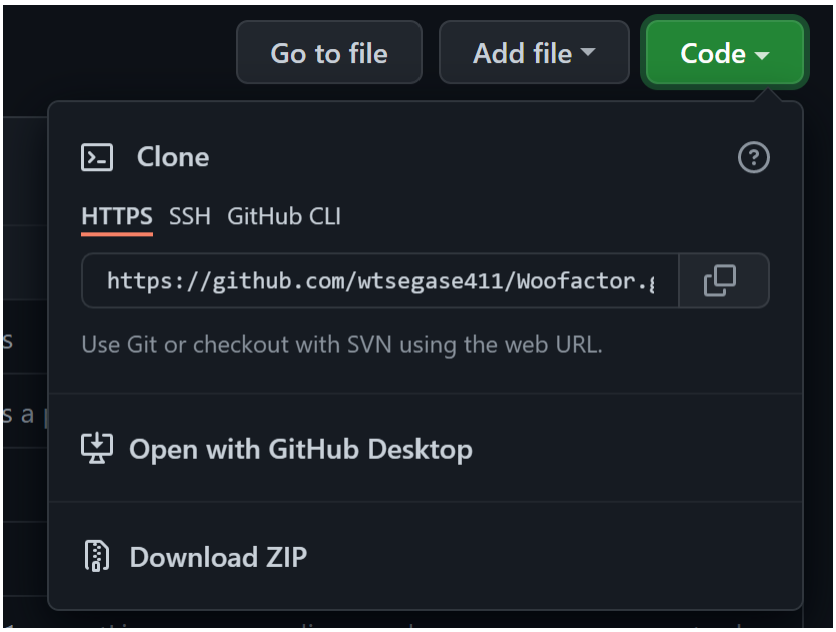
Software User Guide

Howton, J., Kim, J., Osmon, L., Tsegaselassie, W.

Installation Steps

Clone or download the repository



Navigate to the Code tab in the GitHub repository and either download the ZIP or copy the URL for cloning. If cloning, navigate to the desired directory and execute “git clone [url]” to grab the latest version otherwise unzip the downloaded file.

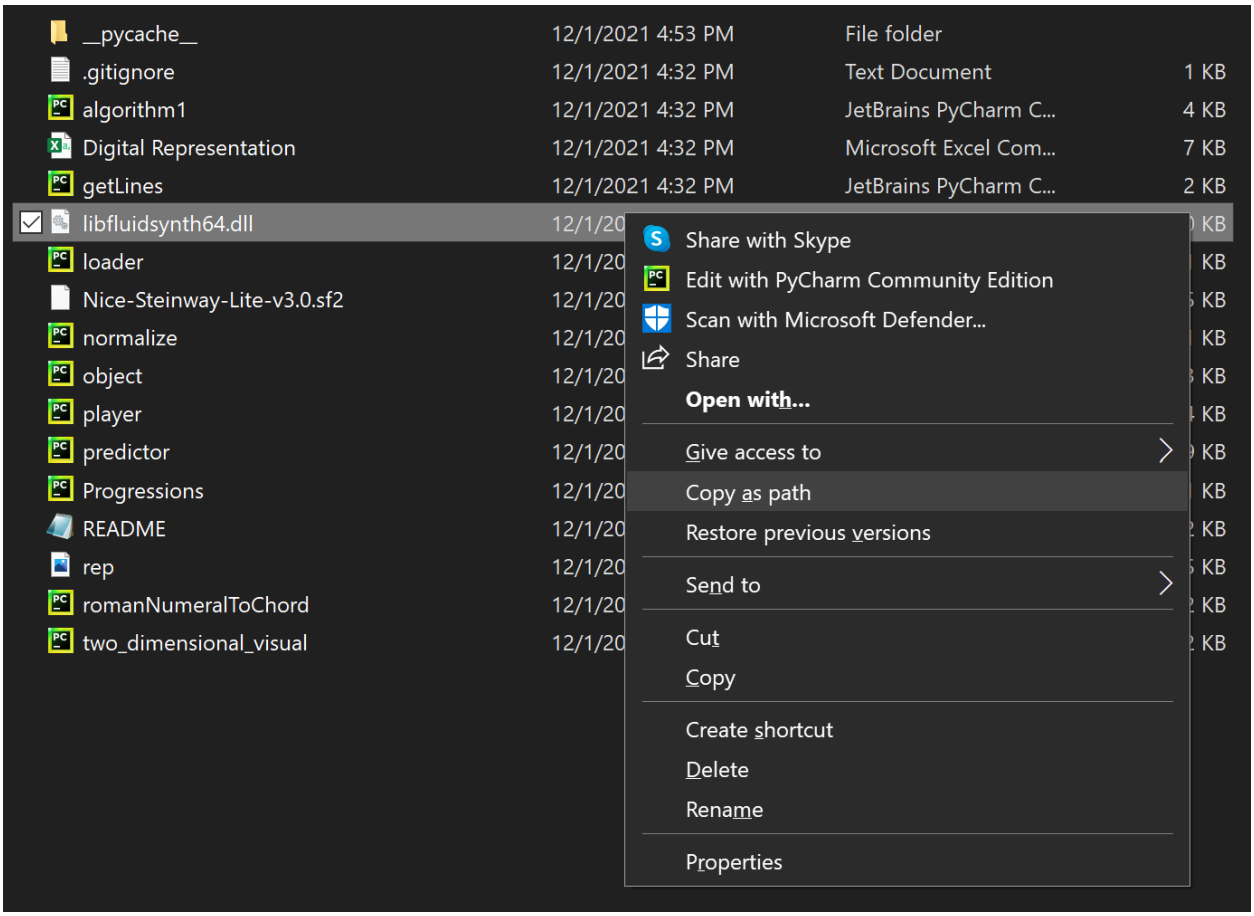
Installing packages

The following packages are required: mingus, matplotlib, and fluidsynth. Either install using the command line with the command “pip install [package name]” or install via an IDE like PyCharm.

