Music Genre Classification

Ximerakis Ioannis, AM 4450
Tsarantanis Dimitrios, AM 4479

Abstract

The objective of this project is to analyze music-audio signals with python and use machine learning techniques to classify music samples into different genres of music. A database of music samples will be used to extract useful features, that are going to be the basis of the classification process.

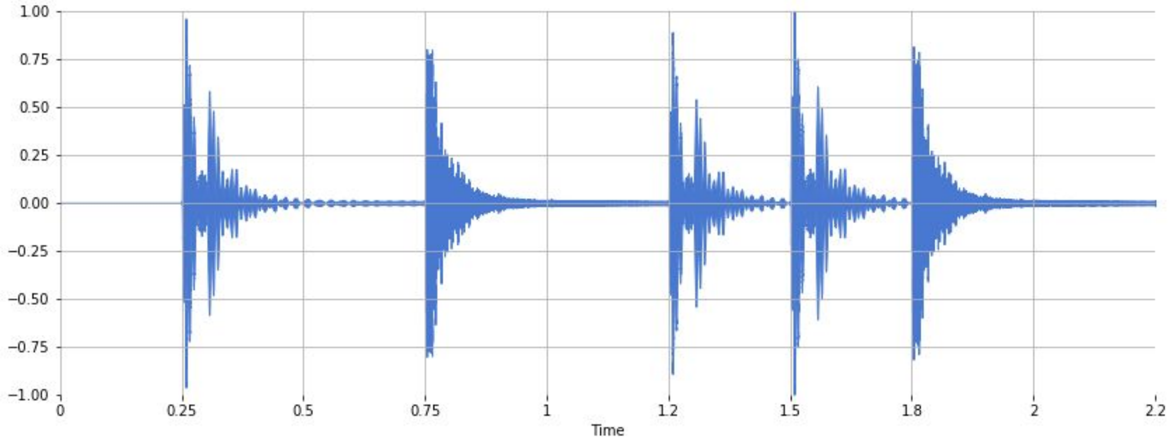
Introduction

Signal Processing

The first step in this process is to find a tool that can read audio files and do some basic audio processing functions. For this purpose the Librosa library from python will be used. Librosa can read audio files and transforms them into audio time series as an array that contains the frequencies in KHz.

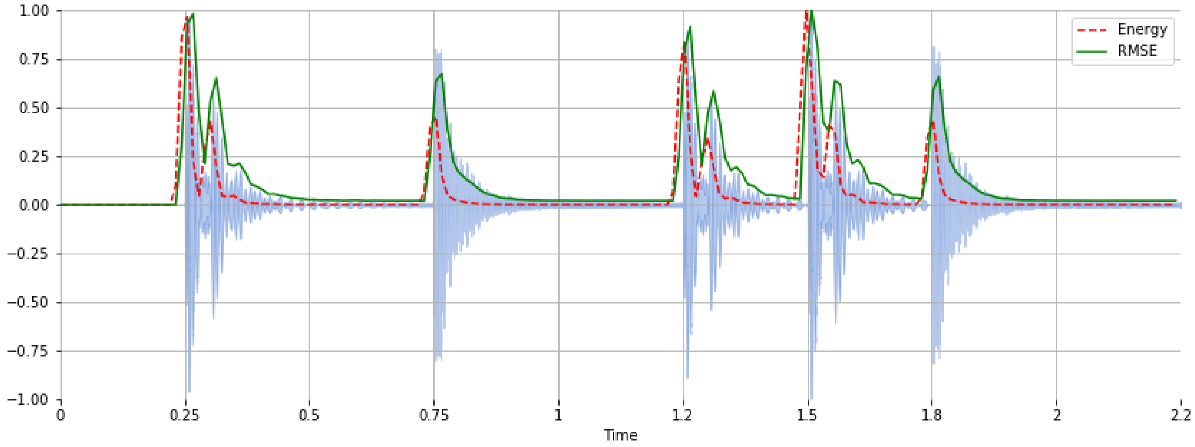
Feature Extraction

Using basic Librosa functions on an audio sample, the following features can be extracted :

