In [1]:

```
%load_ext nb_black
```

Import Section

In [8]:

```
import os
import json
from glob import glob
import numpy as np
import matplotlib.pyplot as plt
import re
import pandas as pd
from matplotlib import cm
import librosa
import pylab
import shutil
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Input
from sklearn.model_selection import train_test_split
import IPython.display as ipd
```

Dataset

In [4]:

```
DATASET_PATH = "COVID-19"
JSON_PATH = "metadata.json"
```

In [5]:

```
len(os.listdir(DATASET_PATH))  # total number of samples in the dataset
```

Out[5]:

```
422
```

In [6]:

```
single_sample = os.listdir(DATASET_PATH)[0]
```

Particular Dataset Information

In [7]:

```
os.listdir(os.path.join(DATASET_PATH, single_sample))
```

Out[7]:

```
['vowel-o.wav',
 'vowel-a.wav',
 'breathing-shallow.wav',
 'cough-shallow.wav',
 'vowel-e.wav',
 'cough-heavy.wav',
 'breathing-deep.wav',
 'counting-normal.wav',
 'metadata.json',
 'counting-fast.wav']
```

In [9]:

```
os.listdir(os.path.join(DATASET_PATH, single_sample))[-2]
```

Out[9]:

```
'metadata.json'
```

In [10]:

```
# read json file
metadata = json.load(open(os.path.join(DATASET_PATH, single_sample, JSON_
```

In [11]:

```
metadata
```

Out[11]:

```
{'a': 22,
 'covid_status': 'no_resp_illness_exposed',
 'dT': 'web',
 'ep': 'y',
 'fV': 2,
 'g': 'male',
 'l_c': 'India',
 'l_l': 'Ahmedabad',
 'l_s': 'Gujarat',
 'rU': 'n',
 'um': 'n'}
```

In [10]:

```
covid_status = set()

for path in os.listdir(DATASET_PATH):
    sample_path = os.path.join(DATASET_PATH, path)
    d = os.listdir(sample_path)
    if len(d) =/= 0:
        metadata = json.load(open(os.path.join(sample_path, JSON_PATH)))
        covid_stats = metadata["covid_status"]
        covid_status.add(covid_stats)
    else:
        os.rmdir(os.path.join(DATASET_PATH, path))
```

```
<IPython.core.display.Javascript object>
```

In [11]:

```
covid_status  # Total Categories
```

Out[11]:

```
{'healthy',
 'no_resp_illness_exposed',
 'positive_asymp',
 'positive_mild',
 'positive_moderate',
 'recovered_full',
 'resp_illness_not_identified'}
```

```
<IPython.core.display.Javascript object>
```

In [12]:

```
len(os.listdir(DATASET_PATH))  # 2 empty folder are removed
```

Out[12]:

```
422
```

```
<IPython.core.display.Javascript object>
```

Following Id's are contained None Signal or 0 length signal that's why we will remove following id's data.

- 9z2XQAVyIkb0saZVigWBr3MsDcr1
- 94OSQGpJCiSuQtHifnlYyOIKL0E2

Processing

In [12]:

```
files = []

# r=root, d=directories, f = files

for r, d, f in os.walk(DATASET_PATH):
    for file in f:
        if ".wav" in file:
            files.append(os.path.join(r, file))
```

In [14]:

```
files[:9]
```

Out[14]:

```
['COVID-19/7DfMFXPDu3W2Fxjs8w0OsLIY8em1/vowel-o.wav',
 'COVID-19/7DfMFXPDu3W2Fxjs8w0OsLIY8em1/vowel-a.wav',
 'COVID-19/7DfMFXPDu3W2Fxjs8w0OsLIY8em1/breathing-shallow.wav',
 'COVID-19/7DfMFXPDu3W2Fxjs8w0OsLIY8em1/cough-shallow.wav',
 'COVID-19/7DfMFXPDu3W2Fxjs8w0OsLIY8em1/vowel-e.wav',
 'COVID-19/7DfMFXPDu3W2Fxjs8w0OsLIY8em1/cough-heavy.wav',
 'COVID-19/7DfMFXPDu3W2Fxjs8w0OsLIY8em1/breathing-deep.wav',
 'COVID-19/7DfMFXPDu3W2Fxjs8w0OsLIY8em1/counting-normal.wav',
 'COVID-19/7DfMFXPDu3W2Fxjs8w0OsLIY8em1/counting-fast.wav']
```

