

Configuration Manual

MSc Research Project
Data Analytics

Nipun Kulshrestha
Student ID: X18190758

School of Computing
National College of Ireland

Supervisor: Manaz Kaleel

**National College of Ireland
Project Submission Sheet
School of Computing**



Student Name:	Nipun Kulshrestha
Student ID:	X18190758
Programme:	Data Analytics
Year:	2020
Module:	MSc Research Project
Supervisor:	Manaz Kaleel
Submission Due Date:	17/08/2020
Project Title:	Configuration Manual
Word Count:	634
Page Count:	9

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	16th August 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Nipun Kulshrestha
X18190758

1 Required Libraries

This project is written in python. The IDE used was Spyder for code writing and Google Colab was used for training for 200 epochs on a GPU, as training could take a lot of time. The list of required libraries to run this project are as follows. Please make sure these libraries are installed on python to have a smooth execution.

- music21
- pygame
- pypianoroll
- pickle
- glob
- numpy
- keras
- tensorflow
- pandas
- matplotlib

2 Running the code

Place all files after unzipping the zip folder for code in a directory. The directory will look like the following.

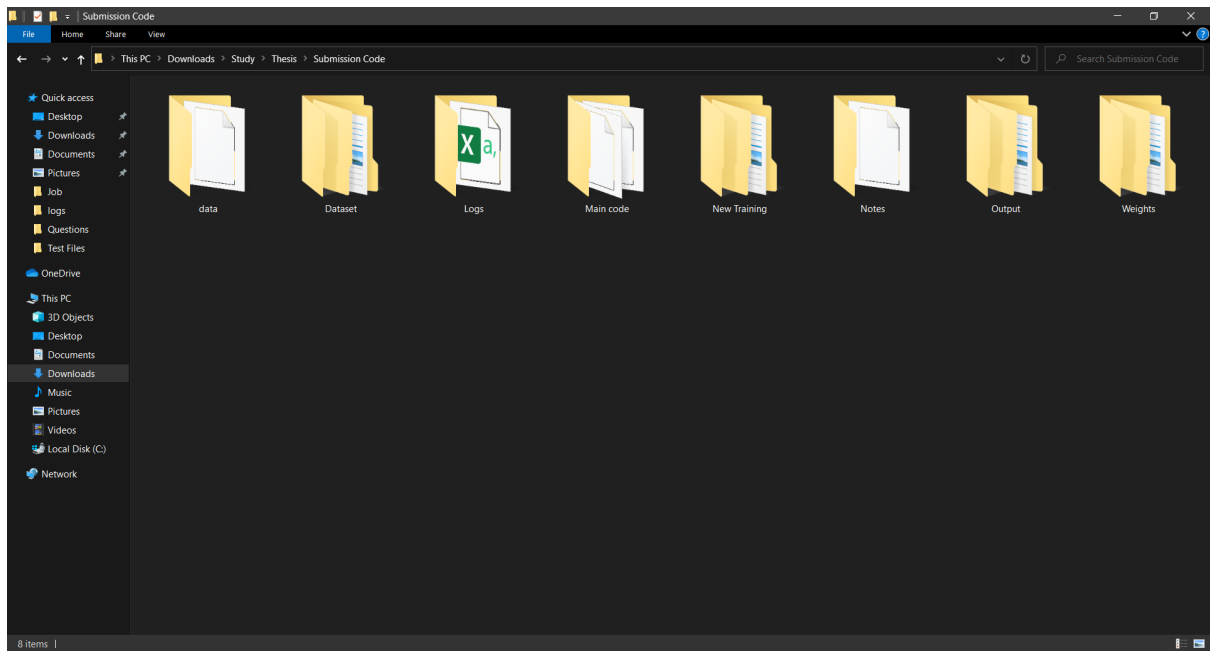


Figure 1: Folder structure

There are 2 datasets provided in the folder. One is a smaller subset for faster training, and other is a larger dataset which would take more time in training.

