Music Genre Classification

ELHAM EMAMI

Winter 2020

Contents

1. Introduction

Music is the most popular art form that is performed and listened to by billions of people every day. There are many genres of music such as pop, classical, jazz, folk etc.

A music genre describes a style of music that has similar characteristics shared by its members and can be distinguished from other types of music. These characteristics are usually related to the instrumentation, rhythm, harmony, and melody of the music. we address a specific multiclass problem that is music genre classification, of which the goal is to classify songs to their correct genre. Generally, the genre classification process of music has two main steps: feature extraction and classification. The first step obtains audio signal information, while the second one classifies the music into various genres according to extracted features.

Our client is one of the digital music services that give access to users millions of songs and other related author's content. We would like to help our client by analyzing music data to classify them based on genre and that would be the first step toward the recommendation system music recognition. This system can help our clients users to listen to the more available songs close to their musical taste.

2. Data Overview

We use the GTZAN dataset which has been the most widely used in the music genre classification task. The dataset contains 30-second audio files including 10 different genres including reggae, classical, country, jazz, metal, pop, disco, hip-hop, rock, and blues.

3. Data Wrangling

3.1 Feature extraction

Every audio signal consists of many features. In order to extract the song's signal features, we can us the Librosa that is a Python package for audio and music analysis. It provides the building blocks necessary to create music information retrieval systems. The extracted features can display in the plots for better demonstration and comparison as well. We have extracted follow features for our project:

- **Zero Crossing Rate**: The zero crossing rate of any signal frame is the rate at which a signal changes its sign during the frame. It denotes the number of times the signal changes value, from positive to negative and vice versa, divided by the total length of the frame.

- **Spectral Centroid**: The spectral centroid is a measure used in digital signal processing to characterize a spectrum. It indicates where the center of mass of the spectrum is located and is calculated as the weighted mean of the frequencies present in the sound. Perceptually, it has a robust connection with the impression of the brightness of a sound.
- **Spectral Rolloff**: It is a measure of the shape of the signal. It represents the frequency below which a specified percentage of the total spectral energy.
- **Mel-Frequency Cepstral Coefficients**: The Mel frequency cepstral coefficients (MFCCs) of a signal are a small set of features (usually about 10–20) which describe the overall shape of a spectral envelope.
- **Chroma Frequencies** is the representation for music audio in which the entire spectrum is projected onto 12 bins representing the 12 distinct semitones (or chroma) of the musical octave.