In [15]:

```
ipd.Audio(files[0]) # vowel-o.wav
```

Out[15]:

0:08 / 0:12

In [16]:

```
ipd.Audio(files[1])  # vowel-a.wav
```

Out[16]:

0:05 / 0:11

In [17]:

```
ipd.Audio(files[2])  # breathing-shallow.wav
```

Out[17]:

0:02 / 0:13

In [18]:

```
ipd.Audio(files[3])  # cough-shallow.wav
```

Out[18]:

0:04 / 0:04

In [19]:

```
ipd.Audio(files[4])  # vowel-e.wav
```

Out[19]:

0:03 / 0:12

In [20]:

```
ipd.Audio(files[5])  # cough-heavy.wav
```

Out[20]:

0:04 / 0:04

In [21]:

```
ipd.Audio(files[6])  # breathing-deep.wav
```

Out[21]:

0:16 / 0:16

In [22]:

```
ipd.Audio(files[7])  # counting-normal.wav
```

Out[22]:

0:16 / 0:16

In [23]:

```
ipd.Audio(files[8])  # counting-fast.wav
```

Out[23]:

0:04 / 0:04

Feature Exploration

In [120]:

```
vowel_o = files[0]
```

<IPython.core.display.Javascript object>

In [121]:

```
y, sr = librosa.load(vowel_o, sr=None)
```

<IPython.core.display.Javascript object>

In [122]:

```
rmse = librosa.feature.rms(y)[0]
```

<IPython.core.display.Javascript object>

In [154]:

```
chroma_stft = librosa.feature.chroma_stft(y=y, sr=sr)[0]
spectral_centroid = librosa.feature.spectral_centroid(y=y, sr=sr)[0]
spectral_bandwidth = librosa.feature.spectral_bandwidth(y=y, sr=sr)[0]
spectral_rolloff = librosa.feature.spectral_rolloff(y=y, sr=sr)[0]
zcr = librosa.feature.zero_crossing_rate(y=y)[0]
```

<IPython.core.display.Javascript object>

In [155]:

```
def calculalte_t(feature):
    frames = range(len(feature))
    t = librosa.frames_to_time(frames)
    return t
```

<IPython.core.display.Javascript object>

In [156]:

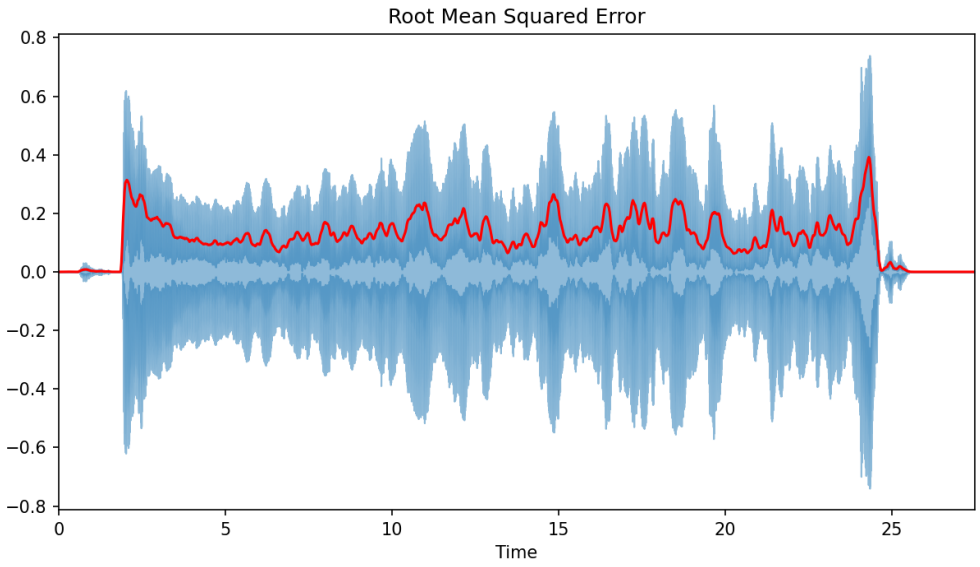
```
def plot_feature(feature, title, c="r"):
#     plt.figure(figsize=(8,10))
    t = calculalte_t(feature)
    librosa.display.waveplot(y, alpha=0.5)
    plt.plot(t, feature, color=c)
    plt.title(title)
    plt.show()
```