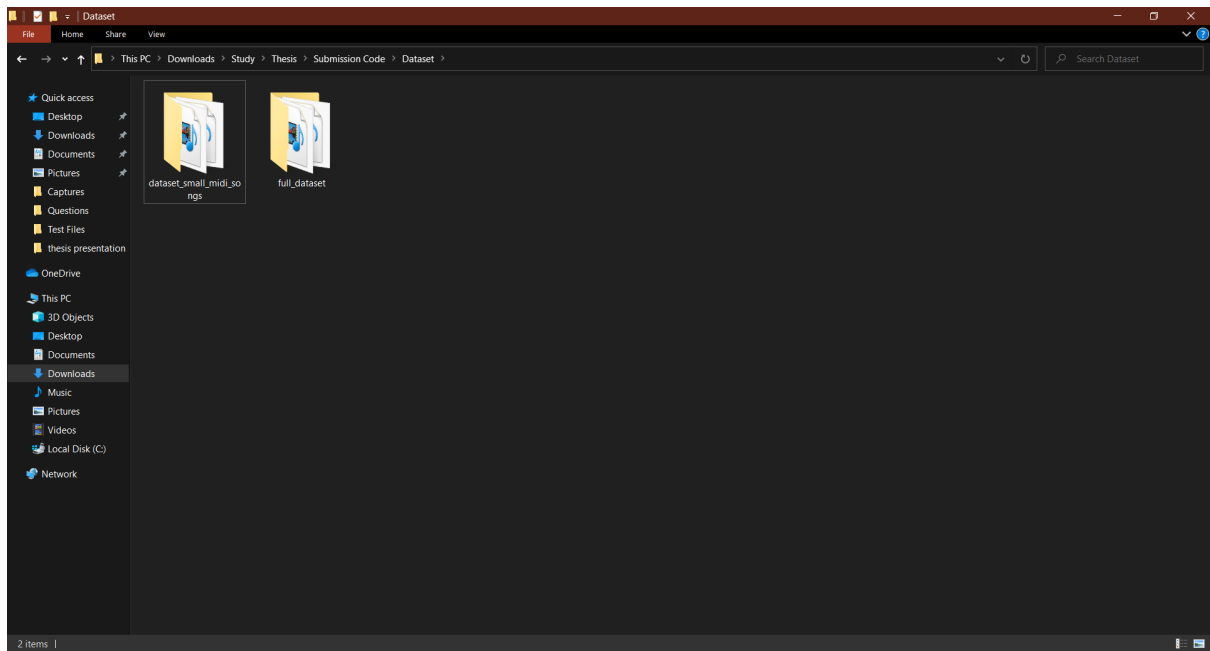


Figure 2: Dataset

There are 5 main code files which are present in the folder named ""Main Code"" as shown below

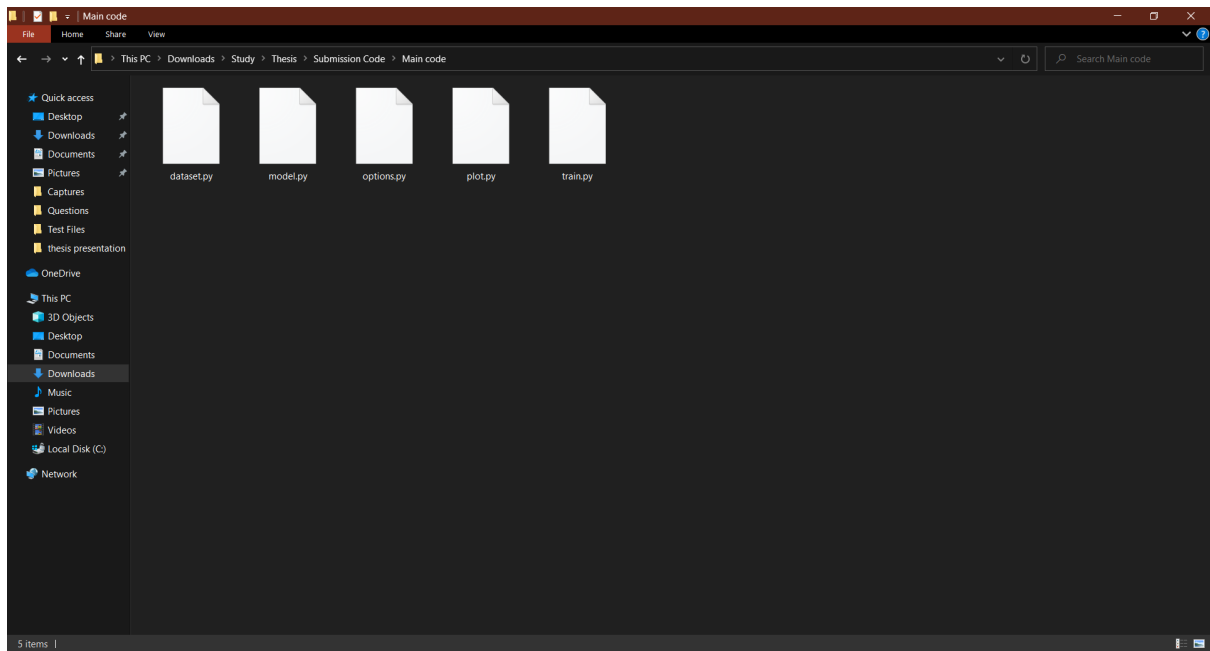


Figure 3: python files

The order to open codes is : Options.py>dataset.py>model.py>train.py>plot.py
The options file will be as shown below