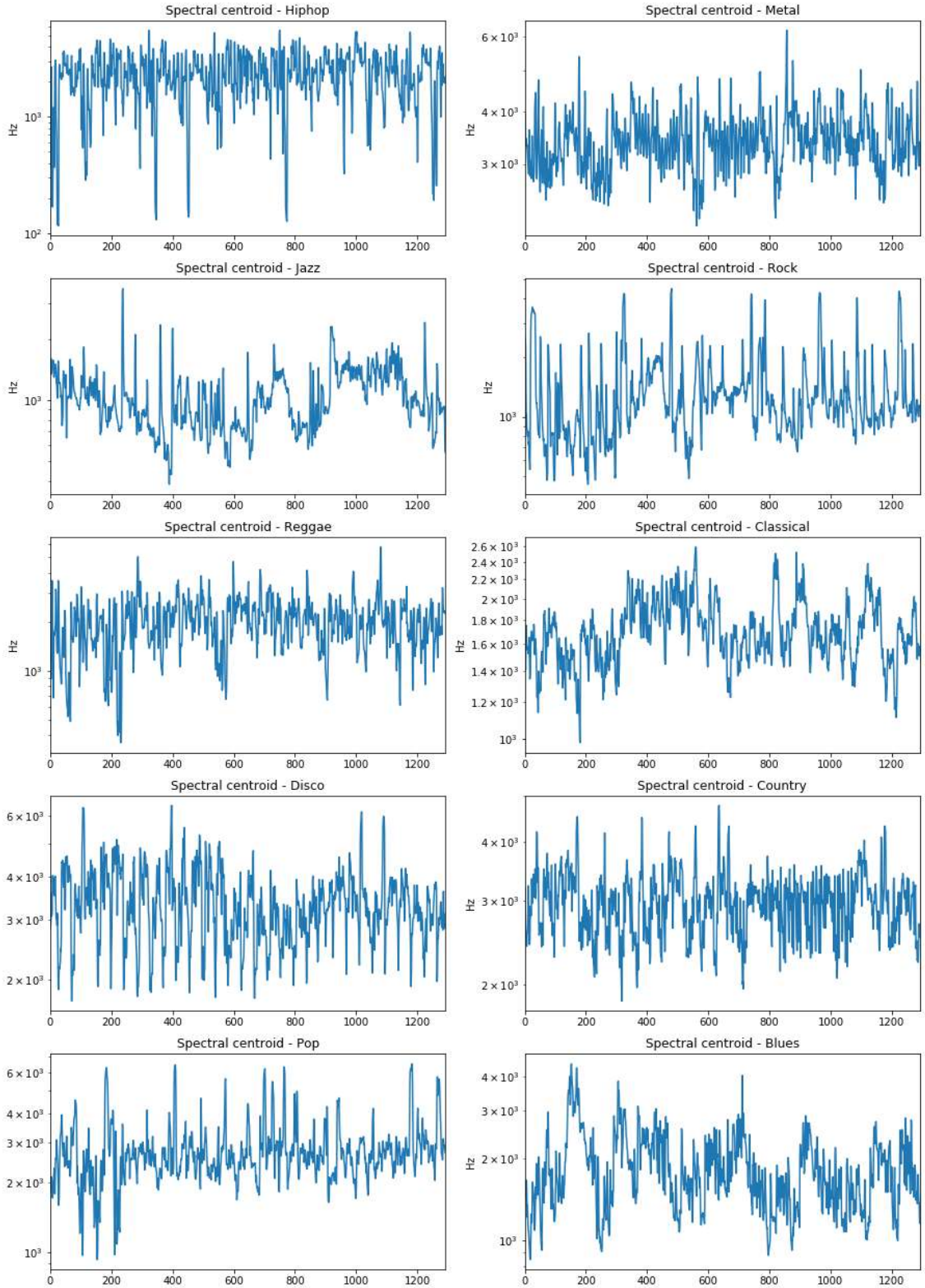


Fig 1 – Spectral centroid for each genre

The above figure displayed the spectral centroid for a song of each genre. As mentioned earlier, the centroid is calculated as the weighted mean of the frequencies present in the sound. As we compare the centroid for the blues and metal genre, we can imply that the spectral centroid lies somewhere in the middle of the blues song while that for the metal song would be toward the end of it. According to the above figure, spectral centroid for the jazz music is close to the start of the song while the pattern for the pop and disco songs are almost in the middle of the song.

The following bar graph demonstrates the mean value of spectral centroid for each genre songs, the bar graph shows the pop and metal has the highest average value of the spectral centroid among the other genres.