<IPython.core.display.Javascript object>

In [143]:

```
plot_feature(rmse, "Root Mean Squared Error")
```

<IPython.core.display.Javascript object>



<IPython.core.display.Javascript object>

In [158]:

```
plot_feature(chroma_stft, "Chroma Type Short Time Fourier Transform", "gr
```

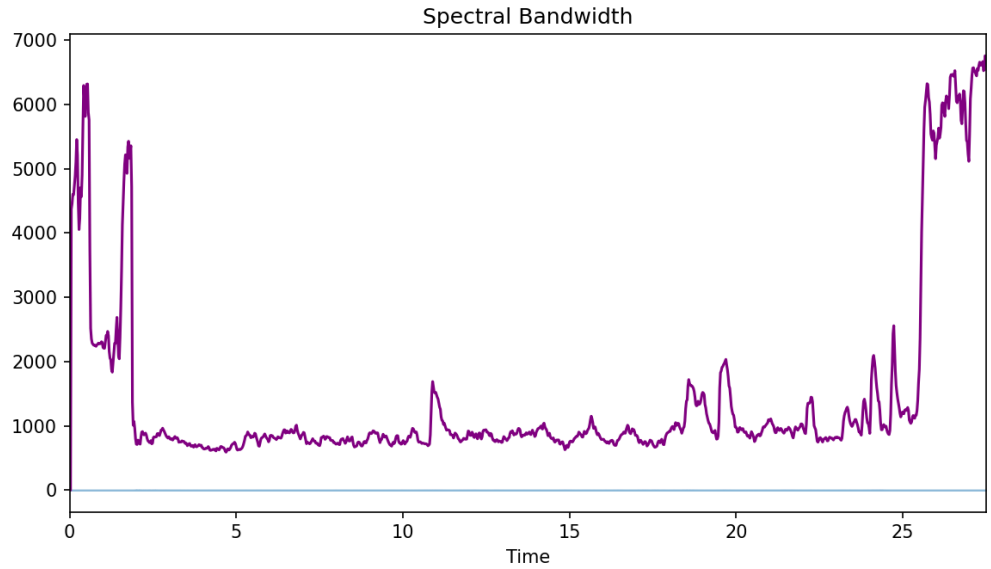
<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>

In [159]:

```
plot_feature(spectral_bandwidth, "Spectral Bandwidth", "purple")
```

<IPython.core.display.Javascript object>

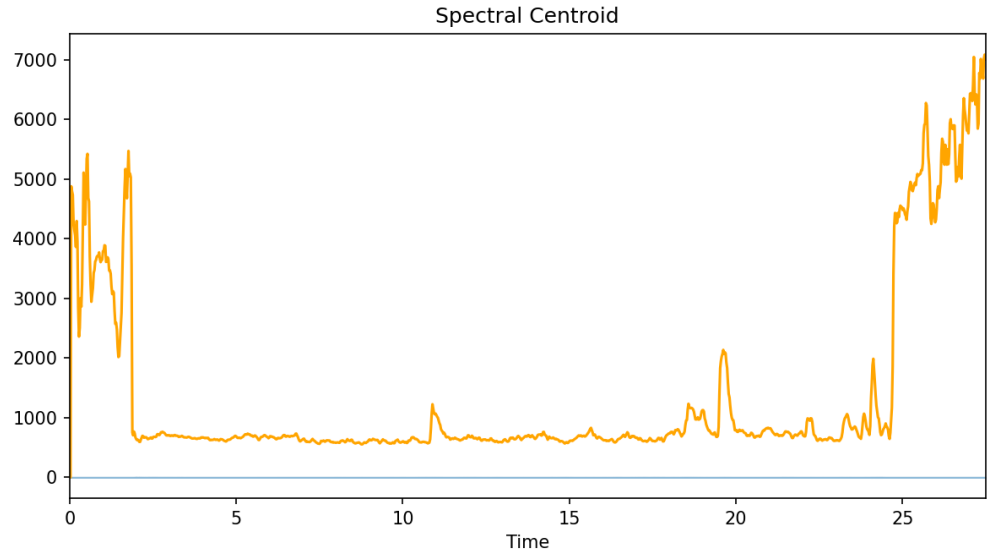


<IPython.core.display.Javascript object>

In [160]:

```
plot_feature(spectral_centroid, "Spectral Centroid", "orange")
```

