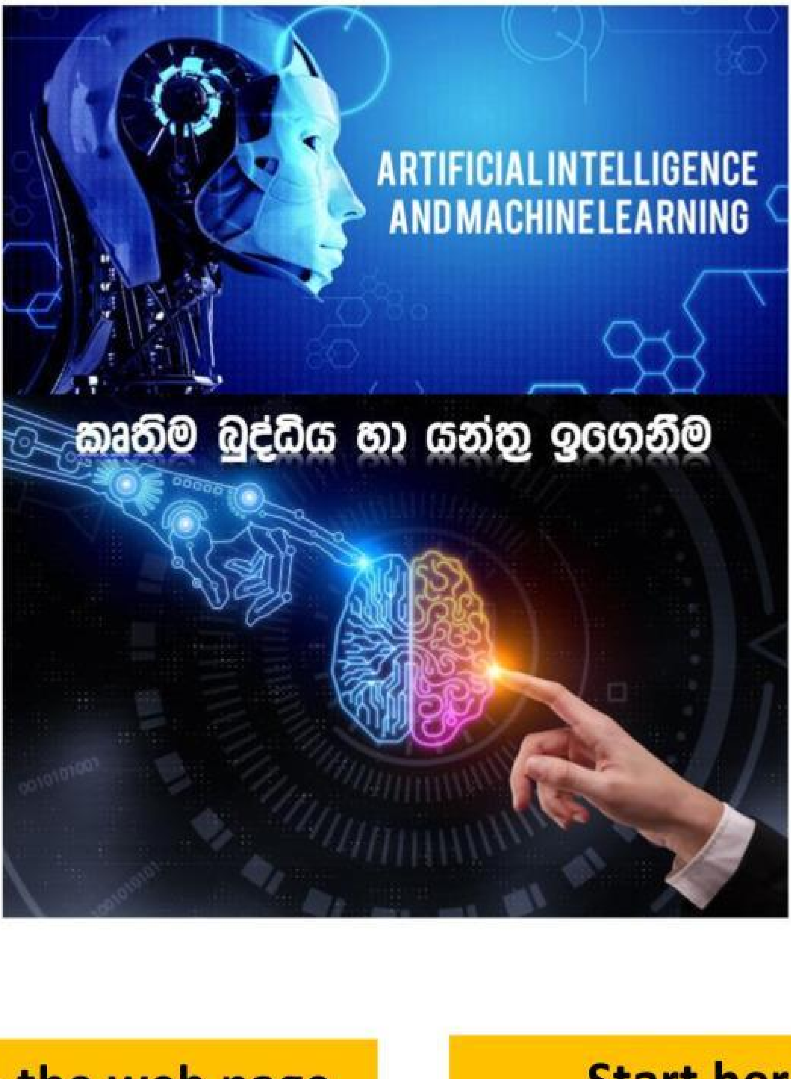
Project 181

181

DP EDUCATION

Coding School

AI and Machine Learning

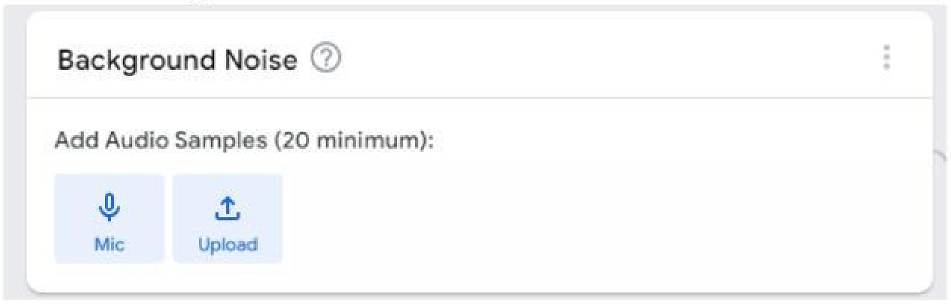


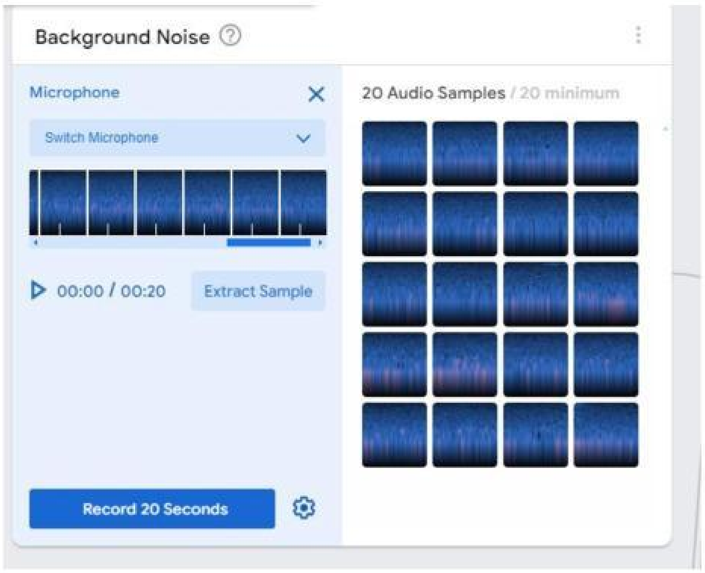
See the web page

Start here

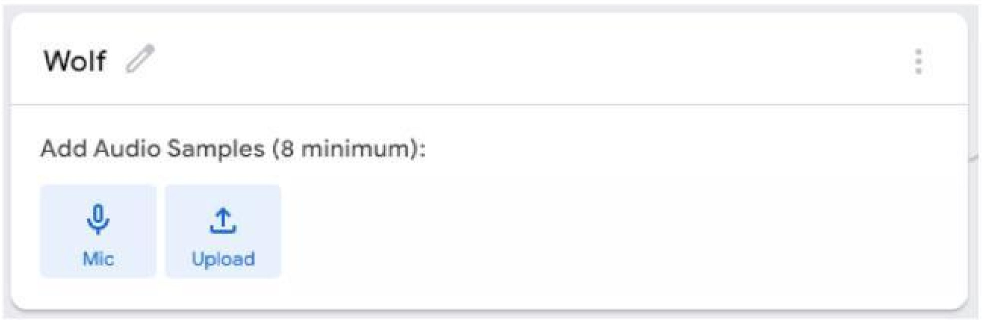
<u>Let's create an animal sound recognition app using the Teachable Machine Audio Project</u>.