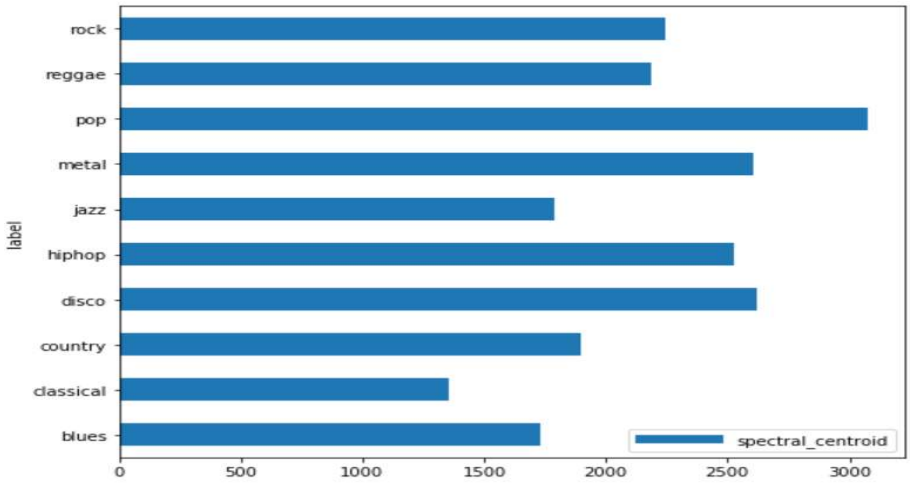


Fig 2 – Mean spectral centroid for each genre

Chroma features are an interesting and powerful representation for music audio in which the entire spectrum is projected onto 12 bins representing the 12 distinct semitones (or chroma) of the musical octave. The idea behind that is that octave plays very important role in music composition and perception. The chroma features of the songs have been displayed in figure 4. There is similarity between the classical and jazz music in term of chroma feature and they follow the linear pattern.