<IPython.core.display.Javascript object>

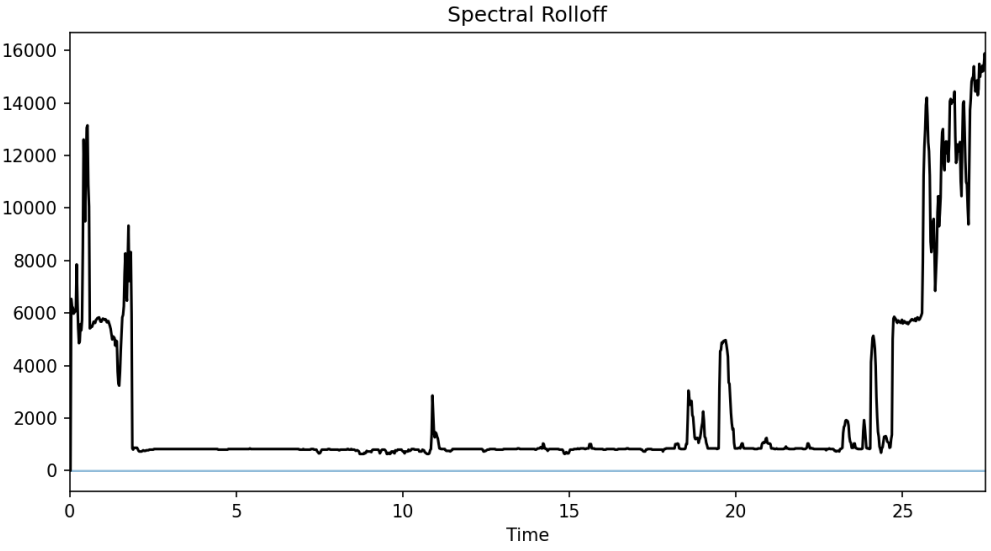


<IPython.core.display.Javascript object>

In [162]:

```
plot_feature(spectral_rolloff, "Spectral Rolloff", "black")
```

<IPython.core.display.Javascript object>

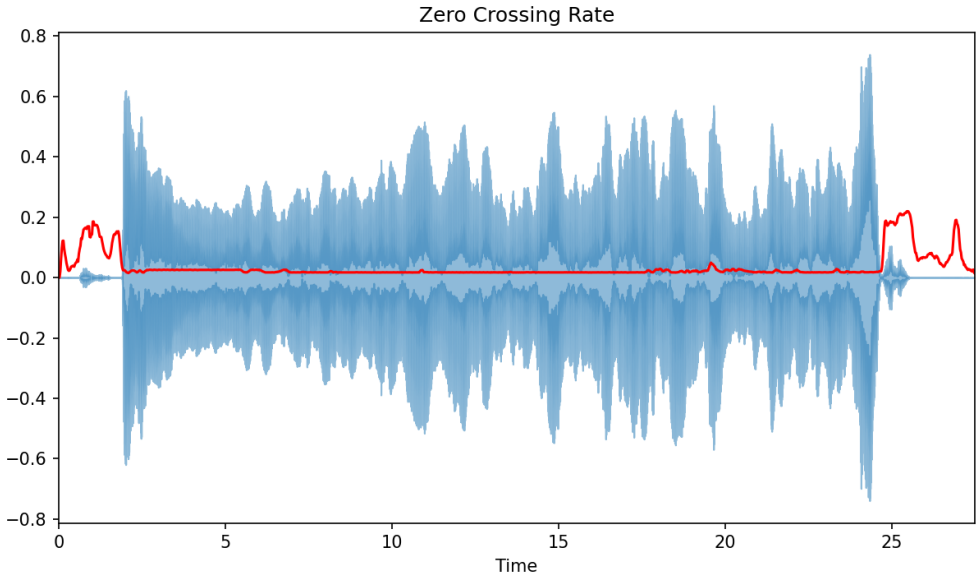


<IPython.core.display.Javascript object>

In [163]:

```
plot_feature(zcr, "Zero Crossing Rate", "red")
```

<IPython.core.display.Javascript object>



<IPython.core.display.Javascript object>

Basic Feature Extraction

In [15]:

```
# RMSE = []
# chroma_stft = []
# spec_cent = []
# spec_bw = []
# rolloff = []
# zcr = []
# user_id = []
# covid_status = []

# for f in files:

#     y, sr = librosa.load(f, sr=None)

#     if y is None or len(y) == 0:
#         continue

#     else:
#         user_id.append(f.split("/")[1])

#         metadata = json.load(open("/".join(f.split("/")[:2]) + "/metada

#         covid_status.append(metadata["covid_status"])

#         # Root Mean Squared Error
#         RMSE.append(np.mean(librosa.feature.rms(y)[0]))

#         # Chroma Based Short Time Fourier Transform
#         chroma_stft.append(np.mean(librosa.feature.chroma_stft(y=y, sr=

#         # Spectral Centroid
#         spec_cent.append(np.mean(librosa.feature.spectral_centroid(y=y,

#         # Spectral Bandwidth
#         spec_bw.append(np.mean(librosa.feature.spectral_bandwidth(y=y,

#         # Spectral RollOff
#         rolloff.append(np.mean(librosa.feature.spectral_rolloff(y=y, sr

#         # Zero Crossing Rate
#         zcr.append(np.mean(librosa.feature.zero_crossing_rate(y)))

# #         mfccs = librosa.feature.mfcc(y=y, sr=sr)
# #         m = []

# #         for e in mfccs:
# #             m.append(np.mean(e))
# #         mfcc.append(m)
```

<IPython.core.display.Javascript object>

In [16]:

```
# data = pd.DataFrame(
#     {
#         "user_id": user_id,
#         "filepath": files,
#         "RMSE": RMSE,
#         "chroma_stft": chroma_stft,
#         "spec_cent": spec_cent,
#         "spec_bw": spec_bw,
#         "spec_rolloff": rolloff,
#         "zcr": zcr,
#     }
# )
```

<IPython.core.display.Javascript object>

In [17]:

```
# data["covid_status"] = covid_status  # target label
```

