

SPEECH EMOTION RECOGNITION USING PYTHON

Mainly, we need to use the Supervised Machine Learning Algorithms of Classification and regression for Speech Emotion Recognition, as there are labeled data of audio with an author and all.

Algorithms Can be Used –

1. Convolutional Neural Network (CNN)- CNN requires the labeled training data to learn the patterns among input data, using this, it will make predictions.

In Speech Emotion Recognition, **Audio Recordings** which are labeled with **corresponding emotions** can be given to the CNN as a **labeled dataset** for training. So, by **extracting features and classifying** them into particular emotions, this can be self-learnt by the CNN.

2. Random Forest Algorithm- This technique builds many decisions trees during the training phase and outputs the class that represents the mean of the classes(classification) or the mean prediction(regression) of the individual trees.

It handles missing values with both categorical and continuous data.

Python Libraries used in the Project –

1. NumPy-

Used to perform **mathematical and logical operations** on Arrays.

Contains Multi-dimensional Arrays and Matrices.

2. Scikit-learn-

Most Powerful Library with efficient tools for **ML and statistical modeling** including range of supervised and unsupervised learning algorithms.

3. Librosa-

Library for **analyzing and processing audio signals**.

Used for **loading, manipulating and extracting features from audio signals**.

4. Glob-

This python module is mainly used when we want **to extract multiple files** from parent-directory and sub-directory which are **of same type of file** like, .mkv, .txt, .mp3, .mp4.,etc.

Steps need to follow while making model –

1. First, we need to **collect data** from the given **RAVDESS** dataset and then **extract features** from them.

If you want to learn more about how files are named refer this dataset –

<https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio>

2. In our program, we have first call to **load_data ()** –
In this, we take one by one audio recordings from stored location using **glob()** function -

```
glob.glob("RAVDESS/Actor_*/*.wav")
```

and before this, we create the **directory for the emotions** with keys(no.) and values(emotion_name) pair and splitting the emotion by using index on file-name separately-

```
emotion = emotions[file_name.split("-")[2]]
```

For extracting features, it will get called to the **extract_feature()** function.

3. In **extract_feature()** function we extract the features from passed audio recording like its **chroma feature, mel feature, mfcc feature** by loading this audio recording to **librosa** gives us a **sample rate of audio and main audio**.

4. And then, we assign the **emotions** for the particular feature of **Audio Recording** by combining them into the different array.

```
x.append(feature)
y.append(emotion)
```

5. **Split the Train & Test DataSet** by using **train_test_split** from **sklearn.model_selection** – This is returning by the **load_data()** function

```
train_test_split(np.array(x), y, test_size=test_size, random_state=0)
```

6. Then we finally **apply the Algorithm** in our case, we are applying **Random Forest Algorithm** by firstly, **train algorithm** by passing **x_train** and **y_train** to the **object of RandomForestClassifier**–

```
RFClassifier.fit(x_train, y_train)
```

7. Then, our training to the algorithm is over, then we are able **to predict using x_test** i.e. test data used for testing our model –

```
Y_pred = RFClassifier.predict(x_test)
```

8. We can find our **accuracy by providing original y_test** to the **accuracy_score(y_test, y_pred)**

In our case, this algorithm gives **67% accuracy**.