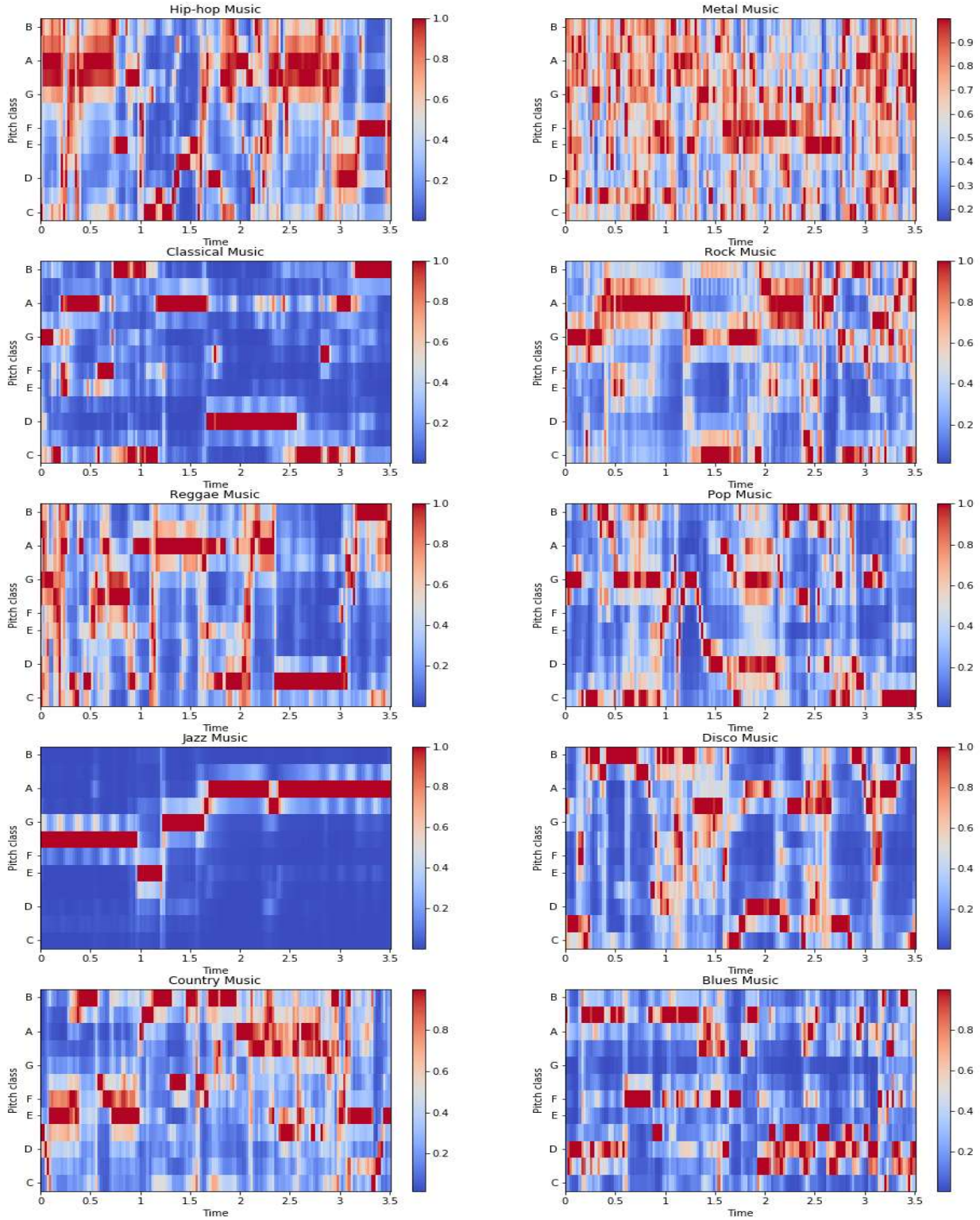


Fig 3 - Chroma stft for each genre

The figure 4 shows mel frequency 1..20 has been displayed as following, Each audio file was converted into a spectrogram which is a visual representation of the spectrum of frequencies over time. A regular spectrogram is a squared magnitude of the short-term Fourier transform (STFT) of the audio signal. This regular spectrogram is squashed using mel scale to convert the audio frequencies into something a human is more able to understand.

