Project Report (Major Project- II)

On

Music Genre Classification using ML

Submitted in partial fulfillment of the requirements

for the award of the degree of

Master of Technology

in

Software Technology

By

Pankaj Malik

Roll No.: - 2K16/SWT/510

Under the guidance of

Dr. Kapil Sharma

Professor (HoD)

Department of Information Technology

<u>Delhi Technological University</u>

(Formerly Delhi College of Engineering)

Bawana Road, Delhi 110042

2019

Delhi Technological University

(Formerly Delhi College of Engineering)

Bawana Road, New Delhi-42

DECLARATION

I hereby declare that the thesis entitled "**Music Genre Classification using ML**" which is being submitted to the Delhi Technological University, in partial fulfillment of the requirements for the award of the degree of Master of Technology in *Software Technology* is an authentic work carried out by me. The material contained in this thesis has not been submitted to any university or institution for the award of any degree.

DATE:

SIGNATURE: Pankaj

PANKAJ MALIK

2K16/SWT/510

CERTIFICATE

Delhi Technological University

(Formerly Delhi College of Engineering)

Bawana Road, New Delhi-42

This is to certify that project report entitled "**Music Genre Classification using ML**"done by me for the Major Project 2 for the award of degree of Master of Technology Degree in Software Technology in the Department of Computer Science & Engineering, Delhi Technological University, New Delhi is an authentic work carried out by me.

Signature:

Student Name

Pankaj Malik Pankaj

2K16/SWT/510

Above Statement given by Student is Correct.

[illegible] 10/07/2020

Project Guide:-

Dr. Kapil Sharma
Professor (HoD)
Department of Information Technology
Delhi Technological University, Delhi

Acknowledgement

No volume of words is enough to express my gratitude towards my guide **Dr. Kapil Sharma**, HoD of Information Technology, Delhi Technological University, Delhi, who has been very concerned and has aided for all the materials essentials for the preparation of this project report. He has helped me to explore this vast topic in an organized manner and provided me all the ideas on how to work towards a research-oriented venture.

I am also thankful to **Dr. Rajni Jindal**, HoD of Computer Science & Engineering Department and **Dr. Ruchika Malhotra**, Coordinator, for the motivation and inspiration that triggered me for the project work.

I would also like to thank the staff members and my colleagues who were always there at the need of hour and provided with all the help and facilities, which I required, for the completion of my project work.

Most importantly, I would like to thank my parents and the almighty for showing me the right direction, to help me stay calm in the oddest of the times and keep moving even at times when there was no hope.

Pankaj

Pankaj Malik

(2K16/SWT/510)

ABSTRACT

Music genre classification is well known problem in Music Information Retrieval systems. It has various applications which start from autonomous tagging of unknown music files to provide best recommendations in several applications. Here in this project, we solved the music genre classification problem using different machine learning algorithms such as k-nearest neighbours (k-NN), Support Vector Machine (SVM), Random Forest and XGboost.For experimental purpose we used GTZAN dataset. In this paper the featureshas been extractedwhich includes MFCC, Chroma,Spectral Centroid, Mel spectrogram and Spectral Contrast from GTZAN dataset. The extracted features are high dimensionaland traditional dimension reduction techniques are inefficient on these features, so we solved the dimensionality reduction problem using statistical operations namely mean, median,maximum, mean absolute deviation, standard error of mean, standard deviation,cumulative sum and cumulative maximum on features. This solution not only reduces the dimensions but also keeps the valuable information intact. In this paper, we have also presented the comparative analysis on different combinations of features, by applying various algorithms of machine learning. After performing various experiments, it was found that SVM with Poly kernel has given the best results. The overall improvement in classification that we got is an average accuracy of 80.25% and maximum accuracy of 91% on all 10 musicgenre.

LIST OF FIGURES

List of Tables

Table of Contents

CHAPTER-01

INTRODUCTION

Music Information Retrieval (MIR)

This is a technology used for extracting the information from music, In this study, our goal is to enhance our understanding about the usefulness of music data, through the research, tools development and application of computational approaches. MIR can be used for categorization, manipulation and even for music creation.

Music can be represented in symbolic format (for example- MIDI file), in audio format (for example- an mp3 file), or in vector format (for example- a scanned score).

In MIR process, we first extract the useful information from a music file, analyze that information and then finally use the collected information in classification of music data. One of the challenges in this task is that we have Music file in various formats. So we need a system that can read the music raw data after discarding the encoding bits. To perform operation on raw data would be easy as the information retrieval becomes easy and accurate.

Machine learning technologies and signal processing knowledge is useful for MIR research.

Conceptual MIR Dimensions

- Stages
    1. Representation/Hearing
    2. Analysis/Learning
    3. Interaction/Action

- Specificity
    1. Genre classification
    2. Mood classification

- Music Recommendation
    1. Playlist Generation
    2. Automatic Tagging

1.1 Music Genre Classification:

A music genre is a traditional category which identifies some pieces of music belongs to a shared tradition or set of conventions.

New genres can be formed by the development of new styles of music and also simply by creating a new categorization. Although it is feasible to generate a musical style without having any relation with an existing genre, new styles generally appear under the influence of pre-existing genres.

1.1.1 Type of Genre:

- Hip hop
- Country
- Dance
- Classical crossover
- R&B
- Folk
- Samba
- Rock
- Pop
- Soul

Classification of Music Genre is a crucial task now a day because it has many applications in this multimedia world. Now a day hundred of songs are being released everyday in different countries and on different websites. So we have huge amount of Music data available for us and that keeps on increasing day by day.

On Internet there are different web-sites which keep various collections of data and provide to user on their demand. So now developer of music database management system requires more accurate meta-data so that search and storage of music in the database can be performed very efficiently and in short amount of time.

Therefore for a streaming service provider classification of song becomes an important task to maintain its play-list or music library which has enormous variety of songs. So to classify the song based on its genre has a significant importance in current time to meet the requirements.

The main aim of our paper is to focus on correct classification of Music genre in minimum time span. We have performed experiments using different machine learning algorithms with 80:20 or 90:10 split of training and testing data.

CHAPTER-02

Literature Review

On automatic music genre classification a lots of promising work has already been done.Dong Myung Kim and Miguel Francisco constructed a system using chroma features andMFCC that had an accuracy of 35.5% on all 10 genres of GTZAN dataset. Michael Haggblade Yang Hong and Kenny Kao made a system that differentiates 4 genres using neural net,SVM, k-means and knn. Kalpesh Patil, Rishabh Raj, Chandrakanth BKC, Prof. Preethi Jyothi have also worked on 4 genres and used Vector Quantization,DNN, GPPS-SVM, GPPS-NN and CNN as the machine learning models. Tzanetakiset al has published its research that used various low-level features to achieve the success rates of 61% while classifying 10 genres.Zain Ahmed Siddiquiand MuhammadAsim Ali used MFCC as their primary feature and tried SVM and k-Nearest Neighbor and the results were great by just using MFCC. Archit Rathore and Margaux Dorido used MFCC, chroma, zero crossing rate,spectral centroid, spectral roll off as their features of choice and performedthe experiment on varieties of classifiers and showed, how poly SVM outclassed others with distinguishing 6 genres with accuracy of 82% but accuracy decreased to 51% while using all10 genres. Nilesh M. Patil and Dr. Milind U.Nemade used Chroma,MFCC, spectralcentroid,spectral roll-off, zero-crossing rate features and achieved maximum classification accuracy of 77.8% with poly SVM and with linear SVM accuracy would be 60%.

We studied various approached and found most of them revolved around extracting the various features of audio file and perform experiments with different machine learning algorithms in order to achieve thehighly possible accuracy. All machine learning approaches that we used are helpful as they showed howclassification accuracies can be impacted by using various combinations of features with different machine learning models.

CHAPTER-03

Proposed Work

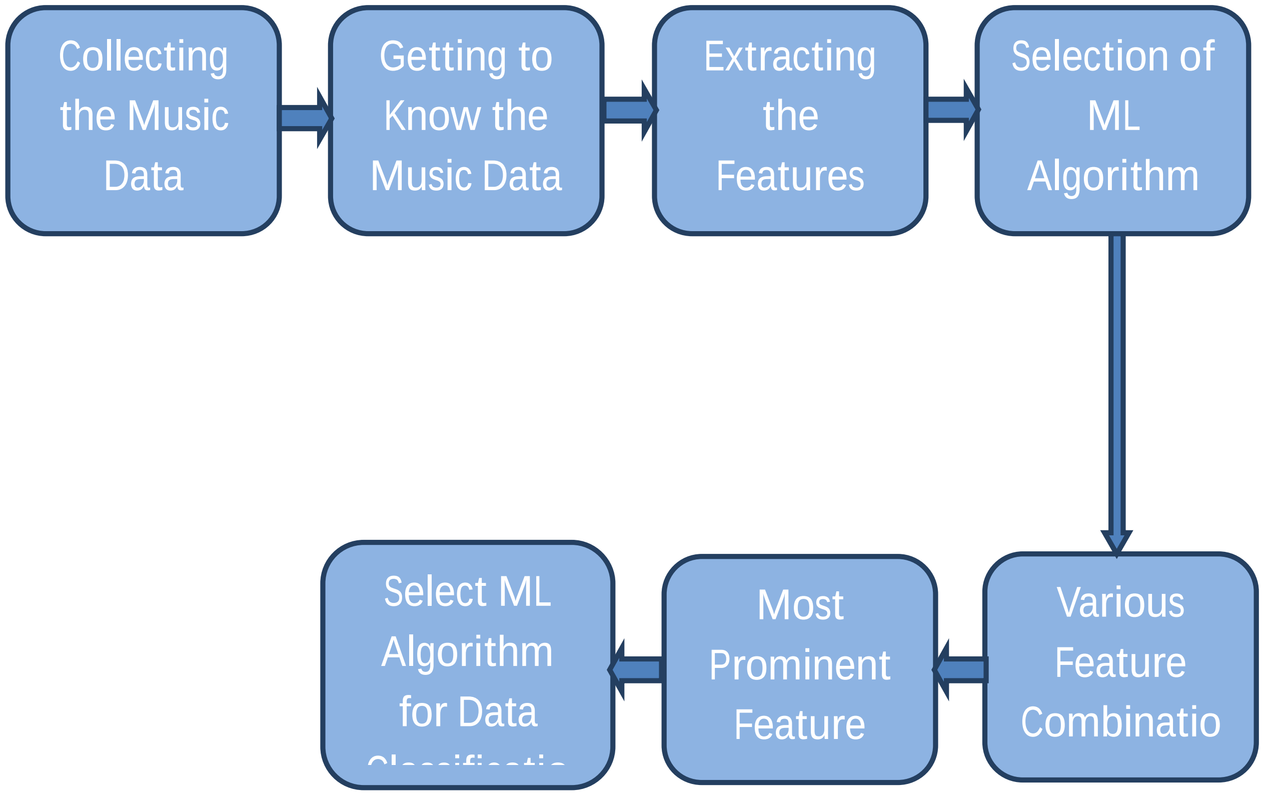

Collecting the Music Data
Getting to Know the Music Data
Extracting the Features
Selection of ML Algorithm
Various Feature Combinatio
Most Prominent Feature
Select ML Algorithm for Data

Figure1. Algorithm for Music Genre Classification

Classification of music is an interesting problem because of its subjective nature but at the same time it has many applications also. The time consuming, expensive and impractical approaches for annotation of each music file is classifying manually or by using semi-automated approach. The music can be classified into different genres like rock,disco, classical, country, pop,jazz etc. Sometimes these genres may be considered arbitrary, but they are very useful in categorizing and organizing large datasets of music files.