Configuring fluidsynth– WINDOWS ONLY

**For Windows users only, others can skip to running the predictor.

There is currently an issue with fluidsynth on windows running machines where the installation is missing a specific .dll file that is needed. To fix this the user must manually add it to fluidsynth's path and the following steps explain how to do that.



Navigate to the WooFactor directory and locate the "libfluidsynth64.dll" file. Shift + Right-Click and select "Copy as path." Paste the path in a text file or notepad to save for later.

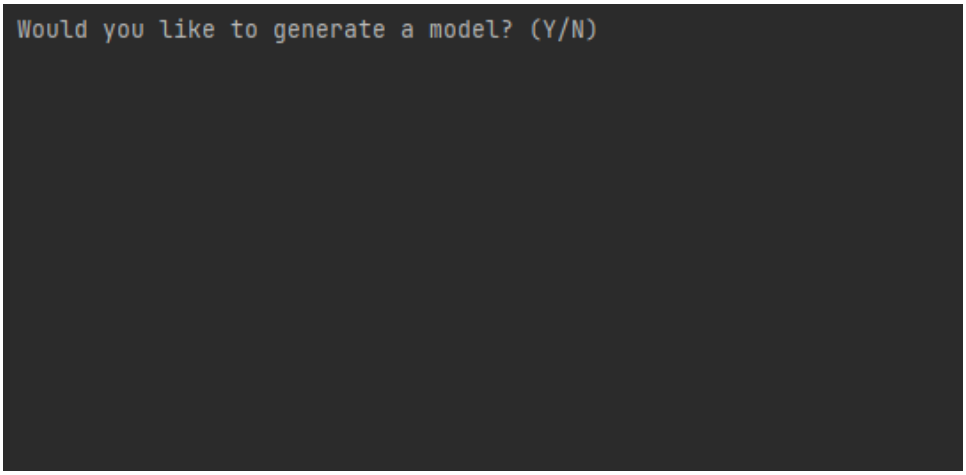
Next in the search bar type "%Appdata%" and navigate to the Local directory. Locate your Python installation directory inside the Packages directory. The path looks like this C:\Users\[USER]\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_XXX\

Navigate to: LocalCache\local-packages\Python39\site-packages\mingus\midi\ and open pyfluidsynth.py in a text editor.

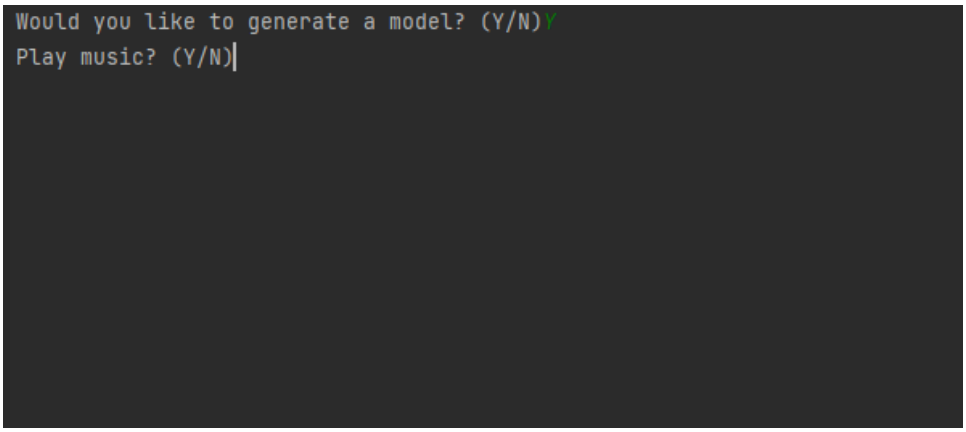
Replace “libfluidsynth” inside of the find_library() field with the path you copied earlier. Make sure that the “\” characters are replaced with “/” and the path is inside quotations.

Running the Predictor

The main algorithm that the user will interface with is predictor.py. Inside of the directory or through an IDE run predictor.py.



The user is prompted to generate a model. If this is the first time predictor.py is run the user will need to input “Y” to create a model based on the existing data. Any time data is adjusted a new model should be generated however given an invalid prompt the value will default to N.



The next prompt determines whether the output from the algorithm will be played aloud by the synthesizer. The default given an invalid prompt is N.

```
Would you like to generate a model? (Y/N)Y
Play music? (Y/N)N
What key would you like the final out put to be in?
```

The user is then given the choice of the key of the output. Any major or minor key is valid and the user will be prompted again if the input is invalid.

```
Would you like to generate a model? (Y/N)Y
Play music? (Y/N)N
What key would you like the final out put to be in? C
Comma separated list of chords in roman numeral form:
```

The next prompt is for the initial data provided to the algorithm. The user must input a comma separated list of chords in the form of Roman numerals. This input will be used to predict the next chord progression.

```
Would you like to generate a model? (Y/N)Y
Play music? (Y/N)N
What key would you like the final out put to be in? C
Comma separated list of chords in roman numeral form: I, V, II
| I | V | II | V | V | V | V | IV | I | I |
| C | G | Dm | G | G | G | G | F | C | C |
Generate another? ('stop') to end:
```

The final prompt allows the user to generate another progression from new input otherwise the program will terminate when the user inputs “stop”