SPEECH EMOTION RECOGNITION

USING LIBROSA

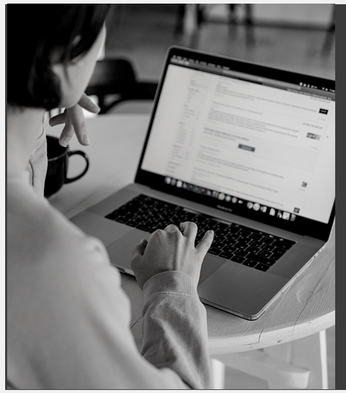
INDEX

Key topics for discussion

Introduction and Motivation for this Project

Speech emotion recognition, the best ever python mini project. The best example of it can be seen at call centers. If you ever noticed, call centers employees never talk in the same manner, their way of pitching/talking to the customers changes with customers. Now, this does happen with common people too, but how is this relevant to call centers? Here is your answer, the employees recognize customers' emotions from speech, so they can improve their service and convert more people. In this way, they are using speech emotion recognition.

Problem Statement and definition

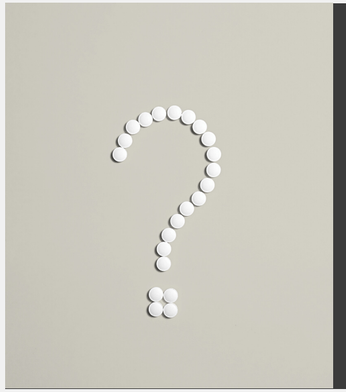


SPEECH EMOTION RECOGNITION

Speech Emotion Recognition, abbreviated as SER, is the act of attempting to recognize human emotion and affective states from speech.

Product comparisons

This is capitalizing on the fact that voice often reflects underlying emotion through tone and pitch. This is also the phenomenon that animals like dogs and horses employ to be able to understand human emotion.



WHY IS SER A CHALLENGE?

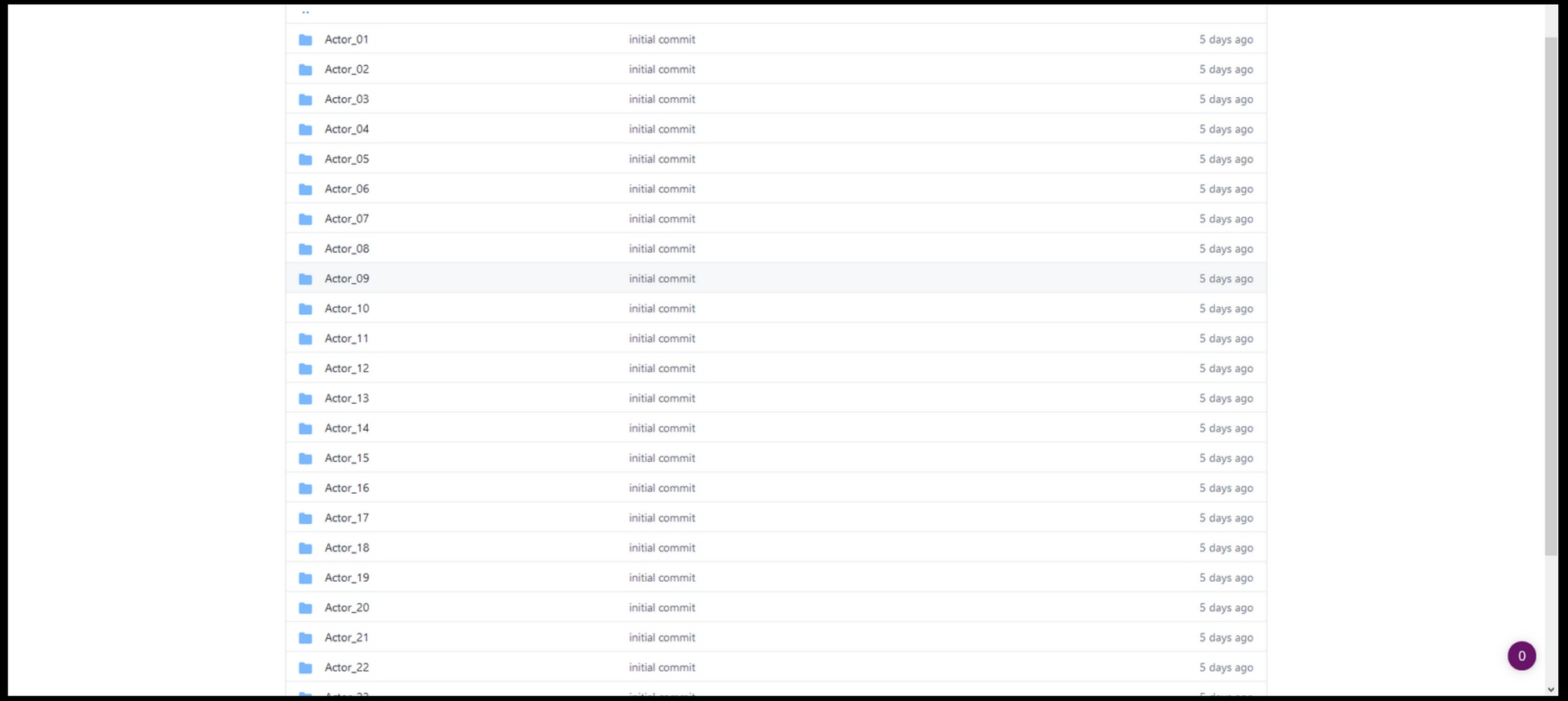
SER is tough because emotions are subjective and annotating audio is challenging.