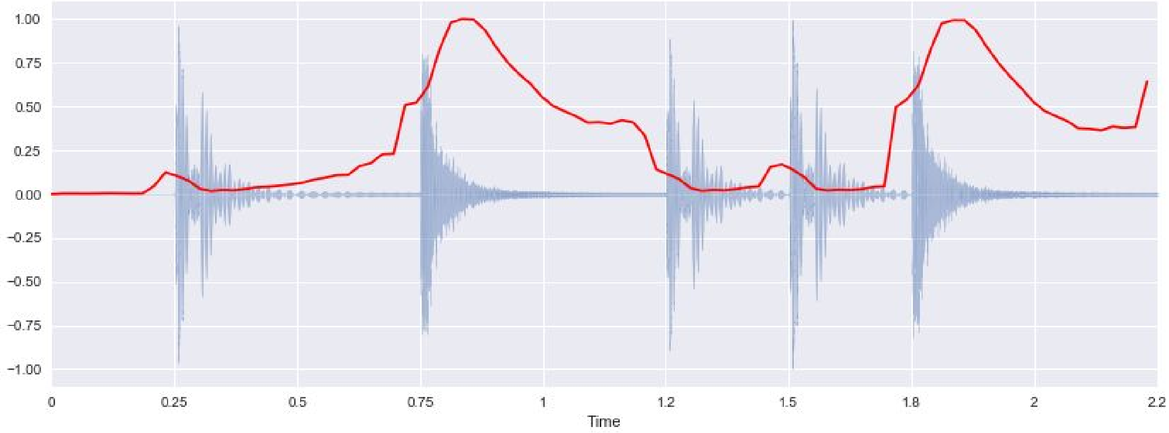


- **Rmse**

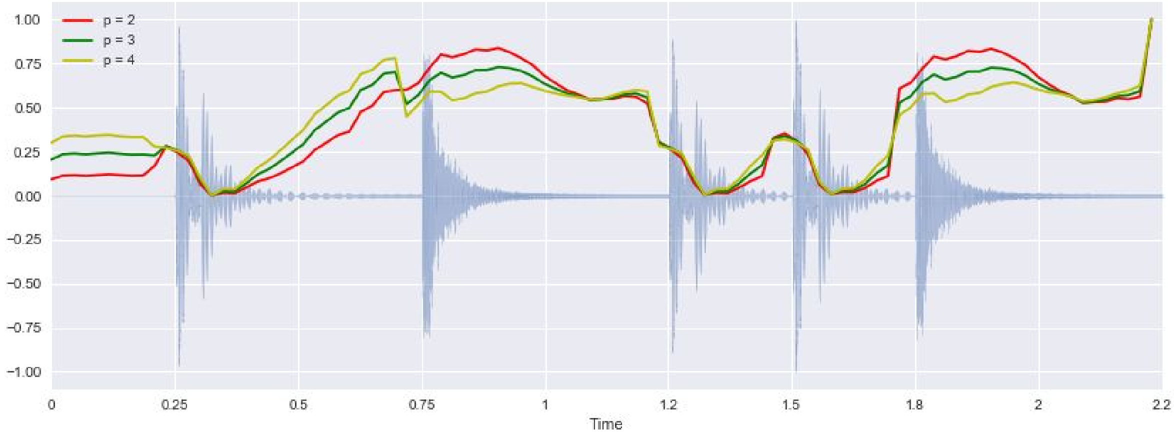
  RMS is the root-mean-square value of a signal and represents its average energy.



