Winter Semester 2020-21

CSE3031 - Artificial Intelligence - Project

Review – 3

Prepared under the guidance of

Dr. NITIN SINGH RAJPUT

Assistant Professor, SCOPE,

VIT, Vellore

Team Members:

1) SHIVAM CHAHAR 19BCT0234
2) TANWI AGRAWAL 19BCE2489
3) VEERA VIJAYAPRASAD C 19BCE2127
4) SAYUJYA SAMIR MALKAN 19BCE2559

TITLE:

COVID-19 testing through the cough sounds using machine learning.

Keywords:

COVID-19, Cough sounds, Location

Type of work:

Project work

Introduction

The inability to test at scale has become humanity's Achilles heel in this COVID-19 pandemic. The symptoms of COVID-19 are similar to more than thirty non-COVID-19 medical conditions. So even if the person is having symptoms of COVID-19 still there remains the possibility that it isn't COVID-19 and it's some other non-COVID-19 medical condition. Now here COVID-19 testing becomes crucial and risky also for the person. The in-person testing method puts the medical staff and the person itself at serious risk because there are more chances of getting infected in the hospital or where the in-person testing is being conducted because those places are the hotspots where all people with or without COVID-19 comes.

Objective

We propose coVITal which is a web based application for COVID-19 testing through the cough sounds combined with the person's location data using machine learning. The testing can be done at home with a smartphone hence reducing the need and the risk of going for the in-person testing. Our model is able to predict COVID-19 with high accuracy. The app can also be used to find bed availability in nearby hospitals along with the contact details of the doctors who could give advice to potential patients via phone or email.

Approach

The whole procedure of this approach is divided into steps as following

1) First step is to take a person's cough sound recordings through a microphone and check it through the machine learning model (already trained to recognize COVID-19 cough sounds). At first, we will keep the weightage of analyzing cough sounds, in the final result, lower because of the lack of enough data to train the model accurately. Once we have enough data to train the model accurately, we can increase the weightage of this metric in final results.
2) Then the application will take all these metrics into account and produce a result that if the person actually needs to go to the hospital for further checking or not.
3) If the result turns out to be that the person needs to go to the hospital then the application will display the list of nearby hospitals along with the number of beds available. The application will also give the user an option to take advice from doctors via phone or email.

Resources which we are going to use for this Project:

1) Coswara COVID cough audio dataset by IISc Bangalore for training and testing.
2) https://www.covid19india.org/ api for categorizing places in red, yellow, green zones and the number of infected people in certain places.

Work Plan

Start: 23/02/2021 End: 4/06/2021

Week	Notes
Week 1	Data Collection: Searching and collecting data related to respiratory sounds (breathing, cough) of healthy and COVID-19 affected people.
Week 2	Data Preparation: Preparing data for actual use in machine learning model.
Week 3	Model Building: Devising and planning algorithm to check for COVID-19 through respiratory sounds (choosing a model)
Week 4	Further improving model.
Week 5	Model Training: Initial model training on collected dataset.
Week 6	Evaluation: Evaluating results from initial model training.
Week 7	Parameter tuning.
Week 8	Further improving model and evaluating.
Week 9	Final predictions and evaluation.
Week 10	Building Frontend and Backend for application.
Week 11	Final brush up of application interface and improving it.

Literature Review

Researchers and scientists have long recognized the value of sound as a possible indicator of behavior and wellbeing. To track sounds from the heart or lungs, optical stethoscopes have used purpose-built external microphone recorders. This, too, need highly qualified physicians to listen and interpret, and they are being rapidly replaced by various devices such as a range of imaging methods (e.g., MRI, sonography) that are easier to observe and interpret. Recent advances in digital audio interpretation and simulation, on the other hand, have the ability to revers this pattern and provide sound as a low-cost, widely dispersed substitute.

Imran Ali et al. [1] proposed, developed, and tested an AI (Artificial Intelligence) based screening solution to detect COVID that could be transferred through a smart cell phone application. The AI4COVID-19 smartphone app captures and sends triple 3-second cough sounds and comeback reactions to AI-based clouds running in the cloud within two minutes. Cough is a common symptom of more than 30 medical conditions linked to non-COVID-19. Cough alone is an extremely complex multidisciplinary problem to diagnose COVID illness. The accuracy of 88.76 percent is achieved by investigating morphological path variations with dissimilarities from cough respiratory.

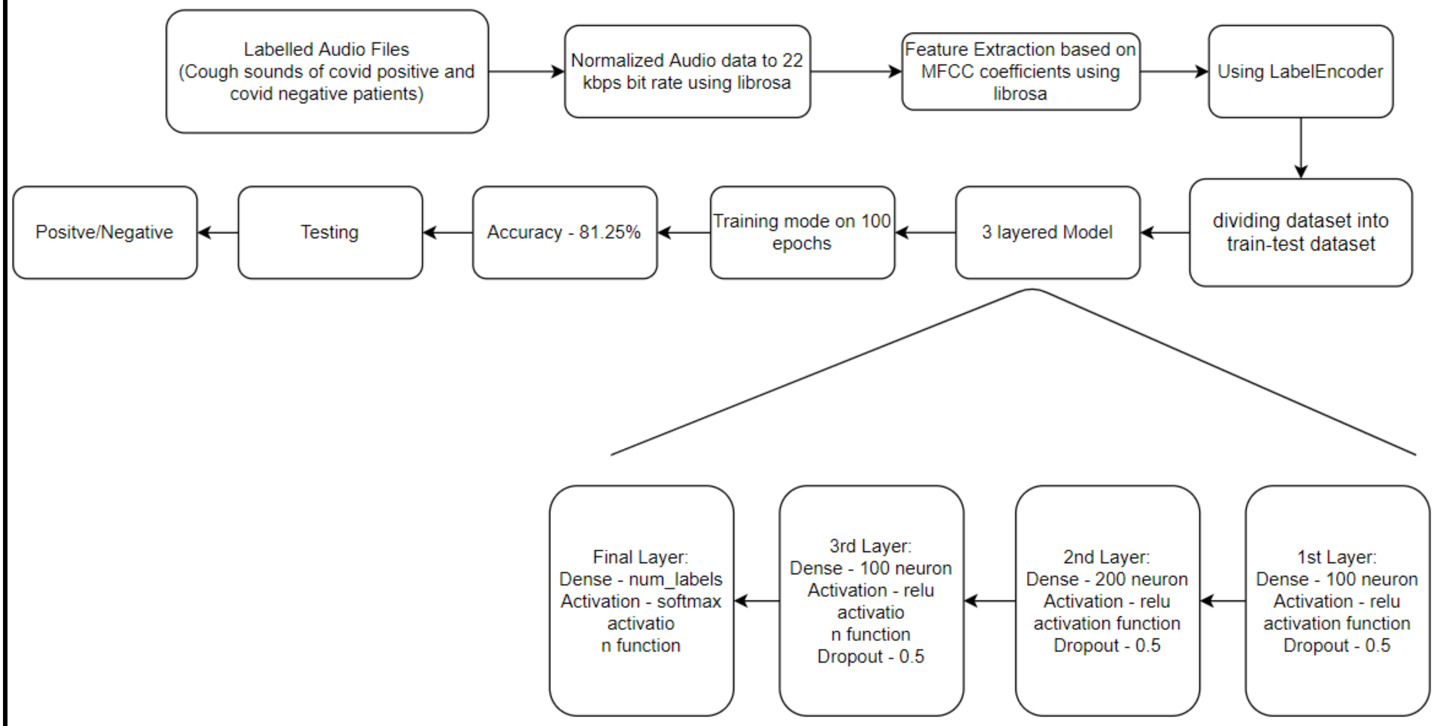
Brown Chloe et al. [2] suggest an Android/iOS app to gather COVID-19 sounds data from crowdsourced sounds respiratory data of more than 200 COVID-19 positives from more than 7k unique users in [9]. Brown Chloe et al. have taken several general parameters and three big set COVID-19 tasks focused on breath and cough tone. The parameters are: I COVID-positive/non-COVID, ii) COVID-positive with cough/non-COVID with cough, iii) COVID-positive with cough/non-COVID with cough, iv) COVID positive with cough/non-COVID asthma cough; task one achieved 80% accuracy for 220 users with cough + breath modality; task two achieved 82 percent accuracy for 29 users with cough only. Finally, task three reached a precision of 80% for 18 users using the modality of breath. Since there is no advanced net to detect any COVID-19 cough, the recall function is marginally lower (72 percent).

Hassan Abdelfatah et al. [3] used the RNN model to implement a system to diagnose COVID positive; authors demonstrated the major impact of the RNN (Recurrent Neural Network) with the use of SSP (Speech Signal Processing) to detect the disease, and specifically, this LSTM (Long Short-Term Memory) used to evaluate the acoustic characteristics of patients' cough, breathing, and voice, in the process. The model results show low accuracy in the speech test as compared to both coughing and breathing sound recordings.

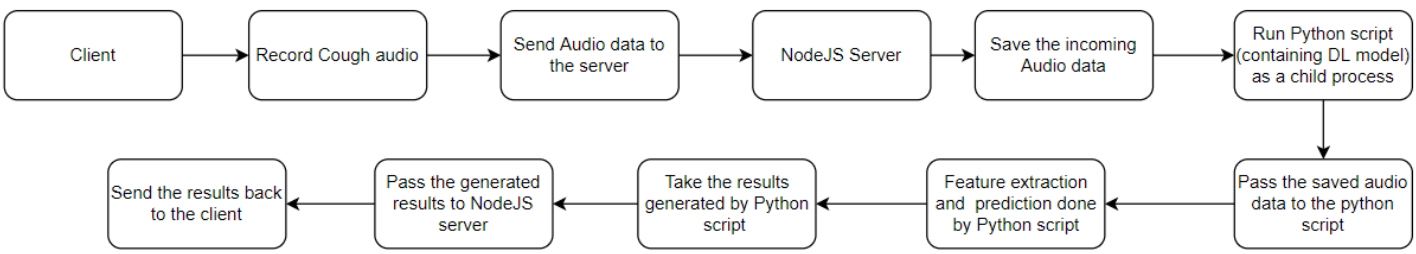
Laguarta Jord et al. [4] suggested an AI model based on cough sound recordings to diagnose COVID symptoms; this model allows for a cost-free prescreening of COVID-19 sound samples around the world. Based on cough sounds from 5320 datasets, it achieves 97.1 percent precision in predicting COVID positive symptoms and 100 percent accuracy in detecting asymptomatic.

The existence of signals in speech evidence regarding the COVID-19 disorder was suggested and studied by Kota Venkata Sai Ritwik et al. [5], and it is a very similar path for the speakers to agree. Help vectors are used to reflect each sentence of Mel filter bank features for each phoneme. The characteristics of COVID-19 speech are extracted from normal speech using a two-class classifier. The limited size of video data obtained from YouTube revealed that an SVM classifier would achieve 88.6 percent accuracy and 92.7 percent F1-Score on this dataset. Further investigation reveals that some telephone classes can distinguish the two classes better than the others (stops, mid vowels, and nasals).

Model Architecture



Application Architecture



Implementation Details

Dataset

Dataset was made using 3 sources:

1. Coswara
2. Virufy COVID-19 Open Cough Dataset (GitHub Repository [6])
3. Hernanmd COVID-19 cough dataset (GitHub Repository [7])

Dataset contained 76 cough audio recordings out of which 30 were of COVID positive.

Screenshot of labelled dataset

	A	B	C	D	E	F	G	H	I
1	date	corona_test	age	gender	medical_history	folder	smoker	patient_reported_symptoms	cough_filename
2	21-Apr	negative	53	male	none,	neg	yes	none,	neg-0421-083-cough-m-53.mp3
3	21-Apr	positive	50	male	Congestive heart failure,	pos	no	Shortness of breath,	pos-0421-084-cough-m-50.mp3
4	21-Apr	negative	43	male	none,	neg	no	Sore throat,	neg-0421-085-cough-m-43.mp3
5	21-Apr	positive	65	male	Asthma or chronic lung disease,	pos	no	Shortness of breath,New or worsening cough,	pos-0421-086-cough-m-65.mp3
6	21-Apr	positive	40	female	none,	pos	no	Sore throat,Loss of taste,Loss of smell,	pos-0421-087-cough-f-40.mp3
7	21-Apr	negative	66	female	Diabetes with complications,	neg	no	none,	neg-0421-088-cough-f-66.mp3
8	21-Apr	negative	20	female	none,	neg	no	none,	neg-0421-089-cough-f-20.mp3
9	21-Apr	negative	17	female	none,	neg	no	Shortness of breath,Sore throat,Body aches,	neg-0421-090-cough-f-17.mp3
10	21-Apr	negative	47	male	none,	neg	no	New or worsening cough,	neg-0421-091-cough-m-47.mp3
11	21-Apr	positive	53	male	none,	pos	no	Fever, chills, or sweating,Shortness of breath,New or worsening cough,Sore throat,Loss of taste,Loss of smell,	pos-0421-092-cough-m-53.mp3
12	21-Apr	positive	24	female	none,	pos	no	none,	pos-0421-093-cough-f-24.mp3
13	21-Apr	positive	51	male	Diabetes with complications,	pos	no	Fever, chills, or sweating,New or worsening cough,Sore throat,	pos-0421-094-cough-m-51.mp3
14	22-Apr	negative	53	male	none,	neg	yes	none,	neg-0422-095-cough-m-53.mp3
15	22-Apr	positive	31	male	none,	pos	no	Shortness of breath,New or worsening cough,	pos-0422-096-cough-m-31.mp3
16	22-Apr	negative	37	male	none,	neg	no	none,	neg-0422-097-cough-m-37.mp3
17	22-Apr	negative	24	female	none,	neg	no	New or worsening cough,	neg-0422-098-cough-f-24.mp3
18	23-Apr	positive	25	female	none,	pos	no	none,	pos-cough-heavy-1.wav
19	24-Apr	positive	26	female	none,	pos	no	none,	pos-cough-heavy-2.wav