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Jul 8 00:48:13 2020
4
5 @author: pc
6 """
7
8 #hyperparameters
9 number_of_epochs=200 #number of epochs to train on
10 batch_size_mid=128 #batch size
11 learning_rate=0.01 #Learning rate
12
13 number_of_notes=500 #number of notes to produce in output
14
15 basePath=r"C:\Users\pc\Downloads\Study\Thesis\Submission Code" #change basepath to wherever you're saving all the files without ba
16
17 input_folder_MIDI=basePath+r"\Dataset\dataset_small_midi_songs\*.mid" #input file path for midi
18 notes_filePath = basePath+r"\data\notes" #path where notes file will be created if you are training full network on new weights
19 notes_filePath_weights = basePath+r"\Notes\notes" #path where notes file included will be kept to load already trained weights
20
21 LSTMweights_path=basePath+r"\Weights\LSTM\weights-improvement-206-0.1950-bigger2.hdf5" #path of trained weight file of lstm
22 LSTMweightsSavePath=basePath+r"\New Training\LSTM\Weights" #path to save new trained weights
23 LSTMlogpath=basePath+r"\New Training\LSTM\Logs\log.csv" #path where logs will be saved for lstm while training
24
25 GRUweights_path=basePath+r"\Weights\GRU\weights-improvement-201-1.5093-bigger2.hdf5" #path of trained weight file of gru
26 GRUweightsSavePath=basePath+r"\New Training\GRU\Weights" #path to save new trained weights
27 GRUlogpath=basePath+r"\New Training\GRU\Logs\log.csv" #path where logs will be saved for gru while training
28
29 LSTMoutput_fileName=basePath+r"\Output\LSTM\LSTM_output.mid" #output file name of midi created by lstm
30 GRUoutput_fileName=basePath+r"\Output\GRU\GRU_output.mid" #output file name of midi created by gru
31
32 commonLogPath=basePath+r"\Logs\logcom.csv" #commonlog file path for graph creation, included in code under logs folder.
33
34 loadTrainedWeights="yes" #works on test "yes" and "no" : change to no if you want to train model again from scratch and are not usi

```

Figure 4: options

This file contains all the customisable variables. The file mainly loads the paths needed for the code to work and hyperparameters. There is a variable called basepath. Change this to the folder where the zip folder structure is present. Other paths are made relative to this path. There is a number of notes variable which decides how many number of notes are to be generated by output. This is set as 500 for now.

Also, there is a variable which works on text "yes" and "no". It is by default set as yes to load pretrained weights which have been trained for 200 epochs. You can change to no if you want to train model again from scratch and are not using included trained weight files.

Run this python file after changing the values. This will load variables into memory. Then run the next set of files dataset to parse dataset, model to create model, train to train the model (it will not start epoch training to the model in case loaded weights are being used) and then plot to finally plot the output.

After plot file is executed, it will display some graphs and analysis of the code by default.

The Generated midi will be saved in the following folders inside Output folder under respective GRU and LSTM folders.

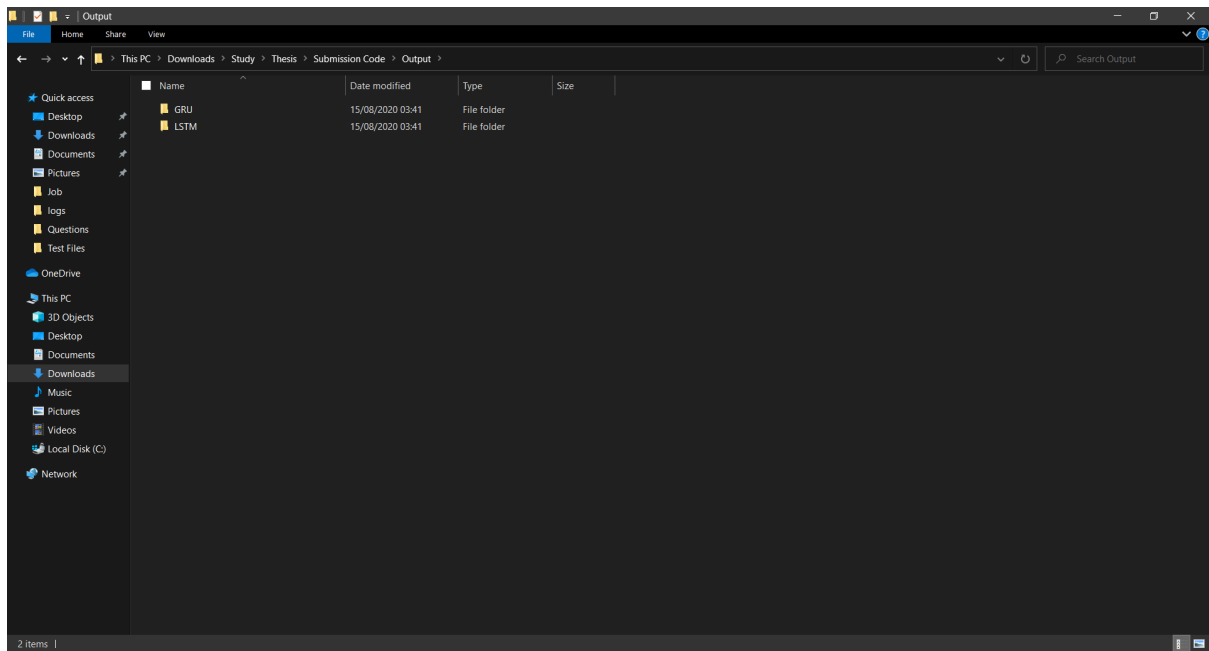


Figure 5: Output folder

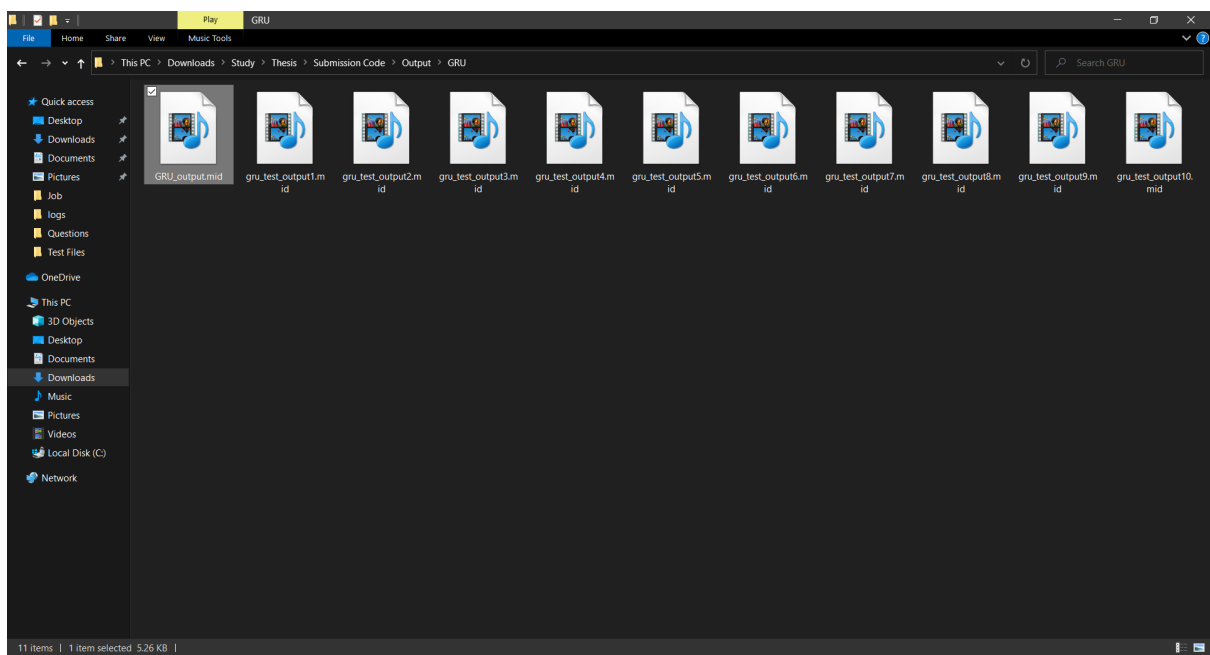


Figure 6: GRU folder

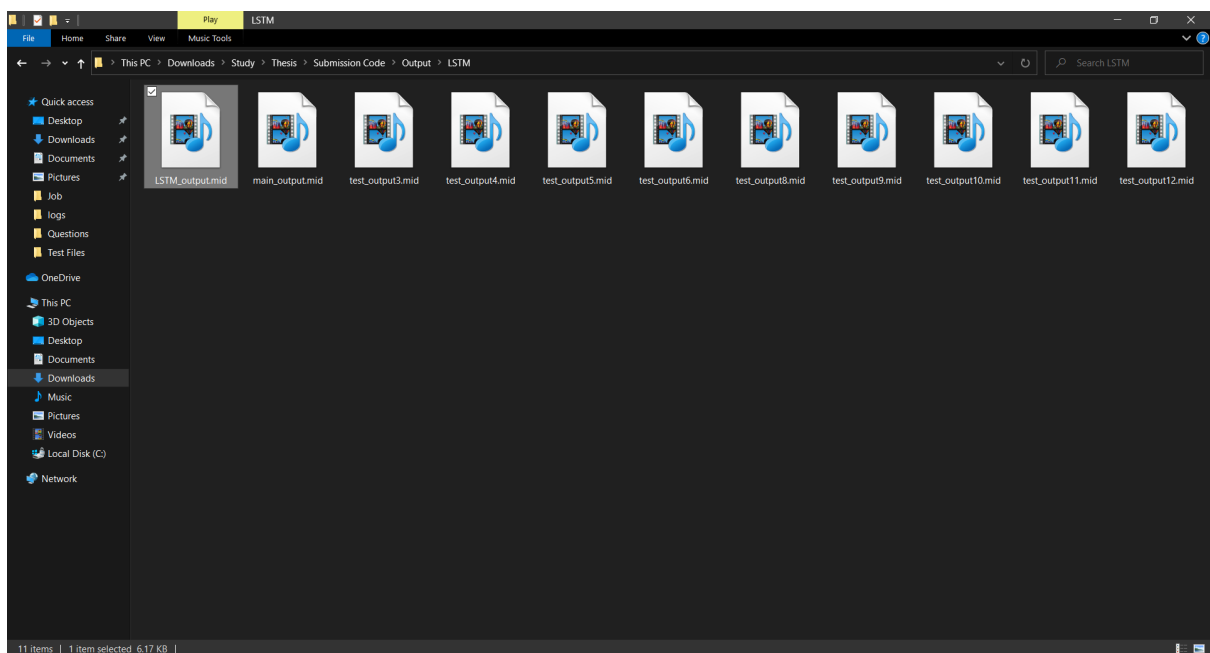


Figure 7: LSTM Folder

3 Output

The output produced is a MIDI file which has music notes. This file can be played either in python console itself, or on an online MIDI player such as here ¹. Select the generated MIDI file from folder and choose next and play. It will bring a piano roll in front of you and you can press play button to play and pause the melody. For example :

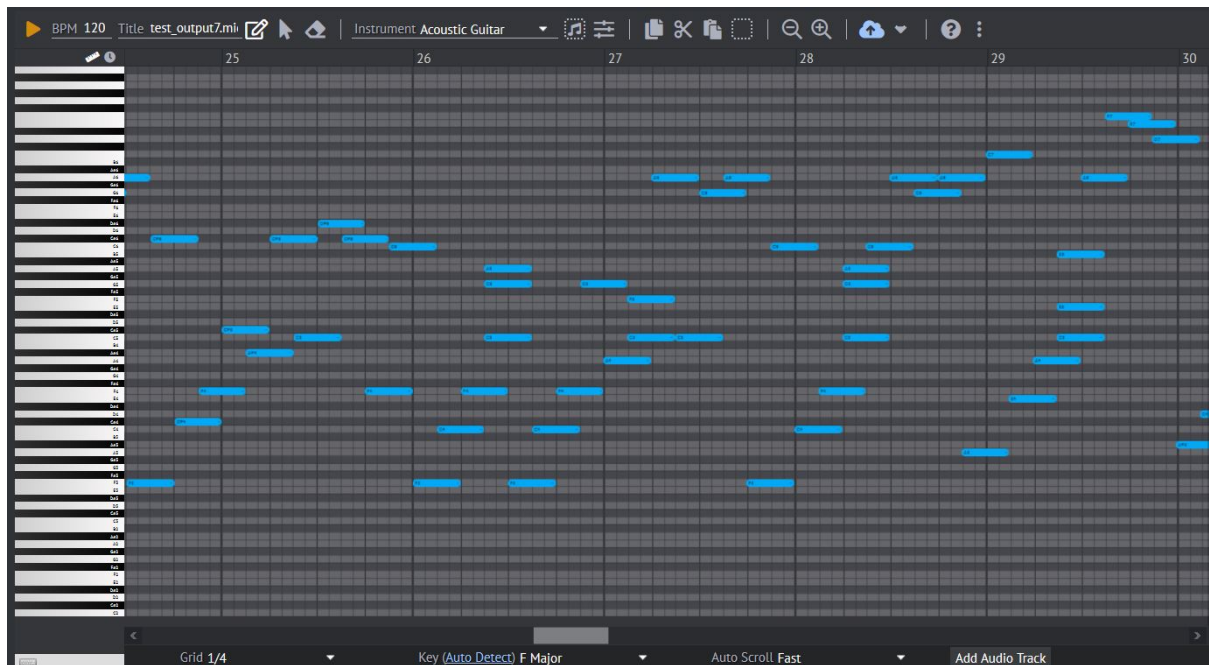


Figure 8: website player

To play on python console itself, after you run the plot.py file, go to the location on code that reads "play the clips generated". For reference:

¹<https://onlinesequencer.net/import>

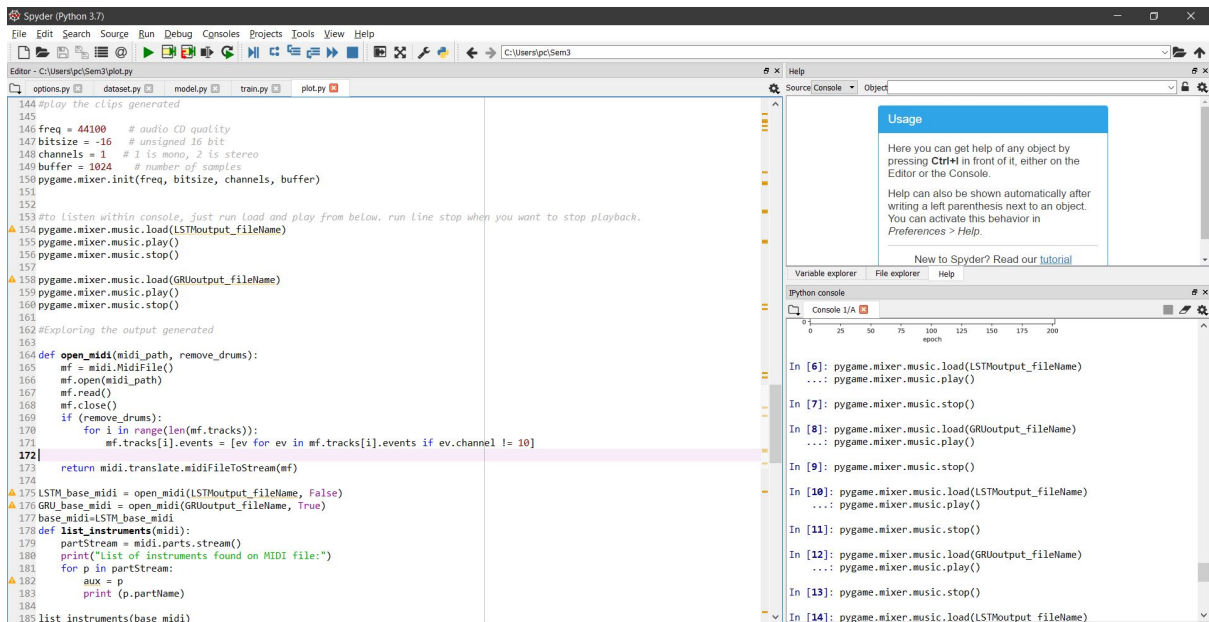


Figure 9: console player

Execute these lines on the code :

```
pygame.mixer.music.load(LSTMoutput_fileName) pygame.mixer.music.play()
```

The console will start playing the melody. To stop, execute the line below it for stop:

```
pygame.mixer.music.stop()
```

You can do the same for playing GRU file as well.

References