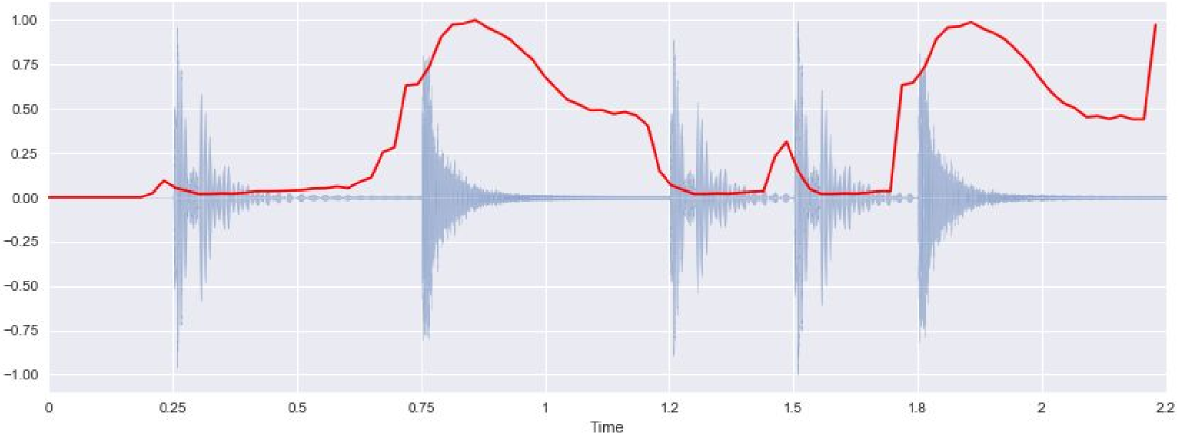
- **Spectral Centroid**
  The spectral centroid is a measure used in digital signal processing to characterise a spectrum. It indicates where the “center of mass” of the spectrum is located.



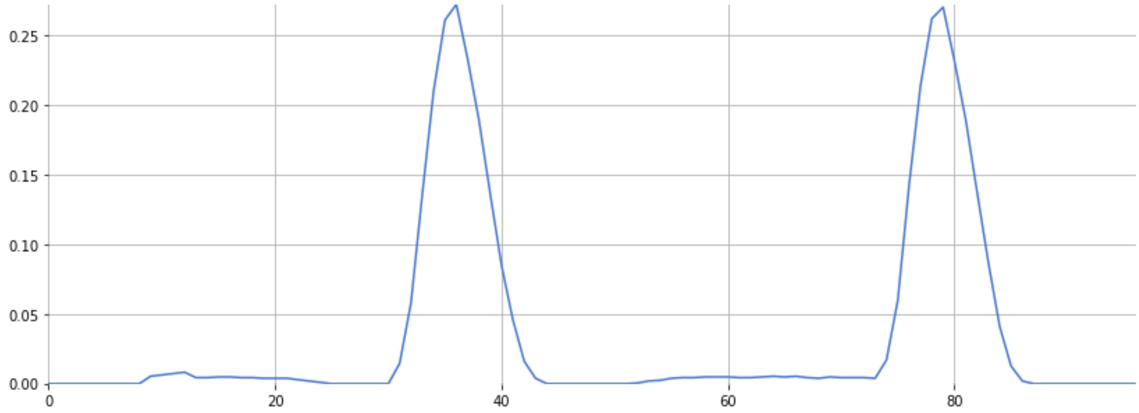
- **Spectral Bandwidth**
  The spectral bandwidth is the FWHM and represents the frequency range at half-maximum of a peak.



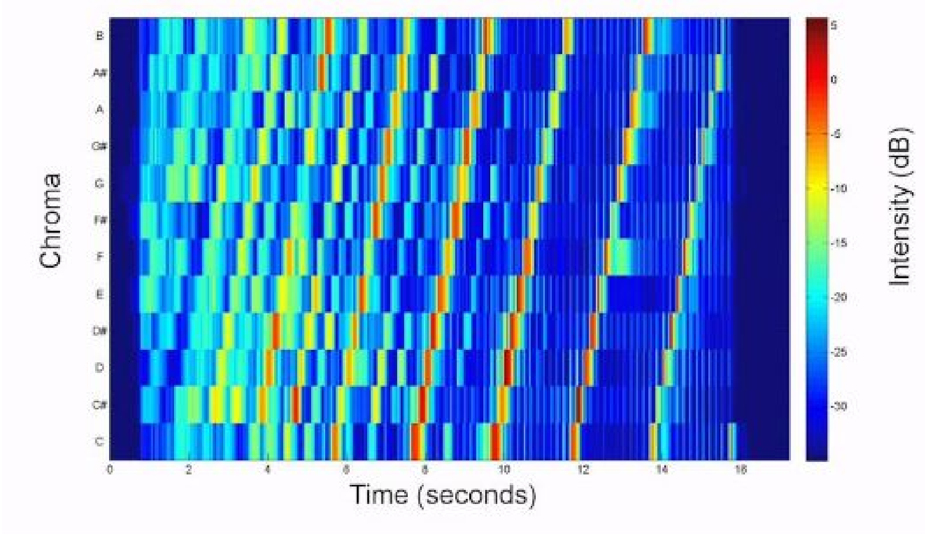
- **Spectral Roll-off**
  Spectral rolloff is the frequency below which a specified percentage of the total spectral energy lies.



