++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

README

Vanessa Job and Xiaomeng Li 11/14/2017

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

This project runs using Python 2. To run this project, you will need the following packages installed:

- `librosa`
- `numpy`
- `scikit-learn`
- `scikits.audiolab`
- `scikits.talkbox`
- `scipy`

To install these, for each package, issue the command:

```
pip2 install PACKAGE_NAME
```

where `PACKAGE_NAME` is the name of the package you want to install.

**File Structure**

You have been provided with a directory called `Code`. In the `Code` directory, you will find the following subdirectories:

- `src`
- `data`
- `output`
- `something`

The source code for the feature extractor and classifiers are in the directory `src`. All of these programs write their output files to the directory `output`. Note: the feature extractor is expecting all of its training data in `Code/data/genres` and all of its validation data in `Code/something/validation`. (Sorry, this is cludgy. It won't work without the extra `something` directory in there. It's because I designed the extractor with the directory structure of the training data in mind, not the validation data.)

**Feature Extraction and Classification for FFT**

To run the feature extractor for FFT, move to the `src` directory and issue the command:

```
python2 preprocesor.py fft
```

This produces two output files: `train.csv` and `validate.csv`, which are the files that contain the extracted features for the training data and the validation data using the FFT to extract. (These files end up in the `output` directory.) To classify this data with the SVM classifier, issue the command:

```
python2 svmmethod.py
```

To classify this data with the logistic regression method, issue the command:

```
python2 logisticmethod.py
```

The output of the classification, `result.csv`, contains classified validation data in a format suitable for submitting to Kaggle. You can find it in the `output` directory.

**Feature Extraction and Classification for MFCC and Custom Classifier**

To run the feature extractor for MFCC or the custom extractor, issue the command

```
python2 preprocesor.py mfcc
```

or

```
python2 preprocesor.py custom
```

respectively. The commands to run the classifiers are the same as in the FFT case.

**Warnings:**

Sometimes the MFCC function produces output that is NaN or inf. You will see a warning on the screen. You can ignore this warning. (We cull this data.)

**Optional:**

If after running the feature extractor one time (to get the training data) you find you want to run it on different validation data, issue the command:

```
Python2 preprocesor.py --v fft
```

The `--v` option tells the preprocessor just to do the extraction for the validation data. (That is, skip extraction for test data.) You will probably have no cause to use this.