Genre generally assumes large weight in music recommender systems. As described in the report lots of researches have already been performed on music genre classification.Standard approach for automatic music genre classification includes feature extraction & using machine learning algorithms.

Extracted features represent the class and nature of audio files which helps in differentiating the music in different genres. Various features can be extracted from a music files. Major research work involves the extracting spectral features like MFCC, Chroma,Spectral bandwidth, zero crossing rate, Spectral Centroid, spectral roll off, Spectral contrast and Mel Spectrogram.The most challenging task in music classification is feature selection therefore music genre classification is still a significant research area. Goal of our research is to classify the music solely based on audio rather than extraneously appended information. So our algorithm is independent of any music encoding techniques by which any raw audio data is encoded.

In our project, we have tried various combinations of different spectral features and various machine learning algorithms such as K-NN which stands for k-nearest neighbor, random Forest, Support Vector Machine (SVM) and XGboost. Extracted features from GTZAN dataset have large dimensions therefore mentioned machine learning algorithms did not performed as expected.Zain Ahmed Siddiqui and Muhammad Asim Ali had tried by decreasing dimensions using PCA algorithm but with this approach music genre classification accuracy also got decreased.

So in this paper, instead of using traditional techniques for dimensionality reduction such as PCA, some other statistical operations namely mean, ,median, standard deviation, standard error of the mean, maximum, mean absolute deviation,cumulative sum and cumulative maximum are used by us. Our main aim is to design a machine learning model that works just like humans, while categorizing the music file based on genre. Completed work on music genre classification has already been described in literature survey. In this paper we will discuss and explains the GTZAN dataset used for experimental purpose. In Chapter 4 we will discuss the audio features extracted from the dataset. Various features has extracted from audio music file such as Mel spectro, chroma, MFCC etc. Chapter 4 contains the detail definition of audio feature and describes the dataset that we used in our project.

Section 5 describes the detailed methodology that is used for the purpose of genre classification. This chapter also contains different machine learning algorithms and intuition behind using them in section 6 contains the results of all experiment performed on GTZAN dataset. Section 7 has conclusion and future work.

CHAPTER-04

4.1 Features and Dataset

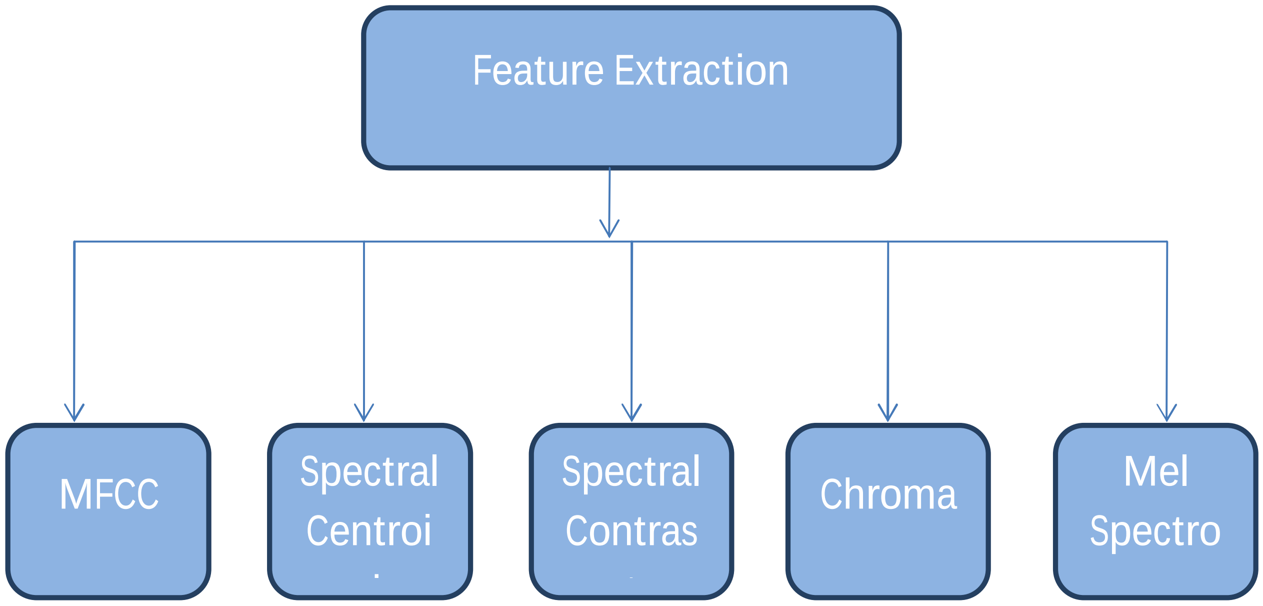


Figure2. Audio Data features

4.1.1 MFCC

When working along with audio data the widely used features are MFCC because its derivation is inspired by the perception of audio by human ear. Short term power spectrum of audio signals is represented by MFCC. The procedure for calculating the MFCC includes framing the signal into short frames and then calculates the periodgram estimate of power spectrum.On these power spectramel-filterbankis applied and energy is summed in each filter. Discrete cosine Transform (DCT) is taken after applying logarithms on thesemel-filterbanks.We usually keep 2-13 coefficients and discarded the others. Eqn (1) and Eqn (2) are used to apply conversion between frequency (f) and Mel (m).

m = 2595 × $log_{10}(1 + \frac{f}{700})$Eqn(1)

f = 700 × $(10^{\frac{m}{2595}} - 1)$Eqn(2)

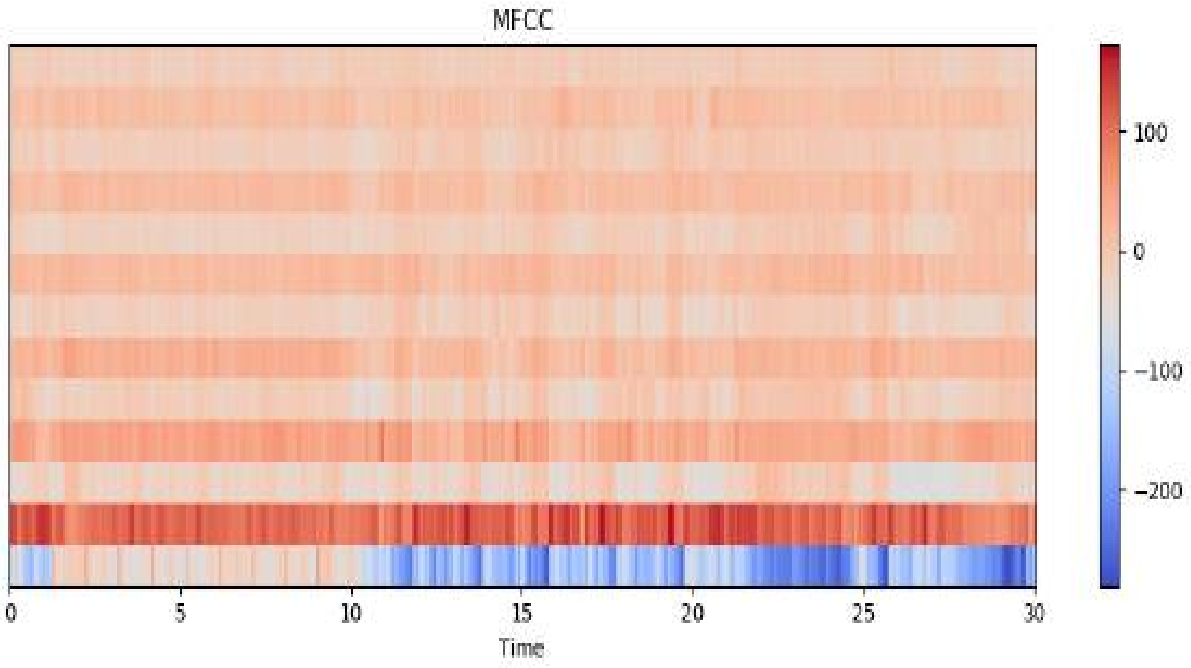


Figure 3 MFCC Plot for Rock Genre Music

To understand the implementation of MFCC in short we follow below mentioned steps. There is lots of concept understanding required to implement MFCC in efficient way. Here we are just mentioning the steps to implement MFCC without digging the details.

1. In first step we do framing of audio signal and convert signal in short frames.

2. In next step we calculate the periodogram estimation of the power spectrum for each short frame..

3. Calculate Mel – spaced filterbank. These are the 26 standard triangular filter (range is 20 to 40) which are appled to power spectral estimate of step 2 output. To calculate filter bank energy, each of the filter bank is multiplied with the power spectrum and then coeff are added together. Once this operation gets completed, we are left with only twenty six numbers which represents the energy in each of the filter bank. For further details, please see the below figure.

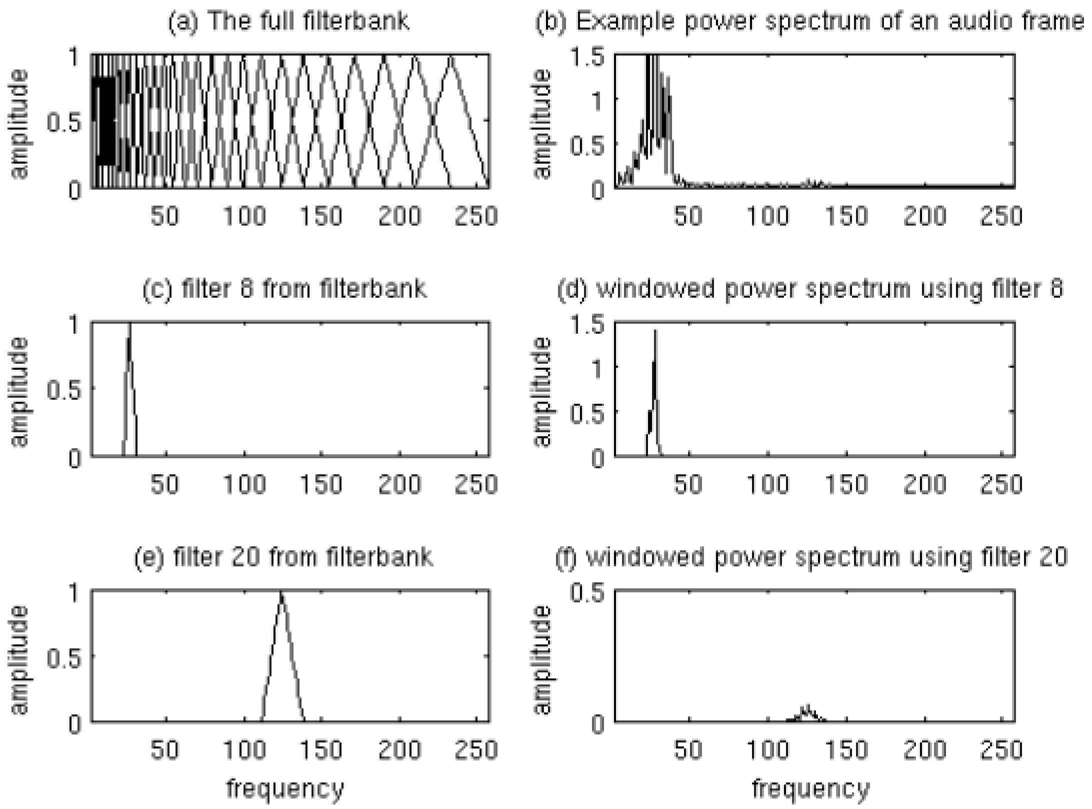


Figure 4. Plot for Mel Filterbank and windowed power spectrum

4. In this step we take the logarithm all twenty six energies computed from step 3 and we get the twenty-six log filter-bank energies.