- **The attached jupyter notebook pdf file contains everything including code and results and the implementation details**

Exploratory Data Analysis

[2]:
```
import matplotlib.pyplot as plt
%matplotlib inline
import librosa
import librosa.display
import IPython.display as ipd
import os
import pandas as pd
import numpy as np
```

Labelled Cough audio dataset

[5]:
```
metadata = pd.read_csv('clinical\\labels.csv')
metadata.head(10)
```

[5]:

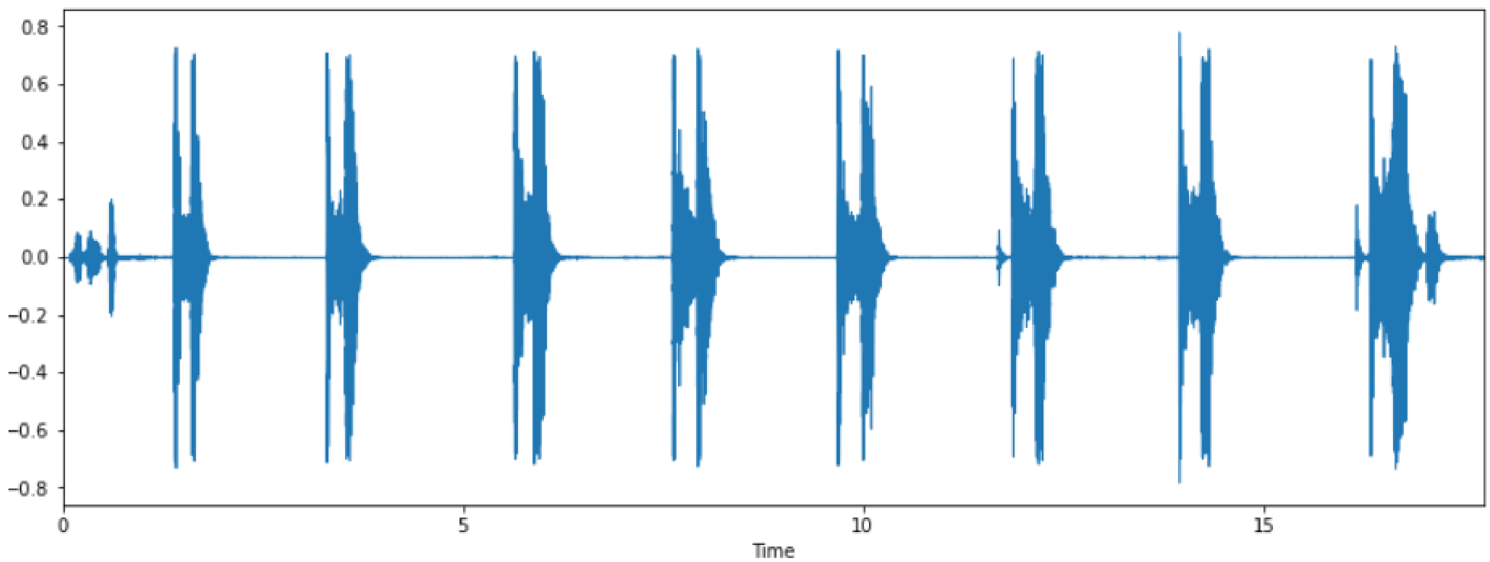
	date	corona_test	age	gender	medical_history	folder	smoker	patient_reported_symptoms	cough_filename
0	21-Apr	negative	53	male	none,	neg	yes	none,	neg-0421-083-cough-m-53.mp3
1	21-Apr	positive	50	male	Congestive heart failure,	pos	no	Shortness of breath,	pos-0421-084-cough-m-50.mp3
2	21-Apr	negative	43	male	none,	neg	no	Sore throat,	neg-0421-085-cough-m-43.mp3
3	21-Apr	positive	65	male	Asthma or chronic lung disease,	pos	no	Shortness of breath,New or worsening cough,	pos-0421-086-cough-m-65.mp3
4	21-Apr	positive	40	female	none,	pos	no	Sore throat,Loss of taste,Loss of smell,	pos-0421-087-cough-f-40.mp3
5	21-Apr	negative	66	female	Diabetes with complications,	neg	no	none,	neg-0421-088-cough-f-66.mp3
6	21-Apr	negative	20	female	none,	neg	no	none,	neg-0421-089-cough-f-20.mp3
7	21-Apr	negative	17	female	none,	neg	no	Shortness of breath,Sore throat,Body aches,	neg-0421-090-cough-f-17.mp3
8	21-Apr	negative	47	male	none,	neg	no	New or worsening cough,	neg-0421-091-cough-m-47.mp3
9	21-Apr	positive	53	male	none,	pos	no	Fever, chills, or sweating,Shortness of breath...	pos-0421-092-cough-m-53.mp3

COVID Positive cough audio waveplot

[4]:
```
filename_pos="clinical\original\pos\\pos-0421-084-cough-m-50.mp3"
plt.figure(figsize=(14,5))
data_pos,sample_rate_pos=librosa.load(filename_pos)
librosa.display.waveplot(data_pos,sr=sample_rate_pos)
ipd.Audio(filename_pos)
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\librosa\core\audio.py:165:
UserWarning: PySoundFile failed. Trying audioread instead.
  warnings.warn("PySoundFile failed. Trying audioread instead.")
```

[4]: <IPython.lib.display.Audio object>



Normalized Sample Rate

[18]:
```
sample_rate_pos
```

[18]: 22050

Audio data converted to array

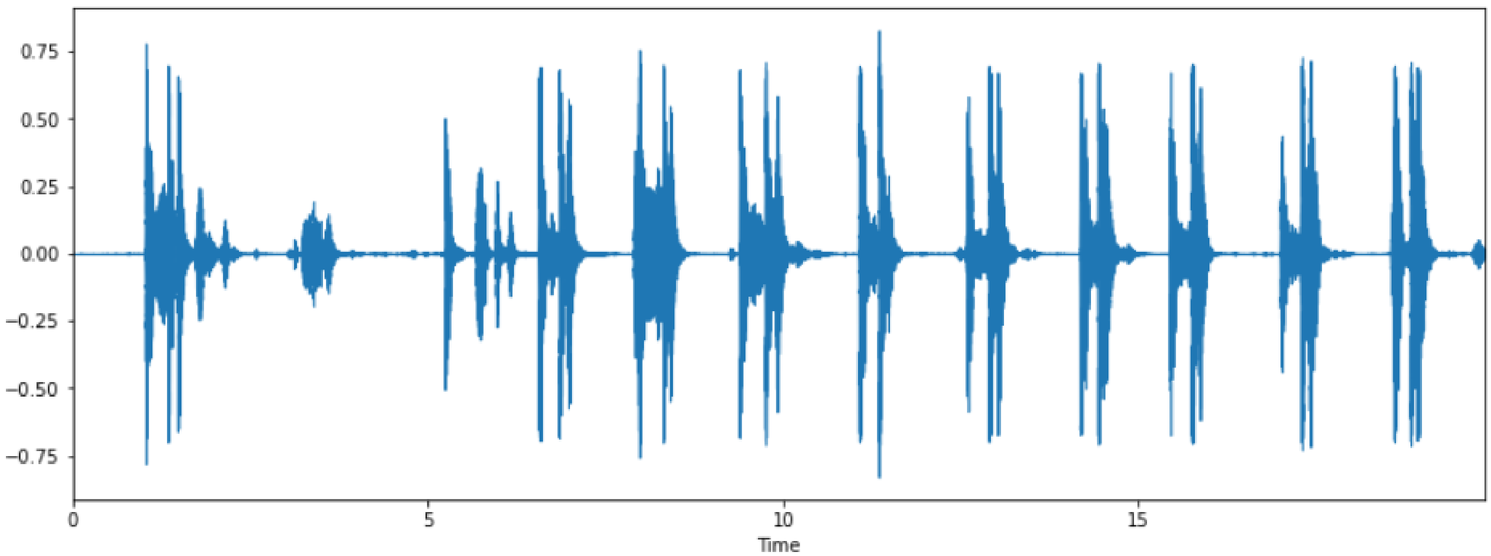
[19]:
```
data_pos
```

[19]:
```
array([ 0.0000000e+00,  0.0000000e+00,  0.0000000e+00, ...,
       -1.9726438e-04,  1.1065308e-05,  0.0000000e+00], dtype=float32)
```

COVID Negative cough audio waveplot

[7]:
```
filename_neg="clinical\\original\\neg\\neg-0421-088-cough-f-66.mp3"
plt.figure(figsize=(14,5))
data_neg,sample_rate_neg=librosa.load(filename_neg)
librosa.display.waveplot(data_neg,sr=sample_rate_neg)
ipd.Audio(filename_neg)
```

[7]: <IPython.lib.display.Audio object>



Normalized Sample Rate

[20]:
```
sample_rate_neg
```

[20]: 22050

Audio data converted to array

[35]:
```
data_neg
```

[35]:
```
array([0.       , 0.       , 0.       , ..., 0.0018331, 0.0038525,
       0.       ], dtype=float32)
```

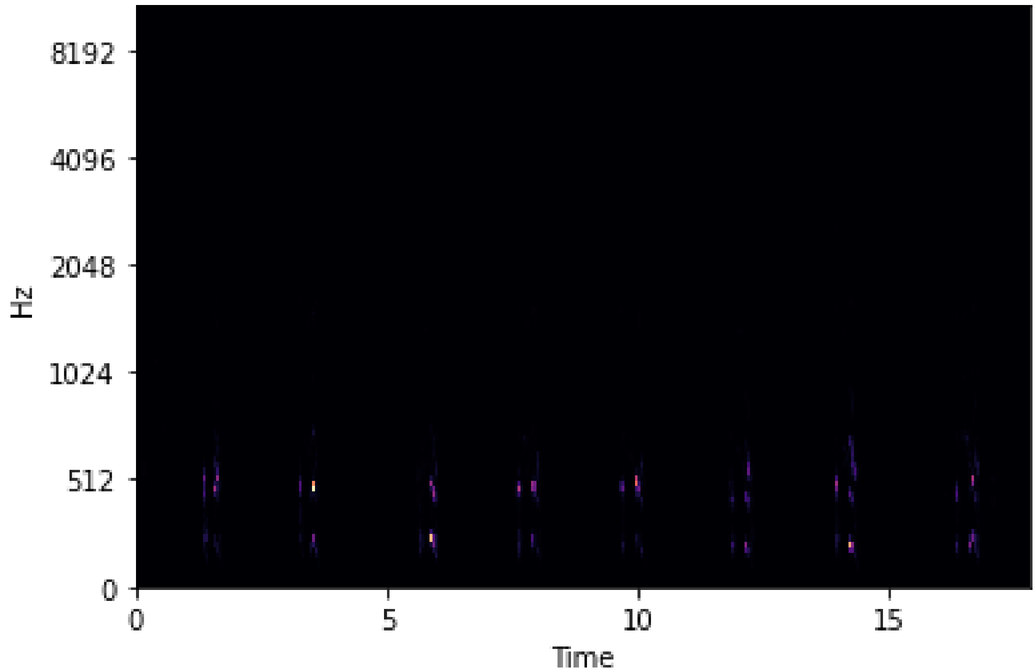
Mel spectogram of COVID positive cough audio

[27]:
```
y, sr = librosa.load(filename_pos)
ps = librosa.feature.melspectrogram(y=y, sr=sr)
ps.shape
```

[27]: (128, 765)

[28]:
```
librosa.display.specshow(ps,y_axis='mel',x_axis='time')
```

[28]: <matplotlib.collections.QuadMesh at 0x165217bbc10>



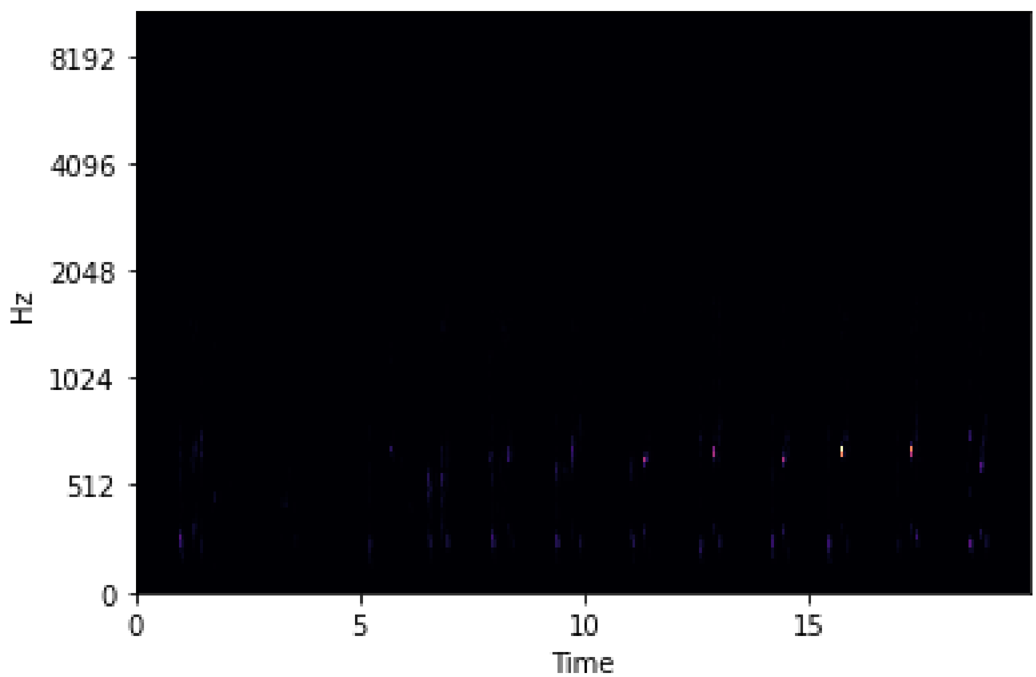
Mel spectogram of COVID Negative cough audio

[10]:
```
y, sr = librosa.load(filename_neg)
ps = librosa.feature.melspectrogram(y=y, sr=sr)
ps.shape
```

