aids, other advantages are that our technology is compatible with any device that includes a microphone and is extremely easy to utilize.

Imagine a world in which one billion people can't listen to music, or parents who can't hear their children speak their first words. One of the one billion people can be you, one of your colleagues, or your loved ones.

Use Quiet Zone.

Quiet Zone

Earplugs are required • now

If not available, withdrawal from t...