5. After this we calculate the DCT (Discrete Cosine Transform) of the twenty-six log filter-bank energies to get twenty-six cepstral coefficients Here point to be noted is that for automatic speech recogintion we can also use MFCC feature. But for ASR application only initial 12 to 13 coeffs are being used and rest upper coeffs are discarded.

To get filter-banks as shown in figure 4 (a), first we have to choose upper and lower frequency. Appropriate value for lower frequency is 300Hz and for that of upper frequency is 8000Hz.

We choose upper frequency at 8000Hz as if speech is being sampled at 8000Hz then our upper frequency is limited to 4000Hz. Then by using eq (1) we convert lower and upper frequencies to MELs. To convert Mels back in Hz we can use eq (2). After that we can create the filter-banks. The 1st filter-bank will start from the first point, and touches the peak at the 2nd point, and again returns to the zero at next point. The 2nd filter-bank starts at the second point and touches the peak at third point, and again comes to zero at the 4th point etc. We have a formula to calculate all this and formula is given as below:

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \leq k \leq f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \leq k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases}$$

.

Where M represents the total filters and f() be the list of M + 2 MEL-spaced freqs.

4.1.2 Spectral Centroid

Center of mass of the spectrum is denoted by spectral centroid. For calculating spectral centroid conversion of audio data from time domain to frequency domain is required which is done using Fast Fourier Transform (FFT). Spectral centroid is the measure of the brightness of the sound.

It computes the shape of the spectrum, higher centroid value represent high brightness and highfrequency. Therefore, it is used in automatic measureof musical timbre.

Spectral centroid can be computed using below Eqn (3)

$$SpectralCentroid = \frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n=0}^{N-1} x(n)}$$

Wherex (n) show the weighted freq value or magnitude of bin number n, and f (n) denotes the centerfrequency of thatparticular bin.

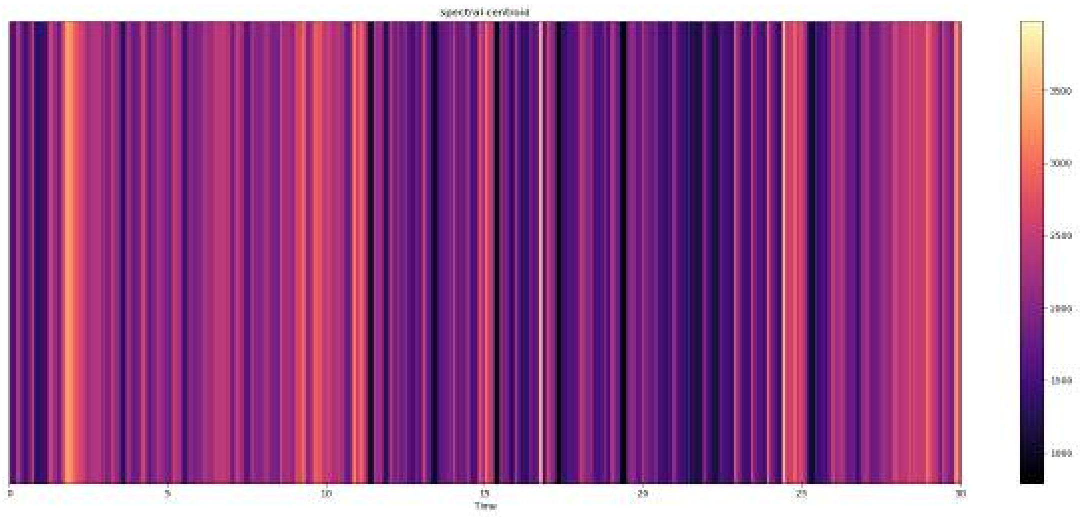


Figure 5 Spectral Centroid for Rock Genre Music

4.1.3 Spectral Contrast

Spectral contrast is a feature is used to represent spectral characteristics of music file. It is defined as the spectral peak, spectral valley and their difference in each frequency sub_band.

In most of music data, the spectral peaks are roughly represents the harmonic component,and spectral valleys represent the noises present in audio data or non-harmonic components. Therefore, this feature can roughly give us the relative distribution of the non - harmonic and harmonic components in the audio spectrum.

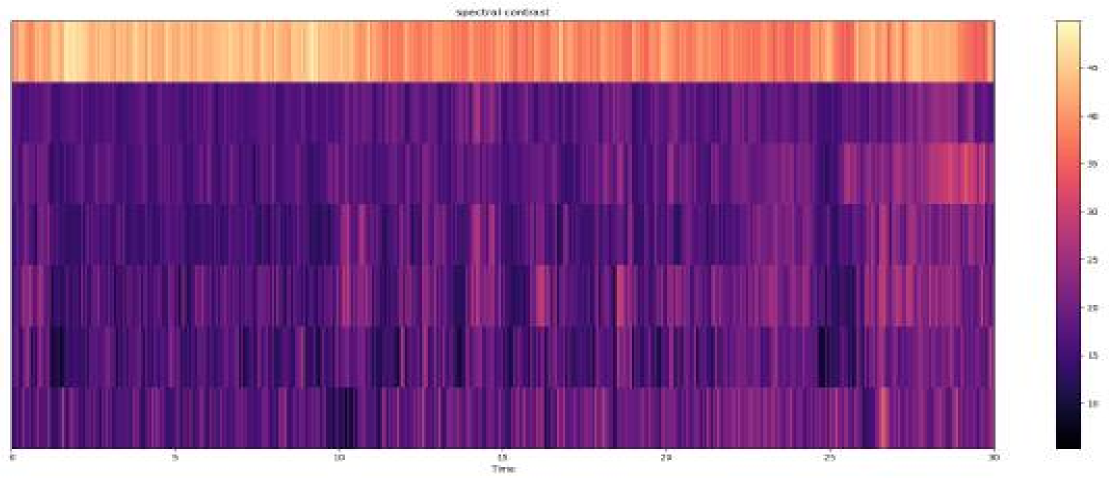


Figure 6. Spectral Contrast for Rock Genre Music

To find spectral Contrast we first divide the Music signal in to short Frames and then on every frame we perform the FFT operation to find the spectral components and after that spectral components are divided in 6 octave-based sub bands.At last, Spectral Contrast is calculated from the octave sub bands.

Let us assume that Fast Fourier Transform vector for k sub band is $(\mathbf{x_{k1}}, \mathbf{x_{k2}}, \mathbf{x_{k3}} \ldots\ldots\ldots \mathbf{x_{kn}})$

And this FFT vector is sorted in decending order which means.

$$x_{k1} \quad > \quad x_{k2} \quad > \cdots \quad > \quad x_{kn}$$

So we can calculate the spectral valley and spectral peaks by using below equations.

$$Peak_k = \log\{\frac{1}{\alpha N}\sum_{i=1}^{\alpha N} x'_{k,i}\}$$

$$Valley_k = \log\{\frac{1}{\alpha N}\sum_{i=1}^{\alpha N} x'_{k,N-i+1}\}$$

Where N is total k sub bands.

Below figure shows the Spectral Contrast and MFCC coefficient calulation procedure.

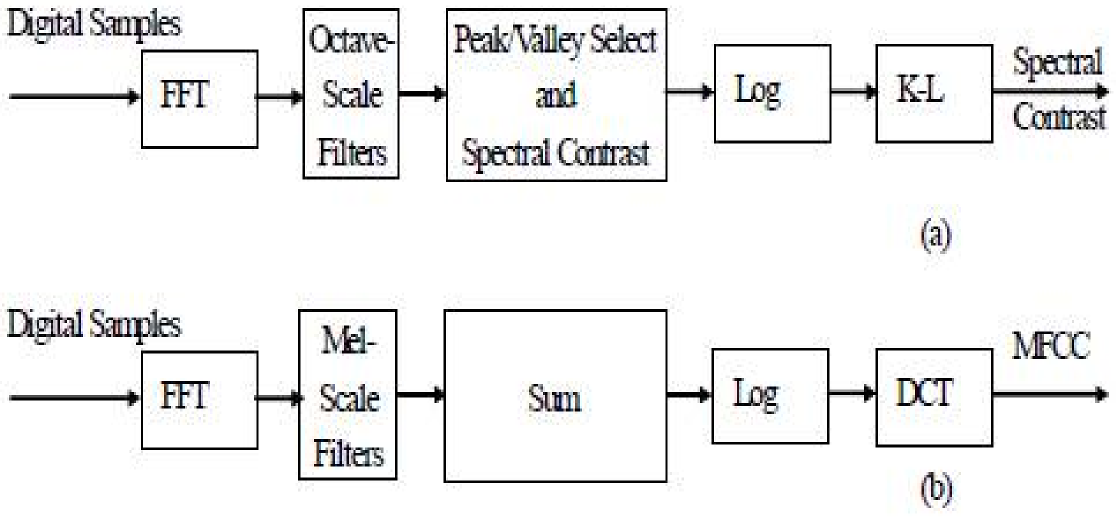


Figure 7. Spectral Contrast vs MFCC

4.1.4 Chroma

Pitch is a property of sound which is associated with musical melodies and categories the sound as low sound or high soud. It order the sound on the frequecny scale.

Croma feature is a property of pitch which indicates the color of musical sound.that can be divided in octave invariant value, known as Chroma and pitch height.

A chroma vector has length 12 and it is a feature vector that represents the energy of pitch classes C, C#, D, D#, E, F, F#, G, G#, A, A#, B. Melodic characteristics & harmonic of music is also represented by chroma it also uses cyclic helix for the representation of musical pitch.

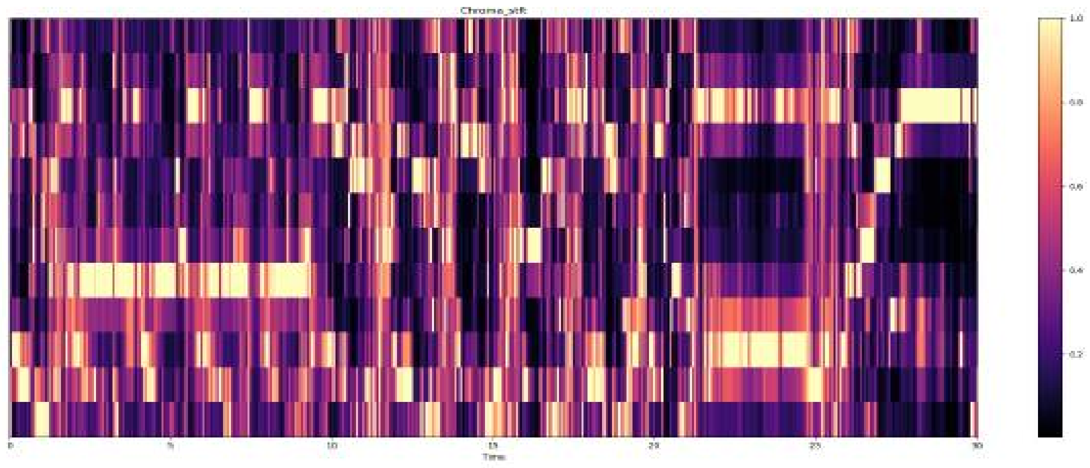


Figure 8. Chroma Spectrum for Rock Genre Music

4.1.5 Mel Spectrogram

Mel Spectrogram is defined as the time-frequency representation ofsignal. The Mel frequency scale isrepresented by usingEqn (1).

To calclulate Mel Spectrogram below are the steps:

1. We do sampling of input audio signal with a fixed window size (say 1024 or 2048).
2. Apply FFT algorithm on sampled signal of step1.
3. Consider the whole frequecny spectrum and convert it in evenly spaced frequecies.
4. Then generate the spectrogram for all frequencies of MEL-sacle.

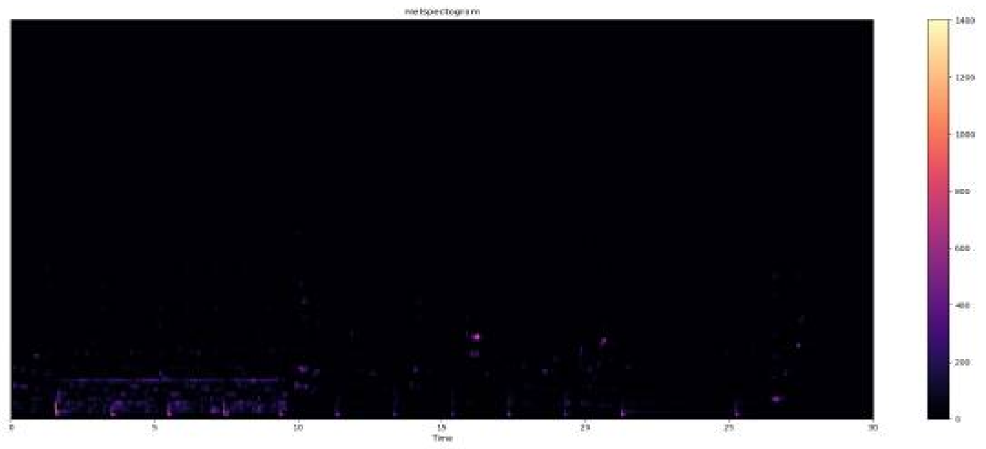


Figure 9. Mel Spectrogram for Rock Genre Music

4.2 DATASET

The famous GTZANdataset has been used for our experiment which is downloaded from http://marsyas.info/downloads/datasets.htmlwebsite.

For Music Information Retrieval (MIR) systems this dataset has been used it contains total one thousand audio tracks. Each audio track length is 30 second. GTZAN dataset contains 10genre and each genre contains100 tracks. Genres in this dataset are Metal, Rock,Classical, Hip-hop, Jazz, Blues, Reggae,Pop, Countryand Disco. These tracks are in .au format and consist of 16-bit depth single channel file having frequency 22050Hz.

CHAPTER-05

Methodology

From audio files features are extracted in the first step.The features MFCC, Mel Spectrogram, Zero crossing rate (ZCR), Chroma stft, Spectral centroid, RMSE, Spectral contrast, Spectral roll off and poly features are extracted using the python librosa library and these features are available in matrix form.In the second step these features are taken individually and applied the operations upon them to reduce dimensionality of the data with keeping the valuable information intact.

For this purpose, we used the librosa library to extract the first12 MFCC coefficient(using hop length = 512, sampling rate = 22050hz and FFT window size of 2048) and get the values in matrix form having dimension 12 1293 for each song and in the same way use default parameter features values mentioned in section 4 using librosa library in python. In next step features are divided into smaller chunks (let takes first 100 columns, 12 100 instead of 12 1293) and the operations which are mentioned in section 5.1 are being applied on the divided chunks.

5.1 Statistical Operations Used In Dimension Reduction

Let X is the set $(x_1, x_2, x_3 \ldots\ldots\ldots x_n)Andx_i$denotes the i^{th}element in the set.

5.1.1 Mean

Mean in calculated by taking sum of the all elements in the given set and divided the sum by total number of elements. It is calculated using below equation.

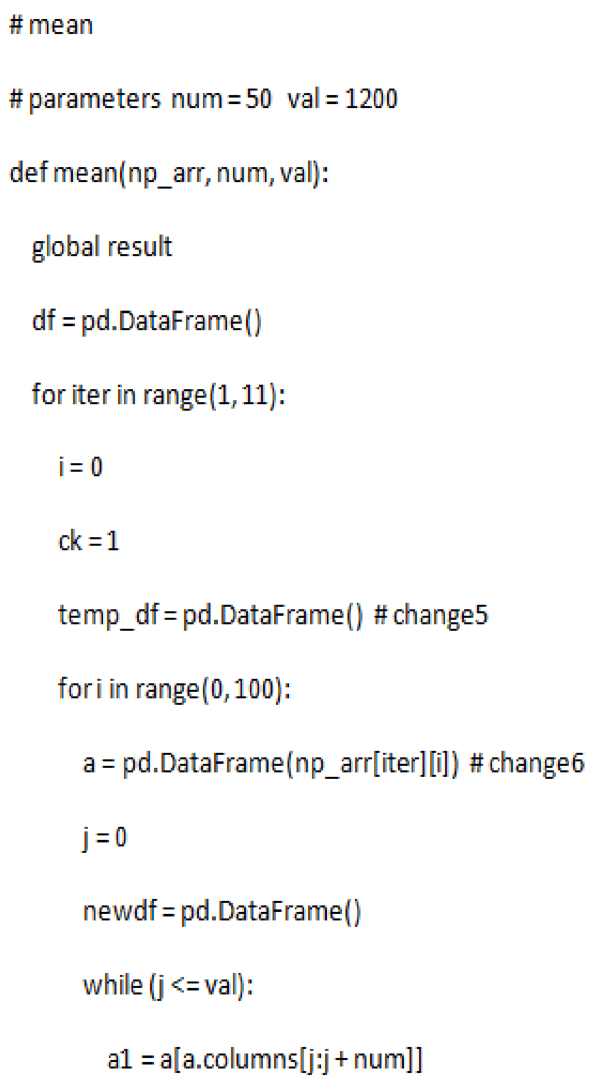
$$\text{Mean} = \frac{\sum_{i=1}^{N} x_i}{N}$$

Other terms that are used for Mean value is averge values of given data-set or also called as expected value. The formula given above is for finding out the airthmatic mean.

In our project we have calculated the mean value of data that we extracted from Music files.

Python libraries are being used for mean calculation to make task easier.

Below is the sample Python code used by in our code for find out airthmatic mean value.



```
# mean

# parameters  num = 50   val = 1200

def mean(np_arr, num, val):

    global result

    df = pd.DataFrame()

    for iter in range(1, 11):

        i = 0

        ck = 1

        temp_df = pd.DataFrame()  # change5

        for i in range(0, 100):

            a = pd.DataFrame(np_arr[iter][i])  # change6

            j = 0

            newdf = pd.DataFrame()

            while (j <= val):

                a1  = a[a.columns[j:j + num]]
            np_a = a1.mean(axis=1)

            # np_a = np_a.mean(axis=1)

            pd_a = pd.DataFrame(np_a)

            pd_a = pd_a.T

            newdf = pd.concat([newdf, pd_a], axis=1, ignore_index=True)

        temp_df = pd.concat([temp_df, newdf])  # change7

    df = df.append(temp_df)


from sklearn import preprocessing

x = df.values  # returns a numpy array

min_max_scaler = preprocessing.MinMaxScaler()

x_scaled = min_max_scaler.fit_transform(x)

df = pd.DataFrame(x_scaled)

print('mean\n')

result = pd.concat([result, df], axis=1, ignore_index=True)
```

5.1.2 Standard Deviation

It denotes the data dispersion. A lower value represents the data points which are closer to mean value and a higher value represents the data points which are far from the mean value in the set. Equation for calculating Standard Deviation is mentioned below

$$\textit{Standard Deviation} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - x_{mean})^2}{N-1}}$$

```
# standard deviation
# parameters  num = 50  val = 1200
def std(np_arr, num, val):
    global result
    df = pd.DataFrame()
    for iter in range(1, 11):
        i = 0
        ck = 1
        temp_df = pd.DataFrame()  # change5
        for i in range(0, 100):
            a = pd.DataFrame(np_arr[iter][i])  # change6
            j = 0
            newdf = pd.DataFrame()
            while (j <= val):
                a1 = a[a.columns[j:j + num]]
                j = j + num
                np_a = a1.std(axis=1)
            # np_a = np_a.mean(axis=1)

            pd_a = pd.DataFrame(np_a)

            pd_a = pd_a.T

            newdf = pd.concat([newdf, pd_a], axis=1, ignore_index=True)

        temp_df = pd.concat([temp_df, newdf])  # change7

    df = df.append(temp_df)


from sklearn import preprocessing

x = df.values  # returns a numpy array

min_max_scaler = preprocessing.MinMaxScaler()

x_scaled = min_max_scaler.fit_transform(x)

df = pd.DataFrame(x_scaled)

print('standard deviation\n')

result = pd.concat([result, df], axis=1, ignore_index=True)
```

5.1.3 Cumulative Sum

Cumulative Sum indicates the total contribution of data so far. This is a sequence of partial sums in the set. Cumulative sum sequence till i,(say S_i) is calculated using below equation.

$$S_i = \sum_{Z=1}^{Z=i} x_i$$

```
# Cumulative sum

# parameters   num = 50       val = 1200

def cumulative(np_arr, num, val):

   global result

   df = pd.DataFrame()

   for iter in range(1, 11):

      i = 0

      ck = 1

      temp_df = pd.DataFrame()  # change5

      for i in range(0, 100):

         a = pd.DataFrame(np_arr[iter][i])  # change6

         j = 0

         newdf = pd.DataFrame()

         while (j <= val):

            a1  = a[a.columns[j:j + num]]

            j = j + num
            np_a = a1.cumsum(axis=1)

            np_a = np_a.mean(axis=1)

            pd_a = pd.DataFrame(np_a)

            pd_a = pd_a.T

            newdf = pd.concat([newdf, pd_a], axis=1, ignore_index=True)

        temp_df = pd.concat([temp_df, newdf])  # change7

    df = df.append(temp_df)


from sklearn import preprocessing

x = df.values  # returns a numpy array

min_max_scaler = preprocessing.MinMaxScaler()

x_scaled = min_max_scaler.fit_transform(x)

df = pd.DataFrame(x_scaled)

print('Cumulative sum\n')

result = pd.concat([result, df], axis=1, ignore_index=True)
```

5.1.4 Mean Absolute Deviation

The amount of variation that occurred around the mean in the set is measured by Mean Absolute Deviation. In simple words, it can be defined as the average distance from the mean. This is calculated by using below equation

$$MeanAbsoluteDeviation = \frac{1}{N}\sum_{i=1}^{N} |x_i - \ x_{mean}|$$

5.1.5 Median

Median is the middle value in the set it divides the data in the set into lower half and upper half. It has significance because it is less affected by the outliers and skewed data.

```
# median
# parameters num = 50   val = 1200
def median(np_arr, num, val):
  global result
  df = pd.DataFrame()
  for iter in range(1, 11):
    i = 0
    ck = 1
    temp_df = pd.DataFrame()  # change5
    for i in range(0, 100):
      a = pd.DataFrame(np_arr[iter][i])  # change6
      j = 0
      newdf = pd.DataFrame()
      while (j <= val):
        a1 = a[a.columns[j:j + num]]
        j = j + num
```

```
            np_a = a1.median(axis=1)

             pd_a = pd.DataFrame(np_a)

            pd_a = pd_a.T

            newdf = pd.concat([newdf, pd_a], axis=1, ignore_index=True)

          temp_df = pd.concat([temp_df, newdf])  # change7


      df = df.append(temp_df)

  from sklearn import preprocessing

  x = df.values  # returns a numpy array

  min_max_scaler = preprocessing.MinMaxScaler()

  x_scaled = min_max_scaler.fit_transform(x)

  df = pd.DataFrame(x_scaled)

  print('median\n')

  result = pd.concat([result, df], axis=1, ignore_index=True)
```

5.1.6 Standard Error of the Mean

Standard error of mean is used to find the effect of random changes in the data. It denotes the mean variation with different kind of experiments computed the same value. Standard error of mean would be high if the effect of random changes is significant standard error of mean would be zero if in the repeated experiments there is no change in data points. This is calculated using below equation

$$StandardErroroftheMean = \frac{\text{Standard Deviation}}{\sqrt{N}}$$

5.1.7 Maximum

Maximum is denoted by the largest value among the all element of data in the set.

```
# max

# parameters    num = 50     val = 1200

def max1(np_arr, num, val):

    df = pd.DataFrame()

    for iter in range(1, 11):

        i = 0

        ck = 1

        temp_df = pd.DataFrame()  # change5

        for i in range(0, 100):

            a = pd.DataFrame(np_arr[iter][i])  # change6

            j = 0

            newdf = pd.DataFrame()

            while (j <= val):

                a1  = a[a.columns[j:j + num]]

                j = j + num

                np_a = a1.max(axis=1)
            pd_a = pd.DataFrame(np_a)

            pd_a = pd_a.T

            newdf = pd.concat([newdf, pd_a], axis=1, ignore_index=True)

        temp_df = pd.concat([temp_df, newdf])  # change7

    df = df.append(temp_df)


from sklearn import preprocessing

x = df.values  # returns a numpy array

min_max_scaler = preprocessing.MinMaxScaler()

x_scaled = min_max_scaler.fit_transform(x)

df = pd.DataFrame(x_scaled)

print('max\n')

result = pd.concat([result, df], axis=1, ignore_index=True)
```

5.1.8 Cumulative Maximum

Cumulative Maximum returns an array of non-decreasing elements which are cumulative maxima of the values in the numeric vector or sequence.

Once we applied all operations mentioned in section 5.1 on given small chunks of data, next step is to concatenate these chunks to get the desired dataset that is to be passed through classification models. For other features also the same process will be repeated. Using this method, we are able to reduce the data by many folds, keeping the vital information intact in the data.

After following above steps, we got our required data then we start concatenating these features and begin the experiment using machine learning models that are mentioned in section 5.2

These machine learning models were also used in previous work and we got the promising results. For these algorithms we used scikitlearn library in python.

```
# Cumulative Maximum

# parameters   num = 50   val = 1200

def cummax(np_arr, num, val):

  global result

  df = pd.DataFrame()

  for iter in range(1, 11):

    i = 0

    ck = 1

    temp_df = pd.DataFrame()   # change5

    for i in range(0, 100):

      a = pd.DataFrame(np_arr[iter][i])  # change6

      j = 0

      newdf = pd.DataFrame()

      while (j <= val):

        a1 = a[a.columns[j:j + num]]

        j = j + num

        np_a = a1.cummax(axis=1)

        np_a = np_a.mean(axis=1)
```

```
            pd_a = pd.DataFrame(np_a)

            pd_a = pd_a.T

            newdf = pd.concat([newdf, pd_a], axis=1, ignore_index=True)

        temp_df = pd.concat([temp_df, newdf])  # change7

    df = df.append(temp_df)


from sklearn import preprocessing

x = df.values  # returns a numpy array

min_max_scaler = preprocessing.MinMaxScaler()

x_scaled = min_max_scaler.fit_transform(x)

df = pd.DataFrame(x_scaled)

print('Cumulative  Maximum\n')

result = pd.concat([result, df], axis=1, ignore_index=True)
```

5.2 Machine Learning Algorithms Used In Experiments

5.2.1Support Vector Machine (SVM)

SVM was introduced first time by Vapnik, Guyon, and Boser and this ML approach is used for both regression and classification problems. In other words, SVM is tool used for prediction and it automatically avoid the over-fit data to maximize the prediction accuracy.

In past ML algo’s main focus was to learn the simple function to achieve the goal of learning and output was a hypothesis which is used for correct classification of training data. These algorithms were used to find a accurate fit to the given data. Here the term generalization comes that is ability of output hypothesis to classify the testing data correctly.

Another task is to find-out the best trade off in trading complexity and no of epochs. The diagram below illustrate the same.

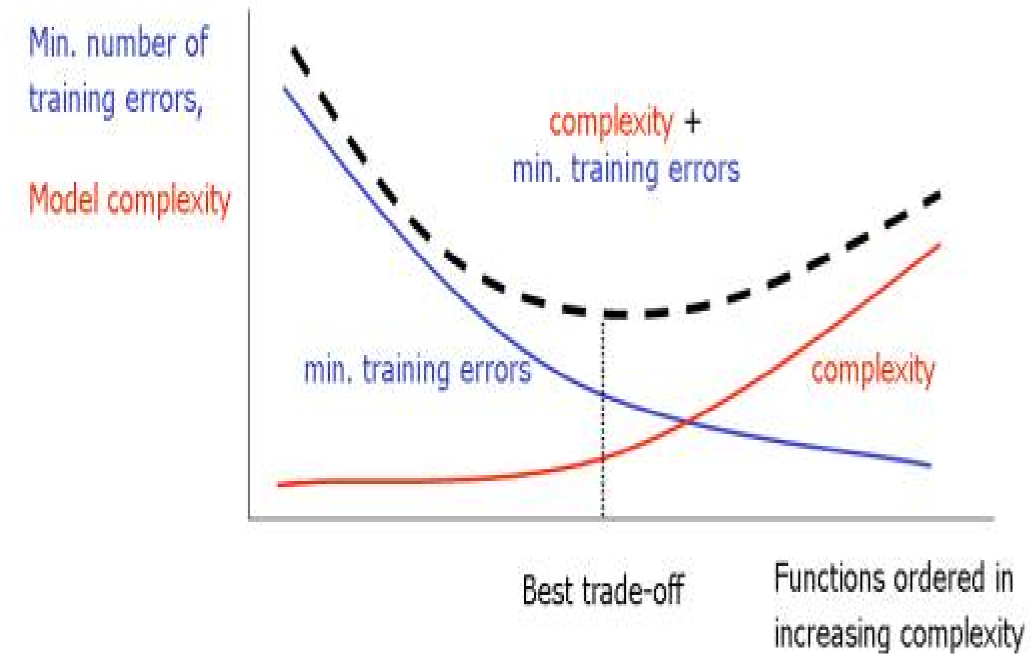
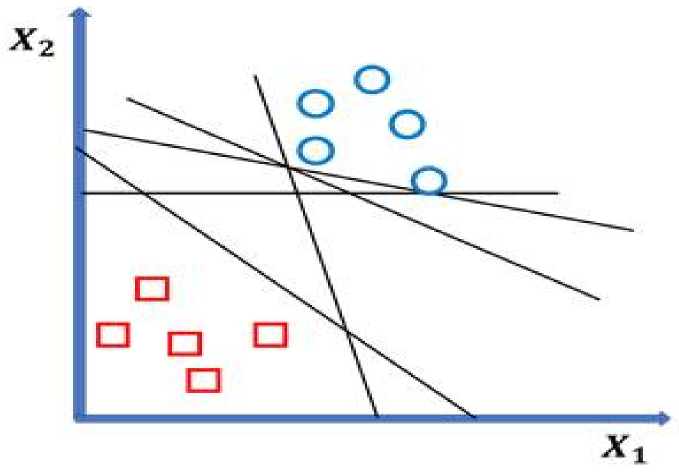

Min. number of training errors,
Model complexity
complexity + min. training errors
min. training errors
complexity
Best trade-off
Functions ordered in increasing complexity

Figure 10. complexity vs Number of epoch

SVM is a supervised classification approach that highly used.In the N dimensional space, SVM is used to find a optimal hyper plane which has maximum distance between the data points and the plane. SVM used various kernels like poly, linear and rbf. SVM is highly used because it has low computational usage and SVM effectively distinguish the linear as well as complex data. SVM is suitable algorithm for audio data because audio data is complex in nature. SVM with poly kernel is used for audio data to get better result.



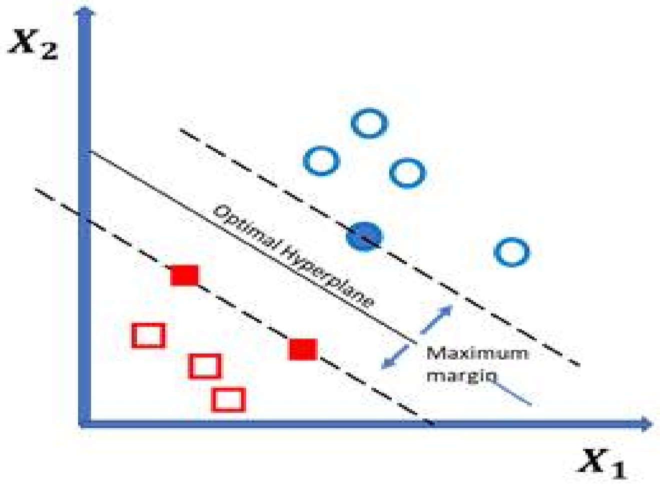


Figure 11. SVM simplified

Mathematically to represents SVM QP_formulation for SVM classification.

We are using the simple representation for elaboration purpose here.

SVM classification:

$$\min_{f,\xi_i} \|f\|_K^2 + C\sum_{i=1}^{l}\xi_i \qquad y_i f(\mathbf{x}_i) \geq 1 - \xi_i, \text{ for all } i \quad \xi_i \geq 0$$

Also we have equation for SVM classification and dual formulation.

$$\min_{\alpha_i} \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \qquad 0 \leq \alpha_i \leq C, \text{ for all } i; \qquad \sum_{i=1}^{l}\alpha_i y_i = 0$$

Where ξ_i is called slack variable & it is a measure of error that occurs at (x_i, y_i).

Training the SVM becomes a huge challenge when no of training points are very large. There are various ways to fast train the SVM model available today.

5.2.2 Random Forest

Random Forest is widely used for solving classification and regression problems. This is supervised learning approach. Random Forest is a collection of decisions trees. This is an ensemble learning approach. The major advantage with random forest is this adds an additional randomness when tree grows. For getting better results, random forest search the best feature instead of searching a most important feature when it splits the tree.

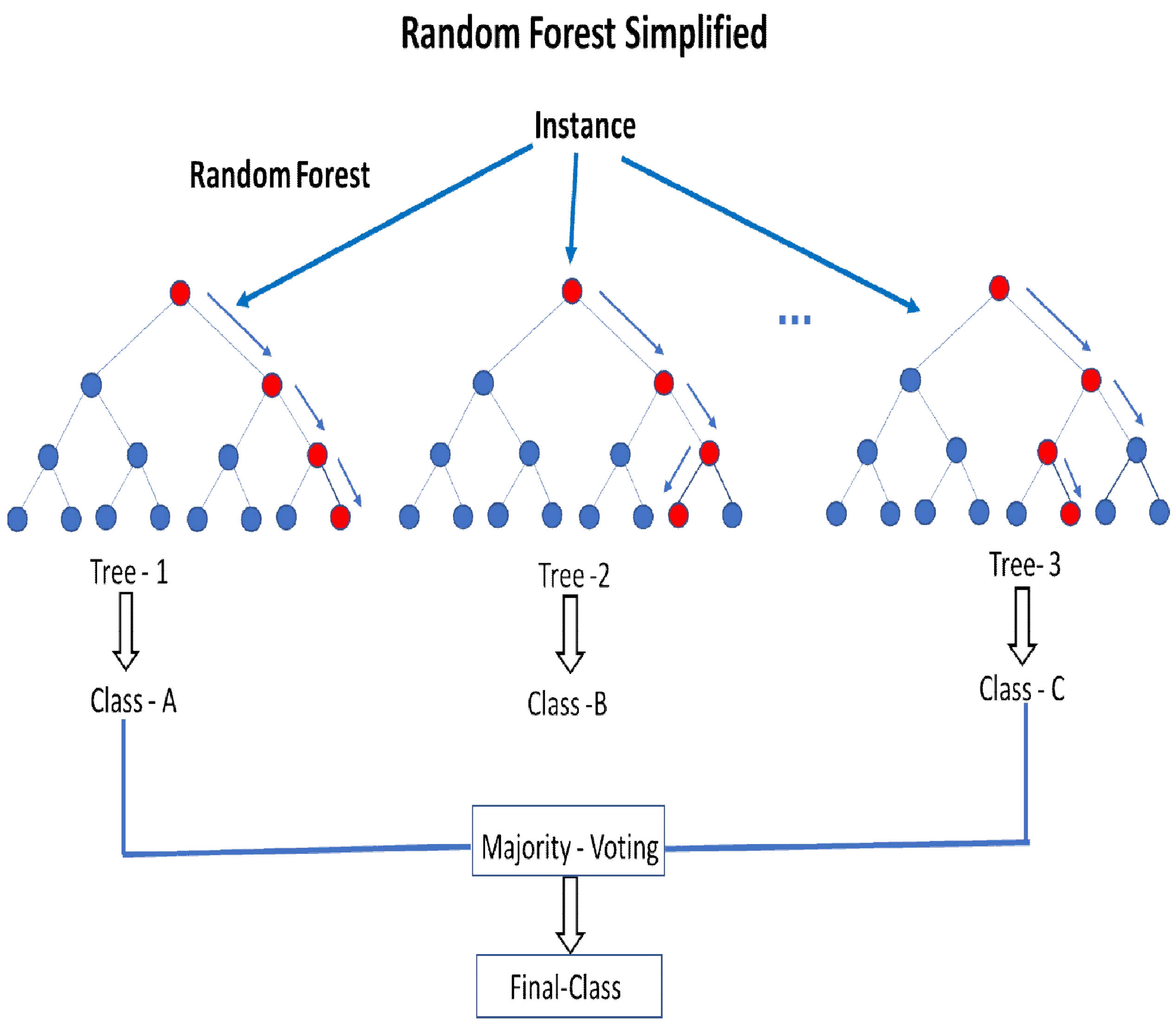


Figure 12. Random Forest Simplified

While node splitting, therandom forest selects the random features. Over fitting is major problem in machine learning algorithms that can be overcome by increasing the depth of the trees. Also R. F. works efficiently when data set contains large amount of items also its variance is quite low as compared to single decision tree. R.F. are flexible also so provide us better accuracy. One of the imp feature that R.F have is it does not required scaled input. It can give accurate results for un-scaled input as well.

With lots of advantages there are few disadvantages also for Random Forests. The major one is its high complexity and to construct the R. F. are more hard and time consuming than that of the decision trees. Also advanced resources are required for R.F. computation. Time complexity in R.F are very high if compared with other algorithms.

5.2.3 k-Nearest Neighbours (k-NN)

k-NN is easy and simple supervised ML algorithm that is useful in solving classification and regression problem. In k-NN algorithm the challenge is to find out the particular value of ‘k’ for a given data for which predictions are accurate.

This algorithm is Lazy algorithm because it does not generalize the training data for building any model. It performs the task like finding nearest neighbours at the time of prediction.

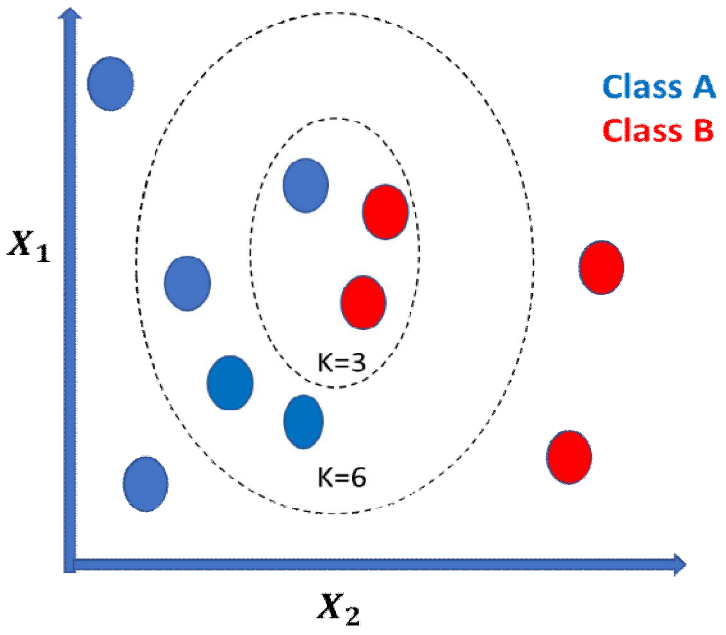


Figure 13. K-NN Simplified

5.2.4 XGBoost

XGBoostis Extreme Gradient Boosting. Boosting means to convert week learners to strong learners. To correct its previous predictor, gradientboosting adds the predictors sequentially. To correct the errors of previous predictors, gradient boosting replaces the new predictor in place of previous predictor which causes errors. This is very slow algorithm for computation because of its sequential model training. XGBoost is new implementation for gradient boosting that improves performance and computational speed of gradient boosting algorithm. In this paper we experimented with different machine learning models along with different combination of the features and combination of operations applied on them.

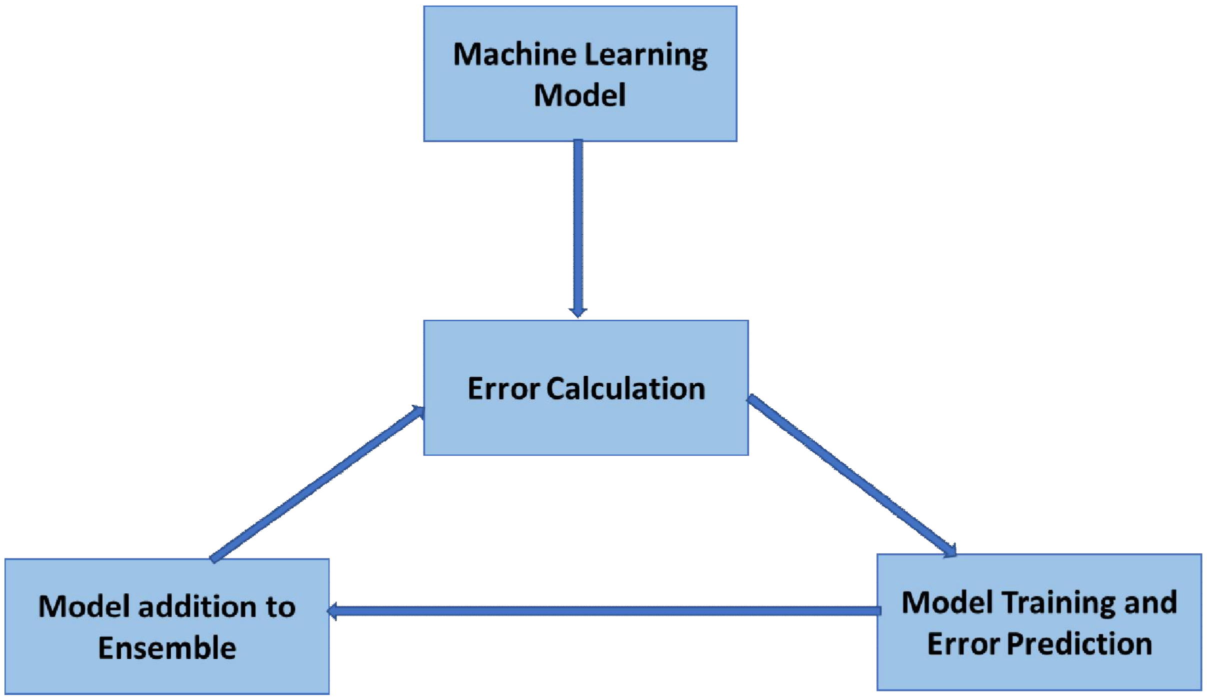

Machine Learning Model
Error Calculation
Model Training and Error Prediction
Model addition to Ensemble

Figure 14. XGBoost Algorithm

CHAPTER-06

Result

For the evaluation purpose, GTZAN dataset is divided in 80:20 split that means 80% of data is used for model training and 20% is used for testing, and also divided in 90:10 split that means 90% of data is used for model training and 10% is used for testing and evaluation is done using the accuracy score, precision, confusion matrix,recall and f1 score.

Models Used	Random Forest (in %)	SVM Linear (in %)	SVM Poly (in %)
MFCC	52.5	54.5	56.0
Chrome_stft	33.0	34.5	35.0
RMSE	40.4	26.0	28.5
ZCR	32.4	21.0	19.0
Poly Features	38.0	30.0	35.5
Spectral Centroid	36.0	27.5	29.0

Table 1.Simple approach by taking features without applying any operation.

The testing and training dataset is changed for each iteration for calculating average accuracy. If we change machine learning models and combination of feature used, classification accuracy also gets changed.

Model Used	Random Forest (in %)	SVM (in %)
Chroma_stft, MFCC	49.5	56.0
Chrome_stft,MFCC,Spectral Centroid	48.5	53.5

Chrome_stft,MFCC,Spectral Centroid, Poly Features	51.5	54.0
Chrome_stft,MFCC,Spectral Centroid, ZCR,Poly Features, RMSE	54.0	58.0

Table 2.Combining Features but no individual operation on features

In our experiment we have used various classification models like SVM with variationin kernels, k-NearestNeighbour, Random Forest andXGboosting. Table 1 indicates the results of various models on using spectral features and without applying any operations. Table 2 indicates the results of combination of various features but without applying dimensionality reduction. After trying various combinations of features the best feature set comes out to be MFCC, chroma stft, Mel spectrogram, spectral centroidand spectral contrast.

Feature Used	Average of 70 iterations (in %)	Maximum of 70 iteration (in %)
MFCC	70.46	77
Spectral Contrast	52.4	60
Mel spectrogram	56.14	61
Chrome_stft	40.43	46
MFCC+Contrast+Mel spectrogram + Centroid + Chrome_stft	79	88.5

Table 3. Results of SVM using poly and linear kernel after dimensionality Reduction

	Precision	Recall	FI-Score	Support
Blues	0.94	1.00	0.97	16
Classical	1.00	1.00	1.00	9
Country	0.92	1.00	0.96	11
Disco	0.71	1.00	0.83	10
Hiphop	0.92	0.92	.92	13
Jazz	1.0	1.0	1.0	9
Metal	1.0	0.80	0.89	10
Pop	1.0	0.88	0.93	8
Reggae	0.71	0.71	0.71	7
Rock	1.00	0.57	0.73	7
Micro average	0.91	0.91	0.91	100
Macro average	0.92	0.89	0.89	100
Weighted average	0.92	0.91	0.91	100

Table 4. Classification Report of 90:10 split

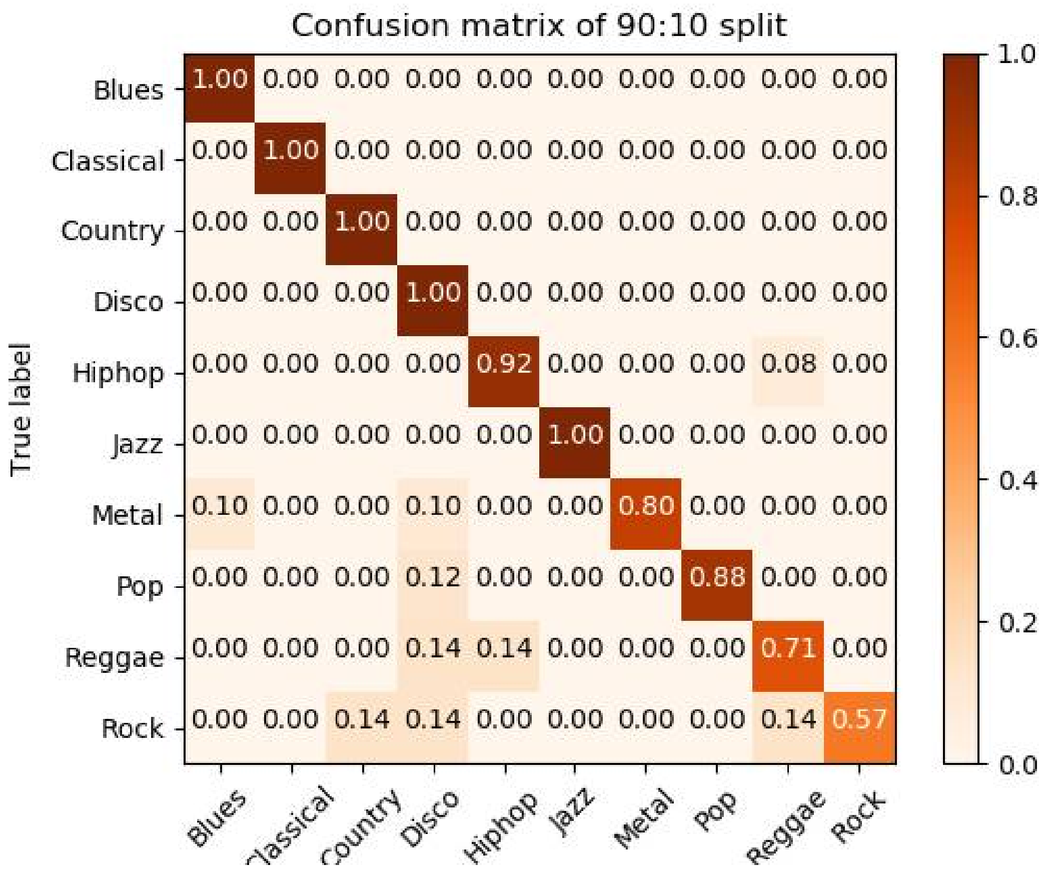


Figure 15. Confusion matrix with 90:10 dataset split

Using the above features and after applying statistical operation on these features, the data set gives highest accuracy with SVM poly or linear kernel algorithm as these two are almost equivalent.

Model Used	Average (in %)	Maximum (in %)
SVM poly	80.25	91.0
SVM linear	79.0	88.0
Random Forest	70.13	78.0
k-NN	70.62	81.0
XGboost	71.0	78.0

Table 5. Results with different Models with 90:10 split

But poly kernel is slightly better than linear kernel. After dimensionality reduction of feature and applying SVM algorithm, results are shown in Table 3.

Confusion matrix with 80:20 split is given in Figure 9 and classification report for the same is given in Table 6. Results indicate that the approach of dimensional reduction produces the best result and improves overall accuracy.

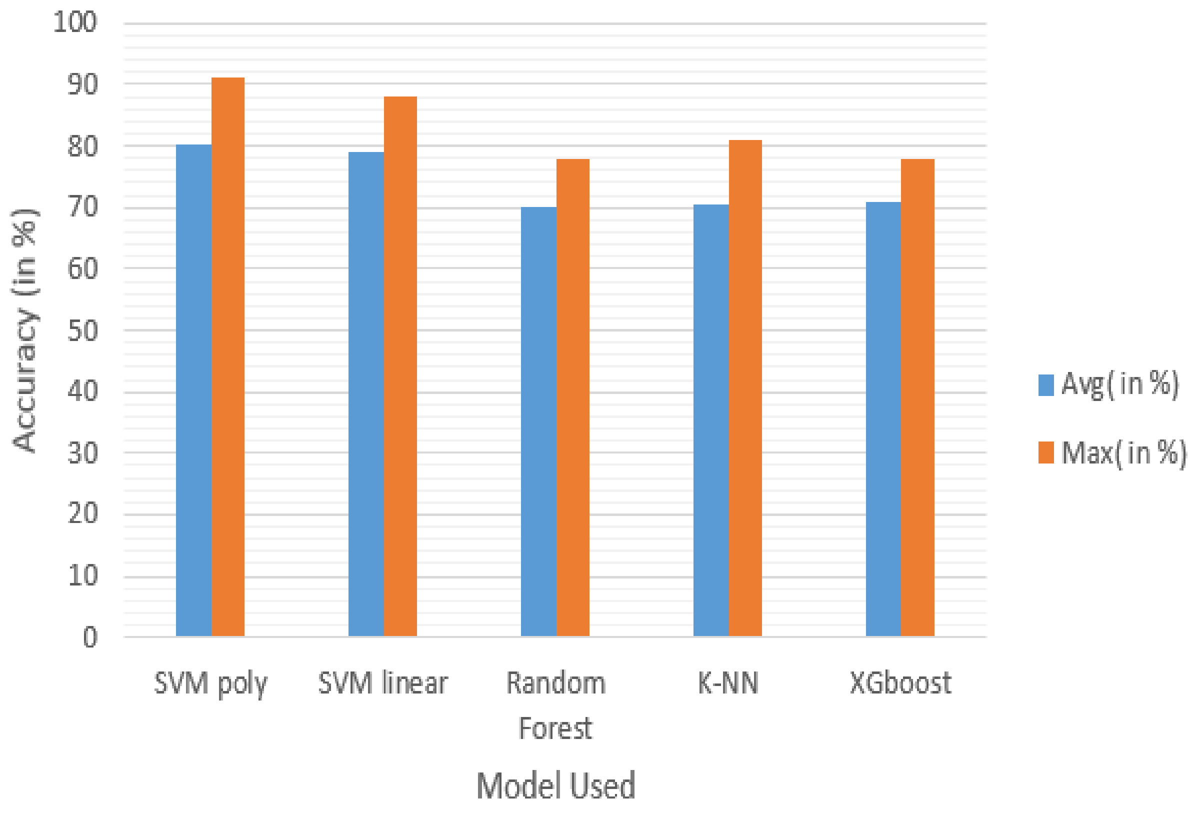


Figure 16 Pictorial representation with 90:10 dataset split

Confusion matrix with 90:10 split is given in Figure 7 and classification report for the same is given in Table 4

After changing the dataset in training and testing phase, we have also calculated the average accuracy for different models and explained as below:

1. Using MFCC feature with SVM poly kernel, average accuracy obtained is 70.46% and maximum accuracy is 77%.

2. Using spectral contrast feature,with SVM poly kernel, average accuracy obtained is 52.4% and maximum accuracy is 60%.

	Precision	**Recall**	**F1-Score**	**Support**
Blues	0.83	1.00	0.91	15
Classical	0.96	0.96	0.96	26
Country	0.91	0.88	0.89	24
Disco	0.90	0.86	0.88	22
Hip-hop	0.94	0.74	0.83	23
Jazz	0.87	0.95	0.91	21
Metal	1.0	1.0	1.0	15
Pop	0.94	0.94	0.94	16
Reggae	0.76	0.89	0.82	18
Rock	0.74	0.70	0.72	20
Micro avg	0.89	0.89	0.89	200
Macro avg	0.89	0.89	0.89	200
Weighted avg	0.89	0.89	0.88	200

Table 6. Classification Report of 80:20 split

Using spectral contrast feature,with SVM poly kernel, average accuracy obtained is 52.4% and maximum accuracy is 60%.

3. Using Mel spectrogram,with liner SVM, average accuracy obtained is 52.4% and maximum accuracy is 60%.

4. Using chroma stft,with liner SVM, average accuracy obtained is 40.43% and maximum accuracy is 46%.

5. After combining all these features, accuracy is improved and obtained as high as 88.5%, and the average accuracy is also improved that comes out to be 79% using poly kernel SVM. With

using linear kernel algorithm with all features average accuracy obtained is 78% and max value is 84% which is almost equivalent that we get from SVM poly.

With using Random forest algorithm, average accuracy obtained is 68.9% and max value is 76% whereas after using k-NN average accuracy obtained is 69.2% and max value is 75%.

After using XGboost algorithm, average accuracy obtained is 72.6% and max value is 74%.

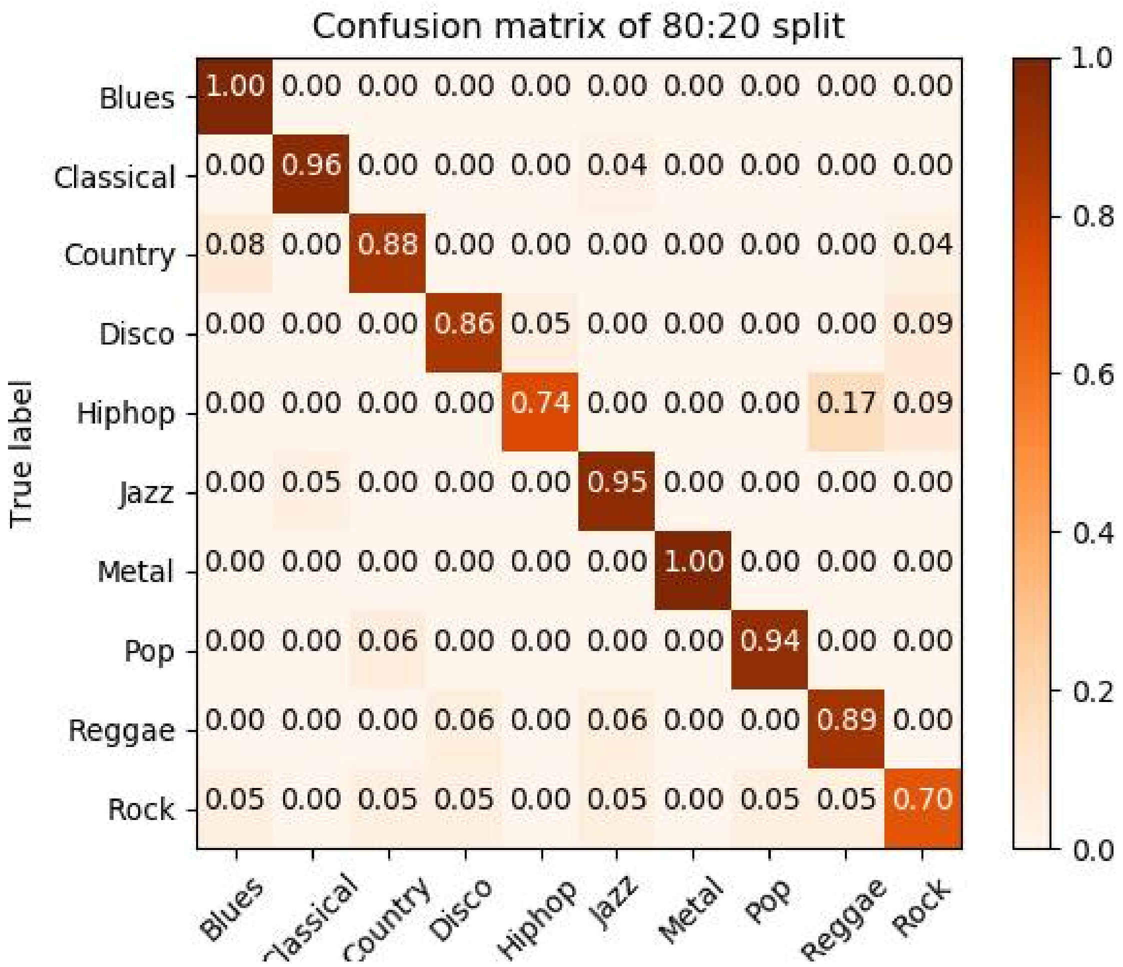


Figure 17. Confusion matrix with 80:20 dataset split

The pictorial representation with 80:20 dataset split and differentmodels is shown in Figure 10.

The results with 80:20 dataset split and different model is shown in Table 7. 90:10 dataset split is also used with increased training data to verify the impact on accuracy. After using poly SVM on 90:10 dataset split maximum accuracy that we obtained is high as 91%, but the average accuracy obtained is around 80.25%. Pictorialrepresentation of different models with 90:10 dataset split is shown in Figure 7.

Model Used	Average (in %)	Maximum (in %)
SVM poly	79.0	88.5
SVM linear	78.0	84.0
Random Forest	68.90	76.0
k-NN	69.20	75.0
XGboost	72.60	74.0

Table 7. Result with different Models with 80:20 split

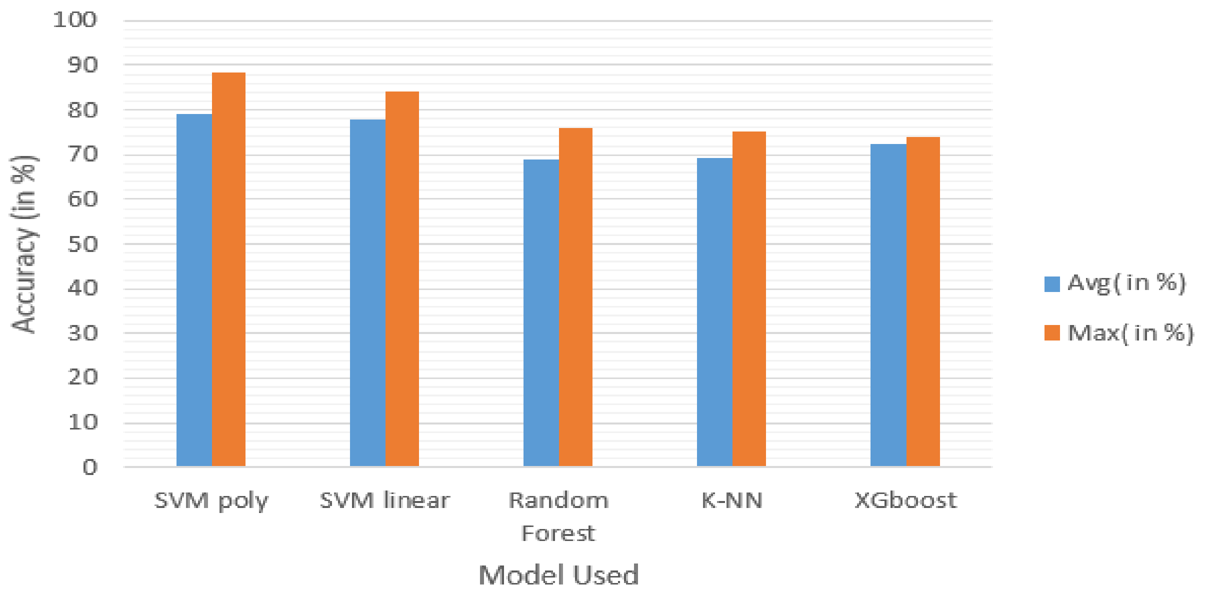


Figure 18. Pictorial representation with 80:20 dataset split

and the results with different models with 90:10 dataset are shown in Table 5.

CHAPTER-07

Conclusion and Future Scope

After various experiments, the result that we have obtained aboveindicates that accuracy in output is highly dependent on various machine learning algorithms that we have used in our experiment. Our work shows that SVM poly and linear, gives best results for the given dataset.The major task is how we process the input data and pass to these algorithms. The dimensionality reduction approach using statistical operations that is mentioned in section 5.1has huge impact on accuracy improvement rather than state of the art PCA on all 10 genres.When we divided data in 80:20 split, average accuracy obtained is 79% and maximum is 88.5%. On the other hand if we split the dataset to 90:10, average accuracy is increased to 80% and maximum accuracy obtained is 91%. In our approach we tried various combinations of features on different machine learning models. The major problem that we faced is increase in dimension size when we combine the features together that is resolved by dimensionality reduction technique which are mentioned above. The model accuracy proves that we are very close to design an autonomous system on different genre for music genre classification.

The main purpose of this research is to provide an approach for designing an autonomous system that classifies genres of GTZAN dataset with considerable accuracy.The deep learning approaches can be explored if we have machines with high computational power and huge size datasets. As some songs are derived from other genres which cause problems for giving just one class to that song so this problem can be further extended to multi-label genre classification.

CHAPTER-08

References

[1] N. Ahmed, T. Natarjan, and K. R. Rao. Discrete cosine transform. In IEEE Transactions on Computer, pages 90–93, January 1974.

[2] Muhammad Asim Ali and Zain Ahmed Siddiqui. Automatic music genres classification using machine learning. International Journal of Advanced Computer Science and Applications(IJACSA), 8, 2017.

[3] G. Biau. Analysis of a random forests model. The Journal of Machine Learning Research, page 1063–1095, 2012.

[4] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. CoRR, abs/1603.02754, 2016.

[5] James W. Cooley, Peter A. W. Lewis, and Peter D. Welch. The fast fourier transform and its applications. In IEEE Transactions on Education, volume 12, pages 27–34, March 1969.

[6] S. A. Dudani. The distance-weighted k-nearestneighbor rule. In IEEE Transactions on Systems, Man

and Cybernetics, volume 6(4), page 325–327.

[7] Miguel Francisco and Dong Myung Kim. Music genre classification and variance comparison on number of genres.

[8] Michael Haggblade, Yang Hong, and Kenny Kao. Music genre classification, 2011.

[9] D.-N. Jiang, L. Lu, H.-J. Zhang, J.-H. Tao, and L.-H. Cai. Music type classification by spectral contrast feature. In IEEE International Conference on Multimedia and Expo (ICME)), August 2002.

[10] Tao Li, MitsunoriOgihara, and Qi Li. A comparative study on content-based music genre classification. SIGIR Forum (ACM Special Interest Group on Information Retrieval), (SPEC. ISS.):282–289, 2003.

[11] GjorgjiMadjarov, Goran Pesanski, Daniel Spasovski, and DejanGjorgjevikj. Automatic music classification into genres. 2013.

[12] M. I. Mandel, G. E. Poliner, and D. P.W. Ellis. Support vector machine active learning for music retrieval. In Multimedia Systems, volume 12, page 3–13, 2006.

[13] Abdul Fattah Mashat, Mohammed M. Fouad, Philip S. Yu, and Tarek F. Gharib. A decision tree classification model for university admission system. International Journal of Advanced Computer Science and Applications(IJACSA), III, 2012.

[14] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto. librosa: Audio and music signal analysis in python. In 14th Python in Science Conference, 2015.

[15] Lindasalwa Muda, Mumtaj Begam, and I. Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. CoRR, abs/1003.4083, 2010.

[16] Meinard Muller, Frank Kurth, and Michael Clausen. Audio matching via chroma-based statistical features. In 6th International Conference on Music Information Retrieval, page 288–295, 2005.

[17] Kalpesh Patil, Rishabh Raj, and Chandrakanth. Music genre classification.

[18] Nilesh M. Patil and Milind U. Nemade. Music genre classification using mfcc, k-nn and svm classifier. International Journal Of Computer Engineering In Research Trends, IV:43–47, 2017.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. rettenhofer,R. Weiss, V. Dubourg, J. Vanderplas, A. Passos,D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay.

Scikit-learn: Machine learning in python. Journalof Machine Learning Research, page 2825–2830,2011.

[20] Archit Rathore and Margaux Dorido. Music genre classification.

[21] Bernhard Sch¨olkopf, Patrice Simard, Alex Smola, andVladimirVapnik. Prior knowledge in support vectorkernels. In Proceedings of the 1997 Conference onAdvances in Neural Information Processing Systems10, NIPS ’97, pages 640–646, Cambridge, MA, USA,1998. MIT Press.

[22] G. Tzanetakis and P. Cook. Automatic musical genreclassification of audio signals. In Proceedings of theInternational Symposium on Music Information Retrieval,2001.

[23] G. Tzanetakis and P. Cook. Musical genre classificationof audio signals. In IEEE Transactions on Speechand Audio Processing, volume 10, page 293–302,2002.