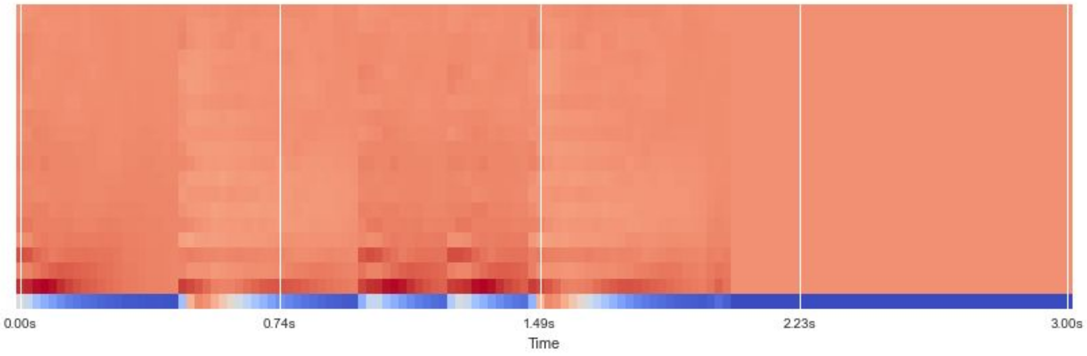
- **Zero Crossing Rate**
  The zero-crossing rate is the rate of sign-changes along a signal, i.e., the rate at which the signal changes from positive to zero to negative or from negative to zero to positive. This is a key feature to classify percussive sounds.



- **Chroma Features**
  In music, the term chroma feature or chromagram closely relates to the twelve different pitch classes. Chroma-based features are a powerful tool for analyzing music whose pitches can be meaningfully categorized (often into twelve categories). One main property of chroma features is that they capture harmonic and melodic characteristics of music.



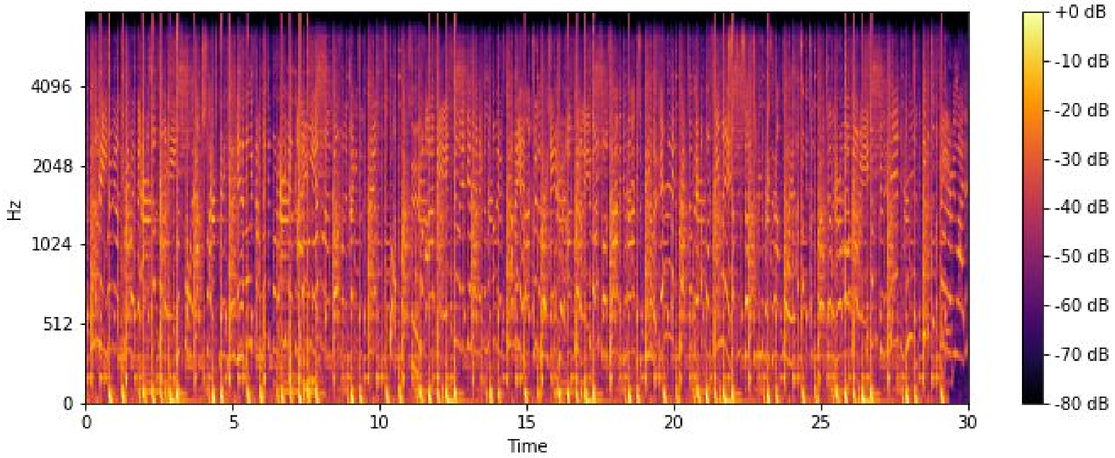
- **Mel-frequency cepstral coefficients**
  The mel frequency cepstral coefficients (MFCCs) of a signal are a small set of features (usually about 10-20) which concisely describe the overall shape of a spectral envelope.