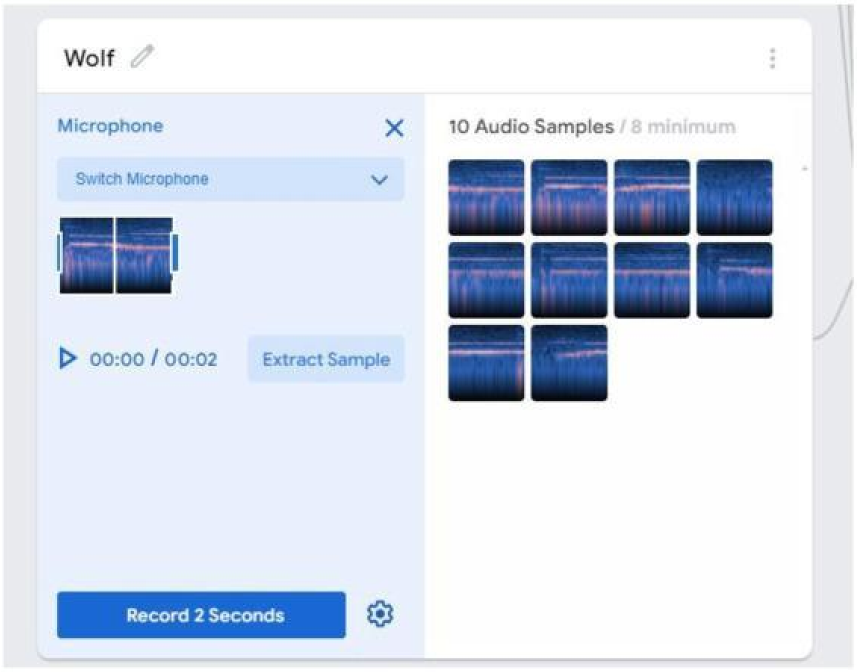
- ❖ First, https://teachablemachine.withgoogle.com Go to the website and start training by getting started. Go into the Audio project and start training the model.
- ❖ First let's add samples related to background noise. For that, use the mic to record a background sound of more than 20 seconds.