[10]: (128, 857)

[11]:
```
librosa.display.specshow(ps,y_axis='mel',x_axis='time')
```

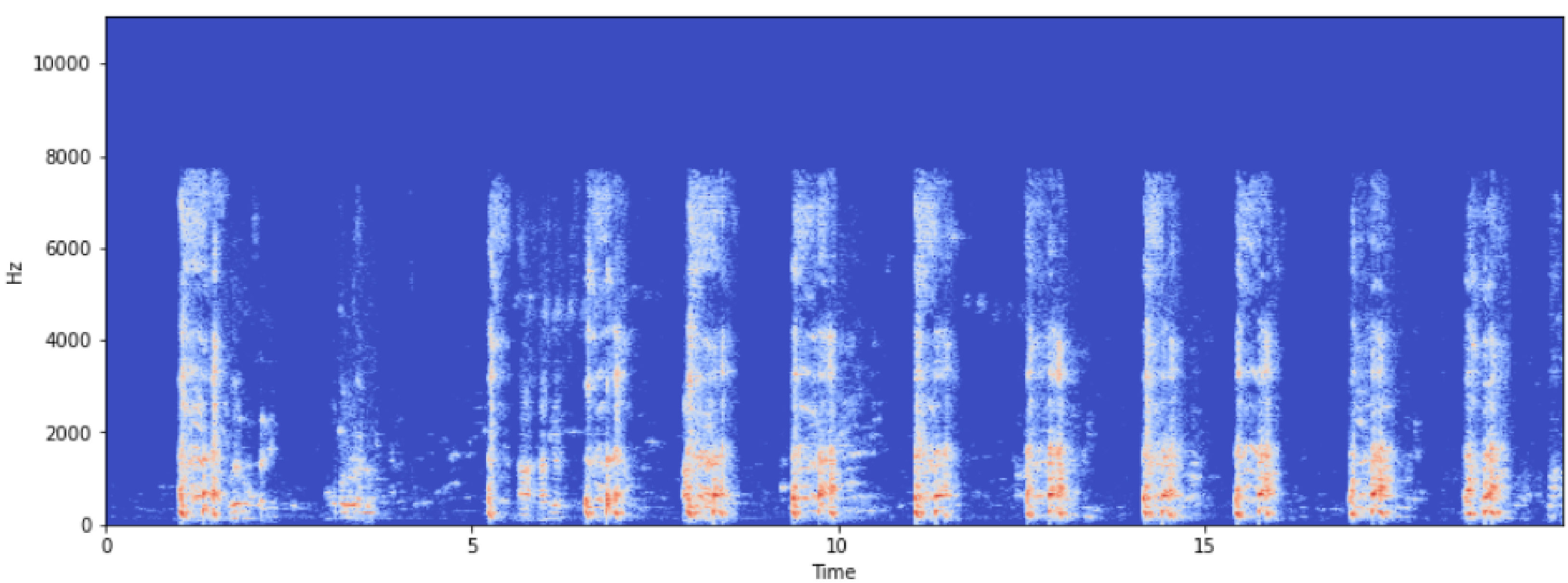
[11]: <matplotlib.collections.QuadMesh at 0x1652116cbb0>



COVID Negative cough audio spectrum

[13]:
```
X = librosa.stft(y)
Xdb = librosa.amplitude_to_db(abs(X))
plt.figure(figsize=(14,5))
librosa.display.specshow(Xdb,sr=sr,x_axis='time',y_axis='hz')
```

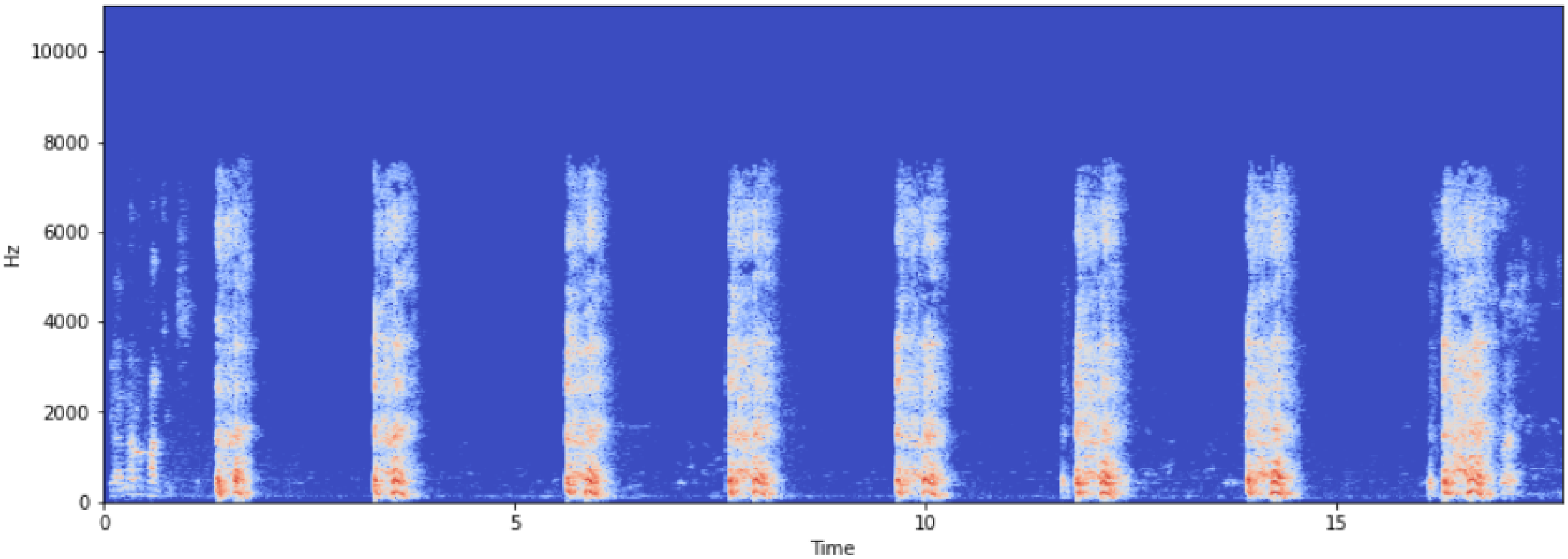
[13]: <matplotlib.collections.QuadMesh at 0x16521203a30>



COVID Positive cough audio spectrum

[15]:
```
y, sr = librosa.load(filename_pos)
X = librosa.stft(y)
Xdb = librosa.amplitude_to_db(abs(X))
plt.figure(figsize=(14,5))
librosa.display.specshow(Xdb,sr=sr,x_axis='time',y_axis='hz')
```

[15]:
```
<matplotlib.collections.QuadMesh at 0x165210beb80>
```



Feature extraction using MFCC

Here we will be using Mel-Frequency Cepstral Coefficients(MFCC) from the audio samples. The MFCC summarises the frequency distribution across the window size, so it is possible to analyse both the frequency and time characteristics of the sound. These audio representations will allow us to identify features for classification.

Extracting features of one file (sample)

[6]:
```
filename_pos="clinical\original\pos\\pos-0421-084-cough-m-50.mp3"
data_pos,sample_rate_pos=librosa.load(filename_pos)
mfccs = librosa.feature.mfcc(y=data_pos, sr=sample_rate_pos, n_mfcc=40)
print(mfccs.shape)
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\librosa\core\audio.py:165:
UserWarning: PySoundFile failed. Trying audioread instead.
  warnings.warn("PySoundFile failed. Trying audioread instead.")
```

```
(40, 765)
```

[7]:
```
mfccs
```

[7]: array([[-5.5126825e+02, -5.5126825e+02, -5.4601251e+02, ...,
          -5.1879736e+02, -5.1567249e+02, -5.1970978e+02],
         [0.0000000e+00, 0.0000000e+00, 6.8114338e+00, ...,
           2.5116863e+01, 2.8397551e+01, 3.0157677e+01],
         [0.0000000e+00, 0.0000000e+00, 5.3188505e+00, ...,
           1.8088528e+01, 2.1304604e+01, 1.7772816e+01],
         ...,
         [0.0000000e+00, 0.0000000e+00, -1.2575887e+00, ...,
           4.6400058e-01, 2.9210353e+00, 4.5062594e+00],
         [0.0000000e+00, 0.0000000e+00, -2.1845088e+00, ...,
           8.3374043e+00, 8.7302361e+00, 5.9929790e+00],
         [0.0000000e+00, 0.0000000e+00, -2.7988749e+00, ...,
          -2.4354792e+00, -2.5820913e+00, -2.5259297e+00]], dtype=float32)

Extracting features of all audio files

[8]:
```
def features_extractor(file):
    audio, sample_rate = librosa.load(file_name, res_type='kaiser_best')
    mfccs_features = librosa.feature.mfcc(y=audio, sr=sample_rate, n_mfcc=40)
    mfccs_scaled_features = np.mean(mfccs_features.T,axis=0)

    return mfccs_scaled_features
```

[9]:
```
from tqdm import tqdm
### Now we iterate through every audio file and extract features
### using Mel-Frequency Cepstral Coefficients
audio_dataset_path='clinical\\original\\'
extracted_features=[]
for index_num,row in tqdm(metadata.iterrows()):
    file_name = os.path.join(os.path.
 ↪abspath(audio_dataset_path),str(row["folder"])+'\\',str(row["cough_filename"]))
    final_class_labels=row["corona_test"]
    data=features_extractor(file_name)
    extracted_features.append([data,final_class_labels])
```

```
0it [00:00, ?it/s]C:\Users\ADMIN\anaconda3\lib\site-
packages\librosa\core\audio.py:165: UserWarning: PySoundFile failed. Trying
audioread instead.
  warnings.warn("PySoundFile failed. Trying audioread instead.")
76it [04:41,  3.70s/it]
```

[10]:
```
### converting extracted_features to Pandas dataframe
extracted_features_df=pd.
 ↪DataFrame(extracted_features,columns=['feature','class'])
extracted_features_df.head(20)
```

[10]:
```
                                               feature     class
0   [-362.26675, 85.50063, -27.324495, 19.882082, ...  negative
1   [-419.73032, 59.81665, -6.4413686, 14.772525, ...  positive
2   [-392.7497, 60.903584, -1.5468708, 12.362222, ...  negative
3   [-439.70718, 69.86176, -8.996845, 13.870843, -...  positive
4   [-391.8664, 57.311222, -15.042628, 11.311216, ...  positive
5   [-373.94162, 80.459946, -9.110424, -1.4120208,...  negative
6   [-440.20844, 50.426956, -6.9619546, 14.114877,...  negative
7   [-401.80597, 89.8483, -21.094421, 9.897526, -2...  negative
8   [-440.315, 70.1227, 4.528007, 6.696279, -5.389...  negative
9   [-367.23538, 77.67865, -11.57527, 9.818056, -2...  positive
10  [-390.46765, 54.457733, -9.511929, 21.605019, ...  positive
11  [-351.85452, 67.796425, -28.611973, 24.644041,...  positive
12  [-345.9613, 87.219955, -24.261772, 16.657543, ...  negative
13  [-343.87643, 77.41182, -28.409334, 17.075321, ...  positive
14  [-363.11862, 75.484665, -16.34377, 5.6816907, ...  negative
15  [-363.81082, 74.34234, -14.45791, 13.4616585, ...  negative
16  [-415.84595, 66.48398, 2.7919796, 24.260683, -...  positive
17  [-527.7733, 39.76033, -19.772854, 21.303574, -...  positive
18  [-300.78442, 76.68725, -14.724839, 22.668074, ...  positive
19  [-351.25974, 66.97575, -7.3544335, 34.0258, 4...  positive
```

[11]:
```python
### Split the dataset into independent and dependent dataset
X=np.array(extracted_features_df['feature'].tolist())
y=np.array(extracted_features_df['class'].tolist())
```

[12]:
```python
X.shape
```

[12]:
```
(76, 40)
```

[13]:
```python
y
```

[13]:
```
array(['negative', 'positive', 'negative', 'positive', 'positive',
       'negative', 'negative', 'negative', 'negative', 'positive',
       'positive', 'positive', 'negative', 'positive', 'negative',
       'negative', 'positive', 'positive', 'positive', 'positive',
       'positive', 'positive', 'positive', 'positive', 'positive',
       'positive', 'positive', 'positive', 'positive', 'positive',
       'negative', 'negative', 'negative', 'negative', 'negative',
       'negative', 'negative', 'negative', 'negative', 'negative',
       'negative', 'negative', 'negative', 'negative', 'negative',
       'negative', 'negative', 'negative', 'negative', 'negative',
       'negative', 'negative', 'negative', 'negative', 'negative',
       'negative', 'negative', 'negative', 'negative', 'negative',
       'negative', 'negative', 'negative', 'negative', 'negative',
       'negative', 'negative', 'negative', 'negative', 'negative',
       'negative', 'negative', 'negative', 'negative', 'negative',
```

```
'negative'], dtype='<U8')
```

Label Encoding

[14]:
```
### Label Encoding
###y=np.array(pd.get_dummies(y))
### Label Encoder
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelEncoder
labelencoder=LabelEncoder()
y=to_categorical(labelencoder.fit_transform(y))
```

[15]:
```
y
```

[15]:
```
array([[1., 0.],
       [0., 1.],
       [1., 0.],
       [0., 1.],
       [0., 1.],
       [1., 0.],
       [1., 0.],
       [1., 0.],
       [1., 0.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [1., 0.],
       [0., 1.],
       [1., 0.],
       [1., 0.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [1., 0.],
       [1., 0.],
```

```
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.]], dtype=float32)
```