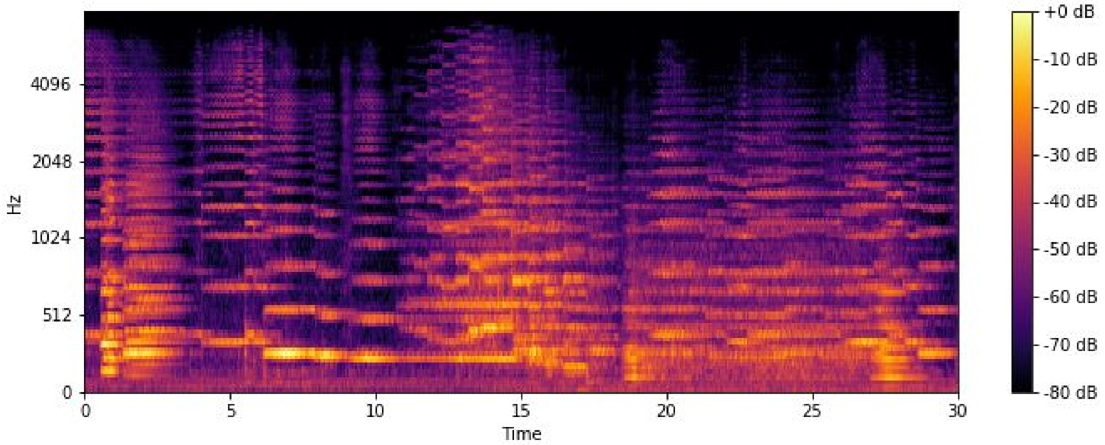
Spectrogram

A spectrogram is a visual representation of the spectrum of frequency of a signal as it varies with time. In 2-D arrays, the vertical axis is frequency while the horizontal axis is time. Here are some typical spectrograms from four different genres. Taking into account the features that were mentioned, it’s easy to see that different patterns emerge for each genre. These patterns are going to give the extracted features their unique identity.

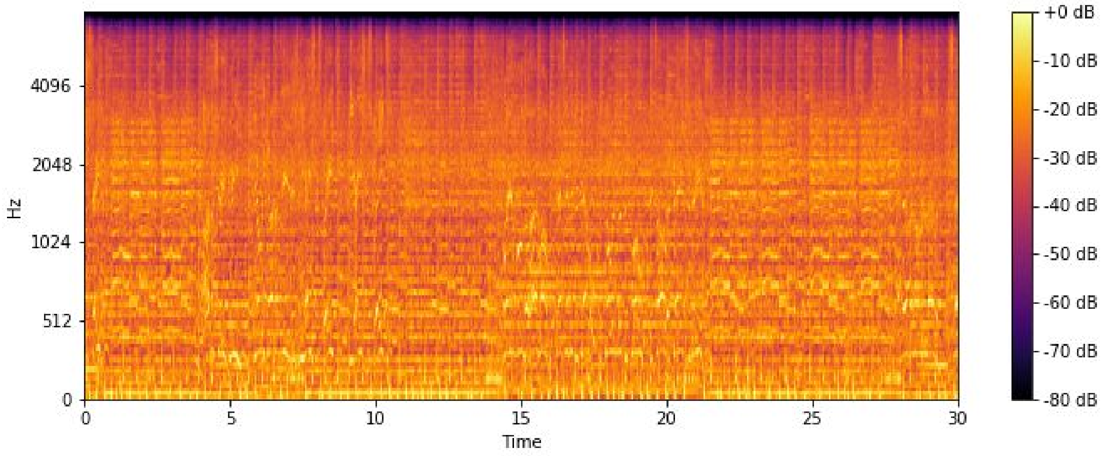
Hip-Hop



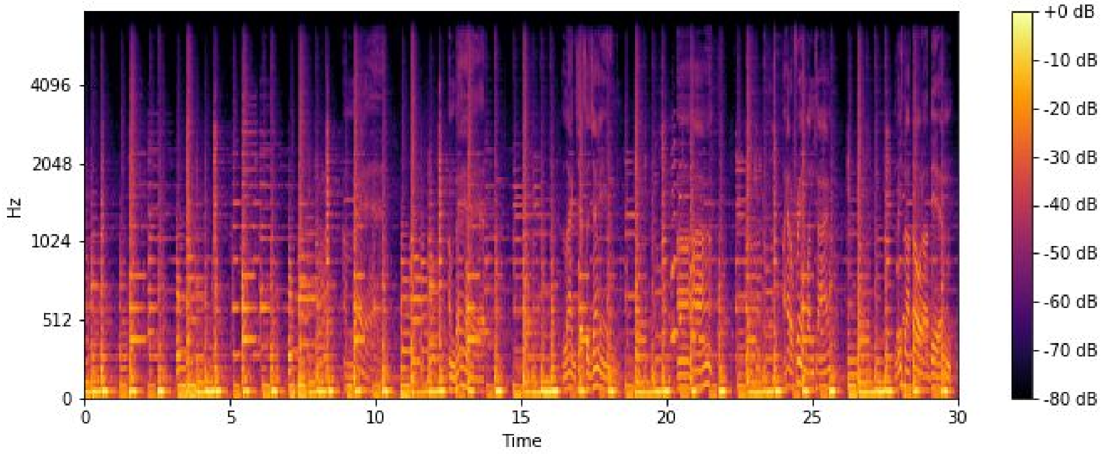
Classical



Metal



Blues



Dataset

The dataset used in this project is the GTZAN dataset , which consists of 1000, 30 second long, audio clips evenly distributed among 10 genres of music (blues, classical, country, disco, hiphop, jazz, reggae, rock, metal and pop).

Classification

There are two basic ways to perform the classification, either by using the song spectrograms and building an image classification neural network, or by using the extracted song features through the use of classifiers or neural networks.

The first method is expected to give better results, but requires the use of RNN or parallel CNN-RNN models to achieve adequate accuracy levels. Due to computational limitations this method is not going to be used in this project.

For this project the second method was chosen. The song features are extracted and stored in a .csv file and a selection of imported classifiers are used for the classification. A different approach using Deep Neural Networks can be used.

Classifiers

The following classifiers were imported from sklearn:

- **Softmax Regression**
  Logistic Regression is commonly used to estimate the
  probability that an instance belongs to a particular class. This can be generalized to support multiple classes directly. This is called Softmax
  Regression, or Multinomial Logistic Regression.

- **Support Vector Machine**
  A Support Vector Machine (SVM) is a very powerful and versatile Machine Learning model, capable of performing linear or nonlinear classification, regression, and even outlier detection.
  - **Soft Margin Classification**
  - **Nonlinear SVM Classification**
  - **Polynomial Kernel**

- **Decision Tree Classifier**
  A Decision Tree is a simple representation for classifying examples. It is a Supervised Machine Learning where the data is continuously split according to a certain parameter.

- **Random Forest Classifier**
  An ensemble is a group of predictors. A random forest is an ensemble of decision trees. A group of Decision Tree Classifiers can be trained and then depending on the majority of the votes the prediction can be made.

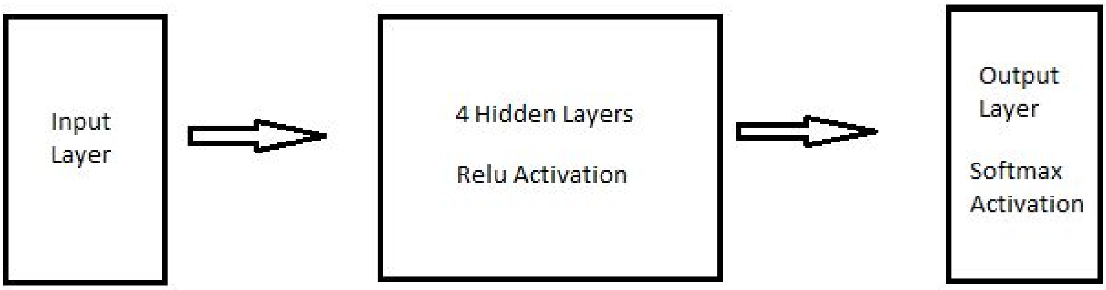
- **Extra Trees Classifier**
  It is possible to make trees even more random by also using random thresholds for each feature rather than searching for the best possible thresholds. A forest with extreme random trees is called Extra Trees.

- **MLP Classifier**
  MLPClassifier stands for Multi-layer Perceptron classifier and uses some underlying neural networks for the classification process.

The dataset was split into a train-set and test-set, which contains the extracted features of the songs, labeled with their corresponding genres. Using the imported classifiers, different models were fitted for each one and evaluated, resulting in different accuracies.

Deep Neural Network

The same dataset was then used on a DNN with the following structure:



Input Layer

4 Hidden Layers

Relu Activation

Output Layer

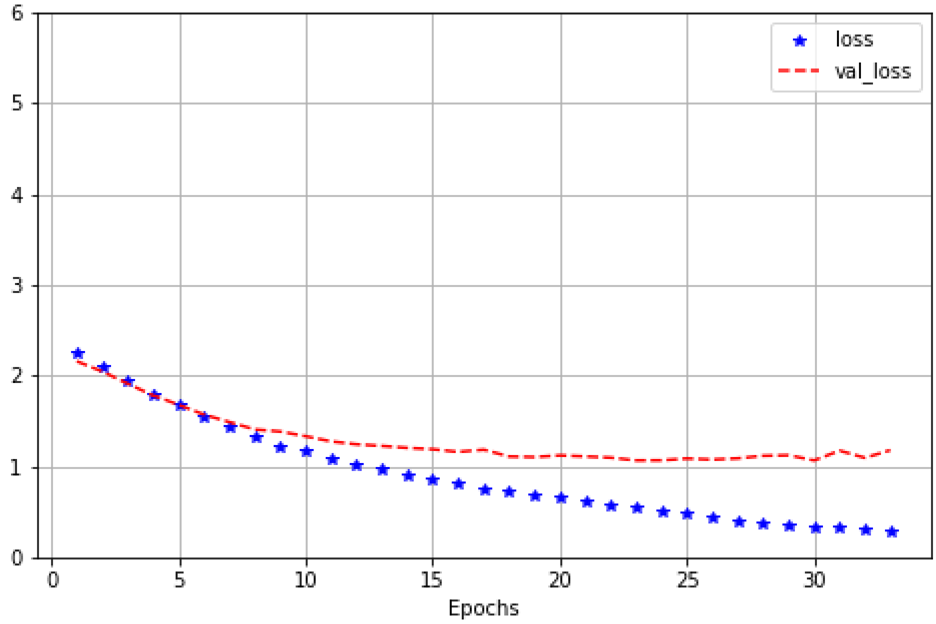
Softmax Activation

Results and Conclusions

In the following table the classifiers used and their respective accuracies are displayed.

Classifier	***Accuracy Score***
Softmax Regression	63%
SVM Regression	67%
SoftMarginClassifier	62%
Non-Linear SVM	60%
Polynomial Kernel	69%
DecisionTreeClassifier	52%
RandomForestClassifier	60%
ExtraTreesClassifier	66%
MLPClassifier	67%

The Deep Neural Network built resulted in 72% accuracy. From the loss-val_loss graph below, no signs of overfitting are visible.



Amongst the classifiers used, it is clear that those oriented towards multiclass classification yield better results. Comparing them with the DNN, although the accuracy difference isn't so significant, the DNN seems to outperform the rest of the classifiers. Fine tuning impacts accuracy greatly for both the classifiers and the DNN, so these results are not definite.
It's also important to take into account the number of songs that the GTZAN database has. There are 100 songs per genre, which is not ideal for neural networks. There are databases with millions of songs, so trying to recreate the process with such datasets would give a much clearer view of the relative performance of the classifiers.

The next step...

Although the method used has relatively good accuracy, even better accuracy can be achieved by using the spectrograms directly as the classification features. Spectrograms are really faithful representations of sound and thus a neural network can learn to identify recurring patterns in different genres' spectrograms. By implementing RNN or parallel CNN-RNN models this can be done quite effectively. So this could be the next step for a project, to further improve accuracy and efficiency.

Bibliography

- Books

  Hands on Machine Learning with Scikit Learn and Tensorflow, 2017, Aurélien Géron

- Websites

  https://musicinformationretrieval.com
  https://github.com/ageron/handson-ml
  https://github.com/priya-dwivedi/Music_Genre_Classification
  https://github.com/parulnith/Music-Genre-Classification-with-Python
  https://github.com/claytonblythe/neuralMusic

- GTZAN Dataset

  ”Musical genre classification of audio signals” by G. Tzanetakis and P. Cook in IEEE Transactions on Audio and Speech Processing 2002

  http://marsyas.info/downloads/datasets.html