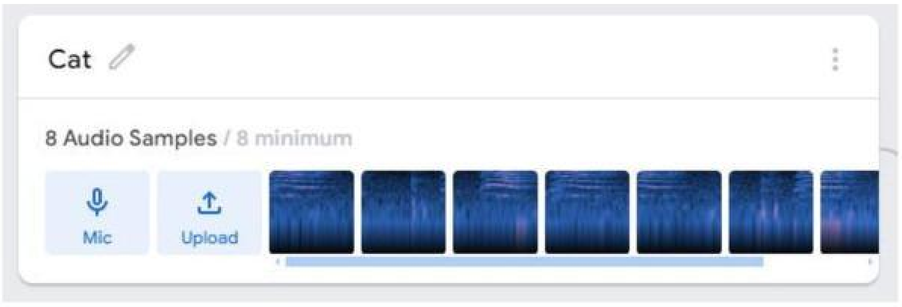
- ❖ Prepare a class to add samples related to the sound of each animal in the model as follows.
- ❖ Use the animal sounds in the glitch project's assets to get you started.



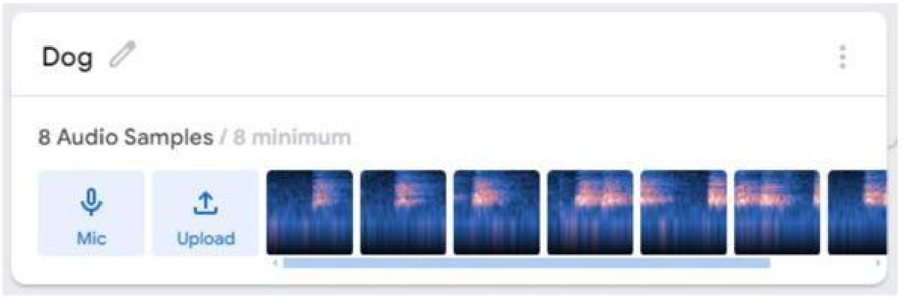


❖ Record samples for 2 seconds each and add more than 8 samples using the extract sample button.

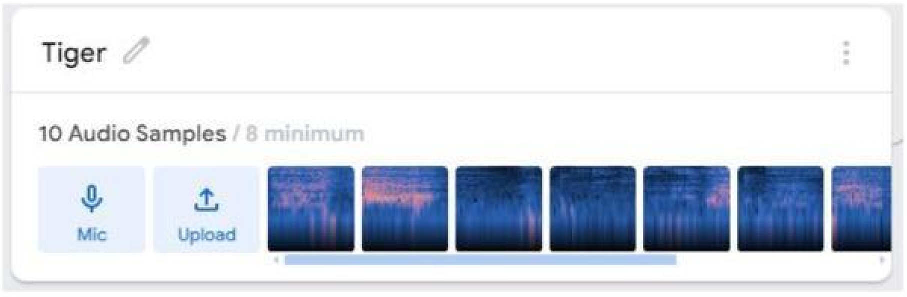
❖



❖



❖

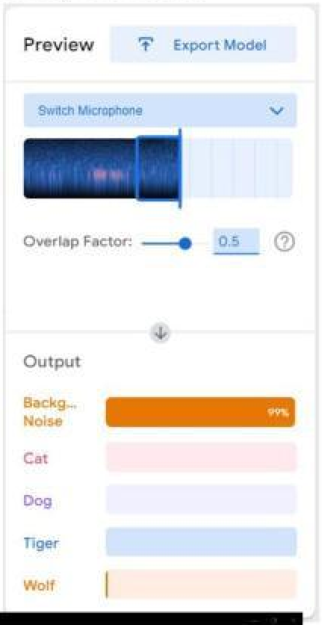


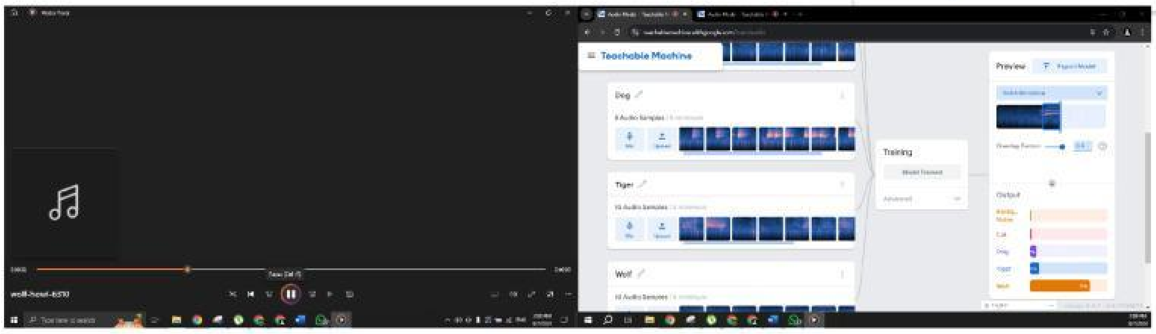
❖ After adding all the samples to the classes as above, train the model using the train model button.

❖ After the model is trained, you can test it in the teachable machine itself. When an animal sound is not played, the background noise should be predicted as the output as follows.

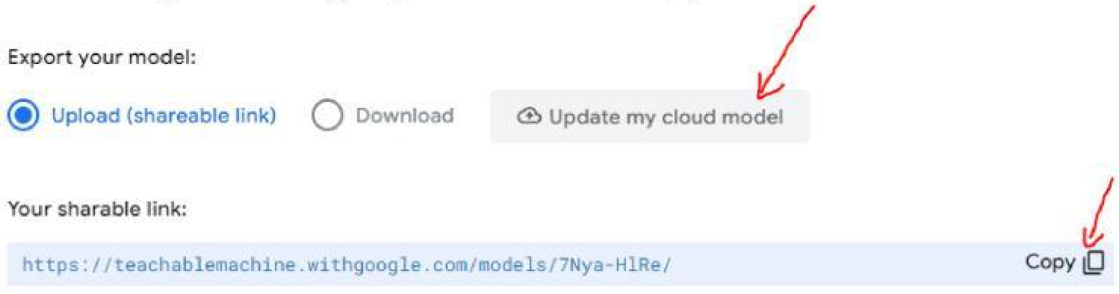
❖ It is when an animal sound is played as below

❖ Predict should be When an animal sound is played as follows, it should be Predict





❖After testing in that way, export the model. Copy the shareable link.



Export your model:

[✓ Upload (shareable link)] Download

Update my cloud model

Your sharable link:

https://teachablemachine.withgoogle.com/models/7Nya-HlRe/ Copy

- ❖ After training the model in this way and getting its link, let's start creating the web as follows.
- ❖ Start coding with the glitch project provided to get you started.
- ❖ Arrange the elements in the Index file as follows.

```
<body>
  <h1>Teachable Machine Audio Model</h1>
  <button type="button" onclick="init()">Start</button>
  <div id="label-container"></div>
```

- ❖ Then prepare the script tags as follows inside the body tag itself.

```
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow-models/speech-commands@0.4.0/dist/speech-commands.min.js"></script>
```

- ❖ Open another script tag and set the functions as below.

```
<script type="text/javascript">

</script>
```

- ❖ In that script tag, first create a variable as follows and store the sharable link you copied in it.

```
const URL = "https://teachablemachine.withgoogle.com/models/7Nya-HlRe/";
```

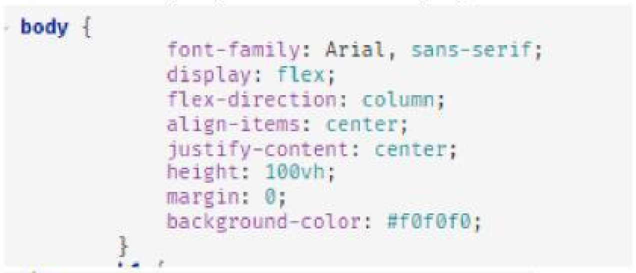
- ❖ Then set the creatModel function in the script tag as follows.

```
async function createModel() {
    const checkpointURL = URL + "model.json"; // model topology
    const metadataURL = URL + "metadata.json"; // model metadata

    const recognizer = speechCommands.create(
        "BROWSER_FFT", // fourier transform type, not useful to change
        undefined, // speech commands vocabulary feature, not useful for your models
        checkpointURL,
        metadataURL);

    // check that model and metadata are loaded via HTTPS requests.
    await recognizer.ensureModelLoaded();

    return recognizer;
}
```