Train - Test Split

[16]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=0)
```

[17]:
```
X_train
```

[17]:
```
array([[-5.2402637e+02,  5.6987354e+01, -4.2589512e+00, ...,
         2.0481347e-01,  3.9402134e+00,  2.0798965e+00],
       [-4.9306253e+02,  1.4614683e+01, -2.3427277e+00, ...,
         1.0731262e+00, -7.8298204e-02,  8.9549601e-01],
       [-3.6803516e+02,  5.0508846e+01,  9.3852711e+00, ...,
        -1.3530470e+00, -1.5751493e+00,  2.4827635e+00],
       ...,
       [-3.9524927e+02,  5.0770729e+01,  2.2054594e+01, ...,
        -3.2736588e-01, -1.2837274e+00,  2.1400109e-02],
       [-4.0264984e+02,  5.8334717e+01,  5.9546204e+00, ...,
         1.3401449e-01, -1.5818532e-01,  2.1612546e-01],
       [-4.6127905e+02,  2.1776873e+01, -9.7883072e+00, ...,
         2.0757441e-01, -1.5606801e-02,  3.0636638e-01]], dtype=float32)
```

[18]:
```
y
```

[18]:
```
array([[1., 0.],
       [0., 1.],
       [1., 0.],
       [0., 1.],
       [0., 1.],
       [1., 0.],
       [1., 0.],
       [1., 0.],
       [1., 0.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [1., 0.],
       [0., 1.],
       [1., 0.],
       [1., 0.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
       [0., 1.],
```

```
[0., 1.],
[0., 1.],
[0., 1.],
[0., 1.],
[0., 1.],
[0., 1.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
```

```
        [1., 0.],
        [1., 0.],
        [1., 0.],
        [1., 0.],
        [1., 0.]], dtype=float32)
```

[19]:
```
X_train.shape
```

[19]:
```
(60, 40)
```

[20]:
```
X_test.shape
```

[20]:
```
(16, 40)
```

[21]:
```
y_train.shape
```

[21]:
```
(60, 2)
```

[22]:
```
y_test.shape
```

[22]:
```
(16, 2)
```

Model Creation

[23]:
```
import tensorflow as tf
print(tf.__version__)
```

```
2.3.0
```

[24]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Dropout,Activation,Flatten
from tensorflow.keras.optimizers import Adam
from sklearn import metrics
```

[25]:
```
### No of classes
num_labels=y.shape[1]
```

[26]:
```
model=Sequential()

###first layer
model.add(Dense(100,input_shape=(40,)))
model.add(Activation('relu'))
model.add(Dropout(0.5))

###second layer
model.add(Dense(200))
model.add(Activation('relu'))
```

```
model.add(Dropout(0.5))
###third layer

model.add(Dense(100))
model.add(Activation('relu'))
model.add(Dropout(0.5))

###final layer
model.add(Dense(num_labels))
model.add(Activation('softmax'))
```

[27]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 100)               4100
_________________________________________________________________
activation (Activation)      (None, 100)               0
_________________________________________________________________
dropout (Dropout)            (None, 100)               0
_________________________________________________________________
dense_1 (Dense)              (None, 200)               20200
_________________________________________________________________
activation_1 (Activation)    (None, 200)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 200)               0
_________________________________________________________________
dense_2 (Dense)              (None, 100)               20100
_________________________________________________________________
activation_2 (Activation)    (None, 100)               0
_________________________________________________________________
dropout_2 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 202
_________________________________________________________________
activation_3 (Activation)    (None, 2)                 0
=================================================================
Total params: 44,602
Trainable params: 44,602
Non-trainable params: 0
_________________________________________________________________
```

[28]:
```
model.
 ↪compile(loss='categorical_crossentropy',metrics=['accuracy'],optimizer='adam')
```

Training model

[29]:
```python
from tensorflow.keras.callbacks import ModelCheckpoint
from datetime import datetime

num_epochs = 100
num_batch_size = 32

checkpointer = ModelCheckpoint(filepath='saved_models/audio_classification.hdf5',
                               verbose=1, save_best_only=True)
start = datetime.now()

model.fit(X_train, y_train, batch_size=num_batch_size, epochs=num_epochs,␣
 ↪validation_data=(X_test, y_test), callbacks=[checkpointer], verbose=1)


duration = datetime.now() - start
print("Training completed in time: ", duration)
```

```
Epoch 1/100
1/2 [==============>...] - ETA: 0s - loss: 66.5800 - accuracy:
0.3438
Epoch 00001: val_loss improved from inf to 7.34220, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 160ms/step - loss: 61.5507 - accuracy:
0.3667 - val_loss: 7.3422 - val_accuracy: 0.8125
Epoch 2/100
1/2 [==============>...] - ETA: 0s - loss: 21.5921 - accuracy:
0.5312
Epoch 00002: val_loss did not improve from 7.34220
2/2 [==============================] - 0s 16ms/step - loss: 28.3465 - accuracy:
0.5500 - val_loss: 12.0373 - val_accuracy: 0.8125
Epoch 3/100
1/2 [==============>...] - ETA: 0s - loss: 16.0657 - accuracy:
0.6250
Epoch 00003: val_loss did not improve from 7.34220
2/2 [==============================] - 0s 18ms/step - loss: 26.9519 - accuracy:
0.5833 - val_loss: 13.8862 - val_accuracy: 0.8125
Epoch 4/100
1/2 [==============>...] - ETA: 0s - loss: 15.1492 - accuracy:
0.6875
Epoch 00004: val_loss did not improve from 7.34220
2/2 [==============================] - 0s 17ms/step - loss: 20.2863 - accuracy:
0.7000 - val_loss: 13.7351 - val_accuracy: 0.8125
Epoch 5/100
1/2 [==============>...] - ETA: 0s - loss: 25.2778 - accuracy:
0.6250
Epoch 00005: val_loss did not improve from 7.34220
```

```
2/2 [==============================] - 0s 19ms/step - loss: 30.5062 - accuracy:
0.6167 - val_loss: 12.7743 - val_accuracy: 0.8125
Epoch 6/100
1/2 [==============>...] - ETA: 0s - loss: 17.1059 - accuracy:
0.7500
Epoch 00006: val_loss did not improve from 7.34220
2/2 [==============================] - 0s 19ms/step - loss: 16.4835 - accuracy:
0.6667 - val_loss: 11.3744 - val_accuracy: 0.8125
Epoch 7/100
1/2 [==============>...] - ETA: 0s - loss: 24.8668 - accuracy:
0.5000
Epoch 00007: val_loss did not improve from 7.34220
2/2 [==============================] - 0s 22ms/step - loss: 22.6841 - accuracy:
0.5667 - val_loss: 10.0507 - val_accuracy: 0.8125
Epoch 8/100
1/2 [==============>...] - ETA: 0s - loss: 24.6187 - accuracy:
0.6562
Epoch 00008: val_loss did not improve from 7.34220
2/2 [==============================] - 0s 17ms/step - loss: 31.6035 - accuracy:
0.5000 - val_loss: 9.2072 - val_accuracy: 0.8125
Epoch 9/100
1/2 [==============>...] - ETA: 0s - loss: 36.0899 - accuracy:
0.5938
Epoch 00009: val_loss did not improve from 7.34220
2/2 [==============================] - 0s 16ms/step - loss: 31.6993 - accuracy:
0.5333 - val_loss: 8.4424 - val_accuracy: 0.8125
Epoch 10/100
1/2 [==============>...] - ETA: 0s - loss: 17.0259 - accuracy:
0.5938
Epoch 00010: val_loss did not improve from 7.34220
2/2 [==============================] - 0s 17ms/step - loss: 18.6734 - accuracy:
0.5500 - val_loss: 7.6305 - val_accuracy: 0.8125
Epoch 11/100
1/2 [==============>...] - ETA: 0s - loss: 25.3690 - accuracy:
0.5312
Epoch 00011: val_loss improved from 7.34220 to 7.14030, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 21ms/step - loss: 22.0894 - accuracy:
0.5500 - val_loss: 7.1403 - val_accuracy: 0.8125
Epoch 12/100
1/2 [==============>...] - ETA: 0s - loss: 10.8027 - accuracy:
0.6562
Epoch 00012: val_loss improved from 7.14030 to 6.75467, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 16ms/step - loss: 19.2331 - accuracy:
0.5167 - val_loss: 6.7547 - val_accuracy: 0.8125
Epoch 13/100
1/2 [==============>...] - ETA: 0s - loss: 24.3298 - accuracy:
```

```
0.3438
Epoch 00013: val_loss improved from 6.75467 to 6.30064, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 18ms/step - loss: 21.7970 - accuracy:
0.4667 - val_loss: 6.3006 - val_accuracy: 0.8125
Epoch 14/100
1/2 [==============>...] - ETA: 0s - loss: 27.3030 - accuracy:
0.5625
Epoch 00014: val_loss improved from 6.30064 to 5.78387, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 11ms/step - loss: 20.3356 - accuracy:
0.5500 - val_loss: 5.7839 - val_accuracy: 0.8125
Epoch 15/100
1/2 [==============>...] - ETA: 0s - loss: 13.9225 - accuracy:
0.6562
Epoch 00015: val_loss improved from 5.78387 to 5.38274, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 12ms/step - loss: 12.7252 - accuracy:
0.6667 - val_loss: 5.3827 - val_accuracy: 0.8125
Epoch 16/100
1/2 [==============>...] - ETA: 0s - loss: 10.0440 - accuracy:
0.6562
Epoch 00016: val_loss improved from 5.38274 to 5.06205, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 11ms/step - loss: 12.3554 - accuracy:
0.6000 - val_loss: 5.0620 - val_accuracy: 0.8125
Epoch 17/100
1/2 [==============>...] - ETA: 0s - loss: 12.1743 - accuracy:
0.6875
Epoch 00017: val_loss improved from 5.06205 to 4.63649, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 19ms/step - loss: 12.9251 - accuracy:
0.6000 - val_loss: 4.6365 - val_accuracy: 0.8125
Epoch 18/100
1/2 [==============>...] - ETA: 0s - loss: 19.9110 - accuracy:
0.5000
Epoch 00018: val_loss improved from 4.63649 to 4.13732, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 20ms/step - loss: 16.2596 - accuracy:
0.5833 - val_loss: 4.1373 - val_accuracy: 0.8125
Epoch 19/100
1/2 [==============>...] - ETA: 0s - loss: 13.1973 - accuracy:
0.5625
Epoch 00019: val_loss improved from 4.13732 to 3.73905, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 22ms/step - loss: 11.2965 - accuracy:
0.5500 - val_loss: 3.7391 - val_accuracy: 0.8125
Epoch 20/100
```

```
1/2 [==============>...] - ETA: 0s - loss: 9.3794 - accuracy: 0.5625
Epoch 00020: val_loss improved from 3.73905 to 3.49211, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 29ms/step - loss: 7.2179 - accuracy:
0.6833 - val_loss: 3.4921 - val_accuracy: 0.8125
Epoch 21/100
1/2 [==============>...] - ETA: 0s - loss: 11.0303 - accuracy:
0.4688
Epoch 00021: val_loss improved from 3.49211 to 3.20805, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 20ms/step - loss: 11.0809 - accuracy:
0.5667 - val_loss: 3.2081 - val_accuracy: 0.8125
Epoch 22/100
1/2 [==============>...] - ETA: 0s - loss: 17.4370 - accuracy:
0.4688
Epoch 00022: val_loss improved from 3.20805 to 3.04916, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 20ms/step - loss: 17.0887 - accuracy:
0.4333 - val_loss: 3.0492 - val_accuracy: 0.8125
Epoch 23/100
1/2 [==============>...] - ETA: 0s - loss: 12.2579 - accuracy:
0.5000
Epoch 00023: val_loss improved from 3.04916 to 2.97421, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 18ms/step - loss: 9.5124 - accuracy:
0.5667 - val_loss: 2.9742 - val_accuracy: 0.8125
Epoch 24/100
1/2 [==============>...] - ETA: 0s - loss: 13.8954 - accuracy:
0.5625
Epoch 00024: val_loss did not improve from 2.97421
2/2 [==============================] - 0s 17ms/step - loss: 13.5872 - accuracy:
0.5167 - val_loss: 2.9755 - val_accuracy: 0.8125
Epoch 25/100
1/2 [==============>...] - ETA: 0s - loss: 13.8293 - accuracy:
0.5625
Epoch 00025: val_loss did not improve from 2.97421
2/2 [==============================] - 0s 16ms/step - loss: 13.2053 - accuracy:
0.5500 - val_loss: 3.0221 - val_accuracy: 0.8125
Epoch 26/100
1/2 [==============>...] - ETA: 0s - loss: 11.0488 - accuracy:
0.6875
Epoch 00026: val_loss improved from 2.97421 to 2.96807, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 18ms/step - loss: 16.2217 - accuracy:
0.6167 - val_loss: 2.9681 - val_accuracy: 0.8125
Epoch 27/100
1/2 [==============>...] - ETA: 0s - loss: 18.0691 - accuracy:
0.3750
```

```
Epoch 00027: val_loss improved from 2.96807 to 2.86407, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 16ms/step - loss: 12.3198 - accuracy:
0.5333 - val_loss: 2.8641 - val_accuracy: 0.8125
Epoch 28/100
1/2 [==============>...] - ETA: 0s - loss: 10.5688 - accuracy:
0.6562
Epoch 00028: val_loss improved from 2.86407 to 2.73149, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 17ms/step - loss: 14.2804 - accuracy:
0.5833 - val_loss: 2.7315 - val_accuracy: 0.8125
Epoch 29/100
1/2 [==============>...] - ETA: 0s - loss: 13.4805 - accuracy:
0.6250
Epoch 00029: val_loss improved from 2.73149 to 2.52921, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 12ms/step - loss: 13.0348 - accuracy:
0.5833 - val_loss: 2.5292 - val_accuracy: 0.8125
Epoch 30/100
1/2 [==============>...] - ETA: 0s - loss: 14.6969 - accuracy:
0.4688
Epoch 00030: val_loss improved from 2.52921 to 2.30720, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 16ms/step - loss: 11.6462 - accuracy:
0.5333 - val_loss: 2.3072 - val_accuracy: 0.8125
Epoch 31/100
1/2 [==============>...] - ETA: 0s - loss: 10.4836 - accuracy:
0.6562
Epoch 00031: val_loss improved from 2.30720 to 2.11066, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 15ms/step - loss: 13.2458 - accuracy:
0.6333 - val_loss: 2.1107 - val_accuracy: 0.8125
Epoch 32/100
1/2 [==============>...] - ETA: 0s - loss: 11.6687 - accuracy:
0.7188
Epoch 00032: val_loss improved from 2.11066 to 1.87172, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 14ms/step - loss: 10.3458 - accuracy:
0.6833 - val_loss: 1.8717 - val_accuracy: 0.8125
Epoch 33/100
1/2 [==============>...] - ETA: 0s - loss: 7.7896 - accuracy: 0.6250
Epoch 00033: val_loss improved from 1.87172 to 1.53973, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 14ms/step - loss: 5.3654 - accuracy:
0.7167 - val_loss: 1.5397 - val_accuracy: 0.8125
Epoch 34/100
1/2 [==============>...] - ETA: 0s - loss: 4.7999 - accuracy: 0.6875
Epoch 00034: val_loss improved from 1.53973 to 1.17503, saving model to
```