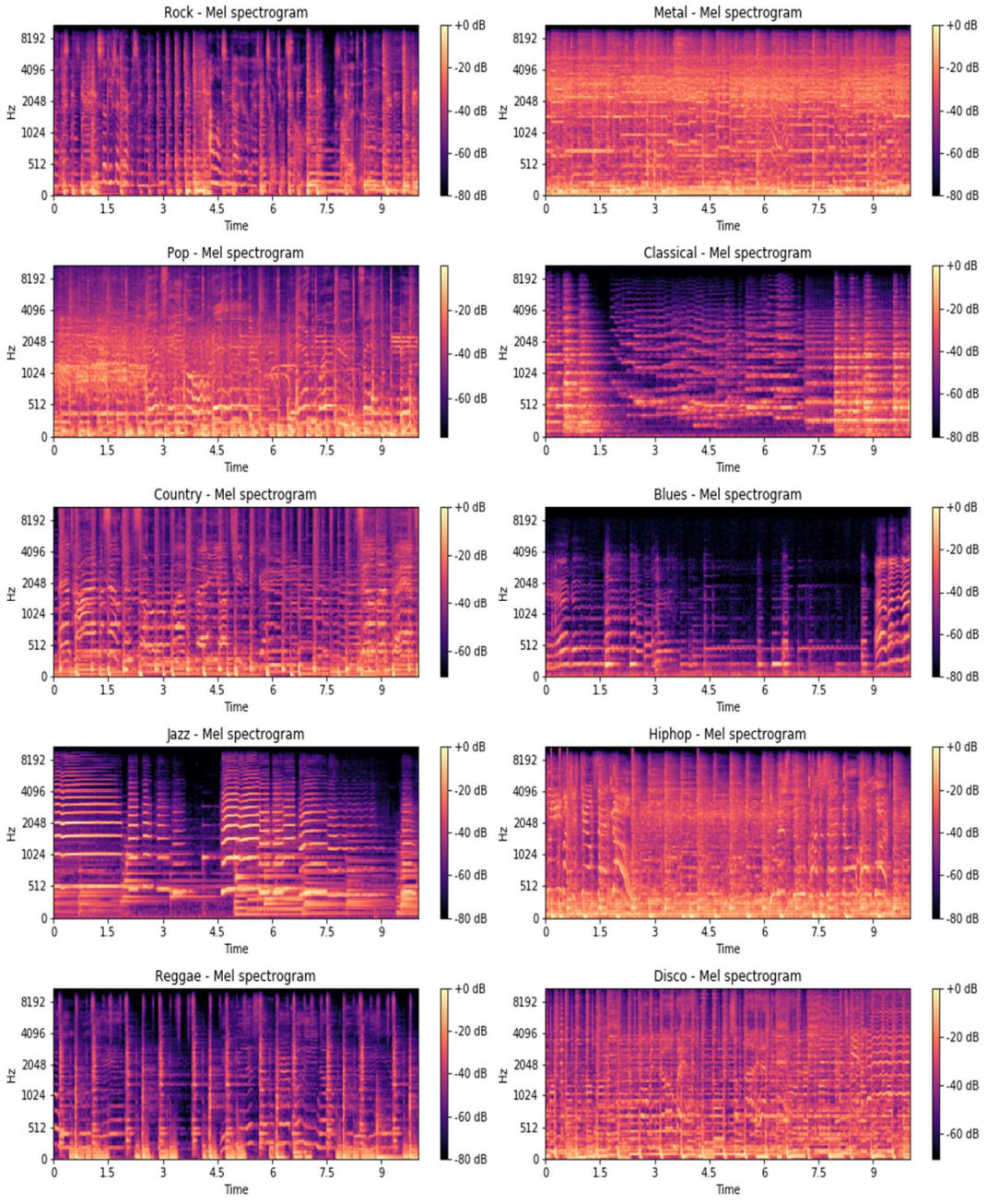


Fig 4 - Mel-Frequency Cepstral Coefficients for each genre

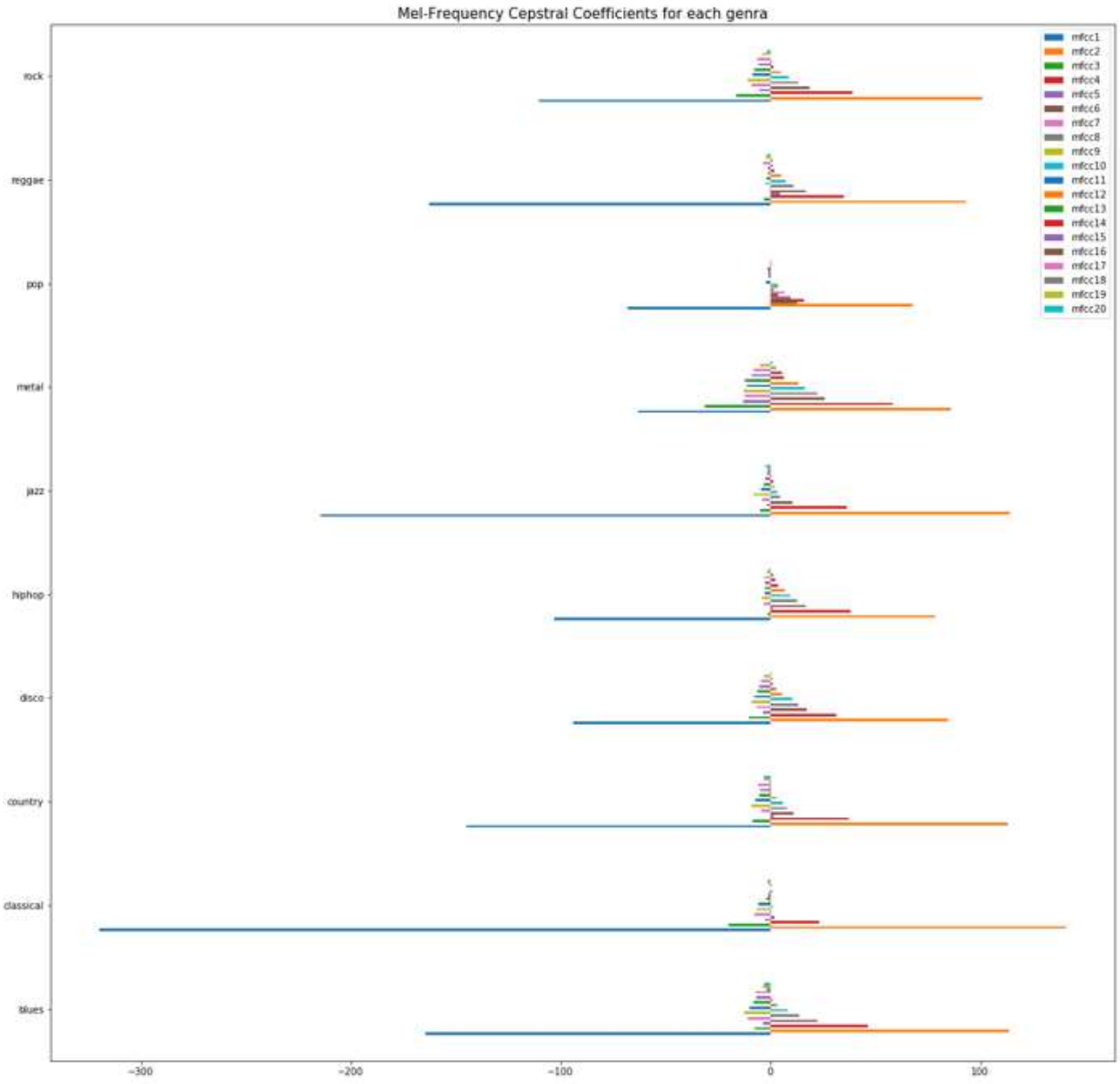


Fig 5 - Mel-Frequency Cepstral Coefficients for each genre

In the above figure, we can see the mel frequency 1..20 for each genre. Classical music has a very extensive difference between the mel frequency features and pop music can be known as dose have least difference of mel- spectrogram feature.