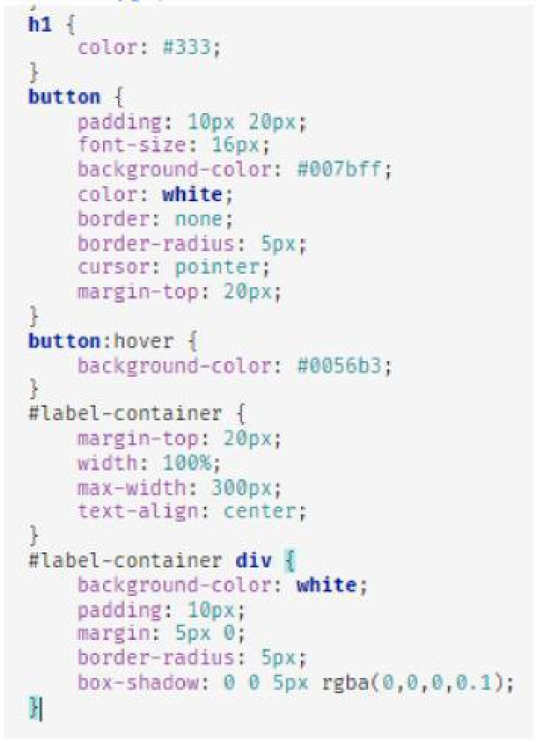
- ❖ Finally, configure the init function as follows.

```
async function init() {
    const recognizer = await createModel();
    const classLabels = recognizer.wordLabels(); // get class labels
    const labelContainer = document.getElementById("label-container");
    for (let i = 0; i < classLabels.length; i++) {
        labelContainer.appendChild(document.createElement("div"));
    }
    recognizer.listen(result => {
        const scores = result.scores;

        for (let i = 0; i < classLabels.length; i++) {
            const classPrediction = classLabels[i] + ": " + result.scores[i].toFixed(2);
            labelContainer.childNodes[i].innerHTML = classPrediction;
        }
    }, {
        includeSpectrogram: true,
        probabilityThreshold: 0.75,
        invokeCallbackOnNoiseAndUnknown: true,
        overlapFactor: 0.50
    });
```

❖ Let's prepare the web page with css as below in the style file.

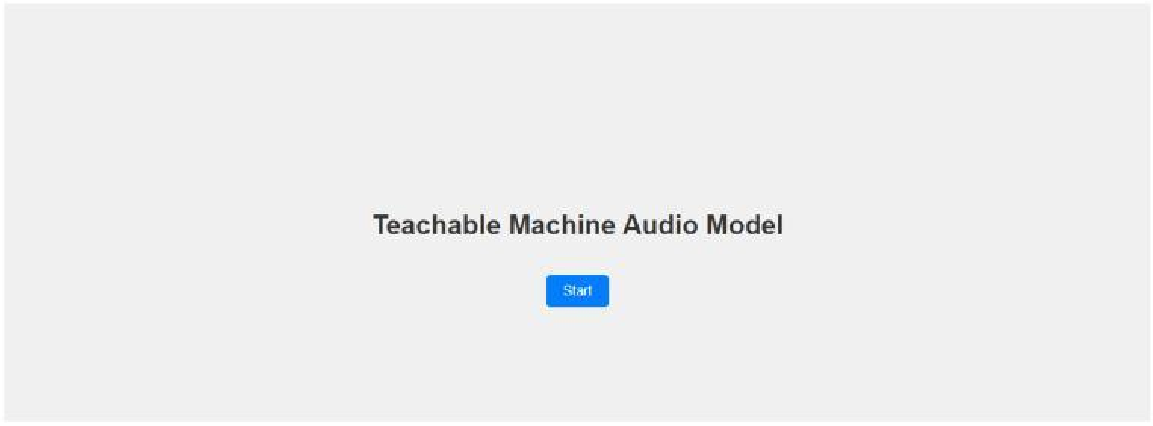


```
body {
            font-family: Arial, sans-serif;
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
            margin: 0;
            background-color: #f0f0f0;
        }
```

