# Dataset

The dataset is a collection of three different types of data on different genres of music. The genres of music in the dataset are as follows:

* Blues: 100 audio files, 100 image files
* Classical: 100 audio files, 100 image files
* Country: 100 audio files, 100 image files
* Disco: 100 audio files, 100 image files
* Hiphop: 100 audio files, 100 image files
* Jazz: 100 audio files, 100 image files
* Metal: 100 audio files, 100 image files
* Pop: 100 audio files, 100 image files
* Reggae: 100 audio files, 100 image files
* Rock: 100 audio files, 100 image files

Other than the above-mentioned data, there are two .csv files. One .csv file contains all the statistical information of all the 3 second snippets of each audio file for each genre. The other .csv file contains the same information but for 30 second snippets instead of the 3 second ones.

The features in both the above-mentioned .csv files are the same which are as follows:

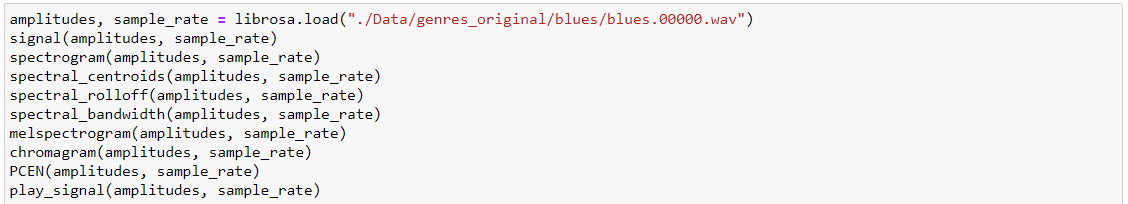
* Filename: name of the music file to which the row belongs to
* Length: length of the song in microseconds. (3 seconds in one .csv file and 30 seconds in the second .csv file for all instances).
* chroma\_stft\_mean: mean of the short-time fourier features and chroma features
* chroma\_stft\_var: variance of the short-time fourier features and chroma features
* rms\_mean: root mean square value
* rms\_var: variance for root mean square value
* spectral\_centroid\_mean: mean of the spectral centroid of the music spectrum graph
* spectral\_centroid\_var: variance of the spectral centroid of the music spectrum graph
* spectral\_bandwidth\_mean: mean of the spectral bandwidth of the music spectrum graph
* spectral\_bandwidth\_var: variance of the spectral bandwidth of the music spectrum graph
* rolloff\_mean: mean of the slope of the section just past the cutoff corner frequency in dB per octave.
* rolloff\_var: variance mean of the slope of the section just past the cutoff corner frequency in dB per octave.
* zero\_crossing\_rate\_mean: mean of rate at which a signal changes from positive to zero to negative or from negative to zero to positive.
* zero\_crossing\_rate\_var: variance of rate at which a signal changes from positive to zero to negative or from negative to zero to positive.
* harmony\_mean : harmonic mean of the music
* harmony\_var: harmonic variance of the music
* tempo: the rate of speed of a musical piece or passage

following are the Mel-frequency cepstrum mean and variance for each of the 20 parts divided for the Mel-frequency cepstrum.

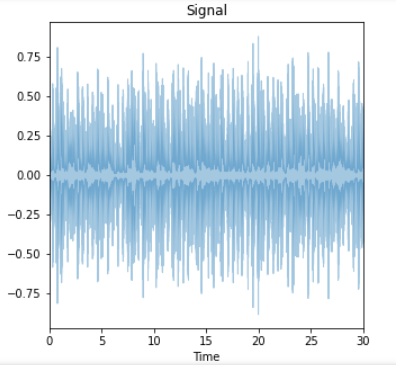
* mfcc1\_mean
* mfcc1\_var
* mfcc2\_mean
* mfcc2\_var
* mfcc3\_mean
* mfcc3\_var
* mfcc4\_mean
* mfcc4\_var
* mfcc5\_mean
* mfcc5\_var
* mfcc6\_mean
* mfcc6\_var
* mfcc7\_mean
* mfcc7\_var
* mfcc8\_mean
* mfcc8\_var
* mfcc9\_mean
* mfcc9\_var
* mfcc10\_mean
* mfcc10\_var
* mfcc11\_mean
* mfcc11\_var
* mfcc12\_mean
* mfcc12\_var
* mfcc13\_mean
* mfcc13\_var
* mfcc14\_mean
* mfcc14\_var
* mfcc15\_mean
* mfcc15\_var
* mfcc16\_mean
* mfcc16\_var
* mfcc17\_mean
* mfcc17\_var
* mfcc18\_mean
* mfcc18\_var
* mfcc19\_mean
* mfcc19\_var
* mfcc20\_mean
* mfcc20\_var
* label: output label for each instance showing the genre