Dataset used

A combination of speech and emotions

Sign Language (SL) is the natural way of communication of mute people. A sign is a movement of one or both hands, accompanied with facial expression, which corresponds to a specific meaning. Although the mute, hard of speaking and speaking signers can communicate without problems amongst themselves, there is a serious challenge for the mute community trying to integrate into educational, social and work environments. The overall goal of this project is to develop a new vision based technology for recognizing and translating continuous sign language to text.

THE DATASET



Actor_01	initial commit	5 days ago
Actor_02	initial commit	5 days ago
Actor_03	initial commit	5 days ago
Actor_04	initial commit	5 days ago
Actor_05	initial commit	5 days ago
Actor_06	initial commit	5 days ago
Actor_07	initial commit	5 days ago
Actor_08	initial commit	5 days ago
Actor_09	initial commit	5 days ago
Actor_10	initial commit	5 days ago
Actor_11	initial commit	5 days ago
Actor_12	initial commit	5 days ago
Actor_13	initial commit	5 days ago
Actor_14	initial commit	5 days ago
Actor_15	initial commit	5 days ago
Actor_16	initial commit	5 days ago
Actor_17	initial commit	5 days ago
Actor_18	initial commit	5 days ago
Actor_19	initial commit	5 days ago
Actor_20	initial commit	5 days ago
Actor_21	initial commit	5 days ago
Actor_22	initial commit	5 days ago

Proposed Methodology

01

You'll need to install the following libraries with pip:

pip install librosa soundfile numpy sklearn pyaudio

02

Make the necessary imports:

librosa, soundfile, os, glob, pickle, numpy, sklearn, accuracy_score

03

Define a function

Define a function extract_feature to extract the mfcc, chroma, and mel features from a sound file. This function takes 4 parameters- the file name and three Boolean parameters

04

Define a Dictionary

Now, let's define a dictionary to hold numbers and the emotions available in the RAVDESS dataset, and a list to hold those we want- calm, happy, fearful, disgust.