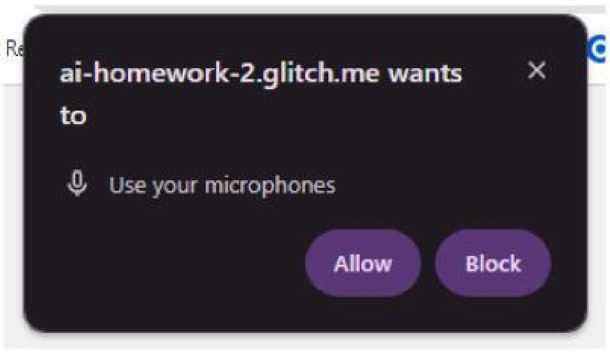


```
h1 {
    color: #333;
}
button {
    padding: 10px 20px;
    font-size: 16px;
    background-color: #007bff;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    margin-top: 20px;
}
button:hover {
    background-color: #0056b3;
}
#label-container {
    margin-top: 20px;
    width: 100%;
    max-width: 300px;
    text-align: center;
}
#label-container div {
    background-color: white;
    padding: 10px;
    margin: 5px 0;
    border-radius: 5px;
    box-shadow: 0 0 5px rgba(0,0,0,0.1);
}
```

❖ After setting the css as above, you should see the web site as below.

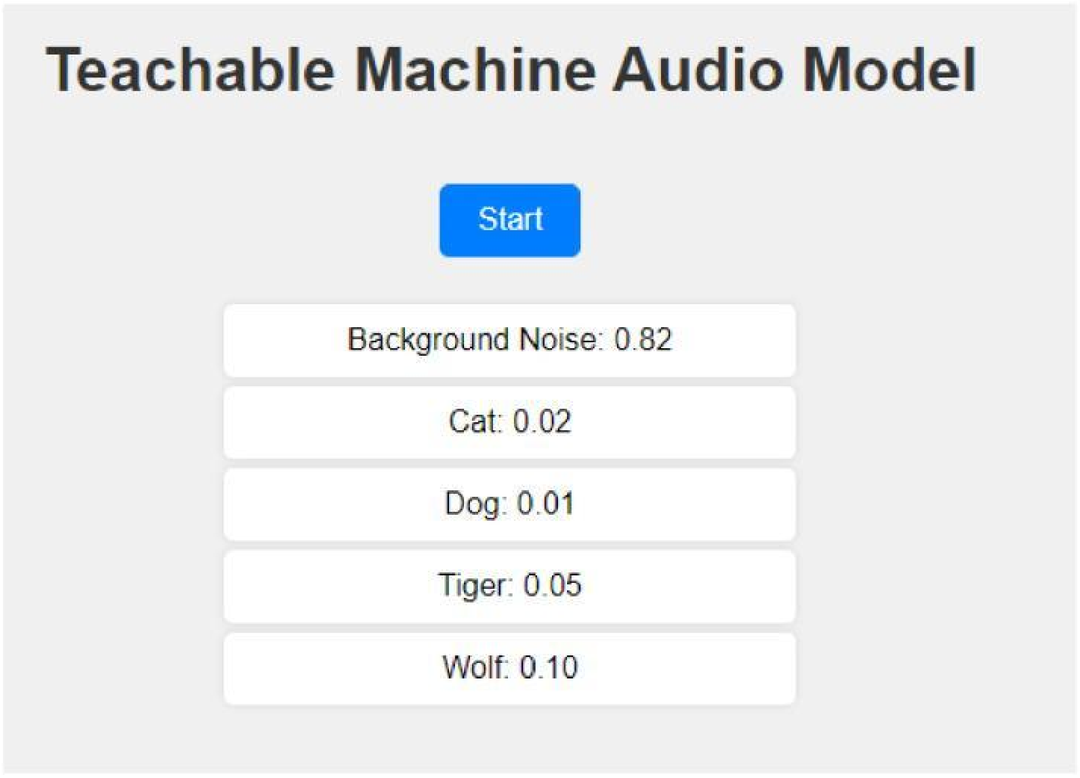


When the start button is clicked, the access browser will make the request for the microphone. Allow it.


ai-homework-2.glitch.me wants to

Use your microphones

❖ Then the prediction model running window will be displayed as below.


Teachable Machine Audio Model

Background Noise: 0.82

Cat: 0.02

Dog: 0.01

Tiger: 0.05

Wolf: 0.10

❖ Now you can play and predict the sounds of each animal from the given or other audio.