<IPython.core.display.Javascript object>

In [18]:

```
data = pd.read_csv("data.csv")
```

<IPython.core.display.Javascript object>

In [19]:

```
# data["covid_status"].value_counts()
```

<IPython.core.display.Javascript object>

In [21]:

```
# data["user_id"].value_counts()
```

<IPython.core.display.Javascript object>

In [22]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3798 entries, 0 to 3797
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Unnamed: 0    3798 non-null   int64  
 1   RMSE          3798 non-null   float64
 2   chroma_stft   3798 non-null   float64
 3   spec_cent     3798 non-null   float64
 4   spec_bw       3798 non-null   float64
 5   spec_rolloff  3798 non-null   float64
 6   zcr           3798 non-null   float64
 7   covid_status  3798 non-null   int64  
 8   new_user_id   3798 non-null   int64  
 9   filename      3798 non-null   int64  
dtypes: float64(6), int64(4)
memory usage: 296.8 KB
```

```
<IPython.core.display.Javascript object>
```

In [23]:

```
3798 / 9
```

Out[23]:

```
422.0
```

```
<IPython.core.display.Javascript object>
```

In [24]:

```
from sklearn.preprocessing import LabelEncoder
```

```
<IPython.core.display.Javascript object>
```

In [25]:

```
# le = LabelEncoder()
# data["new_user_id"] = le.fit_transform(data["user_id"])
```

```
<IPython.core.display.Javascript object>
```

In [26]:

```
# data["filename"] = data["filepath"].apply(lambda x: x.split("/")[2])
```

```
<IPython.core.display.Javascript object>
```

In [28]:

```
# data.drop(["filepath", "user_id"], inplace=True, axis=1)
```

<IPython.core.display.Javascript object>

In [29]:

```
data.head()
```

Out[29]:

	Unnamed: 0	RMSE	chroma_stft	spec_cent	spec_bw	spec_rolloff	zcr	covid_
0	0	0.118379	0.253389	1390.837727	1459.099553	2264.121835	0.034621	
1	1	0.144788	0.246714	1774.427666	2057.234269	3452.113464	0.020360	
2	2	0.000135	0.495409	5502.609625	5720.697350	11688.208532	0.051564	
3	3	0.018265	0.434287	4009.550598	4088.996725	7927.490831	0.065998	
4	4	0.107049	0.297738	1623.703648	2021.015855	3080.055950	0.024948	

<IPython.core.display.Javascript object>

In [ ]:

```
# cs_le = LabelEncoder()  # covid status label Encoder
# file_le = LabelEncoder()  # filename Label Encoder
```

In [ ]:

```
# data["filename"] = file_le.fit_transform(data["filename"])
# data["covid_status"] = cs_le.fit_transform(data["covid_status"])
```

In [31]:

```
cs_le = LabelEncoder()
cs_le.classes_ = np.load("cs_le.npy", allow_pickle=True)

file_le = LabelEncoder()
file_le.classes_ = np.load("file_le.npy", allow_pickle=True)
```

<IPython.core.display.Javascript object>

In [32]:

```
cs_le.classes_
```

Out[32]:

```
array(['healthy', 'no_resp_illness_exposed', 'positive_asymp',
       'positive_mild', 'positive_moderate', 'recovered_full',
       'resp_illness_not_identified'], dtype=object)
```

<IPython.core.display.Javascript object>

In [33]:

```
file_le.classes_
```

Out[33]:

```
array(['breathing-deep.wav', 'breathing-shallow.wav', 'cough-heavy.wa
v',
       'cough-shallow.wav', 'counting-fast.wav', 'counting-normal.wa
v',
       'vowel-a.wav', 'vowel-e.wav', 'vowel-o.wav'], dtype=object)

<IPython.core.display.Javascript object>
```

In [ ]:

```
# data.to_csv("data.csv")
```

In [ ]:

```
# saving both encoder

# np.save("cs_le.npy", cs_le.classes_)
# np.save("file_le.npy", file_le.classes_)
```

In [34]:

```
target = data["covid_status"]
data.drop(["covid_status", "new_user_id"], inplace=True, axis=1)
```

```
<IPython.core.display.Javascript object>
```

In [37]:

```
data = data.iloc[:, 1:]
data.head()
```

Out[37]:

	RMSE	chroma_stft	spec_cent	spec_bw	spec_rolloff	zcr	filename
0	0.118379	0.253389	1390.837727	1459.099553	2264.121835	0.034621	8
1	0.144788	0.246714	1774.427666	2057.234269	3452.113464	0.020360	6
2	0.000135	0.495409	5502.609625	5720.697350	11688.208532	0.051564	1
3	0.018265	0.434287	4009.550598	4088.996725	7927.490831	0.065998	3
4	0.107049	0.297738	1623.703648	2021.015855	3080.055950	0.024948	7

```
<IPython.core.display.Javascript object>
```

In [38]:

```
# (user_id , audio features , audio files)

modified_data = data.values.reshape((422, 7, 9)).astype("float64")
```

```
<IPython.core.display.Javascript object>
```

In [39]:

```
modified_data[0][0]
```

Out[39]:

```
array([1.18379205e-01, 2.53389360e-01, 1.39083773e+03, 1.45909955e+03,
       2.26412184e+03, 3.46209949e-02, 8.00000000e+00, 1.44788490e-01,
       2.46714490e-01])
```

```
<IPython.core.display.Javascript object>
```

In [40]:

```
target_values = []
for v in range(0, len(target), 9):
    target_values.append(target[v])
```

```
<IPython.core.display.Javascript object>
```

In [41]:

```
len(target_values)
```

Out[41]:

```
422
```

```
<IPython.core.display.Javascript object>
```

In [42]:

```
np.save("features_data.npy", modified_data)
np.save("target_data.npy", target_values)
```

```
<IPython.core.display.Javascript object>
```

Data Preparation

- Order of Features Must Be Same
  - RMSE
  - Chroma_Stft
  - Spec_cent
  - Spec_bw
  - Spec_rolloff
  - Zcr
  - Filename (filename => file_le.fit_transform)
- Target:

- covid_status (covid_status => cs_le.fit_transform)

File Information

- **features_data.npy** : our final feature array of shape (422 ,7 ,9)
- **targegt_data.npy** : final target array of shape (422 ,1)
- **cs_le.npy** : covid_status label encoder classes
- **file_le.npy** : filename label encoder classes
- **data.csv** : Informative Featured CSV file.

In [45]:

```
modified_data[0].shape # One Person Paricular Information
```

Out[45]:

```
(7, 9)
```

```
<IPython.core.display.Javascript object>
```

Deep Learning Model

In [47]:

```
n_classes = len(cs_le.classes_)
```

```
<IPython.core.display.Javascript object>
```

In [113]:

```
model = Sequential()

model.add(Input(shape=modified_data[0].shape))
model.add(Flatten())

model.add(Dense(units=16, activation="relu"))
model.add(Dense(units=32, activation="relu"))
model.add(Dense(units=64, activation="relu"))
model.add(Dense(units=64, activation="relu"))
model.add(Dense(units=n_classes, activation="softmax"))
```

```
<IPython.core.display.Javascript object>
```

In [114]:

```
model.summary()
```

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten_3 (Flatten)          (None, 63)                0
_________________________________________________________________
dense_13 (Dense)             (None, 16)                1024
_________________________________________________________________
dense_14 (Dense)             (None, 32)                544
_________________________________________________________________
dense_15 (Dense)             (None, 64)                2112
_________________________________________________________________
dense_16 (Dense)             (None, 64)                4160
_________________________________________________________________
dense_17 (Dense)             (None, 7)                 455
=================================================================
Total params: 8,295
Trainable params: 8,295
Non-trainable params: 0
_________________________________________________________________
```

<IPython.core.display.Javascript object>

In [115]:

```
model.compile(
    optimizer="adam",
    loss=tensorflow.keras.losses.SparseCategoricalCrossentropy(),
    metrics=["accuracy"],
)
```

<IPython.core.display.Javascript object>

Train Test Split

In [105]:

```
X_train, X_test, y_train, y_test = train_test_split(
    modified_data, target_values, test_size=0.2, random_state=0
)
```

<IPython.core.display.Javascript object>

In [106]:

```
X_train.shape
```

Out[106]:

```
(337, 7, 9)
```

<IPython.core.display.Javascript object>

In [107]:

```
X_test.shape
```

Out[107]:

```
(85, 7, 9)
```

```
<IPython.core.display.Javascript object>
```

In [108]:

```
y_train = np.array(y_train)
y_train.shape
```

Out[108]:

```
(337,)
```

```
<IPython.core.display.Javascript object>
```

In [109]:

```
y_test = np.array(y_test)
y_test.shape
```

Out[109]:

```
(85,)
```

```
<IPython.core.display.Javascript object>
```

In [110]:

```
X_train.shape, X_test.shape
```

Out[110]:

```
((337, 7, 9), (85, 7, 9))
```

```
<IPython.core.display.Javascript object>
```