```
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 16ms/step - loss: 6.7195 - accuracy:
0.6667 - val_loss: 1.1750 - val_accuracy: 0.8125
Epoch 35/100
1/2 [==============>...] - ETA: 0s - loss: 8.1551 - accuracy: 0.5312
Epoch 00035: val_loss improved from 1.17503 to 0.90417, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 14ms/step - loss: 10.4978 - accuracy:
0.4667 - val_loss: 0.9042 - val_accuracy: 0.8125
Epoch 36/100
1/2 [==============>...] - ETA: 0s - loss: 7.3576 - accuracy: 0.6250
Epoch 00036: val_loss improved from 0.90417 to 0.73361, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 14ms/step - loss: 7.9317 - accuracy:
0.5500 - val_loss: 0.7336 - val_accuracy: 0.8125
Epoch 37/100
1/2 [==============>...] - ETA: 0s - loss: 11.9080 - accuracy:
0.4375
Epoch 00037: val_loss improved from 0.73361 to 0.66288, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 15ms/step - loss: 9.3931 - accuracy:
0.4667 - val_loss: 0.6629 - val_accuracy: 0.8125
Epoch 38/100
1/2 [==============>...] - ETA: 0s - loss: 7.2464 - accuracy: 0.5312
Epoch 00038: val_loss improved from 0.66288 to 0.59630, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 14ms/step - loss: 5.5952 - accuracy:
0.6000 - val_loss: 0.5963 - val_accuracy: 0.8125
Epoch 39/100
1/2 [==============>...] - ETA: 0s - loss: 10.0022 - accuracy:
0.5000
Epoch 00039: val_loss improved from 0.59630 to 0.57890, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 15ms/step - loss: 8.7332 - accuracy:
0.5167 - val_loss: 0.5789 - val_accuracy: 0.8125
Epoch 40/100
1/2 [==============>...] - ETA: 0s - loss: 9.4766 - accuracy: 0.5625
Epoch 00040: val_loss did not improve from 0.57890
2/2 [==============================] - 0s 14ms/step - loss: 9.4784 - accuracy:
0.6000 - val_loss: 0.5984 - val_accuracy: 0.8125
Epoch 41/100
1/2 [==============>...] - ETA: 0s - loss: 7.2777 - accuracy: 0.5938
Epoch 00041: val_loss did not improve from 0.57890
2/2 [==============================] - 0s 16ms/step - loss: 5.7536 - accuracy:
0.6000 - val_loss: 0.6867 - val_accuracy: 0.8125
Epoch 42/100
1/2 [==============>...] - ETA: 0s - loss: 7.3021 - accuracy: 0.6562
Epoch 00042: val_loss did not improve from 0.57890
```

```
2/2 [==============================] - 0s 17ms/step - loss: 6.6863 - accuracy:
0.6167 - val_loss: 0.7765 - val_accuracy: 0.8125
Epoch 43/100
1/2 [==============>...] - ETA: 0s - loss: 11.1017 - accuracy:
0.4375
Epoch 00043: val_loss did not improve from 0.57890
2/2 [==============================] - 0s 14ms/step - loss: 10.3718 - accuracy:
0.4500 - val_loss: 0.8545 - val_accuracy: 0.8125
Epoch 44/100
1/2 [==============>...] - ETA: 0s - loss: 4.7656 - accuracy: 0.6250
Epoch 00044: val_loss did not improve from 0.57890
2/2 [==============================] - 0s 14ms/step - loss: 5.2127 - accuracy:
0.6500 - val_loss: 0.9068 - val_accuracy: 0.8125
Epoch 45/100
1/2 [==============>...] - ETA: 0s - loss: 2.6990 - accuracy: 0.7188
Epoch 00045: val_loss did not improve from 0.57890
2/2 [==============================] - 0s 13ms/step - loss: 5.5977 - accuracy:
0.6833 - val_loss: 0.8943 - val_accuracy: 0.8125
Epoch 46/100
1/2 [==============>...] - ETA: 0s - loss: 6.3255 - accuracy: 0.5312
Epoch 00046: val_loss did not improve from 0.57890
2/2 [==============================] - 0s 13ms/step - loss: 6.3898 - accuracy:
0.5833 - val_loss: 0.8437 - val_accuracy: 0.8125
Epoch 47/100
1/2 [==============>...] - ETA: 0s - loss: 3.1210 - accuracy: 0.8125
Epoch 00047: val_loss did not improve from 0.57890
2/2 [==============================] - 0s 13ms/step - loss: 5.8009 - accuracy:
0.6500 - val_loss: 0.7846 - val_accuracy: 0.8125
Epoch 48/100
1/2 [==============>...] - ETA: 0s - loss: 7.2163 - accuracy: 0.6875
Epoch 00048: val_loss did not improve from 0.57890
2/2 [==============================] - 0s 14ms/step - loss: 6.0770 - accuracy:
0.5667 - val_loss: 0.7170 - val_accuracy: 0.8125
Epoch 49/100
1/2 [==============>...] - ETA: 0s - loss: 8.3558 - accuracy: 0.5000
Epoch 00049: val_loss did not improve from 0.57890
2/2 [==============================] - 0s 14ms/step - loss: 9.6683 - accuracy:
0.5000 - val_loss: 0.6890 - val_accuracy: 0.8125
Epoch 50/100
1/2 [==============>...] - ETA: 0s - loss: 6.5741 - accuracy: 0.5938
Epoch 00050: val_loss did not improve from 0.57890
2/2 [==============================] - 0s 14ms/step - loss: 5.7107 - accuracy:
0.5500 - val_loss: 0.6579 - val_accuracy: 0.8125
Epoch 51/100
1/2 [==============>...] - ETA: 0s - loss: 5.2222 - accuracy: 0.6250
Epoch 00051: val_loss did not improve from 0.57890
2/2 [==============================] - 0s 17ms/step - loss: 5.7784 - accuracy:
0.5500 - val_loss: 0.6380 - val_accuracy: 0.8125
```

```
Epoch 52/100
1/2 [==============>...] - ETA: 0s - loss: 3.0498 - accuracy: 0.6875
Epoch 00052: val_loss did not improve from 0.57890
2/2 [==============================] - 0s 14ms/step - loss: 2.5694 - accuracy:
0.7500 - val_loss: 0.5970 - val_accuracy: 0.8125
Epoch 53/100
1/2 [==============>...] - ETA: 0s - loss: 3.5603 - accuracy: 0.7500
Epoch 00053: val_loss improved from 0.57890 to 0.55070, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 14ms/step - loss: 4.2912 - accuracy:
0.6667 - val_loss: 0.5507 - val_accuracy: 0.8125
Epoch 54/100
1/2 [==============>...] - ETA: 0s - loss: 7.9270 - accuracy: 0.5000
Epoch 00054: val_loss improved from 0.55070 to 0.51964, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 16ms/step - loss: 6.9573 - accuracy:
0.5000 - val_loss: 0.5196 - val_accuracy: 0.8125
Epoch 55/100
1/2 [==============>...] - ETA: 0s - loss: 7.0801 - accuracy: 0.5625
Epoch 00055: val_loss improved from 0.51964 to 0.49940, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 15ms/step - loss: 6.8827 - accuracy:
0.5500 - val_loss: 0.4994 - val_accuracy: 0.8125
Epoch 56/100
1/2 [==============>...] - ETA: 0s - loss: 4.9611 - accuracy: 0.6875
Epoch 00056: val_loss improved from 0.49940 to 0.48087, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 15ms/step - loss: 5.5080 - accuracy:
0.6167 - val_loss: 0.4809 - val_accuracy: 0.8125
Epoch 57/100
1/2 [==============>...] - ETA: 0s - loss: 3.7283 - accuracy: 0.5625
Epoch 00057: val_loss improved from 0.48087 to 0.46965, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 17ms/step - loss: 4.3224 - accuracy:
0.5667 - val_loss: 0.4696 - val_accuracy: 0.8125
Epoch 58/100
1/2 [==============>...] - ETA: 0s - loss: 6.8397 - accuracy: 0.4688
Epoch 00058: val_loss improved from 0.46965 to 0.46302, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 15ms/step - loss: 4.9102 - accuracy:
0.5333 - val_loss: 0.4630 - val_accuracy: 0.8125
Epoch 59/100
1/2 [==============>...] - ETA: 0s - loss: 4.0426 - accuracy: 0.6562
Epoch 00059: val_loss improved from 0.46302 to 0.45740, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 14ms/step - loss: 3.9101 - accuracy:
0.6000 - val_loss: 0.4574 - val_accuracy: 0.8125
Epoch 60/100
```

```
1/2 [==============>...] - ETA: 0s - loss: 4.2587 - accuracy: 0.6875
Epoch 00060: val_loss improved from 0.45740 to 0.44633, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 15ms/step - loss: 4.4624 - accuracy:
0.6333 - val_loss: 0.4463 - val_accuracy: 0.8125
Epoch 61/100
1/2 [==============>...] - ETA: 0s - loss: 2.9390 - accuracy: 0.6250
Epoch 00061: val_loss improved from 0.44633 to 0.43939, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 14ms/step - loss: 3.3851 - accuracy:
0.6333 - val_loss: 0.4394 - val_accuracy: 0.8125
Epoch 62/100
1/2 [==============>...] - ETA: 0s - loss: 3.5711 - accuracy: 0.5938
Epoch 00062: val_loss improved from 0.43939 to 0.43893, saving model to
saved_models\audio_classification.hdf5
2/2 [==============================] - 0s 16ms/step - loss: 3.4447 - accuracy:
0.5667 - val_loss: 0.4389 - val_accuracy: 0.8125
Epoch 63/100
1/2 [==============>...] - ETA: 0s - loss: 4.6707 - accuracy: 0.5625
Epoch 00063: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 14ms/step - loss: 5.0923 - accuracy:
0.5500 - val_loss: 0.4402 - val_accuracy: 0.8125
Epoch 64/100
1/2 [==============>...] - ETA: 0s - loss: 7.1201 - accuracy: 0.4375
Epoch 00064: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 14ms/step - loss: 5.3719 - accuracy:
0.5333 - val_loss: 0.4436 - val_accuracy: 0.8125
Epoch 65/100
1/2 [==============>...] - ETA: 0s - loss: 5.2278 - accuracy: 0.4688
Epoch 00065: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 15ms/step - loss: 4.5746 - accuracy:
0.5333 - val_loss: 0.4466 - val_accuracy: 0.8125
Epoch 66/100
1/2 [==============>...] - ETA: 0s - loss: 4.1735 - accuracy: 0.5625
Epoch 00066: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 14ms/step - loss: 4.0351 - accuracy:
0.5500 - val_loss: 0.4516 - val_accuracy: 0.8125
Epoch 67/100
1/2 [==============>...] - ETA: 0s - loss: 5.8278 - accuracy: 0.5938
Epoch 00067: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 13ms/step - loss: 3.9869 - accuracy:
0.6833 - val_loss: 0.4572 - val_accuracy: 0.8125
Epoch 68/100
1/2 [==============>...] - ETA: 0s - loss: 3.9165 - accuracy: 0.5938
Epoch 00068: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 12ms/step - loss: 3.8983 - accuracy:
0.5833 - val_loss: 0.4636 - val_accuracy: 0.8125
Epoch 69/100
```

```
1/2 [==============>...] - ETA: 0s - loss: 3.4035 - accuracy: 0.6250
Epoch 00069: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 15ms/step - loss: 3.7185 - accuracy:
0.6333 - val_loss: 0.4649 - val_accuracy: 0.8125
Epoch 70/100
1/2 [==============>...] - ETA: 0s - loss: 5.5184 - accuracy: 0.5625
Epoch 00070: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 14ms/step - loss: 4.2220 - accuracy:
0.6333 - val_loss: 0.4621 - val_accuracy: 0.8125
Epoch 71/100
1/2 [==============>...] - ETA: 0s - loss: 8.2980 - accuracy: 0.5312
Epoch 00071: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 13ms/step - loss: 5.2883 - accuracy:
0.5833 - val_loss: 0.4566 - val_accuracy: 0.8125
Epoch 72/100
1/2 [==============>...] - ETA: 0s - loss: 4.9241 - accuracy: 0.5938
Epoch 00072: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 16ms/step - loss: 4.9263 - accuracy:
0.5500 - val_loss: 0.4566 - val_accuracy: 0.8125
Epoch 73/100
1/2 [==============>...] - ETA: 0s - loss: 4.2102 - accuracy: 0.5938
Epoch 00073: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 14ms/step - loss: 4.6173 - accuracy:
0.5833 - val_loss: 0.4634 - val_accuracy: 0.8125
Epoch 74/100
1/2 [==============>...] - ETA: 0s - loss: 2.9167 - accuracy: 0.6250
Epoch 00074: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 17ms/step - loss: 2.4915 - accuracy:
0.6500 - val_loss: 0.4778 - val_accuracy: 0.8125
Epoch 75/100
1/2 [==============>...] - ETA: 0s - loss: 3.0491 - accuracy: 0.7500
Epoch 00075: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 14ms/step - loss: 4.8684 - accuracy:
0.6667 - val_loss: 0.4917 - val_accuracy: 0.8125
Epoch 76/100
1/2 [==============>...] - ETA: 0s - loss: 3.6178 - accuracy: 0.5938
Epoch 00076: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 15ms/step - loss: 2.9971 - accuracy:
0.6500 - val_loss: 0.5025 - val_accuracy: 0.8125
Epoch 77/100
1/2 [==============>...] - ETA: 0s - loss: 3.6507 - accuracy: 0.5625
Epoch 00077: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 15ms/step - loss: 3.1480 - accuracy:
0.5667 - val_loss: 0.5068 - val_accuracy: 0.8125
Epoch 78/100
1/2 [==============>...] - ETA: 0s - loss: 3.4248 - accuracy: 0.5938
Epoch 00078: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 14ms/step - loss: 3.0981 - accuracy:
```