05

Loading Data

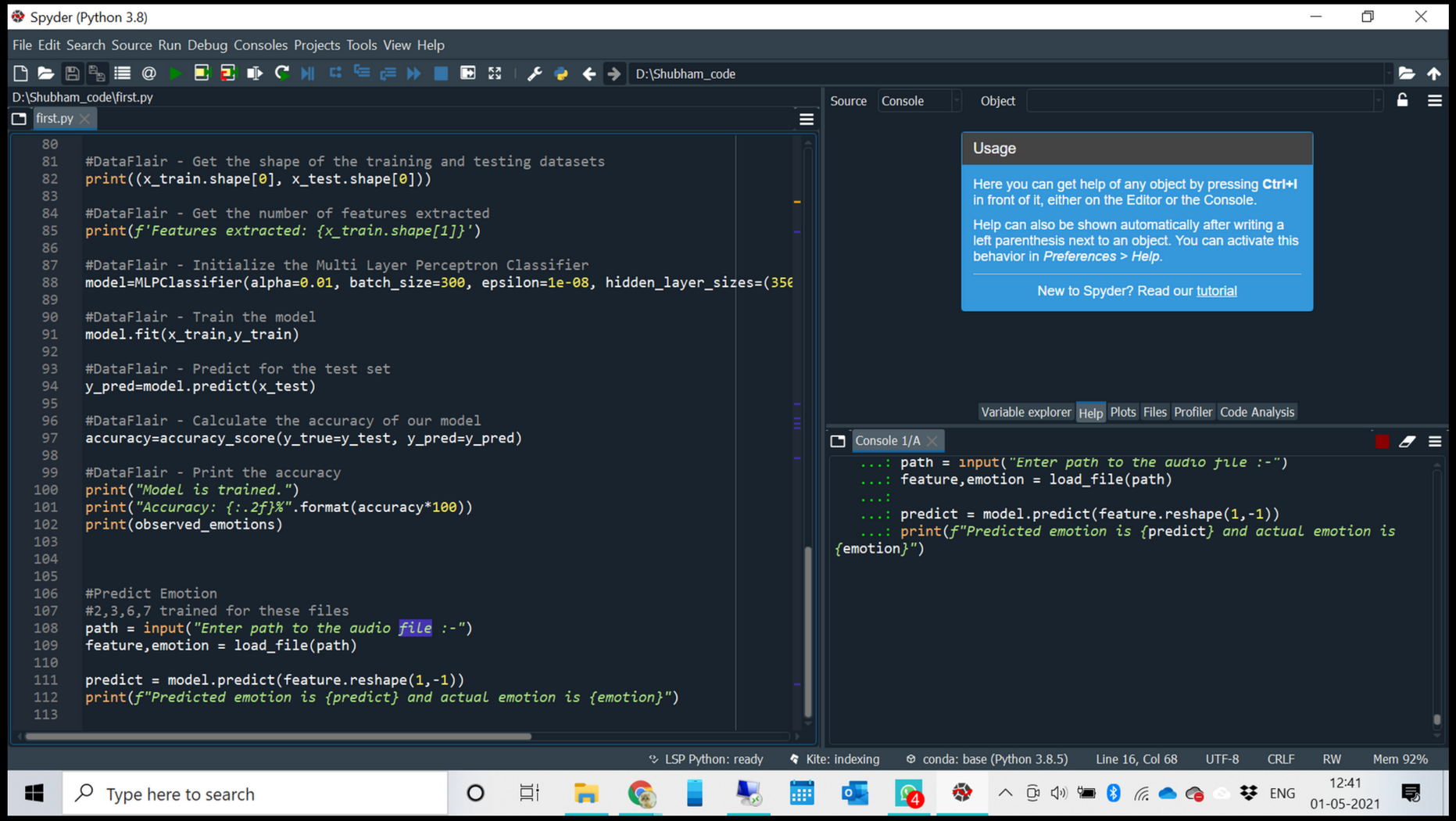
This takes in the relative size of the test set as parameter. x and y are empty lists; we'll use the glob() function from the glob module to get all the pathnames for the sound files in our dataset. The pattern we use for this is: "D:\\DataFlair\\ravdess data\\Actor_**.wav".

06

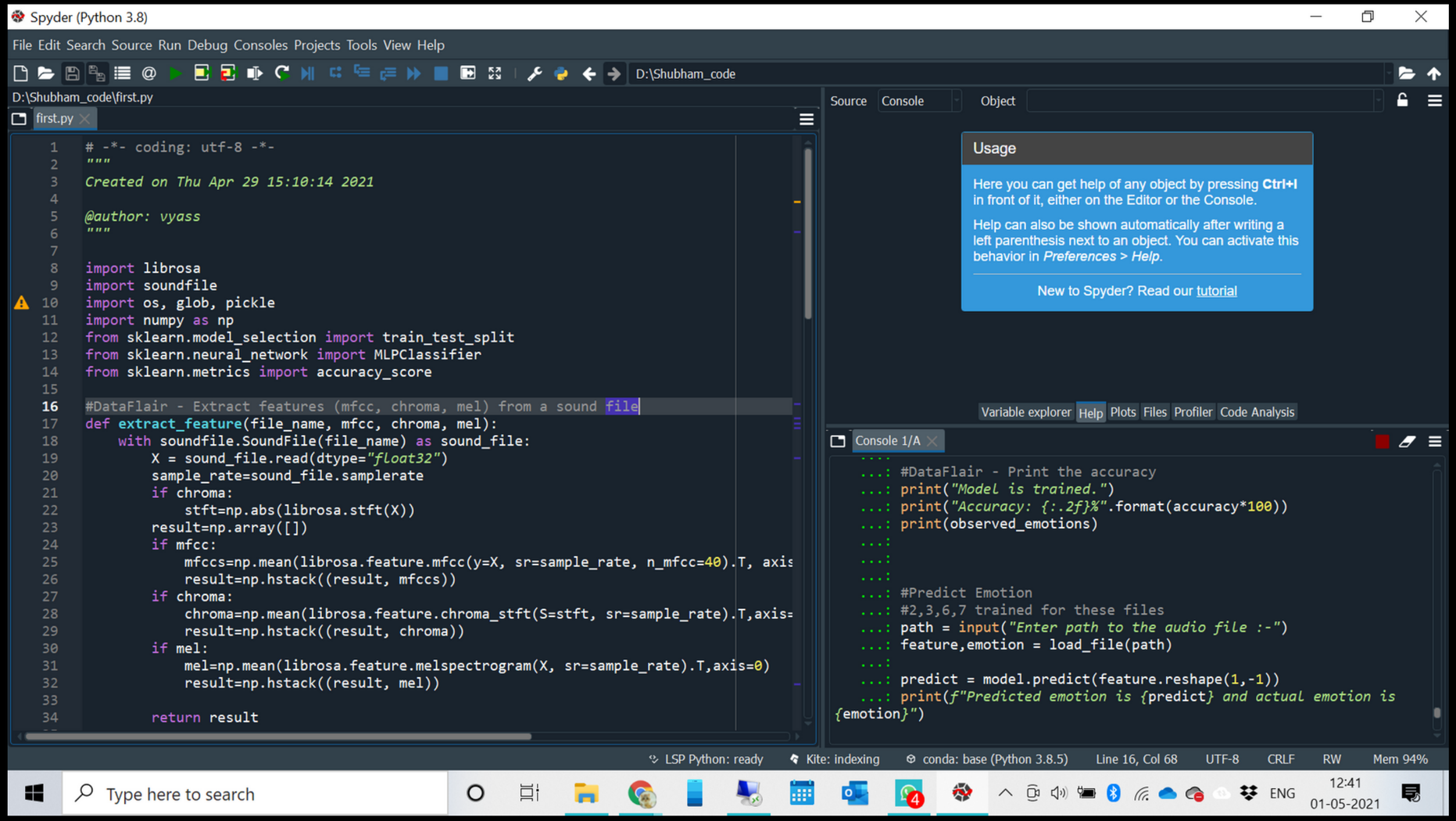
Splitting the Dataset

Time to split the dataset into training and testing sets! Let's keep the test set 25% of everything and use the load_data function for this.

CODE


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\Shubham_code
D:\Shubham_code\first.py
first.py
#DataFlair - Get the shape of the training and testing datasets
print((x_train.shape[0], x_test.shape[0]))
#DataFlair - Get the number of features extracted
print(f'Features extracted: {x_train.shape[1]}')
#DataFlair - Initialize the Multi Layer Perceptron Classifier
model=MLPClassifier(alpha=0.01, batch_size=300, epsilon=1e-08, hidden_layer_sizes=(356
#DataFlair - Train the model
model.fit(x_train,y_train)
#DataFlair - Predict for the test set
y_pred=model.predict(x_test)
#DataFlair - Calculate the accuracy of our model
accuracy=accuracy_score(y_true=y_test, y_pred=y_pred)
#DataFlair - Print the accuracy
print("Model is trained.")
print("Accuracy: {:.2f}%".format(accuracy*100))
print(observed_emotions)
#Predict Emotion
#2,3,6,7 trained for these files
path = input("Enter path to the audio file :-")
feature,emotion = load_file(path)
predict = model.predict(feature.reshape(1,-1))
print(f"Predicted emotion is {predict} and actual emotion is {emotion}")
Source Console Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Variable explorer Help Plots Files Profiler Code Analysis
Console 1/A
...: path = input("Enter path to the audio file :-")
...: feature,emotion = load_file(path)
...:
...: predict = model.predict(feature.reshape(1,-1))
...: print(f"Predicted emotion is {predict} and actual emotion is {emotion}")
LSP Python: ready
Kite: indexing
conda: base (Python 3.8.5)
Line 16, Col 68
UTF-8
CRLF
RW
Mem 92%
Type here to search
ENG
12:41
01-05-2021

CODE


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\Shubham_code
D:\Shubham_code\first.py
first.py
-*- coding: utf-8 -*-
"""
Created on Thu Apr 29 15:10:14 2021
@author: vyass
"""
import librosa
import soundfile
import os, glob, pickle
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
#DataFlair - Extract features (mfcc, chroma, mel) from a sound file
def extract_feature(file_name, mfcc, chroma, mel):
with soundfile.SoundFile(file_name) as sound_file:
X = sound_file.read(dtype="float32")
sample_rate=sound_file.samplerate
if chroma:
stft=np.abs(librosa.stft(X))
result=np.array([])
if mfcc:
mfccs=np.mean(librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=40).T, axis
result=np.hstack((result, mfccs))
if chroma:
chroma=np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T,axis=
result=np.hstack((result, chroma))
if mel:
mel=np.mean(librosa.feature.melspectrogram(X, sr=sample_rate).T,axis=0)
result=np.hstack((result, mel))
return result
Source Console Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Variable explorer Help Plots Files Profiler Code Analysis
Console 1/A
...: #DataFlair - Print the accuracy
...: print("Model is trained.")
...: print("Accuracy: {:.2f}%".format(accuracy*100))
...: print(observed_emotions)
...: #Predict Emotion
...: #2,3,6,7 trained for these files
...: path = input("Enter path to the audio file :-")
...: feature,emotion = load_file(path)
...: predict = model.predict(feature.reshape(1,-1))
...: print(f"Predicted emotion is {predict} and actual emotion is {emotion}")
LSP Python: ready
Kite: indexing
conda: base (Python 3.8.5)
Line 16, Col 68
UTF-8
CRLF
RW
Mem 94%
Type here to search
ENG
12:41
01-05-2021

Proposed Methodology

07

Observe the shape of the training and testing datasets

08

Initialize an MLPClassifier

This is a Multi-layer Perceptron Classifier; it optimizes the log-loss function using LBFGS or stochastic gradient descent.

09

Fit/train the model.

10

Predict the values for the test set

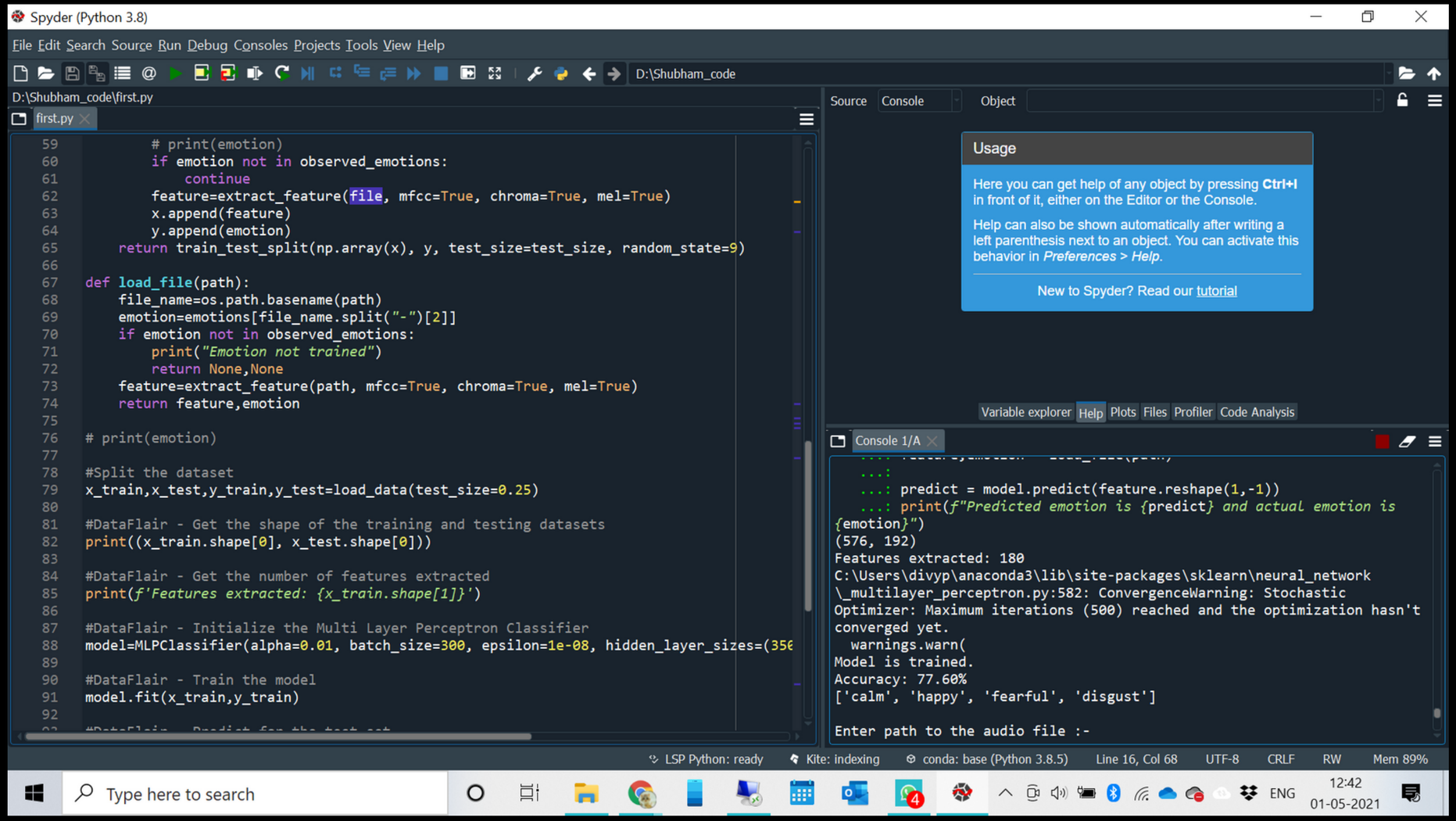
This gives us y_pred (the predicted emotions for the features in the test set).

11

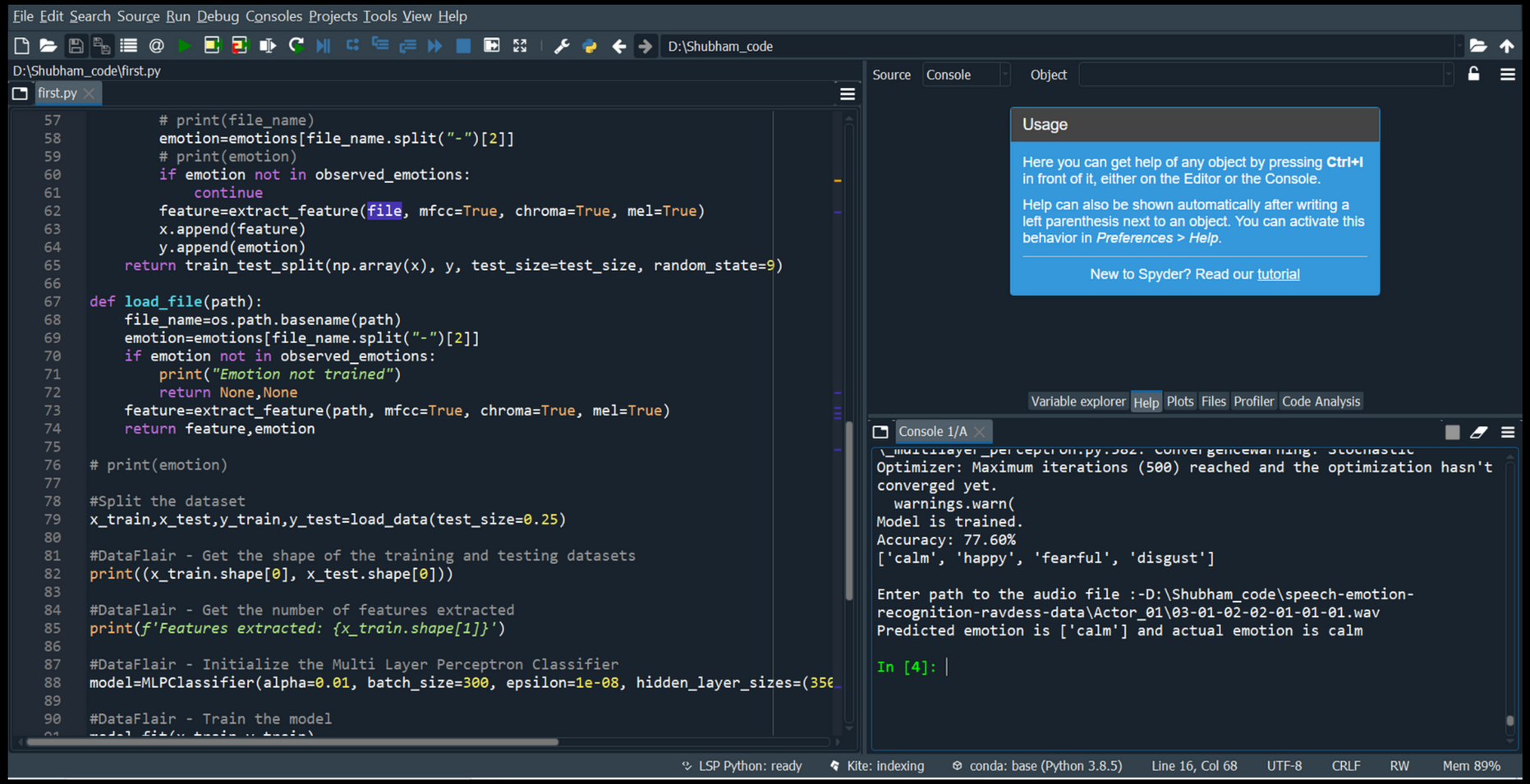
Calculating the Accuracy

To calculate the accuracy of our model, we'll call up the accuracy_score() function we imported from sklearn. Finally, we'll round the accuracy to 2 decimal places and print it out.

CODE


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\Shubham_code
D:\Shubham_code\first.py
first.py
59 # print(emotion)
60 if emotion not in observed_emotions:
61 continue
62 feature=extract_feature(file, mfcc=True, chroma=True, mel=True)
63 x.append(feature)
64 y.append(emotion)
65 return train_test_split(np.array(x), y, test_size=test_size, random_state=9)
66
67 def load_file(path):
68 file_name=os.path.basename(path)
69 emotion=emotions[file_name.split("-")[2]]
70 if emotion not in observed_emotions:
71 print("Emotion not trained")
72 return None,None
73 feature=extract_feature(path, mfcc=True, chroma=True, mel=True)
74 return feature,emotion
75
76 # print(emotion)
77
78 #Split the dataset
79 x_train,x_test,y_train,y_test=load_data(test_size=0.25)
80
81 #DataFlair - Get the shape of the training and testing datasets
82 print((x_train.shape[0], x_test.shape[0]))
83
84 #DataFlair - Get the number of features extracted
85 print(f'Features extracted: {x_train.shape[1]}')
86
87 #DataFlair - Initialize the Multi Layer Perceptron Classifier
88 model=MLPClassifier(alpha=0.01, batch_size=300, epsilon=1e-08, hidden_layer_sizes=(356
89
90 #DataFlair - Train the model
91 model.fit(x_train,y_train)
92
Source Console Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Variable explorer Help Plots Files Profiler Code Analysis
Console 1/A
...:
...: predict = model.predict(feature.reshape(1,-1))
...: print(f"Predicted emotion is {predict} and actual emotion is {emotion}")
(576, 192)
Features extracted: 180
C:\Users\divyp\anaconda3\lib\site-packages\sklearn\neural_network_multilayer_perceptron.py:582: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (500) reached and the optimization hasn't converged yet.
warnings.warn(
Model is trained.
Accuracy: 77.60%
['calm', 'happy', 'fearful', 'disgust']
Enter path to the audio file :-
LSP Python: ready
Kite: indexing
conda: base (Python 3.8.5)
Line 16, Col 68
UTF-8
CRLF
RW
Mem 89%
Type here to search
ENG
12:42
01-05-2021

Results


File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\Shubham_code
D:\Shubham_code\first.py
first.py
# print(file_name)
emotion=emotions[file_name.split("-")[2]]
# print(emotion)
if emotion not in observed_emotions:
continue
feature=extract_feature(file, mfcc=True, chroma=True, mel=True)
x.append(feature)
y.append(emotion)
return train_test_split(np.array(x), y, test_size=test_size, random_state=9)
def load_file(path):
file_name=os.path.basename(path)
emotion=emotions[file_name.split("-")[2]]
if emotion not in observed_emotions:
print("Emotion not trained")
return None,None
feature=extract_feature(path, mfcc=True, chroma=True, mel=True)
return feature,emotion
print(emotion)
#Split the dataset
x_train,x_test,y_train,y_test=load_data(test_size=0.25)
#DataFlair - Get the shape of the training and testing datasets
print((x_train.shape[0], x_test.shape[0]))
#DataFlair - Get the number of features extracted
print(f'Features extracted: {x_train.shape[1]}')
#DataFlair - Initialize the Multi Layer Perceptron Classifier
model=MLPClassifier(alpha=0.01, batch_size=300, epsilon=1e-08, hidden_layer_sizes=(356
#DataFlair - Train the model
Source
Console
Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Variable explorer
Help
Plots
Files
Profiler
Code Analysis
Console 1/A
Optimizer: Maximum iterations (500) reached and the optimization hasn't converged yet.
warnings.warn(
Model is trained.
Accuracy: 77.60%
['calm', 'happy', 'fearful', 'disgust']
Enter path to the audio file :-D:\Shubham_code\speech-emotion-recognition-ravdess-data\Actor_01\03-01-02-02-01-01-01.wav
Predicted emotion is ['calm'] and actual emotion is calm
In [4]:
LSP Python: ready
Kite: indexing
conda: base (Python 3.8.5)
Line 16, Col 68
UTF-8
CRLF
RW
Mem 89%

Future Possible Advancement

- This project can be carried forward and we can develop the model and assemble to a hardware for everyone to use.

- This project can be improved at two levels, first can be made for multiple languages, and second the optimization of the model using multiple theories.

The Team

Artificial Intelligence |
Pandit Deendayal Energy
University

NAMAN PARMAR

18BIT075

ROHAN SHETH

18BIT091

PARAM MODI

18BIT082

SHUBHAM VYAS

18BIT107

SARTHAK PANSURIA

18BIT100

Thank you for your time!