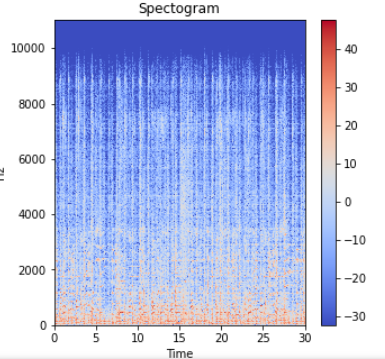
# Machine Learning

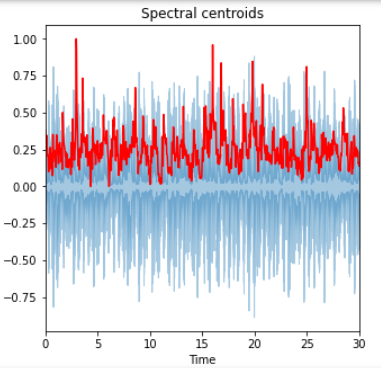
The aim of this research code is to create a machine learning algorithm that can classify different music based on genre. The algorithm used for this is the Deep Neural Network provided by Keras. There are many graphs that pertain to data preprocessing and Machine Learning. The purpose of this document is to explain them.

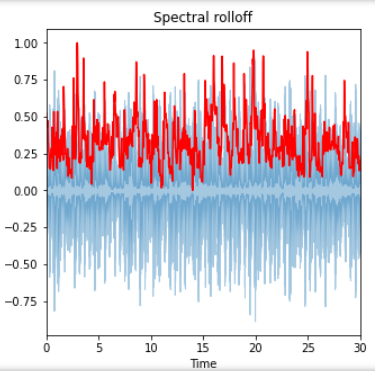


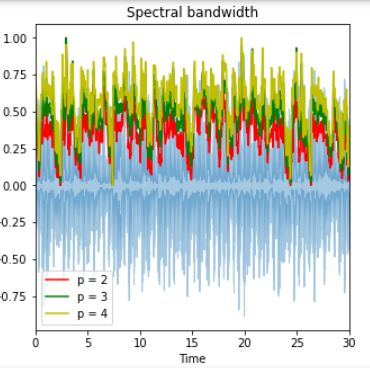
In the above figure, there are 8 functions if the first and the last line of the code is excluded. Each of these functions are meant for data visualization. These present a music file from the dataset in different formats. The functions visualize the audio file using the following graphs in order of the functions:

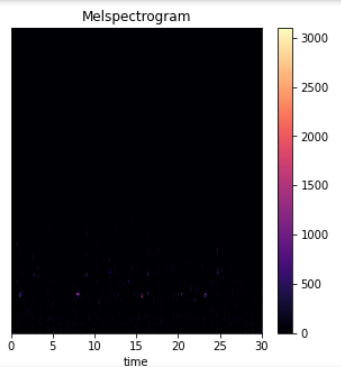


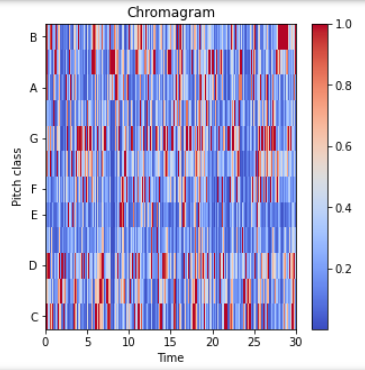


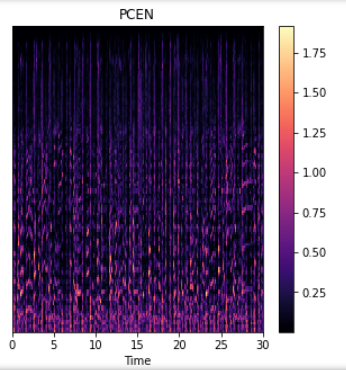




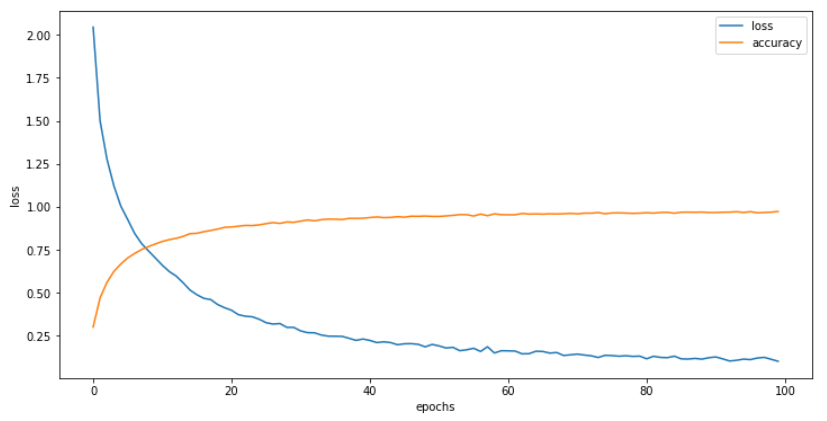


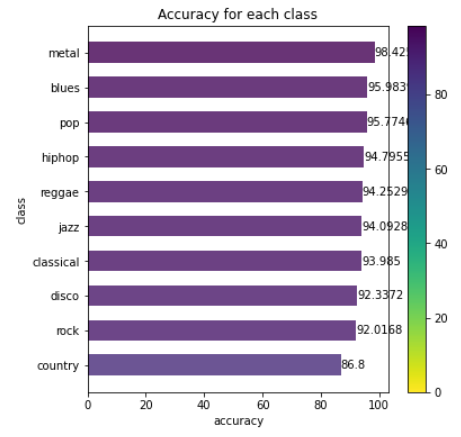






The graph below summarizes the performance of the algorithm graphically. The blue line in the graph below shows the loss of the algorithm and the orange line shows the accuracy of the algorithm. As the number of epochs increase as shown on the x-axis, the loss of the algorithm i.e., the wrong predictions decrease and the accuracy of the algorithm increases. Both the graphs are asymptotic in nature meaning that the number of epochs will not matter after a certain point in the graph.

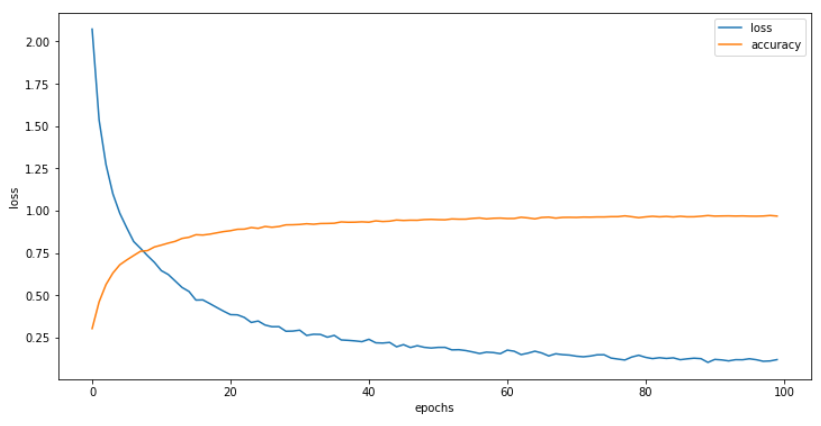




The above graph shows the accuracy of the algorithm for each genre of music.

The same algorithm is also trained with the PCEN form of data. PCEN is the abbreviation of Pre-Channel energy normalization. PCEN provides a normalized time vs frequency distribution of the audio files by performing automatic gain control and nonlinear compression. The data is recorded in a different dataset and is then used to train the algorithm. The 3 second feature file and the PCEN data file are the same except the fact that the PCEN file has 63 columns dedicated to the PCEN values for each instance.

The performance of the algorithm after training is shown by the following two graphs:



The blue line in the graph below shows the loss of the algorithm and the orange line shows the accuracy of the algorithm. As the number of epochs increase as shown on the x-axis, the loss of the algorithm i.e., the wrong predictions decrease and the accuracy of the algorithm increases. Both the graphs are asymptotic in nature meaning that the number of epochs will not matter after a certain point in the graph.