```
0.6500 - val_loss: 0.5146 - val_accuracy: 0.8125
Epoch 79/100
1/2 [==============>...] - ETA: 0s - loss: 4.5334 - accuracy: 0.5312
Epoch 00079: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 16ms/step - loss: 3.5876 - accuracy:
0.5333 - val_loss: 0.5141 - val_accuracy: 0.8125
Epoch 80/100
1/2 [==============>...] - ETA: 0s - loss: 3.2643 - accuracy: 0.5938
Epoch 00080: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 12ms/step - loss: 3.2287 - accuracy:
0.5167 - val_loss: 0.5117 - val_accuracy: 0.8125
Epoch 81/100
1/2 [==============>...] - ETA: 0s - loss: 3.9764 - accuracy: 0.5938
Epoch 00081: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 12ms/step - loss: 3.7867 - accuracy:
0.5667 - val_loss: 0.5015 - val_accuracy: 0.8125
Epoch 82/100
1/2 [==============>...] - ETA: 0s - loss: 1.2729 - accuracy: 0.6875
Epoch 00082: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 14ms/step - loss: 2.8372 - accuracy:
0.6333 - val_loss: 0.4849 - val_accuracy: 0.8125
Epoch 83/100
1/2 [==============>...] - ETA: 0s - loss: 2.3236 - accuracy: 0.6562
Epoch 00083: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 16ms/step - loss: 2.6647 - accuracy:
0.5833 - val_loss: 0.4743 - val_accuracy: 0.8125
Epoch 84/100
1/2 [==============>...] - ETA: 0s - loss: 3.3246 - accuracy: 0.5312
Epoch 00084: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 18ms/step - loss: 2.5988 - accuracy:
0.6500 - val_loss: 0.4732 - val_accuracy: 0.8125
Epoch 85/100
1/2 [==============>...] - ETA: 0s - loss: 1.2611 - accuracy: 0.8125
Epoch 00085: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 14ms/step - loss: 1.7210 - accuracy:
0.7333 - val_loss: 0.4754 - val_accuracy: 0.8125
Epoch 86/100
1/2 [==============>...] - ETA: 0s - loss: 4.1572 - accuracy: 0.5000
Epoch 00086: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 14ms/step - loss: 4.3858 - accuracy:
0.5167 - val_loss: 0.4768 - val_accuracy: 0.8125
Epoch 87/100
1/2 [==============>...] - ETA: 0s - loss: 2.7888 - accuracy: 0.7188
Epoch 00087: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 14ms/step - loss: 2.9916 - accuracy:
0.6667 - val_loss: 0.4818 - val_accuracy: 0.8125
Epoch 88/100
1/2 [==============>...] - ETA: 0s - loss: 1.9977 - accuracy: 0.5938
```

```
Epoch 00088: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 13ms/step - loss: 2.4140 - accuracy:
0.5667 - val_loss: 0.4843 - val_accuracy: 0.8125
Epoch 89/100
1/2 [==============>...] - ETA: 0s - loss: 1.7309 - accuracy: 0.6875
Epoch 00089: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 13ms/step - loss: 1.8502 - accuracy:
0.6500 - val_loss: 0.4868 - val_accuracy: 0.8125
Epoch 90/100
1/2 [==============>...] - ETA: 0s - loss: 3.4124 - accuracy: 0.4688
Epoch 00090: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 13ms/step - loss: 3.1764 - accuracy:
0.5167 - val_loss: 0.4895 - val_accuracy: 0.8125
Epoch 91/100
1/2 [==============>...] - ETA: 0s - loss: 3.3886 - accuracy: 0.5938
Epoch 00091: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 15ms/step - loss: 3.3258 - accuracy:
0.5667 - val_loss: 0.4930 - val_accuracy: 0.8125
Epoch 92/100
1/2 [==============>...] - ETA: 0s - loss: 1.9497 - accuracy: 0.5625
Epoch 00092: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 14ms/step - loss: 3.1150 - accuracy:
0.5667 - val_loss: 0.5013 - val_accuracy: 0.8125
Epoch 93/100
1/2 [==============>...] - ETA: 0s - loss: 4.5654 - accuracy: 0.5000
Epoch 00093: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 15ms/step - loss: 3.3720 - accuracy:
0.6167 - val_loss: 0.5121 - val_accuracy: 0.8125
Epoch 94/100
1/2 [==============>...] - ETA: 0s - loss: 2.3280 - accuracy: 0.6250
Epoch 00094: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 16ms/step - loss: 1.6326 - accuracy:
0.7000 - val_loss: 0.5235 - val_accuracy: 0.8125
Epoch 95/100
1/2 [==============>...] - ETA: 0s - loss: 2.5301 - accuracy: 0.6875
Epoch 00095: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 18ms/step - loss: 2.8884 - accuracy:
0.6500 - val_loss: 0.5353 - val_accuracy: 0.8125
Epoch 96/100
1/2 [==============>...] - ETA: 0s - loss: 5.4936 - accuracy: 0.6875
Epoch 00096: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 13ms/step - loss: 4.4232 - accuracy:
0.6167 - val_loss: 0.5504 - val_accuracy: 0.8125
Epoch 97/100
1/2 [==============>...] - ETA: 0s - loss: 2.1955 - accuracy: 0.5938
Epoch 00097: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 15ms/step - loss: 2.3177 - accuracy:
0.5833 - val_loss: 0.5654 - val_accuracy: 0.8125
```

```
Epoch 98/100
1/2 [==============>...] - ETA: 0s - loss: 1.5019 - accuracy: 0.6875
Epoch 00098: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 15ms/step - loss: 2.1007 - accuracy:
0.6333 - val_loss: 0.5779 - val_accuracy: 0.8125
Epoch 99/100
1/2 [==============>...] - ETA: 0s - loss: 2.5410 - accuracy: 0.4375
Epoch 00099: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 13ms/step - loss: 2.2894 - accuracy:
0.4833 - val_loss: 0.5924 - val_accuracy: 0.8125
Epoch 100/100
1/2 [==============>...] - ETA: 0s - loss: 1.6555 - accuracy: 0.7500
Epoch 00100: val_loss did not improve from 0.43893
2/2 [==============================] - 0s 15ms/step - loss: 2.0222 - accuracy:
0.6500 - val_loss: 0.6109 - val_accuracy: 0.8125
Training completed in time:  0:00:05.541014
```

Model accuracy

[30]:
```
test_accuracy=model.evaluate(X_test,y_test,verbose=0)
print(test_accuracy[1])
```

```
0.8125
```

Testing

[57]:
```
X_train[1]
```

[57]:
```
array([-4.9306253e+02,  1.4614683e+01, -2.3427277e+00,  4.7251754e+00,
       -4.9997411e+00, -5.0991964e+00, -9.8744669e+00, -6.1525526e+00,
       -4.7394080e+00, -8.1517029e-01, -3.9094977e+00, -3.0535016e+00,
       -1.5697505e+00,  1.9981915e+00,  9.9561042e-01,  1.5086237e+00,
        1.0312959e-03,  1.2510941e+00,  6.2790591e-01,  1.3772461e+00,
        4.6870396e-01,  1.7234601e+00, -8.3529478e-01,  1.4567796e+00,
        4.4083098e-01,  5.9845048e-01,  7.2546071e-01,  3.8946551e-01,
        1.9497240e-01,  6.3206202e-01, -4.1609910e-01,  8.3015215e-01,
        1.1107286e-01,  8.3418852e-01,  6.1854827e-01,  1.0216424e+00,
        7.0161372e-01,  1.0731262e+00, -7.8298204e-02,  8.9549601e-01],
      dtype=float32)
```

[56]:
```
np.argmax(model.predict(X_train), axis=-1)
```

[56]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
```

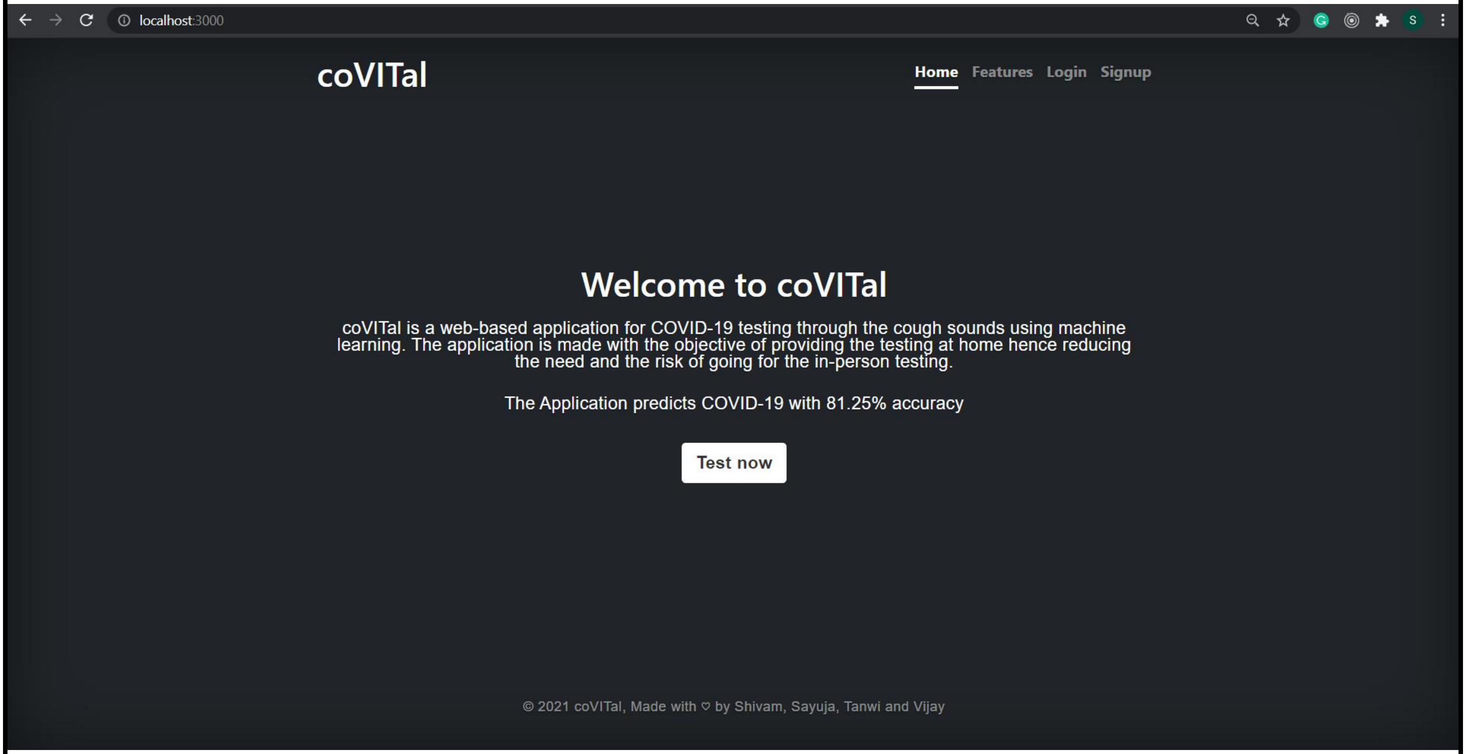
Testing with Some audio recordings of me coughing

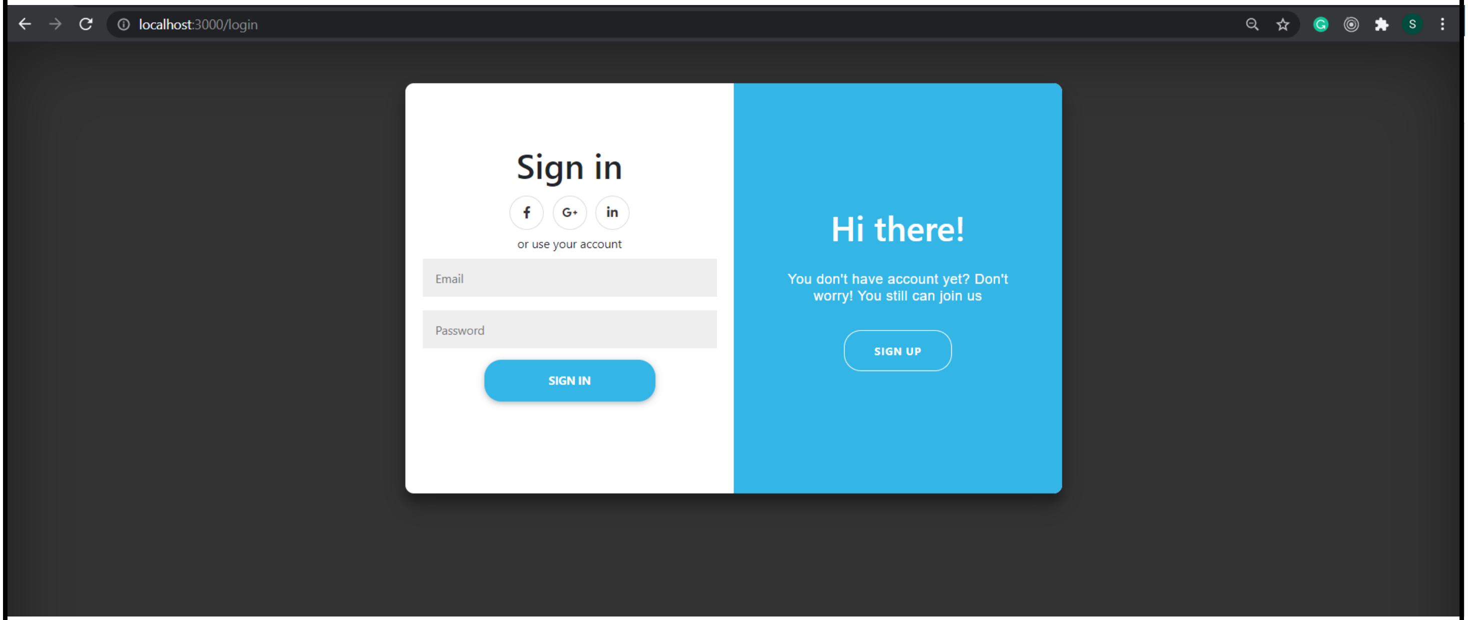
[59]:
```
filename="clinical\\shivam_cough.mp3"
audio, sample_rate = librosa.load(filename, res_type='kaiser_best')
mfccs_features = librosa.feature.mfcc(y=audio, sr=sample_rate, n_mfcc=40)
mfccs_scaled_features = np.mean(mfccs_features.T,axis=0)

print(mfccs_scaled_features)
mfccs_scaled_features=mfccs_scaled_features.reshape(1,-1)
print(mfccs_scaled_features)
print(mfccs_scaled_features.shape)
predicted_label=model.predict_classes(mfccs_scaled_features)
print(predicted_label)
prediction_class = labelencoder.inverse_transform(predicted_label)
prediction_class
```

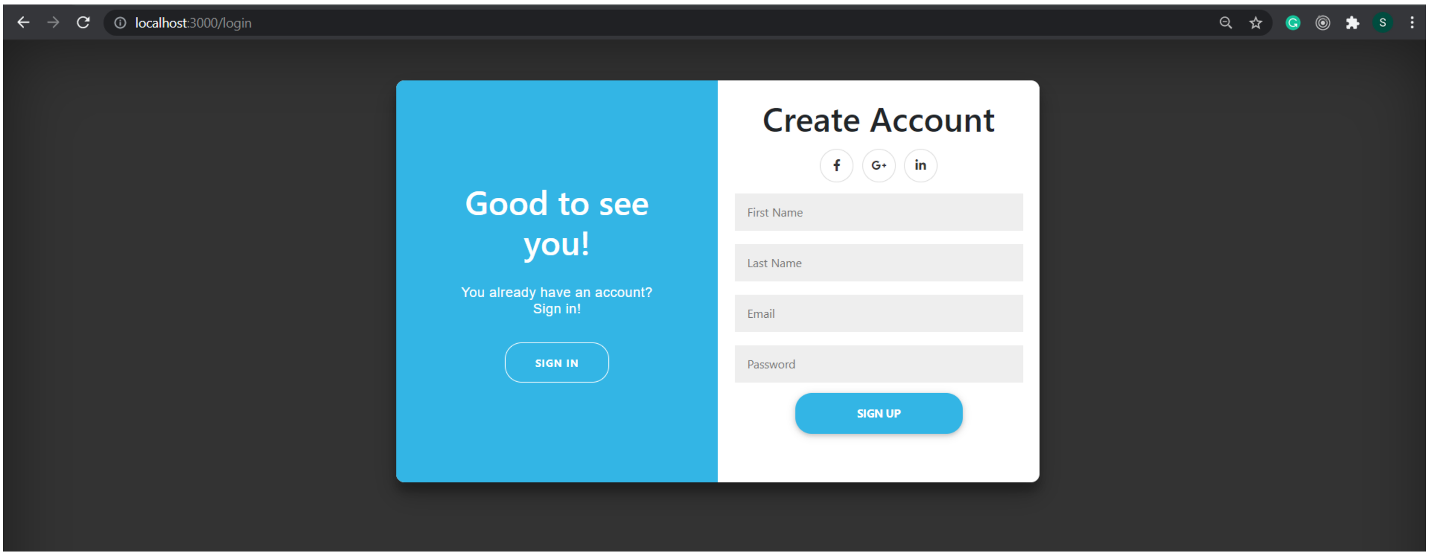
```
[-3.5887982e+02  6.8352638e+01  8.9355345e+00  2.0273815e+01
 -8.8841134e-01  3.2996807e+00 -2.0065159e+01 -5.9433098e+00
 -7.9503412e+00  1.8832532e-01 -1.0156484e+01 -1.2615176e+01
 -1.5022754e+01 -5.7426333e+00 -5.3318629e+00 -6.2207584e+00
 -1.3299288e+01 -3.2086434e+00 -9.2501783e+00 -7.0228310e+00
 -6.7276292e+00 -5.7607741e+00 -4.7696905e+00 -2.0750031e+00
 -5.4737725e+00 -8.8357725e+00 -8.0440655e+00 -1.9138778e+00
 -6.2845793e+00 -1.3863666e-01 -3.5454631e+00 -9.6129227e-01
 -1.4628594e+00 -1.6901464e+00 -1.4948845e+00 -1.6255490e+00
 -6.5643656e-01  4.0409222e-01  4.7367591e-01  1.5002842e+00]
[[-3.5887982e+02  6.8352638e+01  8.9355345e+00  2.0273815e+01
  -8.8841134e-01  3.2996807e+00 -2.0065159e+01 -5.9433098e+00
  -7.9503412e+00  1.8832532e-01 -1.0156484e+01 -1.2615176e+01
  -1.5022754e+01 -5.7426333e+00 -5.3318629e+00 -6.2207584e+00
  -1.3299288e+01 -3.2086434e+00 -9.2501783e+00 -7.0228310e+00
  -6.7276292e+00 -5.7607741e+00 -4.7696905e+00 -2.0750031e+00
  -5.4737725e+00 -8.8357725e+00 -8.0440655e+00 -1.9138778e+00
  -6.2845793e+00 -1.3863666e-01 -3.5454631e+00 -9.6129227e-01
  -1.4628594e+00 -1.6901464e+00 -1.4948845e+00 -1.6255490e+00
  -6.5643656e-01  4.0409222e-01  4.7367591e-01  1.5002842e+00]]
(1, 40)
[0]
```

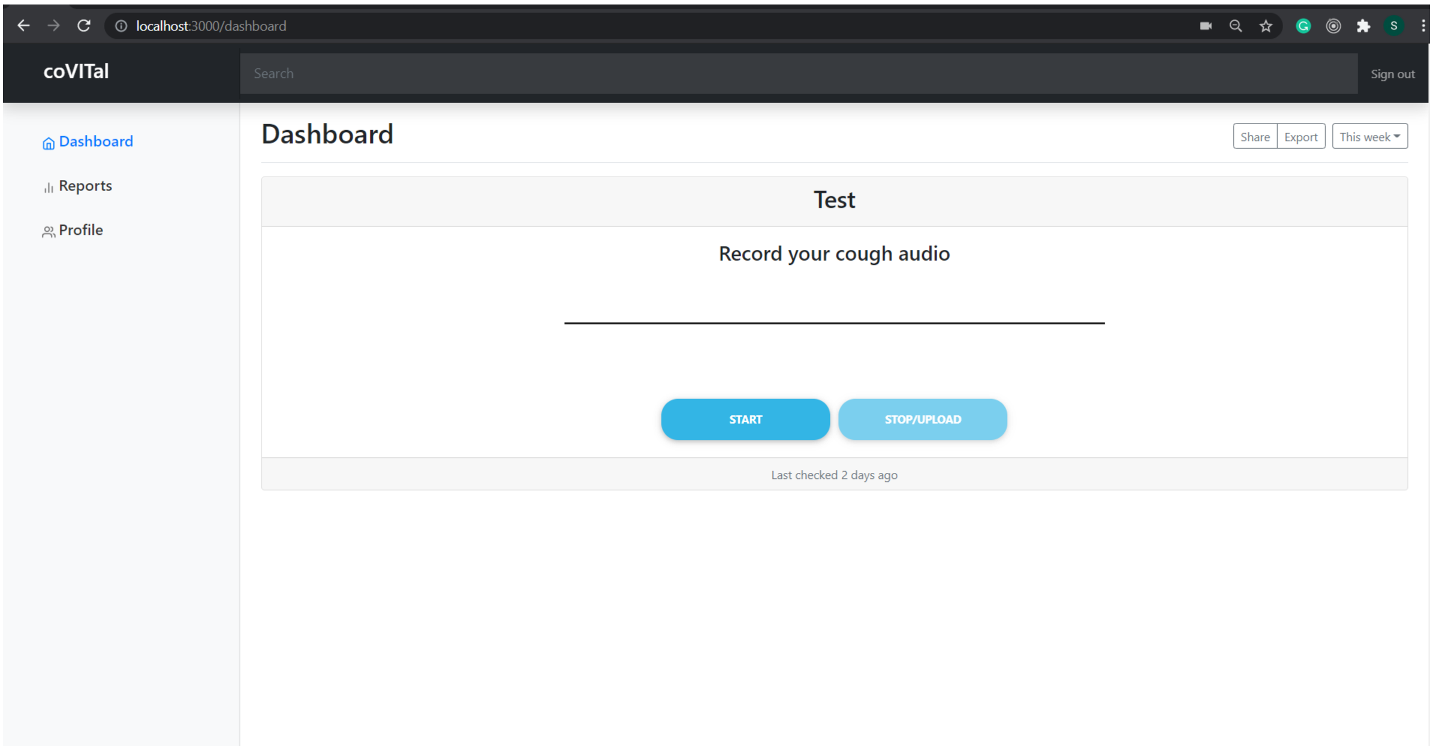
[59]:
```
array(['negative'], dtype='<U8')
```

