
# Audio Pattern Recognition: Musical Genre Classification Using Different Classifiers

Rishav Mondal
Matriculation: 963810

***Abstract***— In this project we build a system using complex operations and various features available to automatically classify music genres. This project is basically divided into two parts. The first part is to extract the various audio features and also visualize the feature space suing K-Means clustering, and, second part is to compare the various classifiers and their performances

## I. INTRODUCTION

Classification as a process has been, since the ancient times, a very important method used by the people to organize and arrange various entities and objects of knowledge to make it easier for future use. Over time the time the process of Classification has gone through various improvements and update, so as to make it easier for people to handle the task of Classification easier, especially since the emergence of Big Data, as Classifying these data using the old approach is nearly impossible considering the size of the task at hand.

Music has been for ages has been a primary source of entertainment. **Genre** of a music can be defined as a category or rather conventional category that recognizes the characteristics or traits of sub-division of the music file belonging to a traditional or any conventional established music form. Categorizing and Classifying music with respect to their **genres** has always been a challenge

The term Music Genre Classification can be explained as **categorizing of music samples**. In this technique we can apply the method of Classification on the musical fields to distinguish between the various available music to according to the genre they belong to. A musical genre is represented by a label created and used by humans to categorize and describe the large universe of music But since there's not a clear distinction between the music of various genres, the job of classifying them according to the genres is a bit complex and hence tough.

In this project we'll create a system to automatically classify the music in a particular dataset according to their genres. This will be accomplished in two parts:

1. In the first part we will extract the various features of the music and visualize them individually, then organize the dataset and visualize the feature space using **K-Means** clustering and considering the music of different genres will essentially have different feature values, hence we can more or less use the feature space to form an idea of how the music files of each genres are distributed.

2. In the second part, we will use four classifiers to automatically classify the music files according to their genres and test the performance of each classifier and compare them with each other. THE Classifiers used are K-Nearest Neighbor, Support Vector Machine, Logistic Regression and Multi-Layer Perceptron.

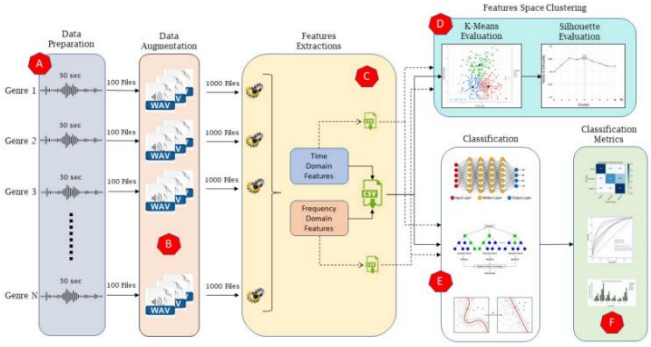
Systems or algorithms like these have been in recent years by music streaming companies like Spotify, Deezer and SoundCloud to properly maintain their database and aid in streaming, purchasing or searching of a particular music in their huge database.

## II. SYSTEM ARCHITECTURE OVERVIEW

The architecture of the implemented system is basically divided into 6 blocks. In the first two we have sample data preparation and data augmentation. The next part of the system is in the third step or the extraction of the features and their choice which is the most essential part of the entire system. At the fourth and fifth step we have the clustering of the features and the classification through various approaches respectively. And at the last step we have the evaluation of the obtained results.

### *I. Data Preparation*

At first and foremost, to create a system for audio genre classification, we need an equally well-equipped audio dataset that would help us in the creation and training of the system. The dataset used here is the GTZAN dataset. The dataset comes with the audio files and two CSV files which contains the extracted features of the audio files. One for the entire audio duration of 30 sec and the other for audio files spilt into 3 sec.




### *II. Data Augmentation*

There are a lot of techniques for audio data augmentations, but in this project, we used one of the simplest one, that is split data in more sub samples for increase the number of data that. Starting from 999 samples of 30 seconds and reaching 9990 samples of 3 seconds.

### *III. Features Extraction*

Feature extraction is the process of computing a compact numerical representation that can be used to characterize a segment of audio. The design of descriptive features for a specific application is the main challenge in building pattern recognition systems. Once the features are extracted standard machine learning techniques which are independent of the specific application area can be used.

### *IV. Features Space Clustering*

In this architecture block, after extracting the characteristics from the audio samples we applied an unsupervised learning technique known as the **K-Means** Clustering to cluster and visualize the feature space. It too, as in any classification

problem, tries to learn in order to associate a label **y** of the set **Y** with a given point **x** of the input space **X**, in our case the correct musical genre. The substantial difference between unsupervised and supervised learning techniques is that of automatically extracting knowledge from the input data. This knowledge is extracted without specific knowledge of the content to be analyzed. The aim of the **K-means** algorithm is to minimize the total inter-cluster variance and to maximize the intra-cluster variance. We first choose **K** initial centroids, where **K** is a user specified hyper parameter, which represents the number of desired clusters. Each point is then assigned to the closest centroid on the basis of a predetermined measure of similarity or distance, and each collection of points assigned to a centroid is a cluster. The centroid of each cluster is then updated for each iteration based on the points assigned to the cluster, until the algorithm converges.

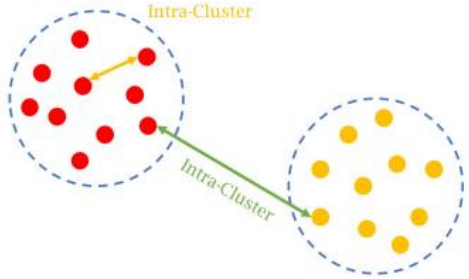



Fig. 2: Schematic example of intra-cluster and inter-cluster distance.

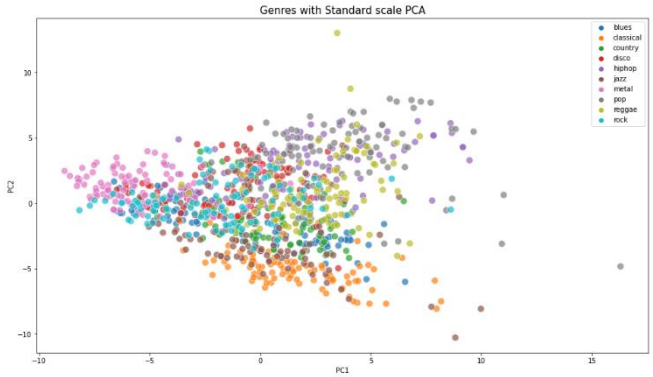



Fig. 3: K-Means Clusters using Standard scale of the dataset.

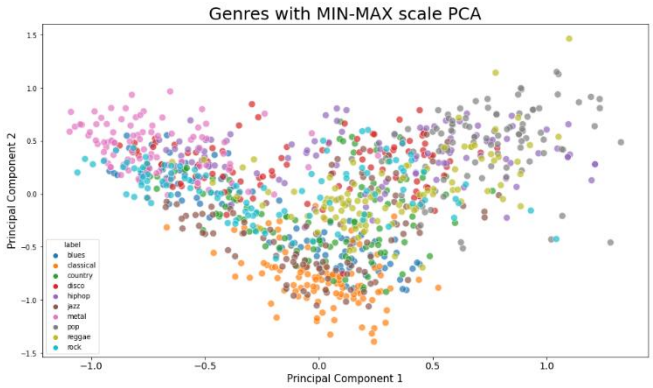



Fig. 4: K-Means Clusters using Min-Max scale of the dataset.

### *V. Classification*

The fifth block is dedicated to the classification of the characteristics extracted with the aid of a different supervised classification methods. The ones we have decided to use to be able to compare them are three, we'll see the main peculiarities of each one in the theoretical background section. The classifiers used are:

1. Support Vector machine (**SVM**)
2. K-Nearest Neighbor (**K-NN**)
3. Logistic Regression (**LR**)
4. Multi-Layer Perceptron (**MLP**)

### *VI. Classification Metrics*

In this section we analyze how to evaluate the goodness of a model or several classification models. The methods used are, the Accuracy measures and the confusion matrix. The Detailed explanation in the section Theoretical background.

## III. THEORITICAL BACKGROUND

### *I. Features Extraction*

This process as explained above briefly is the process of computing a compact numerical representation that can be used to characterize a segment of audio. The features extracted in this process of is divided into two parts:

1) **Time Domain Features:**

a) a) **Tempo**: Particularly useful features to extract from musical sources may be an approximation of tempo as well as the beat onset indices, an array of frame numbers corresponding to beat events

b) **Energy**: This feature provides the so-called power of the signal. Let $s_i(k)$, where $k = 1 \ldots W_L$ be the sequence of audio samples of the ith frame, where $W_L$ is the length of the frame. The short-term energy is computed according to the equation:

$$E(i) = \frac{1}{W_L} \cdot \sum_{n=1}^{W_L} s_i(n)^2 \cdot \qquad (1)$$

c) **Energy Entropy**: The entropy of energy can be interpreted as a measure of abrupt changes in the energy level of an audio signal. In order to compute it, we first divide each short-term frame in $K$ sub-frames of fixed duration. Then, for each sub-frame, $j$, we compute its energy as in Eq.1 and divide it by the total energy. At a final step, the entropy, $H_i$ of the sub-frames sequence is computed according to the equation:

$$H(i) = -\sum_{j=1}^{W_L} e_j \cdot \log_2(e_j) \cdot \qquad (2)$$

d) **Root Mean Square Energy (RMSE)**: Root-Mean-Square Energy is typically denoted RMS energy, is another time domain feature, computed as shown in Eq.3 where $W_L$ represents frame size, i.e., the number of samples in each frame and $s_i(k)$ the signal. As it relates to perceived sound intensity, RMS energy can be used for loudness estimation and

as an indicator for new events in audio segmentation.

$$RMSE = \sqrt{\frac{1}{W_L} \cdot \sum_{n=1}^{W_L} s_i(n)^2 \cdot} \quad (3)$$

e) **Zero-Crossing Rate (ZCR)**: Measures the number of times the amplitude value changes it's sign. In other words, the number of times the signal changes value, from positive to negative and vice versa, divided by the length of the frame, according to the equation:

$$Z(i) = \frac{1}{2W_L} \cdot \sum_{n=1}^{W_L} |sgn[s_i(n)] - sgn[s_i(n-1)]| \cdot \quad (4)$$

2) **Frequency Domain Features:**

a) **Spectral Centroid**: The spectral centroid is a measure to characterize the "center of mass" of a given spectrum. Perceptually, it has a robust connection with the impression of sound brightness, or rather as an indication of the amount of high-frequency content in a sound, obtained according to the following equation, where $m_t(n)$ represents number of frequency bins, i.e., the number of the highest frequency band.

$$SC_i = \frac{\sum_{k=1}^{N} m_t(n) \cdot n}{\sum_{n=1}^{N} m_t(n)} \quad (5)$$

b) **Spectral Bandwidth**: Spectral bandwidth is derived from the spectral centroid and indicates the spectral range of the interesting parts in the signal, i.e., the parts around the centroid. It can be interpreted as variance from the mean frequency in the signal. The definition is given in Eq.6.The average bandwidth of a music piece may serve to describe its perceived timbre.

$$BW_i = \frac{\sum_{n=1}^{N} |n - SC_i| \cdot m_t(n)}{\sum_{n=1}^{N} m_t(n)} \quad (6)$$

c) **Spectral Contrast**: Four each frame of a spectrogram s is divided into sub-bands. For each sub-band, the energy contrast is estimated by comparing the mean energy in the top quantile (peak energy) to that of the bottom quantile (valley energy). High contrast values generally correspond to clear, narrow-band signals, while low contrast values correspond to broad-band noise.Ads

d) **Spectral Rolloff**: This feature is defined as the frequency below which a certain percentage (usually around 85-90%) of the magnitude distribution of the spectrum is concentrated. Therefore, if the $m_{th}$ DFT coefficient corresponds to the spectral rolloff of the $i_{th}$ frame, then it satisfies the following Eq. 7, where C is the adopted percentage (user parameter). The spectral rolloff frequency is usually normalized by dividing it with N, so that it takes values between 0 and 1.

$$\sum_{n=1}^{m} m_t(n) = C \cdot \sum_{n=1}^{N} m_t(n) \quad (7)$$

e) **Mel Frequency Cepstral Coefficient**: Mel frequency cepstral coefficients (MFCCs) have their origin in speech processing but were also found to be suited to model timbre in music. The MFCC feature is calculated in the frequency domain, derived from the signal's spectrogram and for each frame, cepstral coefficients are computed using Mel-filter bank with a variable numbers of Mel filters. In music signal processing, between 13 and 25 MFCCs are typically computed for each frame.

f) **Chromagram**: In music, the term chromagram is attentive to the twelve different pitch classes. This vector of features is computed by grouping the DFT coefficients of a short-term window into 12 bins. Each bin represents one of the 12 equal tempered pitch classes of Western-type music. Each bin produces the mean of log-magnitudes of the respective DFT coefficients. One main characteristic features of color is that they capture the harmonic and melodic features of music, at the same time robust to changes in timbre and instrumentation.

### II. *Features Space Clustering*

As stated in the previous section, Feature Space Clustering is the process in which certain algorithms are used to visualize the feature space of the dataset. In this project the algorithm used was **K-Means Clustering.** The Algorithm for K-Means Clustering is as follows:

**Algorithm 1:** K-means

[1] **Input:** Set N patterns $x_i$, desired number of clusters K
[2] **Output:** Set of K clusters
[3] *Choose K patterns as the $c_j$ centers of clusters;*
[4] **repeat**
[5] *Assign each pattern $x_i$ to the cluster $P_i$ that has the center $c_j$ closest to $x_i$ (that is, the one that minimizes d ($x_i$, $c_j$));*
[6] *Recalculate the $c_j$ centers of the clusters using the patterns that belong to each cluster (mean or centroid);*
**until** *Convergence criterion are satisfied;*
[7] **return**

## IV. The Analysis

Now that the dataset has been prepared and ready to use, all that is left is for us to make use of a few classifiers to predict the genres of the music from the dataset, and to compare the performances of these classifiers.

### I. *Classification*

As mentioned in the previous section, in this block we make use of multiple classifiers, in this case four, for the classification purpose and use the data received to compare their performances. In this project I have use four classifiers:

1) **Support Vector Machine(SVM)**: Support vector machines have shown superb performance at binary

classification tasks and handle large dimensional feature vectors better than other classification methods. Basically, a Support Vector Machine aims at searching for a hyper-plane that separates the positive data points and the negative data points with maximum margin. To extend SVM for multi classification there are two approaches called:

- **One-Vs-Rest (ovr)**: Is a heuristic method for using binary classification algorithms for multi-class classification. It involves splitting the multi-class data set into multiple binary classification problems. This approach requires that each model predicts a class membership probability or a probability-like score. The max of these scores is then used to predict a class.
- **One-Vs-One (ovo)**: Like ovr, ovo splits a multi-class classification data set into binary classification problems. Unlike ovr that splits it into one binary data set for each class, the ovo approach splits the data set into one data set for each class versus every other class. This is significantly more data sets, and in turn, models than the ovr strategy described in the previous section. The formula for calculating the number of binary data sets, is as follows: $K \cdot (K - 1)/2$.

2) **K-nearest neighbors (KNN):** is a type of supervised learning algorithm used for both regression and classification. KNN tries to predict the correct class for the test data by calculating the distance between the test data and all the training points. Then select the K number of points which is closet to the test data. The KNN algorithm calculates the probability of the test data belonging to the classes of 'K' training data and class holds the highest probability will be selected. In the case of regression, the value is the mean of the 'K' selected training points.

3) **Logistic Regression(LR)**: It estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure. Hence logistic regression is used to estimate the relationship between a dependent variable and one or more independent variables, but it is used to make a prediction about a categorical variable versus a continuous one. A categorical variable can be true or false, yes or no, 1 or 0, et cetera. The unit of measure also differs from linear regression as it produces a probability, but the logit function transforms the S-curve into straight line.

4) **Multi-Layer Perceptron(MLP)**: A MLP is a variant of the original Perceptron model proposed by Rosenblatt in the 1950 . It has one or more hidden layers between its input and output layers, the neurons are organized in layers, the connections are always directed from lower layers to upper layers, the neurons in the same layer are not interconnected. The neurons number in the input layer equal to the number of measurement for the pattern problem and the neurons number in the output layer equal to the number of class, for the choice of layers number and neurons in each layers and connections called architecture problem, our main objectives is to optimize it for suitable network with sufficient parameters and good generalisation for classification or regression task.

### *II. Classification metrics*

As mentioned previously, in this section we compare the performances of the various classifiers used in the previous block for classification. The metrics used in this section are the:

1) **Accuracy Score:** Accuracy is a metric used in classification problems used to tell the percentage of accurate predictions. We calculate it by dividing the number of correct predictions by the total number of predictions. In a multiclass problem, we can use the same general definition as with the binary one. However, because we cannot rely on True/False binary definitions, we need to express it in a more general form. We calculate accuracy by dividing the number of correct predictions (the corresponding diagonal in the matrix) by the total number of samples. Multilabel classification problems differ from multiclass ones in that the classes are mutually non exclusive to eachother. In machine learning, we can represent them as multiple binary classification problems.In this project we will using the Multilabel Accuracy Measure.

| CLASSIFIER | ACCURACY SCORE | PARAMETERS |
|---|---|---|
| SVM | 0.8952 | C=500, K = rbf |
| KNN | 0.8899 | NN = 5 |
| LR | 0.7361 | C = 50, M = 5000 |
| MLP | 0.8035 | RS = 5, M = 5000 |

Where,

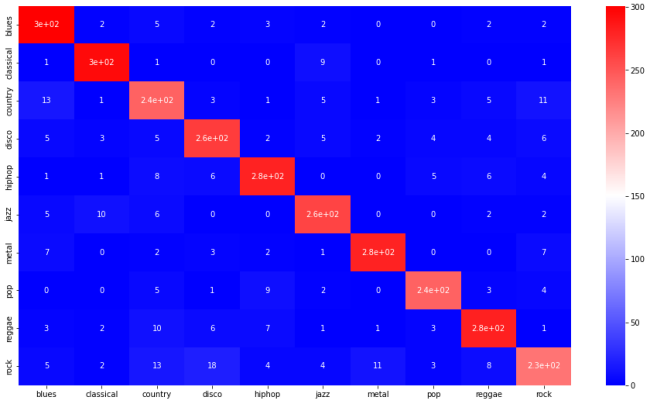
**C** = regularization parameter, strength inversely to its value
**K** = kernel used
**NN** = Nearest Neighbor
**RS** = Random State
**M** = maximum iteration

2) **Confusion Matrix**: A classifier can be described as a function that maps the elements of a set into certain classes or groups. In the case of supervised classification, the set of data to be classified contains a subdivision into classes, with respect to which it's possible to evaluate the quality of the result produced. In a binary classification problem, the set of data to be classified is divided into two classes that we can conventionally indicate as positive (**p**) or negative (**n**). The results of applying a binary classifier fall into one of the following four categories:

1) True Positive (TP).
2) False Positive (FP).
3) True Negative (TN).
4) False Negative (FN)

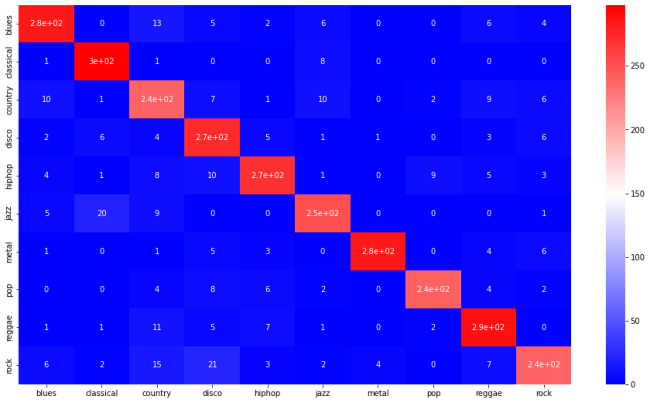
**SVM**

CONFUSION MATRIX FOR SVM



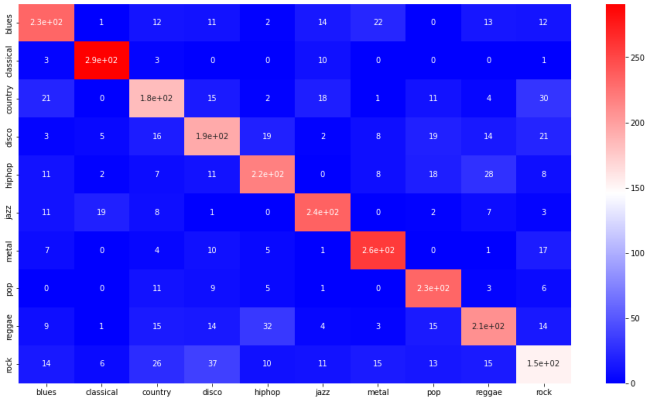

**KNN**

CONFUSION MATRIX FOR KNN



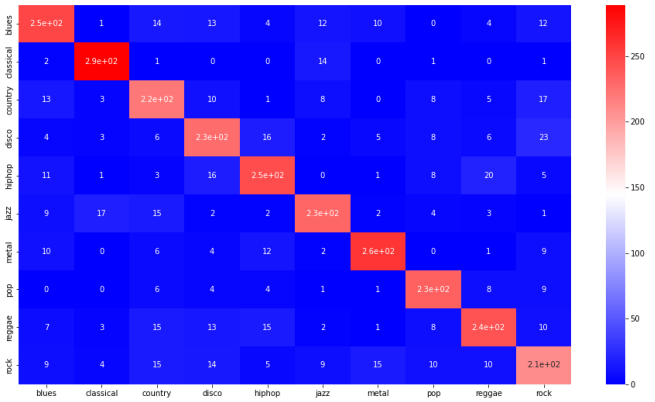

**LR**

CONFUSION MATRIX FOR LR




**MLP**

CONFUSION MATRIX FOR MLP




*III. Miscellaneous*

In this project two separate datasets were also created using only the time domain and frequency domain features respectively. It was done to check the performance of the best classifier on these two separate domain datasets individually. The results were surprising:

1) **Accuracy Score:**

| CLASSIFIER | ACCURACY SCORE |
|---|---|
| SVM_TIME | 0.4933 |
| SVM_FREQUENCY | 0.7867 |

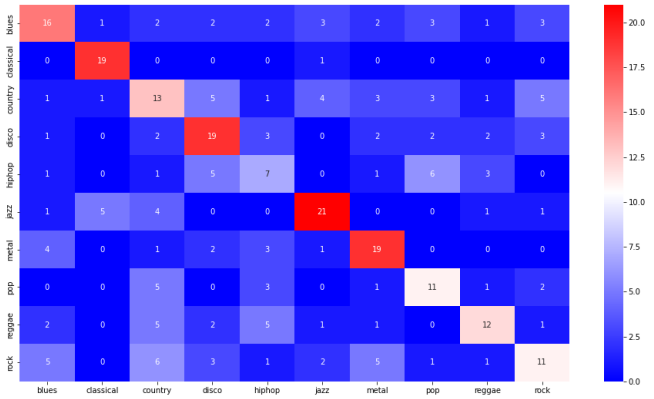
For,

**C** = 1000
**DFS** = ovo
**Kernel** = rbf
**Max_iter** = 5000

2) **Confusion Matrix:**

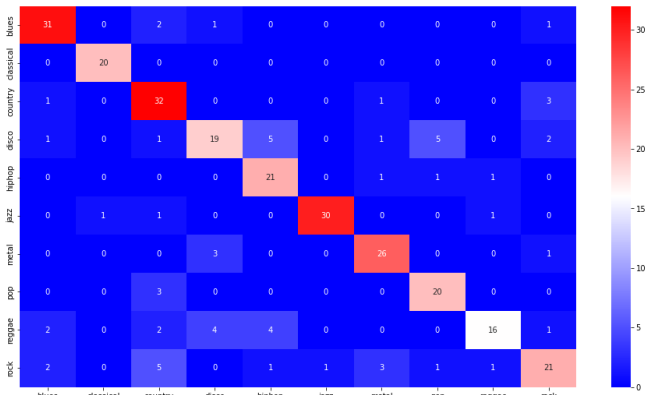
**SVM_TIME**

CONFUSION MATRIX FOR SVM




**SVM_FREQUENCY**

CONFUSION MATRIX FOR SVM



## V. CONCLUSION

In this project we have managed to find out not only about the different types of features but also compared their contribution to the automatic Music Genre Prediction task. We managed to visualize many of the audio features, and their distribution using graphs. We used an unsupervised learning method, that is K-Means clustering, to successfully visualize the feature space. We also we also managed to compare the performance of four classifiers, and we found out that best result does not always come from the most well-known classifier. We also used the data in an unorthodox way, where we separated the Time Domain feature and the frequency domain features to find the contribution of the different features in the prediction of a music genre. We found out that the Frequency domain features has much more effect in the classification of a genre then the Time domain features. This project can be used in the future with more different classifiers and compare their performances. As for the classifiers chose in this project, it turns out Support Vector Machines are the best in this kind of classification problems.

## APPENDIX

Link to the GitHub repository: **GITHUB**