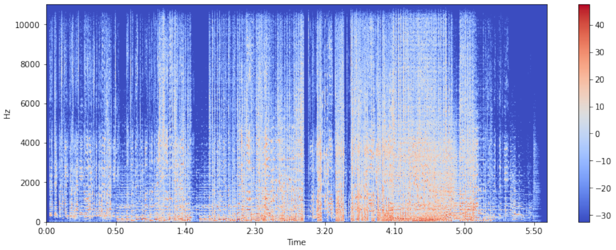
Audio Feature Extraction

Dimitris Petratos

The below notes were inspired from this TDS blog

Introduction

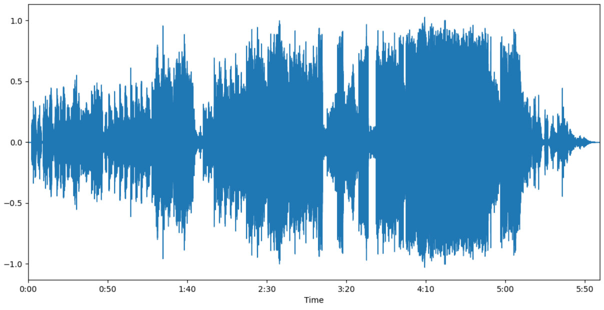
Audio files are composed based on three basic dimensions corresponding to time, amplitude and frequency.
For the porpuse of these notes, we will extract features from Queen's Bohemian Rhapsody.
We start by visualizing the spectogram ouf our audio file:



Spectogram is the representation of the spectrum of frequences that constitute our file.
Moving on, we calculate the Zero Crossing Rate (ZCR). ZCR indicates the rate of signal's change through the time it takes place. In other words, it's the rate that we observe signal's change between positive and negative values. It is formally defined as:

$$ZCR = \frac{1}{T-1}\Sigma_{t=1}^{T-1}\chi_{\mathbb{R}<0}(s_t s_{t-1})$$

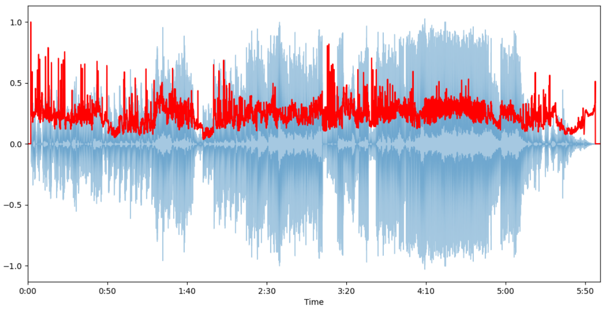
where s is our signal across length T and χ is a characteristic function.
In our case we take the following frequency plot:



which returns us ZCR=840,414.
Continuing, we are about to find the Spectral Centroid (SC). SC indicates where the 'center of mass' is and it can be thought as the weighted average of frequencies in the signal, determined using a Fourier transform with their magnitudes as weights. Formally is defined as:

$$\text{SC}=\frac{\Sigma_{t=1}^{T-1} f(t)s(t)}{\Sigma_{t=1}^{T-1} s(t)}$$

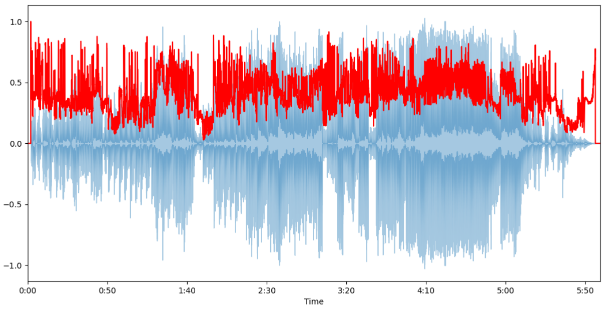
Below we represent the normalized SC, based on the normalized frequency plot:



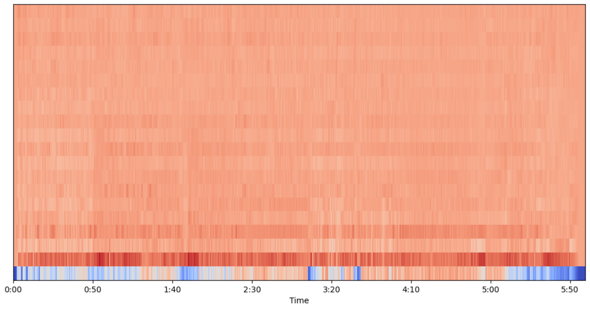
We continue by calculating Spectral roll-off (SRO). SRO is the frequency below which a thresholded percentage of the total spectral energy lies. Commonly that threshold is taken to be 85%. Spectral Energy (SE) is though as the 'power' of the signal and is defined formally as:

$$\text{SE(s)}=\int_{-\infty}^{+\infty}\|s(t)\|^2\chi_T(t)dt$$

χ_T is taken for normalization pursposes.
Below we represent SRO for our case:



Now, we will calculate Mel frequency cepstral coefficients (MFCCs). MFCCs are features, commonly 20, that are used for automate speech and sound recognition in general. Not focusing on details, below we represent the spectrum of MFCCs in our case:



The above could be considered as the most important features for audio files. Next, we will describe the mechanism of a music classifier system, that uses the above and a few more features.

Music Classifier

Below we will describe the pipeline used to make a music genre classifier.

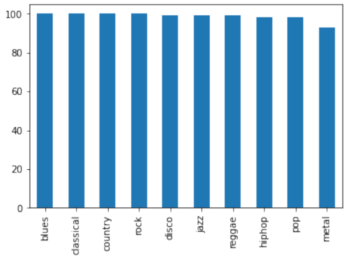
As mentioned before, we will use as features the ones examined earlier. We will also use audio file's Root-Mean-Square (RMS), waveform's Chromagram (CHR) and Spectral's 2-Bandwidth (SBW).

For all of the features that will be extracted, we will compute their average and variance and create a 14-d array which will describe each song used at the training procedure.

As dataset, we will use GTZAN Dataset which constitutes of 10 genres (blues, classical, country, disco, hiphop, jazz, metal, pop, reggae and rock), each containing 100 samples of 30 seconds tracks.

For reasons of storage and execution facility, we will use Google's Colab notebook to convert audio files to the desirable csv format.

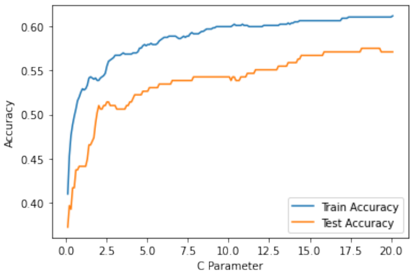
After transforming the audio files and removing dublicates, we get the followting distribution of genres:



taking 986 unique values.

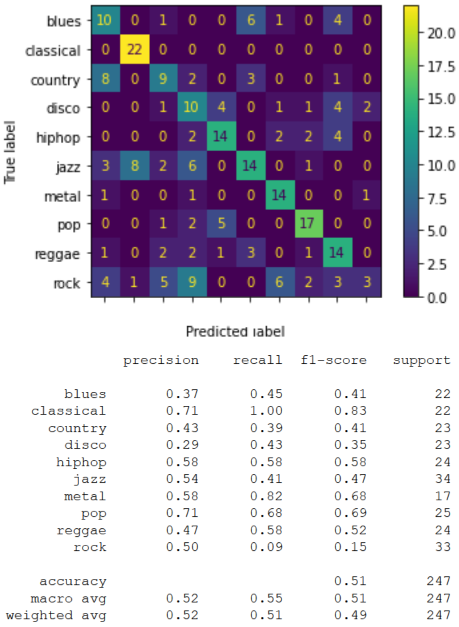
At this report, we will just use some classic classification methods that have implementation at scikit-learn, trying hyperparameter tuning. The fact that there exist 10 classes, makes the task difficult for trivial approaches such as logistic regression.

Using 25/75 test/train split, Standard and MinMax Scaler to (0,1), for logistic regression and different values of hyperparameter C, we get the following results:



We see that we have an accuracy convergence at 55%.

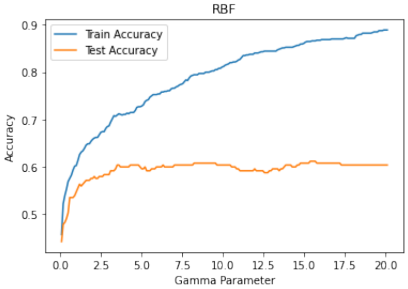
For C=2.5 we get the follow confussion matrix:



blues	10	0	1	0	0	6	1	0	4	0
classical	0	22	0	0	0	0	0	0	0	0
country	8	0	9	2	0	3	0	0	1	0
disco	0	0	1	10	4	0	1	1	4	2
hiphop	0	0	0	2	14	0	2	2	4	0
jazz	3	8	2	6	0	14	0	1	0	0
metal	1	0	0	1	0	0	14	0	0	1
pop	0	0	1	2	5	0	0	17	0	0
reggae	1	0	2	2	1	3	0	1	14	0
rock	4	1	5	9	0	0	6	2	3	3

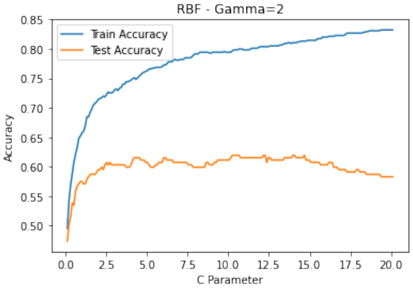
```
              precision    recall  f1-score   support

       blues       0.37      0.45      0.41        22
   classical       0.71      1.00      0.83        22
     country       0.43      0.39      0.41        23
       disco       0.29      0.43      0.35        23
      hiphop       0.58      0.58      0.58        24
        jazz       0.54      0.41      0.47        34
       metal       0.58      0.82      0.68        17
         pop       0.71      0.68      0.69        25
      reggae       0.47      0.58      0.52        24
        rock       0.50      0.09      0.15        33

    accuracy                           0.51       247
   macro avg       0.52      0.55      0.51       247
weighted avg       0.52      0.51      0.49       247
```

We get the best results for classical music, followed by metal and pop. The worst classification is spotted for rock genre. Next we search for different values of gamma parameter, using RBF kernel:



We see that from a specific value of gamma and above, our model ovefitts.

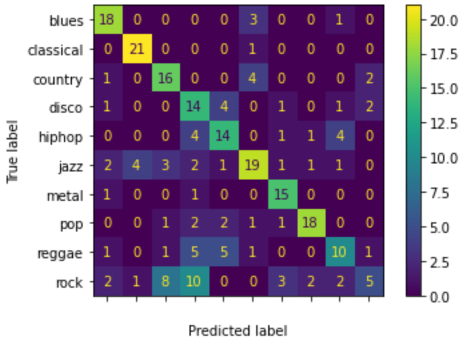
Holding gamma=2 stable and searching for different values of C, we get the below results:



We could do a grid search demonstrating our results on a heat map, but optimization is beyond the scope of these notes.

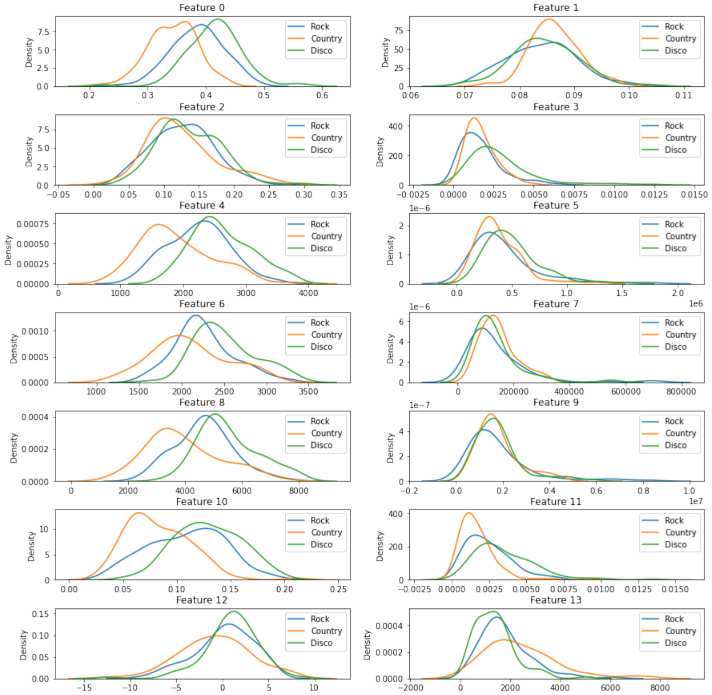
Sigmoid kernel has been tried in the context of SVM classification, but no significant results were shown.

Finally, below we demonstrate the results we got from RBF with C=5 and Gamma=2:

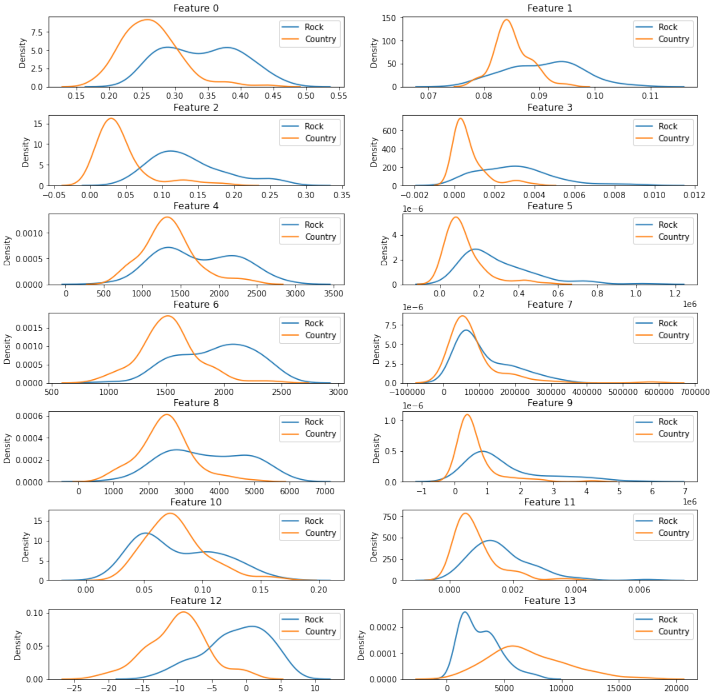


```
              precision    recall  f1-score   support

       blues       0.69      0.82      0.75        22
   classical       0.81      0.95      0.88        22
     country       0.55      0.70      0.62        23
       disco       0.37      0.61      0.46        23
      hiphop       0.54      0.58      0.56        24
        jazz       0.66      0.56      0.60        34
       metal       0.68      0.88      0.77        17
         pop       0.82      0.72      0.77        25
      reggae       0.53      0.42      0.47        24
        rock       0.50      0.15      0.23        33

    accuracy                           0.61       247
   macro avg       0.61      0.64      0.61       247
weighted avg       0.61      0.61      0.59       247
```

We see that accuracy improved by 10%.
At both of the cases we plotted the confussion matrix, we found missclassificasion of rock music that is classified as country and disco. This might indicate that rock has common feature characteristics with these two genres.
Comparing the densities of these three genres, we prove our hypothesis:



as densities seems to overlap.
To have another point of comparison, we demonstarte the densities of blues and classical genres:



For many features we obtain notice almost linear separation.

Future Work

We presented some features that can be extracted from audio files and can be use for machine learning tasks.
At the notebooks that accompany these notes, we use Python and specifically modules librosa and scikit-learn to extract the features described and make classifiers to predict music genres.
Having 10 classes to be predicted, makes the task of classifiaction difficult (as expeted).
We used PCA transfomation after Standard and MinMax Scalers. We didn't conducted feature selection, that could have helped us. At future work, feature selection/extraction methods such as Stepwise Forward Selection, LDA or more extravaganza methods such as autoencoders could be used.
We could also conduct the same task, using spectograms in the context of image classification.