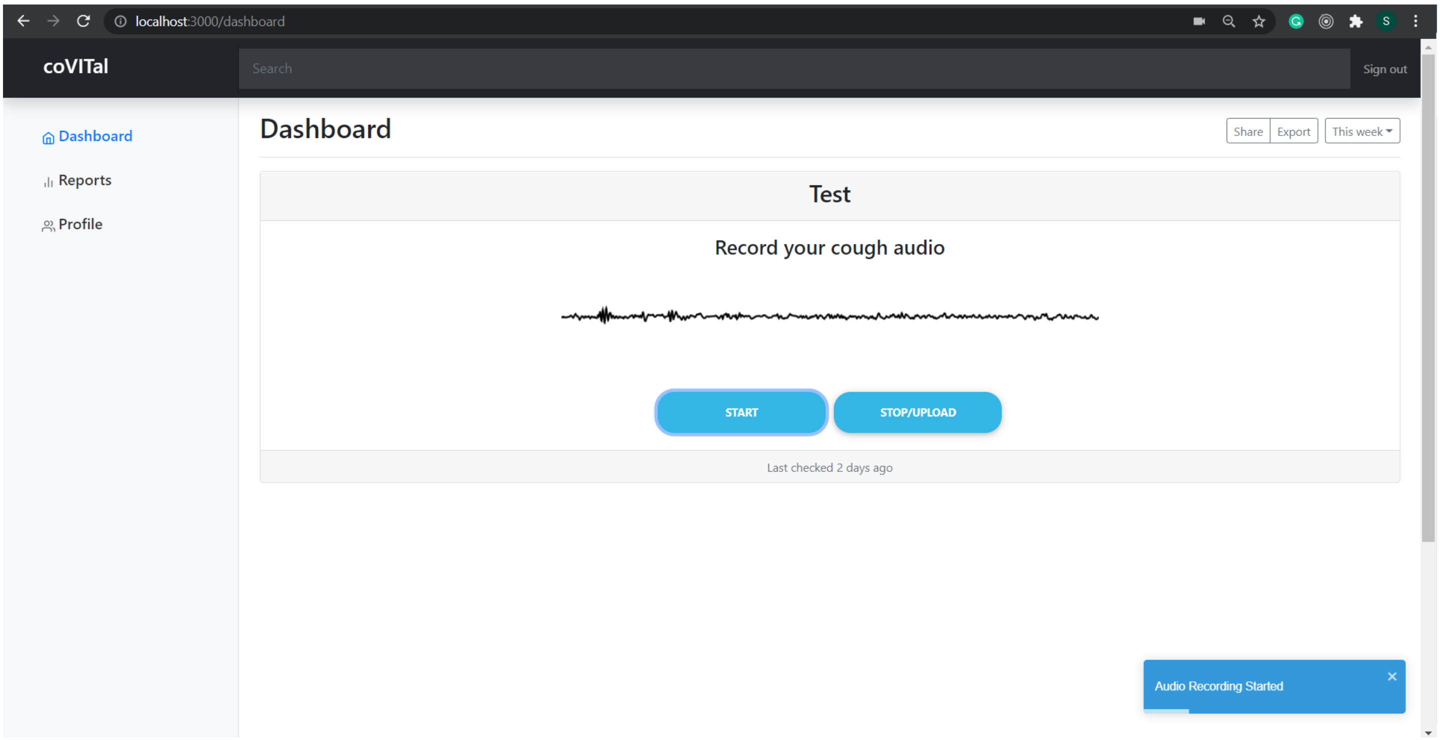
Application Frontend Screenshot

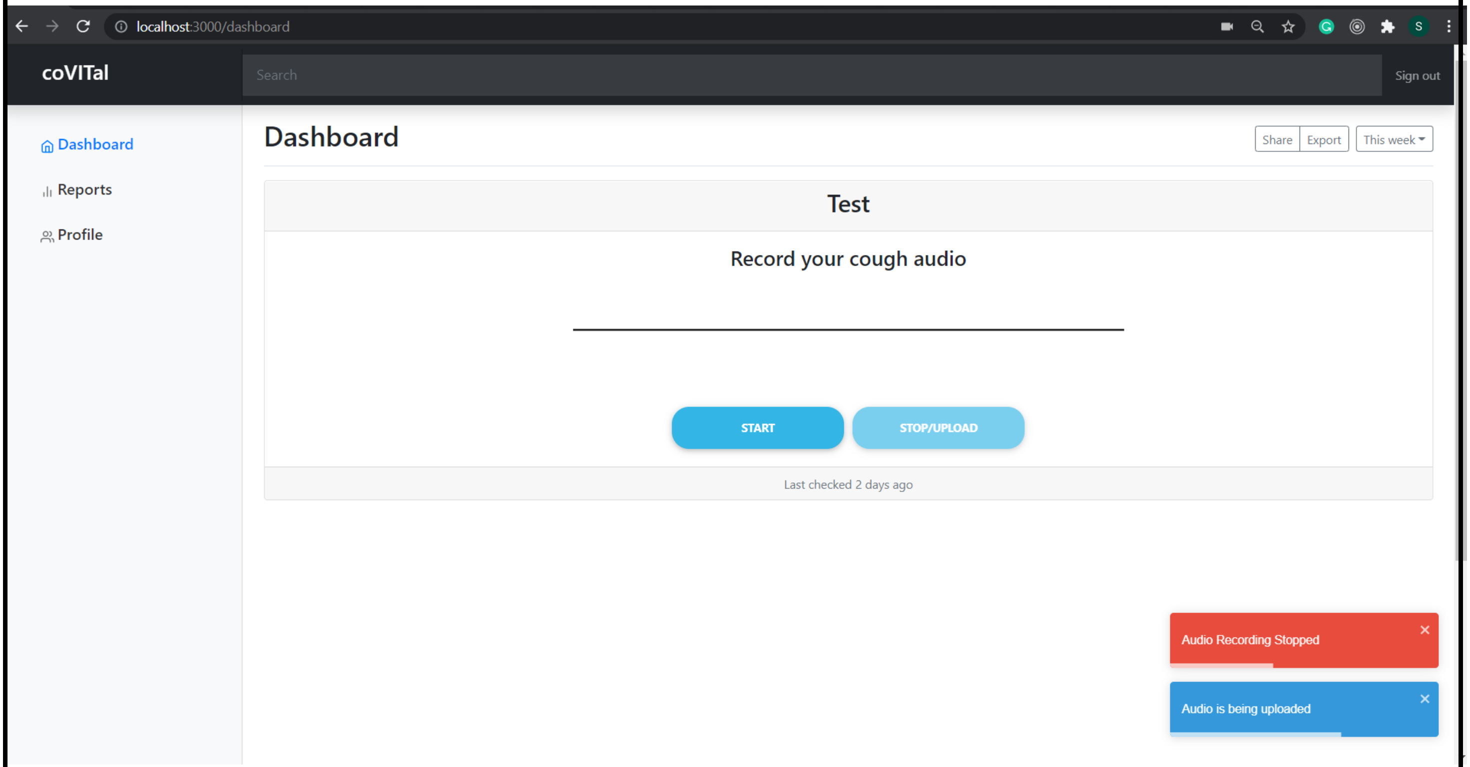


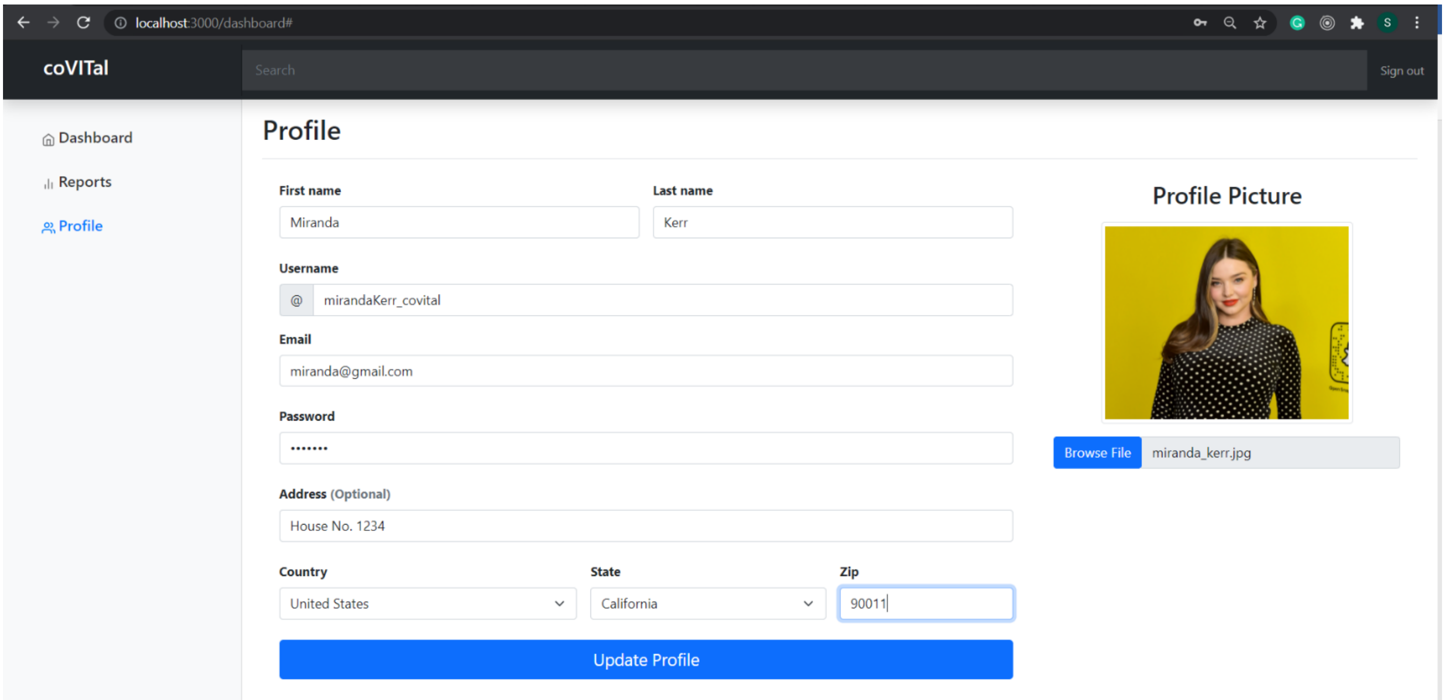


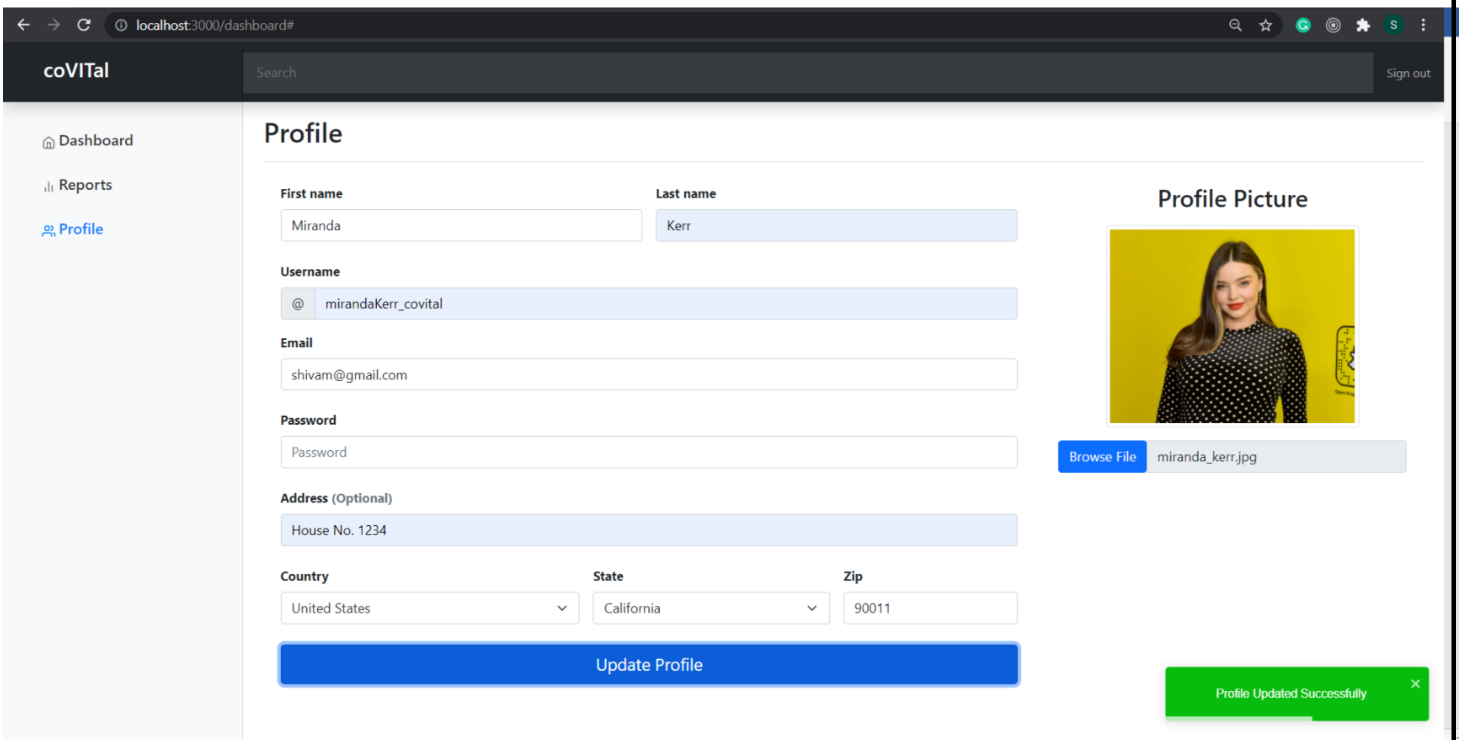

localhost:3000/login
Good to see you!
You already have an account? Sign in!
SIGN IN
Create Account
First Name
Last Name
Email
Password
SIGN UP


localhost:3000/dashboard
coVITal
Search
Sign out
Dashboard
Reports
Profile
Dashboard
Share
Export
This week
Test
Record your cough audio
START
STOP/UPLOAD
Last checked 2 days ago


localhost:3000/dashboard
coVITal
Search
Sign out
Dashboard
Reports
Profile
Dashboard
Share
Export
This week
Test
Record your cough audio
START
STOP/UPLOAD
Last checked 2 days ago
Audio Recording Started


localhost:3000/dashboard
coVITal
Search
Sign out
Dashboard
Reports
Profile
Dashboard
Share
Export
This week
Test
Record your cough audio
START
STOP/UPLOAD
Last checked 2 days ago
Audio Recording Stopped
Audio is being uploaded


localhost:3000/dashboard#
coVITal
Search
Sign out
Dashboard
Reports
Profile
Profile
First name
Miranda
Last name
Kerr
Username
@
mirandaKerr_covital
Email
miranda@gmail.com
Password
Address (Optional)
House No. 1234
Country
United States
State
California
Zip
90011
Update Profile
Profile Picture
Browse File
miranda_kerr.jpg


localhost:3000/dashboard#
coVITal
Search
Sign out
Dashboard
Reports
Profile
Profile
First name
Miranda
Last name
Kerr
Username
@
mirandaKerr_covital
Email
shivam@gmail.com
Password
Password
Address (Optional)
House No. 1234
Country
United States
State
California
Zip
90011
Update Profile
Profile Picture
Browse File
miranda_kerr.jpg
Profile Updated Successfully

Conclusion

There is no one that has not been affected in any way by COVID-19, which has taken over the planet. Researchers, medical industry researchers, and genetic scientists are now working harder than ever to find a cure for the disease. Artificial intelligence (AI) is one tool that could help keep this effort to evaluate researchers and scientists going. We compared findings and techniques introduced by each author in literature review section, which focuses on literature research for COVID-19 disease diagnosis using COVID-19 respiratory sounds data and various AI-based techniques.

The fundamental idea of the application was that cough sound may be utilized as a test medium for diagnosing a range of respiratory disorders using AI, according to our previous independent investigations. To investigate if this concept can be applied to COVID-19, we compare the pathomorphological changes induced by COVID-19 to those generated by other cough-causing medical illnesses. We notice that COVID-19 has a distinct effect on the respiratory system, and cough linked with it is likely to have distinct latent aspects as well.

On the basis of all this and using the medical domain knowledge as a foundation, we propose to develop coVITal which is a web-based application for COVID-19 testing through the cough sounds using machine learning. The application is made with the objective of providing the testing at home hence reducing the need and the risk of going for the in-person testing.

Despite the impressive performance of detecting covid with accuracy of 81.25%, coVITal is not meant to compete with clinical testing. Instead, it provides a one-of-a-kind functional instrument for fast, cost-effective, and most crucially, safe monitoring, tracing, and tracking, and thereby curbing the worldwide pandemic's wild spread by essentially allowing everyone to test. While we seek to improve the coVITal, the purpose of this research is to demonstrate a proof of concept in order to get community support for additional labelled data and large-scale experiments.

References

[1] Imran, A., Posokhova, I., Qureshi, H. N., Masood, U., Riaz, M. S., Ali, K., ... & Nabeel, M. (2020). AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app. *Informatics in Medicine Unlocked*, *20*, 100378.

[2] Brown, C., Chauhan, J., Grammenos, A., Han, J., Hasthanasombat, A., Spathis, D., ... & Mascolo, C. (2020, August). Exploring automatic diagnosis of covid-19 from crowdsourced respiratory sound data. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 3474-3484).

[3] Hassan, A., Shahin, I., & Alsabek, M. B. (2020, November). Covid-19 detection system using recurrent neural networks. In *2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)* (pp. 1-5). IEEE.

[4] Laguarta, J., Hueto, F., & Subirana, B. (2020). COVID-19 Artificial Intelligence Diagnosis using only Cough Recordings. IEEE Open Journal of Engineering in Medicine and Biology.

[5] Ritwik, K. V. S., Kalluri, S. B., & Vijayasenan, D. (2020). COVID-19 Patient Detection from Telephone Quality Speech Data. *arXiv preprint arXiv:2011.04299*.

[6] https://github.com/virufy/virufy-data

[7] https://github.com/hernanmd/COVID-19-train-audio