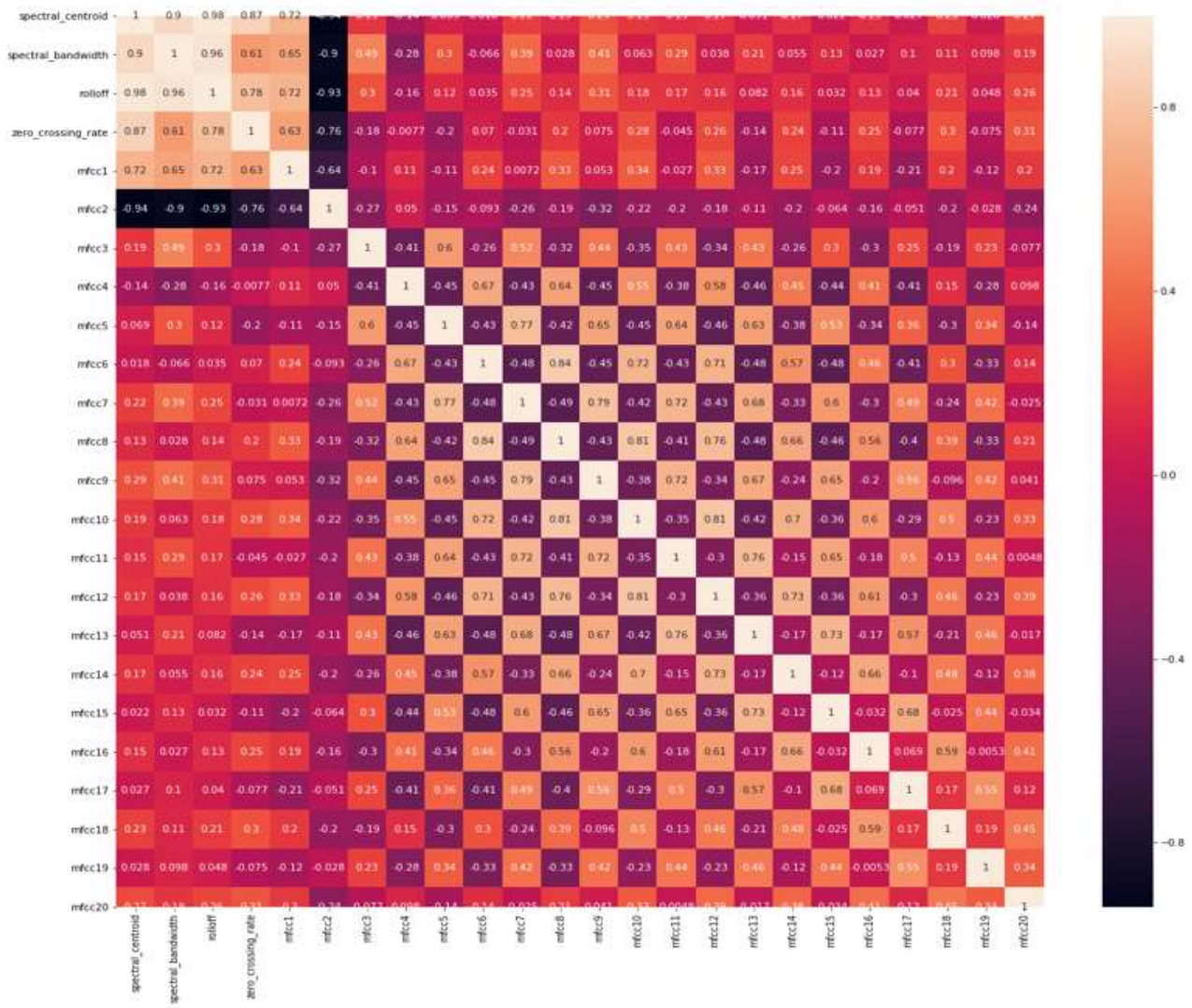
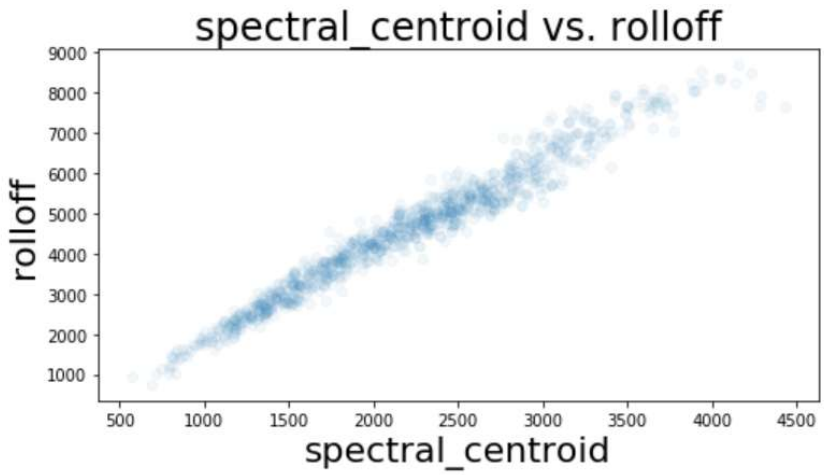


Fig 6 – Correlation matrix

The correlation matrix has been created for the extracted features of music songs. According to the confusion matrix, there are an approximately strong correlation between the mel frequency cepstral coefficients1 and Spectral Centroid, spectral_bandwidth, roll off and zero_crossing_rate. The is correlation decrease with other mel frequency cepstral coefficients. There is very strong correlation between spectra centroid and spectral bandwidth that spectral centroid indicates at which frequency the energy of a spectrum is centered upon.

4. Inferential Statistics

The first test is to consider there is a significant correlation between ‘spectral_centroid’ and ‘rolloff’. The following plot is showing the value of ‘spectral_centroid’ and ‘rolloff’ correlation:



The correlation of these two features are as:

	Spectral centroid	rolloff
Spectral centroid	1	0.979633
rolloff	0.979633	1

The null hypothesis is that ‘Spectral centroid’ and ‘rolloff’ are correlated and alternative hypothesis is they are not correlated. According to the p-value of following table, we cannot reject our null hypothesis which states, ‘Spectral centroid’ and ‘rolloff’ are correlated.

Mean of ‘Spectral centroid’ and ‘rolloff’ correlation	0.9796333818926216
Standard Error	0.00636
z-statistics value	154.1264
p-value using z-statistics	2
Margin of Error	0.00072

Confidence interval is between 0.97891 and 0.98035

The second hypothesis is to test the mean number of zero_crossing_rate is 0.1008, so we will tak e this statement as the null hypothesis. We have calculated the T-test are (statistic=2.1446900710 485095, pvalue=0.03221829991630641). Since the p-value is less than the alpha 0.05, we can re ject the null hypothesis and average value zero crossing rate is not 0.1008.

5. Classification

We have extracted the features in the previous steps. In order to classify the songs, we can use the extracted features or use spectrogram images directly for classification. We have tested both ways in our projects. We applied a convolutional neural network on the spectrogram images and on extracted features.

5.1 Convolutional Neural Network

A Convolutional Neural Network (CNN) is comprised of one or more convolutional layers (often with a subsampling step) and then followed by one or more fully connected layers as in a standard multilayer neural network. The architecture of a CNN is designed to take advantage of the 2D structure of an input image (or other 2D input such as a speech signal). This is achieved with local connections and tied weights followed by some form of pooling which results in translation-invariant features.

A CNN consists of several convolutional and subsampling layers optionally followed by fully connected layers. The input to a convolutional layer is a m x m x r image where mm is the height and width of the image, we have set the height and weight 128 and r is the number of channels, r is set to 3 for RGB image.

We use the Keras library to build the CNN model, Every Keras model is either built using the Sequential class, which represents a linear stack of layers, or the functional Model class, which is more customizable. We have initiated the sequential class in our model. The Sequential constructor takes an array of Keras Layers. We'll use different types of layers for our CNN: Convolutional, Max Pooling, Dropout to prevent the overfitting and Softmax. We do so on

Conv2D. Once this input shape is specified, Keras will automatically infer the shapes of inputs for later layers. The output Softmax layer has 10 nodes, one for each class. We have applied with a pretty good default Adam optimizer. Since we used a Softmax output layer, we had used the Cross-Entropy loss. We have 10 different classes to be distinguished so we had used categorical_crossentropy as a loss function. This is a classification problem, then we just defined the accuracy as the metric of the Keras report. The number of epochs which is iterations over the entire dataset to train for is set up 50 times in our model. The image dataset has been split into 75% of training data and 25% of test data. We have not received very good accuracy which is 10% and is more a random guess with training CNN model over the images and one of the main reasons is the data amount. If we have trained with more data, it can be helpful to improve the accuracy of the model.

We have trained our model over the extracted features as well, this time we have set the epoch size 30 times and our model accuracy is 91%.

6. Conclusion and Future Work

My experience in our project is the deep learning models using CNN can perform
as well as the baseline model. This proves that deep learning can itself extract useful features from raw mel-spectograms. Furthermore, while the models have low accuracy, there might be a result of an insufficient sample. Even 1000 spectrograms per genre maybe a very small sample here since we are training these models from scratch. A bigger data set should improve results. For future work, I would like to try the following:

- Further, improve the performance of these models by trying different convolutional and recurrent architectures and with the bigger size of the dataset.
- Currently, I am converting the full 5 seconds of a song to a single spectrogram is possible that the full 30 seconds make a better improvement determination of genre