Model Training

In [116]:

```
model.fit(X_train, y_train, batch_size=1, epochs=30, validation_data=(X_t
```

```
Epoch 1/30
337/337 [==============================] - 0s 736us/step - loss: 111.9
914 - accuracy: 0.4866 - val_loss: 79.3241 - val_accuracy: 0.1647
Epoch 2/30
337/337 [==============================] - 0s 525us/step - loss: 37.66
66 - accuracy: 0.4837 - val_loss: 24.7656 - val_accuracy: 0.5882
Epoch 3/30
337/337 [==============================] - 0s 524us/step - loss: 18.45
10 - accuracy: 0.5193 - val_loss: 20.5721 - val_accuracy: 0.4235
Epoch 4/30
337/337 [==============================] - 0s 526us/step - loss: 14.31
80 - accuracy: 0.4896 - val_loss: 14.0340 - val_accuracy: 0.6353
Epoch 5/30
337/337 [==============================] - 0s 534us/step - loss: 10.15
24 - accuracy: 0.4866 - val_loss: 6.3630 - val_accuracy: 0.6118
Epoch 6/30
337/337 [==============================] - 0s 587us/step - loss: 7.057
8 - accuracy: 0.4985 - val_loss: 6.1100 - val_accuracy: 0.5412
Epoch 7/30
337/337 [==============================] - 0s 612us/step - loss: 4.692
0 - accuracy: 0.4926 - val_loss: 6.3451 - val_accuracy: 0.5765
Epoch 8/30
337/337 [==============================] - 0s 573us/step - loss: 3.091
1 - accuracy: 0.5163 - val_loss: 2.2992 - val_accuracy: 0.4706
Epoch 9/30
337/337 [==============================] - 0s 581us/step - loss: 2.125
0 - accuracy: 0.5134 - val_loss: 1.8912 - val_accuracy: 0.6706
Epoch 10/30
337/337 [==============================] - 0s 579us/step - loss: 1.536
9 - accuracy: 0.6083 - val_loss: 1.8021 - val_accuracy: 0.7059
Epoch 11/30
337/337 [==============================] - 0s 556us/step - loss: 1.504
9 - accuracy: 0.6499 - val_loss: 1.6542 - val_accuracy: 0.6824
Epoch 12/30
337/337 [==============================] - 0s 569us/step - loss: 1.439
7 - accuracy: 0.6202 - val_loss: 1.6315 - val_accuracy: 0.6588
Epoch 13/30
337/337 [==============================] - 0s 566us/step - loss: 1.551
9 - accuracy: 0.6558 - val_loss: 2.0169 - val_accuracy: 0.6824
Epoch 14/30
337/337 [==============================] - 0s 562us/step - loss: 1.416
1 - accuracy: 0.6409 - val_loss: 1.7174 - val_accuracy: 0.7176
Epoch 15/30
337/337 [==============================] - 0s 592us/step - loss: 1.200
3 - accuracy: 0.6647 - val_loss: 2.8350 - val_accuracy: 0.7176
Epoch 16/30
337/337 [==============================] - 0s 572us/step - loss: 1.302
0 - accuracy: 0.6350 - val_loss: 1.6943 - val_accuracy: 0.6824
Epoch 17/30
337/337 [==============================] - 0s 569us/step - loss: 1.475
9 - accuracy: 0.6439 - val_loss: 1.1555 - val_accuracy: 0.7059
Epoch 18/30
337/337 [==============================] - 0s 543us/step - loss: 1.173
0 - accuracy: 0.6647 - val_loss: 1.1625 - val_accuracy: 0.7059
Epoch 19/30
337/337 [==============================] - 0s 534us/step - loss: 1.316
```

```
3 - accuracy: 0.6528 - val_loss: 1.0372 - val_accuracy: 0.7059
Epoch 20/30
337/337 [==============================] - 0s 572us/step - loss: 1.266
6 - accuracy: 0.6499 - val_loss: 1.0364 - val_accuracy: 0.7059
Epoch 21/30
337/337 [==============================] - 0s 581us/step - loss: 1.192
0 - accuracy: 0.6588 - val_loss: 1.0504 - val_accuracy: 0.7059
Epoch 22/30
337/337 [==============================] - 0s 530us/step - loss: 1.187
5 - accuracy: 0.6617 - val_loss: 1.1530 - val_accuracy: 0.6941
Epoch 23/30
337/337 [==============================] - 0s 520us/step - loss: 1.186
1 - accuracy: 0.6588 - val_loss: 1.0619 - val_accuracy: 0.7059
Epoch 24/30
337/337 [==============================] - 0s 566us/step - loss: 1.177
3 - accuracy: 0.6617 - val_loss: 1.0612 - val_accuracy: 0.6941
Epoch 25/30
337/337 [==============================] - 0s 564us/step - loss: 1.172
8 - accuracy: 0.6588 - val_loss: 1.0654 - val_accuracy: 0.7059
Epoch 26/30
337/337 [==============================] - 0s 580us/step - loss: 1.182
2 - accuracy: 0.6617 - val_loss: 1.0563 - val_accuracy: 0.7059
Epoch 27/30
337/337 [==============================] - 0s 566us/step - loss: 1.181
7 - accuracy: 0.6588 - val_loss: 1.0531 - val_accuracy: 0.7059
Epoch 28/30
337/337 [==============================] - 0s 567us/step - loss: 1.182
5 - accuracy: 0.6588 - val_loss: 1.0609 - val_accuracy: 0.7059
Epoch 29/30
337/337 [==============================] - 0s 569us/step - loss: 1.183
1 - accuracy: 0.6588 - val_loss: 1.0565 - val_accuracy: 0.7059
Epoch 30/30
337/337 [==============================] - 0s 576us/step - loss: 1.183
3 - accuracy: 0.6588 - val_loss: 1.0552 - val_accuracy: 0.7059
```

Out[116]:

```
<tensorflow.python.keras.callbacks.History at 0x7fb01051bb80>
```

```
<IPython.core.display.Javascript object>
```

In [117]:

```
model.evaluate(X_test, y_test)
# [loss , accuracy ]
```

```
3/3 [==============================] - 0s 696us/step - loss: 1.0552 -
accuracy: 0.7059
```

Out[117]:

```
[1.055246353149414, 0.7058823704719543]
```

```
<IPython.core.display.Javascript object>
```

In [118]:

```
model.save("model_1_70.h5")
```

```
<IPython.core.display.Javascript object>
```

In [ ]: